

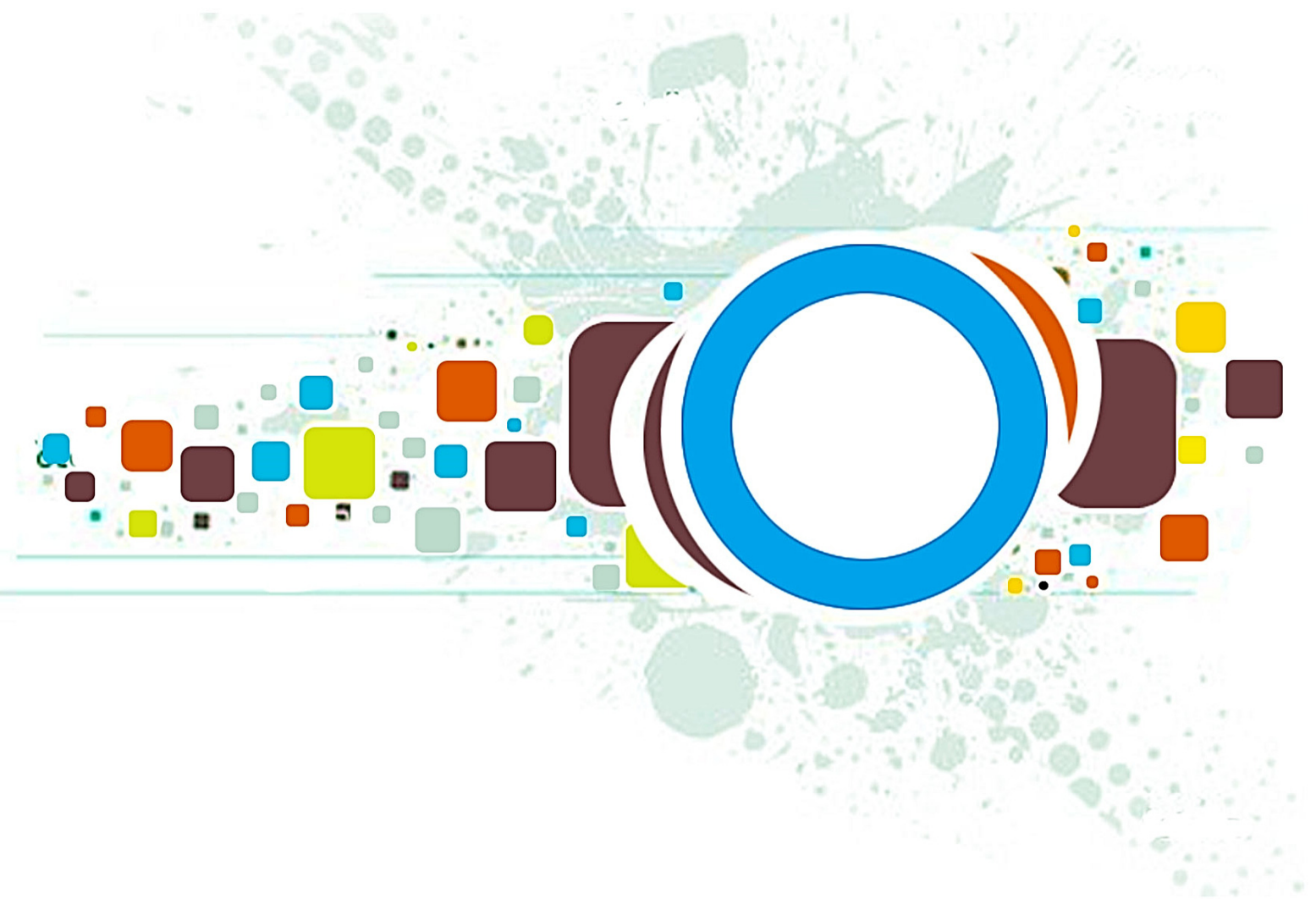
Volume 8 • Issue 5 • August / October 2014

Editor-in-Chief
Professor Hu, Yu-Chen

INTERNATIONAL JOURNAL OF
IMAGE PROCESSING (IJIP)

ISSN : 1985-2304

Publication Frequency: 6 Issues Per Year



CSC PUBLISHERS
<http://www.cscjournals.org>

INTERNATIONAL JOURNAL OF IMAGE PROCESSING (IJIP)

VOLUME 8, ISSUE 5, 2014

**EDITED BY
DR. NABEEL TAHIR**

ISSN (Online): 1985-2304

International Journal of Image Processing (IJIP) is published both in traditional paper form and in Internet. This journal is published at the website <http://www.cscjournals.org>, maintained by Computer Science Journals (CSC Journals), Malaysia.

IJIP Journal is a part of CSC Publishers

Computer Science Journals

<http://www.cscjournals.org>

INTERNATIONAL JOURNAL OF IMAGE PROCESSING (IJIP)

Book: Volume 8, Issue 5, September/October 2014

Publishing Date: 10-10-2014

ISSN (Online): 1985-2304

This work is subjected to copyright. All rights are reserved whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, re-use of illustrations, recitation, broadcasting, reproduction on microfilms or in any other way, and storage in data banks. Duplication of this publication of parts thereof is permitted only under the provision of the copyright law 1965, in its current version, and permission of use must always be obtained from CSC Publishers.

IJIP Journal is a part of CSC Publishers

<http://www.cscjournals.org>

© IJIP Journal

Published in Malaysia

Typesetting: Camera-ready by author, data conversion by CSC Publishing Services – CSC Journals, Malaysia

CSC Publishers, 2014

EDITORIAL PREFACE

The International Journal of Image Processing (IJIP) is an effective medium for interchange of high quality theoretical and applied research in the Image Processing domain from theoretical research to application development. This is the *Fifth Issue* of Volume *Eight* of IJIP. The Journal is published bi-monthly, with papers being peer reviewed to high international standards. IJIP emphasizes on efficient and effective image technologies, and provides a central for a deeper understanding in the discipline by encouraging the quantitative comparison and performance evaluation of the emerging components of image processing. IJIP comprehensively cover the system, processing and application aspects of image processing. Some of the important topics are architecture of imaging and vision systems, chemical and spectral sensitization, coding and transmission, generation and display, image processing: coding analysis and recognition, photopolymers, visual inspection etc.

The initial efforts helped to shape the editorial policy and to sharpen the focus of the journal. Started with Volume 8, 2014, IJIP appears with more focused issues. Besides normal publications, IJIP intends to organize special issues on more focused topics. Each special issue will have a designated editor (editors) – either member of the editorial board or another recognized specialist in the respective field.

IJIP gives an opportunity to scientists, researchers, engineers and vendors from different disciplines of image processing to share the ideas, identify problems, investigate relevant issues, share common interests, explore new approaches, and initiate possible collaborative research and system development. This journal is helpful for the researchers and R&D engineers, scientists all those persons who are involve in image processing in any shape.

Highly professional scholars give their efforts, valuable time, expertise and motivation to IJIP as Editorial board members. All submissions are evaluated by the International Editorial Board. The International Editorial Board ensures that significant developments in image processing from around the world are reflected in the IJIP publications.

IJIP editors understand that how much it is important for authors and researchers to have their work published with a minimum delay after submission of their papers. They also strongly believe that the direct communication between the editors and authors are important for the welfare, quality and wellbeing of the Journal and its readers. Therefore, all activities from paper submission to paper publication are controlled through electronic systems that include electronic submission, editorial panel and review system that ensures rapid decision with least delays in the publication processes.

To build its international reputation, we are disseminating the publication information through Google Books, Google Scholar, Directory of Open Access Journals (DOAJ), Open J Gate, ScientificCommons, Docstoc and many more. Our International Editors are working on establishing ISI listing and a good impact factor for IJIP. We would like to remind you that the success of our journal depends directly on the number of quality articles submitted for review. Accordingly, we would like to request your participation by submitting quality manuscripts for review and encouraging your colleagues to submit quality manuscripts for review. One of the great benefits we can provide to our prospective authors is the mentoring nature of our review process. IJIP provides authors with high quality, helpful reviews that are shaped to assist authors in improving their manuscripts.

Editorial Board Members

International Journal of Image Processing (IJIP)

EDITORIAL BOARD

EDITOR-in-CHIEF (EiC)

Professor Hu, Yu-Chen
Providence University (Taiwan)

ASSOCIATE EDITORS (AEiCs)

Professor. Khan M. Iftekharuddin
University of Memphis
United States of America

Assistant Professor M. Emre Celebi
Louisiana State University in Shreveport
United States of America

Assistant Professor Yufang Tracy Bao
Fayetteville State University
United States of America

Professor. Ryszard S. Choras
University of Technology & Life Sciences
Poland

Professor Yen-Wei Chen
Ritsumeikan University
Japan

Associate Professor Tao Gao
Tianjin University
China

Dr Choi, Hyung Il
Soongsil University
South Korea

EDITORIAL BOARD MEMBERS (EBMs)

Dr C. Saravanan
National Institute of Technology, Durgapur West Benga
India

Dr Ghassan Adnan Hamid Al-Kindi
Sohar University
Oman

Dr Cho Siu Yeung David

Nanyang Technological University
Singapore

Dr. E. Sreenivasa Reddy
Vasireddy Venkatadri Institute of Technology
India

Dr Khalid Mohamed Hosny
Zagazig University
Egypt

Dr Chin-Feng Lee
Chaoyang University of Technology
Taiwan

Professor Santhosh.P.Mathew
Mahatma Gandhi University
India

Dr Hong (Vicky) Zhao
Univ. of Alberta
Canada

Professor Yongping Zhang
Ningbo University of Technology
China

Assistant Professor Humaira Nisar
University Tunku Abdul Rahman
Malaysia

Dr M.Munir Ahamed Rabbani
Qassim University
India

Dr Yanhui Guo
University of Michigan
United States of America

Associate Professor András Hajdu
University of Debrecen
Hungary

Assistant Professor Ahmed Ayoub
Shaqra University
Egypt

Dr Irwan Prasetya Gunawan
Bakrie University
Indonesia

Assistant Professor Concetto Spampinato
University of Catania
Italy

Associate Professor João M.F. Rodrigues

University of the Algarve
Portugal

Dr Anthony Amankwah

University of Witswatersrand
South Africa

Dr Chuan Qin

University of Shanghai for Science and Technology
China

Associate Professor Vania Vieira Estrela

Fluminense Federal University (Universidade Federal Fluminense-UFF)
Brazil

Dr Zayde Alcicek

firat university
Turkey

Dr Irwan Prasetya Gunawan

Bakrie University
Indonesia

TABLE OF CONTENTS

Volume 8, Issue 5, September/October 2014

Pages

- 220 - 224 Novel Approach for Image Restoration and Transmission
Rabab Abdul Rasool
- 225 - 244 Combining Generative And Discriminative Classifiers For Semantic Automatic Image Annotation
Brahim MINAOUI, Mustapha OUJAOURA, Mohammed FAKIR
- 245 - 254 Header Based Classification of Journals Using Document Image Segmentation and Extreme Learning Machine
Kalpana S, Vijaya MS
- 255 - 277 Graph Theory Based Approach For Image Segmentation Using Wavelet Transform
Vikramsingh R. Parihar, Nileshsingh V. Thakur
- 278 - 293 An Approach For Single Object Detection In Images
Kartik Umesh Sharma, Nileshsingh V. Thakur
- 294 - 312 An Application of Eight Connectivity based Two-pass Connected-Component Labelling Algorithm For Double Sided Braille Dot Recognition
Shreekanth T, V. Udayashankara
- 313 - 324 Mixed Language Based Offline Handwritten Character Recognition Using First Stroke Based Training Sets
Magesh Kasthuri, V. Shanthi, Venkatasubramanian Sivaprasatham

- 325 - 342 Unsupervised Classification of Images: A Review
Abass Olaode, Golshah Naghdy, Catherine Todd
- 343 - 354 Connotative Feature Extraction For Movie Recommendation
N. G. Meshram, A. P. Bhagat
- 355 - 383 Vision-Based Localization and Scanning of 1D UPC and EAN Barcodes with Relaxed
Pitch, Roll, and Yaw Camera Alignment Constraints
Vladimir Kulyukin, Tanwir Zaman
- 384 - 396 Lip Reading by Using 3-D Discrete Wavelet Transform with Dmey Wavelet
Sunil S. Morade, Suprava Patnaik

Novel Approach for Image Restoration and Transmission

Rabab Abdul Rasool
Computer and Software Engineering
Al-Mustansirya University
Baghdad, Iraq

rabab_rassol@yahoo.com

Abstract

This paper develops a new technique in image restoration and transmission process, where the image size is halved after transforming it to the frequency domain by applying discrete Fourier transform. The conjugate symmetry and mirror property of transformed image spectrums could be utilized by deleting the redundant spectrums from second half image after tracking and keeping the conjugated locations. Those redundant locations are kept using *one- to- one* relationship. Depending on the halving procedure, the new image size will be divided by two. A reconstructed procedure is created to redistribute the deleted spectrum with their associated locations. The reconstructed image is ready now for restoring again by applying the inverse discrete Fourier transform back to the spatial domain. The restored image is qualified using Peak Signal to Noise Ratio measurement and the result was very satisfied. The advantages of this technique appear in the storage cost, where the memory locations will be reduced to the half. Also, from communication side, this work approved that the image transmission time needs to transmit the halved image is half of the original one.

Keywords: DFT, Conjugate Symmetry, Mirror, One to one.

1. INTRODUCTION

Most of digital image processing methods concern how to process image details in spatial domain. Image details might be color intensity or pixel coordinates (locations). The second domain that digital image could be processed in is the frequency domain. The transformation from spatial domain to the frequency one could be implemented by applying the Two Dimension Discrete Fourier Transform (2D-DFT) or Fast Fourier Transform (FFT) [1] [2]. The transformed image details are spectrums. These spectrums are distributed in a manner that has many properties like: DC coefficients; conjugate symmetry; shifting; and mirror. Transforming from spatial to the frequency domain will still provide same operation and even better because it is an alternative method for low pass and high pass filtering. Also, it could process some particular frequencies efficiency. The convolution in time or spatial domain could be intercepted into multiplication in frequency domain, and for this reason image will be processed as a whole block. This work adopts and develops new technique of considering the half image size in storing and network communication. This technique applies the mirror and conjugate symmetry property in implementing the tracking, mapping, and reconstructing procedure. Image restoration in [1] has different procedure which is based on the intensities of the nearest neighbor of pixel in spatial domain [3][4]. This paper is organized as follows: section 1 is an introduction. Section 2 provides a preview about the 2D DFT and its properties. The tracking, mapping and reconstructing procedure is explained in section 3. Section 4 produces an example of five restored images. The transmitting time for different image size over different networks types is compared in section 5. The final section is the conclusion in section 6.

2. THE TWO DIMENSION DISCRETE FOURIER TRANSFORM

Image with M rows and N columns can be transforming from spatial domain to the frequency domain by applying formula 1. While the inverse Fourier transform could be obtained from applying formula 2.

$$F(U, V) = \sum_{x=1}^M \sum_{y=1}^N f(x, y) \exp^{-2\pi j(\frac{xu}{M} + \frac{yv}{N})} \quad (1)$$

$$f(x, y) = 1/MN \sum_{u=1}^M \sum_{v=1}^N F(U, V) \exp^{2\pi j(\frac{xu}{M} + \frac{yv}{N})} \quad (2)$$

Where $f(x,y)$ is the two dimensional image in the spatial domain and $F(u,v)$ is the transformed image in the frequency domain.

Many properties could be obtained from that transformation to the frequency domain, it includes:

- Similarity
- Seprability
- DC Coefficient
- Shifting
- Conjugate Symmetry or Mirror

The two DFT properties, the DC coefficient and Mirror property is explained in figure 1. This figure clearly explained that the upper half image is the mirror or complex conjugate to the down half and vise versa. Also, the left half side of image is a mirror to right half side and vise versa. Next section will explain how tracking, mapping is, and reconstruction procedure will utilize these properties to obtain a new image with half size, which means half rows or columns [5][6][7].

	a		a*
b*	B*	d*	A*
	C	DC	C*
b	B	d	A

FIGURE 1: Conjugate Symmetry in 2D DFT.

3. ONE - TO - ONE MAP PROCEDURE

A powerful tracking and mapping procedure is applied to track the redundant spectrum locations in the lower half image, keep those 1 to 1 relations, and then delete them. The transformed image size is approximately halved, i.e.; (rows/2 + 1) or (columns/2 + 1). Now, the new image could be stored or transmitted. The restored or received image should be inversed back to the spatial domain after applying the mapping or reconstructing procedure. The restored image is qualified using Peak Signal to Noise Ratio measurement and the result is shown in figure 2.

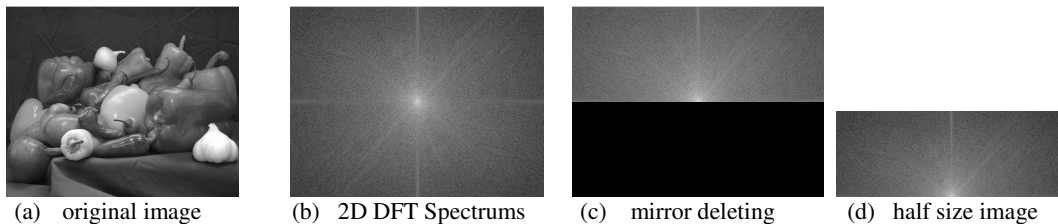
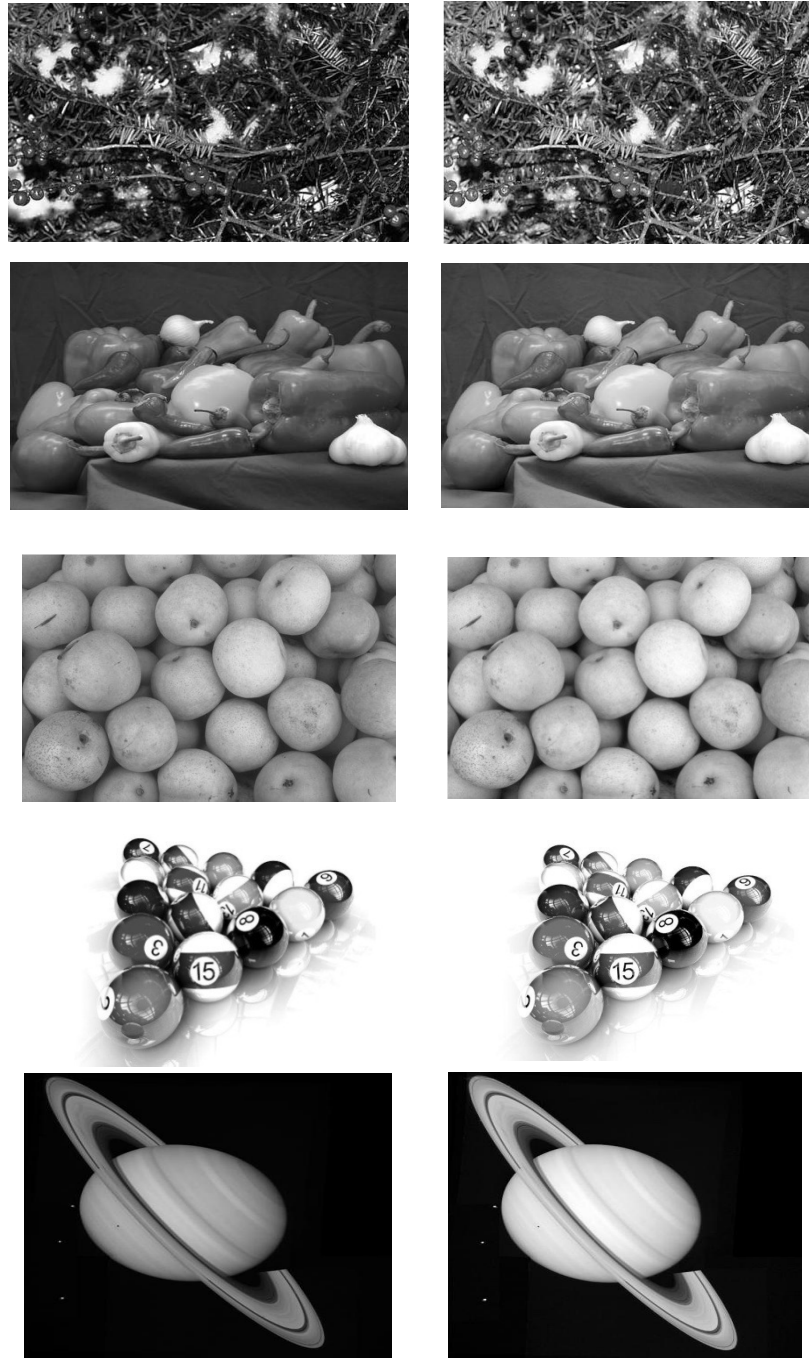


FIGURE 2: Mapping Procedure Sequence State.



(a) Original images

(b) Restored images

FIGURE 3: Five original and restored images, (a) origin images, (b) restored images.

4. THE TRANSMISSION TIME ESTIMATION

The new image file size has been evaluated over data transmission system. Five different images with different file size have been transmitted and the delivery time is calculated as in formula 3.

$$\text{Packet transmission time} = \text{Packet size} / \text{Bit rate} \quad (3)$$

Those five images are transformed to the frequency domain by applying the 2D-DFT. Then the redundant (mirror) information is cut-out. The new image size would be approximately halved of the original one. These new images are transmitted again over data transmission system. The file transmission time is re-calculated. Figure 4 represents the wired computer network with bit rate equals to 250 Mb/s.

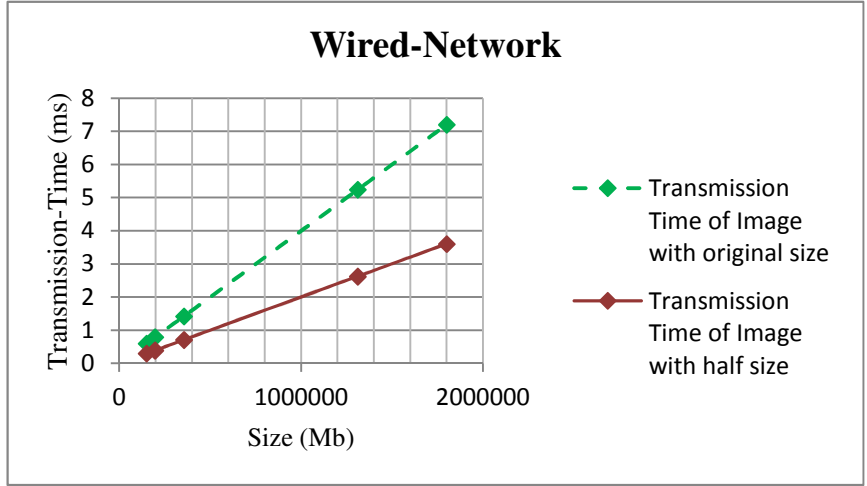


FIGURE 4: Wired Computer Network.

Clearly, the time need to transmit the original size is approximately twice, where the difference for image size 150 Kbyte is 0.3 m sec, while for image size 1.8 M Byte the difference time would be 3.6 m sec and so on as the image file size becomes large the differences time would be larger. Figure 5 belongs to the wireless network with bit rate equals to 600 M b/s. With image file size equals to 150 Kbyte, the difference of the delivery time between the two transmissions is approximately 0.125 msec. for image size 1.8 M Byte the differences time would be 1.5 msec.

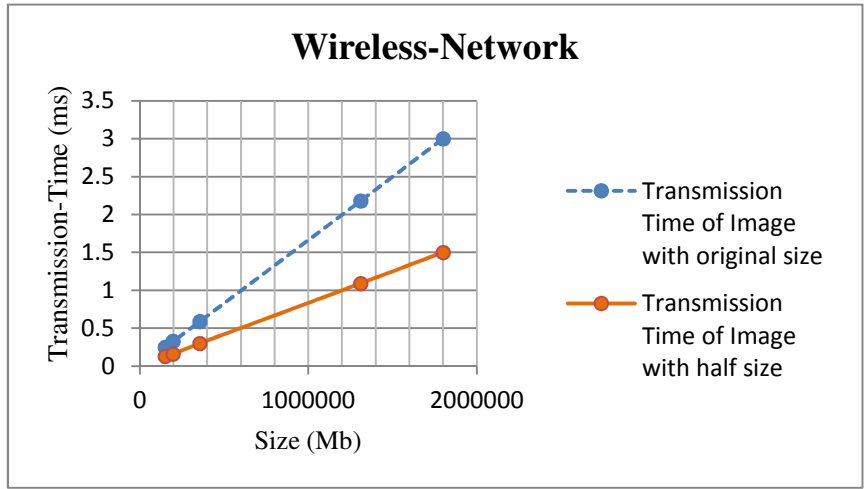


FIGURE 5: Wireless Computer Network.

This result approved that, transmitting half image size will eventually reduce the transmission time to the half under same circumstances like network type, distance, and bit rate.

5. CONCLUSION

A new and novel approach has been considered and implemented. This approach is depending on the two properties of the Two-Dimensional Discrete Fourier Transform, the mirror and the conjugate symmetry. A one-to-one map frequency domain procedure is developed and implemented to half image file and to restore or reconstruct the spatial domain image file again. Five images have been transformed, halved, stored, and restored again. The appearance of those images was very good and the transmission time needed for network delivery is reduced to the half of the original one.

6. REFERENCES

- [1] JAGADISH H. PUJARA. " Novel Approach for Image Restoration via Nearest Neighbor Method, Journal of Theoretical and Applied Information Technology, pp. 76-79, 2010.
- [2] Muthana Hamd. "RGB Image Reconstruction Using Two-Separated Band Reject Filters", Advances in Image and Video Processing, Vol. 2, No. 2, pp.1-7, 2014.
- [3] M. Andrew. An Introduction to Digital Image Processing with Matlab. Victoria University of Technology, 2004, pp. 85-115.
- [4] W. K. PRATT. Digital Image Processing. New York: John Wiley and Sons, 2001, pp. 185-319.
- [5] M. Goparaju and S Mohan. "Design & Implementation of DWT – IDWT Algorithm for Image Compression by using FPGA", International Journal of Scientific and Research Publications, Vol. 3, pp. 1-4, March 2013.
- [6] T. R. Jeyalakshmi and K. Ramar. "A Modified Method for Speckle Noise Removal in Ultrasound Medical Images", International Journal of Computer and Electrical Engineering, Vol. 2, No. 1, pp. 54-58, February, 2010.
- [7] Muthana Hamd and Rabab Rassol. "Quality Measurement for Reconstructed RGB Image via Noisy Environments", Advances in Image and Video Processing, Vol. 2, No.1, pp. 1-8, 2014.

Combining Generative And Discriminative Classifiers For Semantic Automatic Image Annotation

Brahim MINAOUI

*Faculty of Science and Technology,
Computer Science Department,
Sultan Moulay Slimane University.
PO Box. 523, Béni Mellal, Morocco.*

bra_min@yahoo.fr

Mustapha OUJAOURA

*Faculty of Science and Technology,
Computer Science Department,
Sultan Moulay Slimane University.
PO Box. 523, Béni Mellal, Morocco.*

M.Mustapha.Oujaoura@ieee.org

Mohammed FAKIR

*Faculty of Science and Technology,
Computer Science Department,
Sultan Moulay Slimane University.
PO Box. 523, Béni Mellal, Morocco.*

fakfad@yahoo.fr

Abstract

The object image annotation problem is basically a classification problem and there are many different modeling approaches for the solution. These approaches can be classified into two main categories such as generative and discriminative. An ideal classifier should combine these two complementary approaches. In this paper, we present a method achieving this combination by using the discriminative power of the neural networks and the generative nature of Bayesian networks. The evaluation of the proposed method on three typical image's database has shown some success in automatic image annotation.

Keywords: Automatic Image Annotation, Discriminative Classifier, Generative Classifier, Neural Networks, Bayesian Networks.

1. INTRODUCTION

Automatic image annotation help to bridge the semantic gap, that exists between low-level visual features and the high-level abstractions perceived by humans, by producing object labels or keyword annotations which are nearer to the high level semantic descriptions needed for good image retrieval.

In order to overcome this semantic gap, a number of current research efforts focus on robust classifiers achieving automatically multi-level image annotation [1-6]. These classifiers can be characterized as generative and discriminative according to whether or not the distribution of the image and labels is modeled.

It was observed that generatively-trained classifiers perform better with very few training examples and provide a principled way of treating missing information, whereas a classifiers trained discriminatively perform better with sufficient training data and provide a flexible decision boundaries [7]. Motivated by these observations, several researchers have proposed a variety of techniques that combine the strengths of these two types of classifiers. These hybrid methods, which have delivered promising results in the domains of object recognition [8-10], scene classification [11-15] and automatic image annotation [16-17], have been explored in different

ways: [9] and [11] propose a classifier switching algorithm to select the best classifier (generative or discriminative) for a given dataset and availability of label. [10], [14] and [15] propose a technique for combining the two classifiers based on a continuous class of cost functions that interpolate smoothly between the generative strategy and the discriminative one. [8, 12-13] and [16] propose a hybrid generative-discriminative approach in which the features extracted from a generative model are analyzed by a followed discriminative classifier. [17] devise a hybrid generative-discriminative learning approach that includes a Bayesian Hierarchical model (generative model) trained discriminatively.

In this paper, in an attempt to gain the benefit of both generative and discriminative approaches, we propose an approach which combines in a parallel scheme the Bayesian networks for the generative model and the neural networks for the discriminative classifier to accomplish the task of automatic image annotation. The annotation decision is realized by the vote of combined classifiers. Each classifier votes for a given keyword. The keyword that has the maximum of votes will be considered as the proper keyword for the annotation of an object in a query image.

The rest of paper is organized as follows. The various features used in this study are explained in Section 2. Section 3 presents the Bayesian networks and neural networks classifiers. Section 4 describes the experiences adopted to realize the automatic image annotation using these classifiers. Finally, the conclusion of this work is presented in Section 5.

2. FEATURES EXTRACTION

After dividing the original image into several distinct regions that correspond to objects in a scene by using region growing segmentation algorithm [18], the following descriptors are extracted:

2.1 Color Histogram

Typically, the color of an image is represented through some color model. There exist various color models to describe color information. The more commonly used color models are RGB (red, green, blue), HSV (hue, saturation, value) and Y, Cb, Cr (luminance and chrominance). Thus, the color content is characterized by 3 channels from some color models. In this paper, we used RGB color models. One representation of color image content is by using color histogram. Statistically, it denotes the joint probability of the intensities of the three color channels [19].

Color histogram describes the distribution of colors within a whole or within an interest region of image. The histogram is invariant to rotation, translation and scaling of an object but the histogram does not contain semantic information, and two images with similar color histograms can possess different contents.

The histograms are normally divided into bins to coarsely represent the content and reduce dimensionality of subsequent classification and matching phase. A color histogram H for a given image is defined as a vector by:

$$H = \left\{ h[i \in \{1, \dots, k\}] = \frac{\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \delta(f(x, y) - C(i))}{M \times N} \text{ and } (i-1) \times E\left(\frac{256}{k}\right) \leq C(i) < i \times E\left(\frac{256}{k}\right) \right\} \quad (1)$$

Where:

- i represent a color in the color histogram;
- $E(x)$ denotes the integer part of x ;
- $h[i]$ is the number of pixel with color i in that image;
- k is the number of bins in the adopted color model;

And δ is the unit pulse defined by:

$$\delta(x) = \begin{cases} 1 & \text{si } x = 0 \\ 0 & \text{si } x \neq 0 \end{cases} \quad (2)$$

In order to be invariant to scaling change of objects in images of different sizes, color histograms H should be divided by the total number of pixels M x N of an image to have the normalized color histograms.

For a three-channel image, a feature vector is then formed by concatenating the three channel histograms into one vector.

2.2 Legendre Moments

In this paper, the Legendre moments are calculated for each one of the 3 channel in a color image. A feature vector is then formed by concatenating the three channel moments into one vector.

The Legendre moments [20] for a discrete image of M x N pixels with intensity function f(x, y) is the following:

$$L_{pq} = \lambda_{pq} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} P_p(x_i) P_q(y_j) f(x, y) \quad (3)$$

Where $\lambda_{pq} = \frac{(2p+1)(2q+1)}{M \times N}$, xi and yj denote the normalized pixel coordinates in the range of [-1, +1], which are given by:

$$\begin{cases} x_i = \frac{2x - (M - 1)}{M - 1} \\ y_j = \frac{2y - (N - 1)}{N - 1} \end{cases} \quad (4)$$

$P_p(x)$ is the pth-order Legendre polynomial defined by:

$$P_p(x) = \sum_{k=0}^p \left\{ \frac{(-1)^{\frac{p-k}{2}} (p+k)! x^k}{2^p k! \left(\frac{p-k}{2}\right)! \left(\frac{p+k}{2}\right)!} \right\}_{p-k = \text{even}} \quad (5)$$

In order to increase the computation speed for calculating Legendre polynomials, we used the recurrent formula of the Legendre polynomials defined by:

$$\begin{cases} P_p(x) = \frac{(2p-1)x}{p} P_{p-1}(x) - \frac{(p-1)}{p} P_{p-2}(x) \\ P_1(x) = x, \quad P_0(x) = 1 \end{cases} \quad (6)$$

2.3 Texture Descriptors

This Several images have textured patterns. Therefore, the texture descriptor is used as feature extraction method from the segmented image.

The texture descriptor is extracted using the co-occurrence matrix introduced by Haralick in 1973 [21]. So for a color image I of size $N \times N \times 3$ in a color space (C_1, C_2, C_3) , for $(k, l) \in [1, \dots, N]^2$ and $(a, b) \in [1, \dots, G]^2$, the co-occurrence matrix $M_{k,l}^{C,C'}[I]$ of the two color components $C, C' \in \{C_1, C_2, C_3\}$ from the image I is defined by:

$$M_{k,l}^{C,C'}([I], a, b) = \frac{1}{(N-k)(N-l)} \sum_{i=1}^{N-k} \sum_{j=1}^{N-l} \delta(I(i, j, C) - a, I(i+k, j+l, C') - b) \quad (7)$$

Where δ is the unit pulse defined by:

$$\delta(x, y) = \begin{cases} 1 & \text{if } x = y = 0 \\ 0 & \text{else} \end{cases} \quad (8)$$

Each image I in a color space (C_1, C_2, C_3) can be characterized by six color co-occurrence matrix:

$$M^{C_1, C_1}[I], M^{C_2, C_2}[I], M^{C_3, C_3}[I], M^{C_1, C_2}[I], M^{C_1, C_3}[I], M^{C_2, C_3}[I].$$

Matrix $M^{C_2, C_1}[I]$, $M^{C_3, C_1}[I]$ and $M^{C_3, C_2}[I]$ are not taken into account because they can be deduced respectively by diagonal symmetry from matrix $M^{C_1, C_2}[I]$, $M^{C_1, C_3}[I]$ and $M^{C_2, C_3}[I]$

As they measure local interactions between pixels, they are sensitive to significant differences in spatial resolution between the images. To reduce this sensitivity, it is necessary to normalize these matrices by the total number of the considered co-occurrences matrix:

$$M_{k,l}^{C,C'}([I], a, b) = \frac{M_{k,l}^{C,C'}([I], a, b)}{\sum_{i=0}^{T-1} \sum_{j=0}^{T-1} M_{k,l}^{C,C'}([I], i, j)} \quad (9)$$

Where T is the number of quantization levels of the color components

To reduce the large amount of information of these matrices, the 14 Haralick indices [21] of these matrices are used. There will be then 84 textures attributes for six co-occurrence matrices (14×6) .

3. NEURAL NETWORKS AND BAYESIAN NETWORKS CLASSIFIERS

3.1 Neural Networks

Neural networks (or artificial neural networks) learn by experience, generalize from previous experiences to new ones, and can make decisions [22, 23].

A multilayer neural network consists of an input layer including a set of input nodes, one or more hidden layers of nodes, and an output layer of nodes. Fig.1 shows an example of a three layer network used in this paper, having input layer formed by M nodes, one hidden layer formed by L nodes, and output layer formed by N nodes.

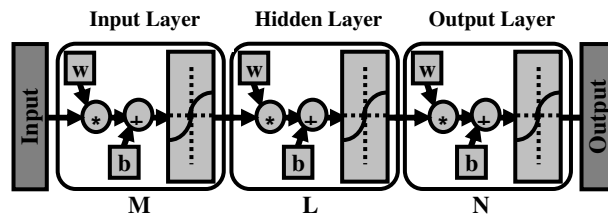


FIGURE 1: The Three Layer Neural Network.

This neural network is trained to classify inputs according to target classes. The training input data are loaded from the reference database while the target data should consist of vectors of all zero values except for a one element, where its index is the class they are to represent. The transfer function used in this tree layer neural network is hyperbolic tangent sigmoid transfer function defined by:

$$f(x) = 2/(1 + \exp(-2x)) - 1 \tag{10}$$

According to authors in [24], the number of neurons in the hidden layer is approximately equal to:

$$L = E(1 + \sqrt{M(N + 2)}) \tag{11}$$

Where:

- $E(x)$ denotes the integer part of x .
- M and N are respectively the number of neurons in the input and output layers.

3.2 Bayesian Networks

The Bayesian networks are based on a probabilistic approach governed by Bayes' rule. The Bayesian approach is then based on the conditional probability that estimates the probability of occurrence of an event assuming that another event is verified. A Bayesian network is a graphical probabilistic model representing the random variable as a directed acyclic graph. It is defined by [25]:

- $G = (X, E)$, Where X is the set of nodes and E is the set of edges, G is a Directed Acyclic Graph (DAG) whose vertices are associated with a set of random variables $X = \{X_1, X_2, \dots, X_n\}$;
- $\theta = \{P(X_i | Pa(X_i))\}$ is a conditional probabilities of each node X_i relative to the state of his parents $Pa(X_i)$ in G .

The graphical part of the Bayesian networks indicates the dependencies between variables and gives a visual representation tool of knowledge more easily understandable by users. Bayesian networks combine qualitative part that are graphs and a quantitative part representing the conditional probabilities associated with each node of the graph with respect to parents [26]. Pearl and all [27] have also shown that Bayesian networks allow to compactly representing the joint probability distribution over all the variables:

$$P(X) = P(X_1, X_2, \dots, X_n) = \prod_{i=1}^n P(X_i | Pa(X_i)) \tag{12}$$

Where $Pa(X_i)$ is the set of parents of node X_i in the graph G of the Bayesian networks.

This joint probability could be actually simplified by the Bayes rule as follows [28]:

$$\begin{aligned}
 P(X) &= P(X_1, X_2, \dots, X_n) = \prod_{i=1}^n P(X_i | Pa(X_i)) \\
 &= P(X_n | X_{n-1}, \dots, X_1) \times P(X_{n-1} | X_{n-2}, \dots, X_1) \times \dots \times P(X_2 | X_1) \times P(X_1) \quad (13) \\
 &= P(X_1) \times \prod_{i=2}^n P(X_i | X_{i-1}, \dots, X_1)
 \end{aligned}$$

The construction of a Bayesian network consists in finding a structure or a graph and estimates its parameters by machine learning. In the case of the classification, the Bayesian network can have a class node C_i and many attribute nodes X_j . The naive Bayes classifier is used in this paper due to its robustness and simplicity. The Fig 2 illustrates its graphical structure.

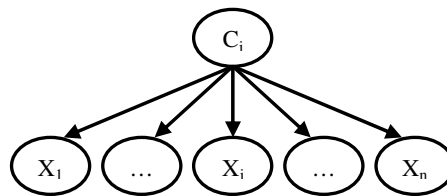


FIGURE 2: Naive Bayes Classifier Structure.

To estimate the Bayesian networks parameters and probabilities, Gaussian distributions are generally used. The conditional distribution of a node relative to its parent is a Gaussian distribution whose mean is a linear combination of the parent's value and whose variance is independent of the parent's value [29]:

$$P(X_i = x_i | Pa(X_i)) = \frac{1}{\sqrt{2\pi\sigma_i^2}} \exp \left\{ -\frac{1}{2\sigma_i^2} \left(x_i - \left(\mu_i + \sum_{j=1}^{n_i} \frac{\sigma_{ij}}{\sigma_j^2} (x_j - \mu_j) \right) \right)^2 \right\} \quad (14)$$

Where,

- $Pa(X_i)$ Are the parents of X_i ;
- μ_i, μ_j, σ_i and σ_j are respectively the means and variances of the attributes X_i and X_j without considering their parents;
- n_i is the number of parents of X_i ;
- σ_{ij} is the regression matrix of weights.

After the parameter and structure learning of a Bayesian networks, The Bayesian inference is used to calculate the probability of any variable in a probabilistic model from the observation of one or more other variables. So, the chosen class C_i is the one that maximizes these probabilities [30]:

$$P(C_i|X) = \begin{cases} P(C_i) \prod_{j=1}^n P(X_j | Pa(X_j), C_i) & \text{if } X_j \text{ has parents .} \\ P(C_i) \prod_{j=1}^n P(X_j | C_i) & \text{else .} \end{cases} \quad (15)$$

For the naive Bayes classifier, the absence of parents and the variables independence assumption are used to write the posterior probability of each class as given in the following equation [31]:

$$P(C_i|X) = P(C_i) \prod_{j=1}^n P(X_j | C_i) \quad (16)$$

Therefore, the decision rule d of an attribute X is given by:

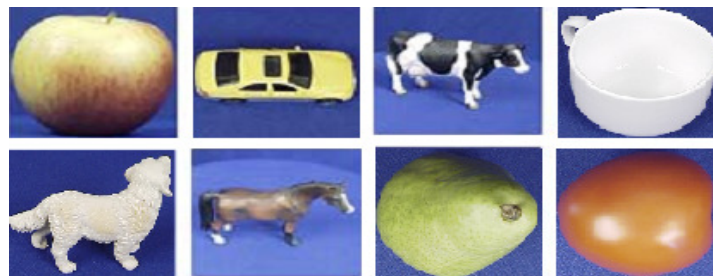
$$d(X) = \arg \max_{C_i} P(C_i|X) = \arg \max_{C_i} P(X|C_i) P(C_i) = \arg \max_{C_i} P(C_i) \prod_{j=1}^n P(X_j | C_i) \quad (17)$$

The class with maximum probability leads to the suitable keyword for the input image.

4. EXPERIMENTS AND RESULTS

In this section, we study and compare the performance of discriminative and generative classifiers for automatic image annotation using in first time each classifier alone and in second time the combination of the two different classifiers [31].

In order to achieve this goal, we conduct two experiments on three image databases ETH-80 [32], COL-100 [33] and NATURE created in this work. The Fig.3 shows some examples of image objects from these three image databases used in our experiments.



ETH-80



COIL-100



FIGURE 3: Some objects images from ETH-80, COL-100 and NATURE databases.

In the phase of learning and classification, we used a training set of 40 images and a test set of 40 images for each image databases.

In all experiments, the features described in Section 2 are extracted after image segmentation by region growing. For each region that represent an object, 10 components of Legendre moments (L00, L01, L02, L03, L10, L11, L12, L20, L21, L30) and 16 elements for RGB color histograms are extracted from each color plane namely R, G and B. The number of input features extracted using Texture extraction method is 14 Haralick indices multiplied by 6 co-occurrence matrices. This gives 84 textures attributes.

4.1 Experiment 1

In this experience, we provide comparative results of image annotation between the two classifiers: discriminative (neural networks) and generative (Bayesian networks). The experimental method adopted in this experience is represented by the figure 4.

In first time, we have used three neural networks classifiers to annotate images of all databases. Each neural networks, receiving as input one of the three extracted descriptors, votes for a given keyword. The keyword that has the maximum of votes is considered as the proper keyword for the annotation of an object in a query image.

In second time, we repeated the same operation with Bayesian networks classifier as shown in figure 4.

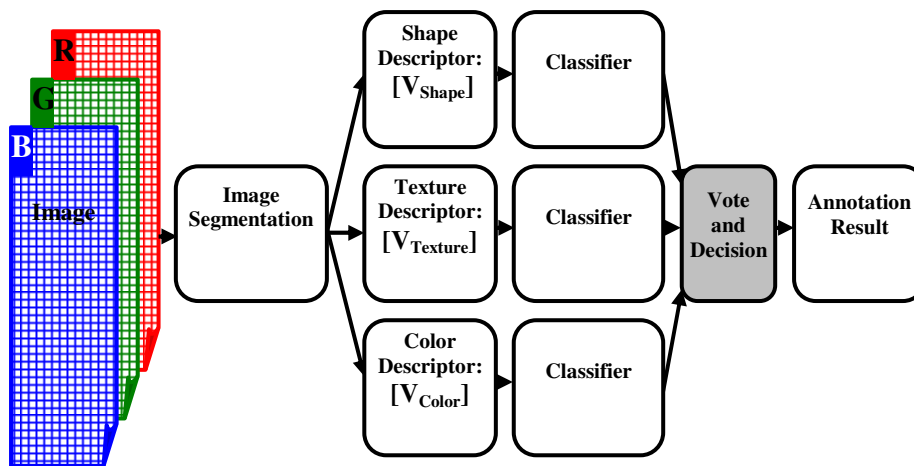


FIGURE 4: Experimental method adopted for image annotation.

4.1.1 Results

Table I summarizes the results of automatic image annotation for each type of classifier and Figures 5,6,7,8, 9 and 10 shows the confusion matrix.

Database	Classification Approach	Average Annotation Rate	Error Rate
ETH-80	neural networks	87.50%	12.50%
	Bayesian networks	90.00%	10.00%
COIL-100	neural networks	82.50%	17.50%
	Bayesian networks	85.00%	15.00%
NATURE	neural networks	90.00%	10.00%
	Bayesian networks	93.33%	6.77%

TABLE 1: Average annotation rate and error rate.

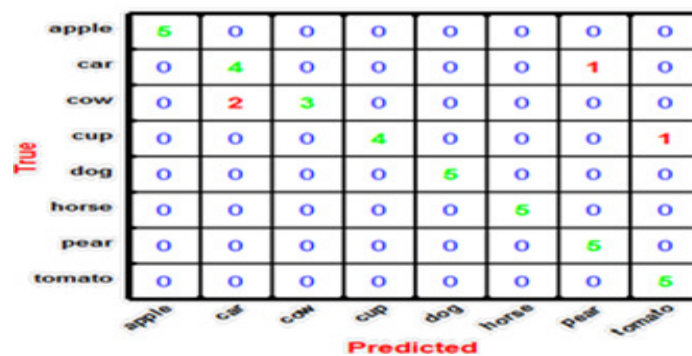


FIGURE 5: Confusion matrix for images of database ETH-80 by using Bayesian networks.

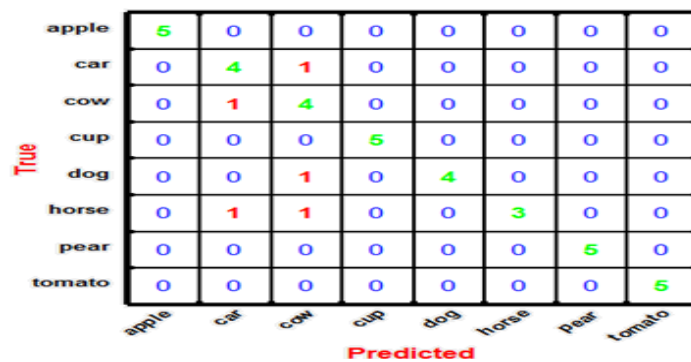


FIGURE 6: Confusion matrix for images of database ETH-80 by using neural networks.

water	5	0	0	0	0	0
sky	0	5	0	0	0	0
sahara	0	1	4	0	0	0
ground	0	0	0	5	0	0
gazon	0	1	0	0	4	0
forest	0	0	0	0	0	5
	water	sky	sahara	ground	gazon	forest

Predicted

FIGURE 7: Confusion matrix for images of database NATURE by using Bayesian networks.

forest	5	0	0	0	0	0
gazon	0	5	0	0	0	0
ground	0	0	5	0	0	0
sahara	0	0	0	5	0	0
sky	0	1	0	0	4	0
water	1	1	0	0	0	3
	forest	gazon	ground	sahara	sky	water

Predicted

FIGURE 8: Confusion matrix for images of database NATURE by using neural networks.

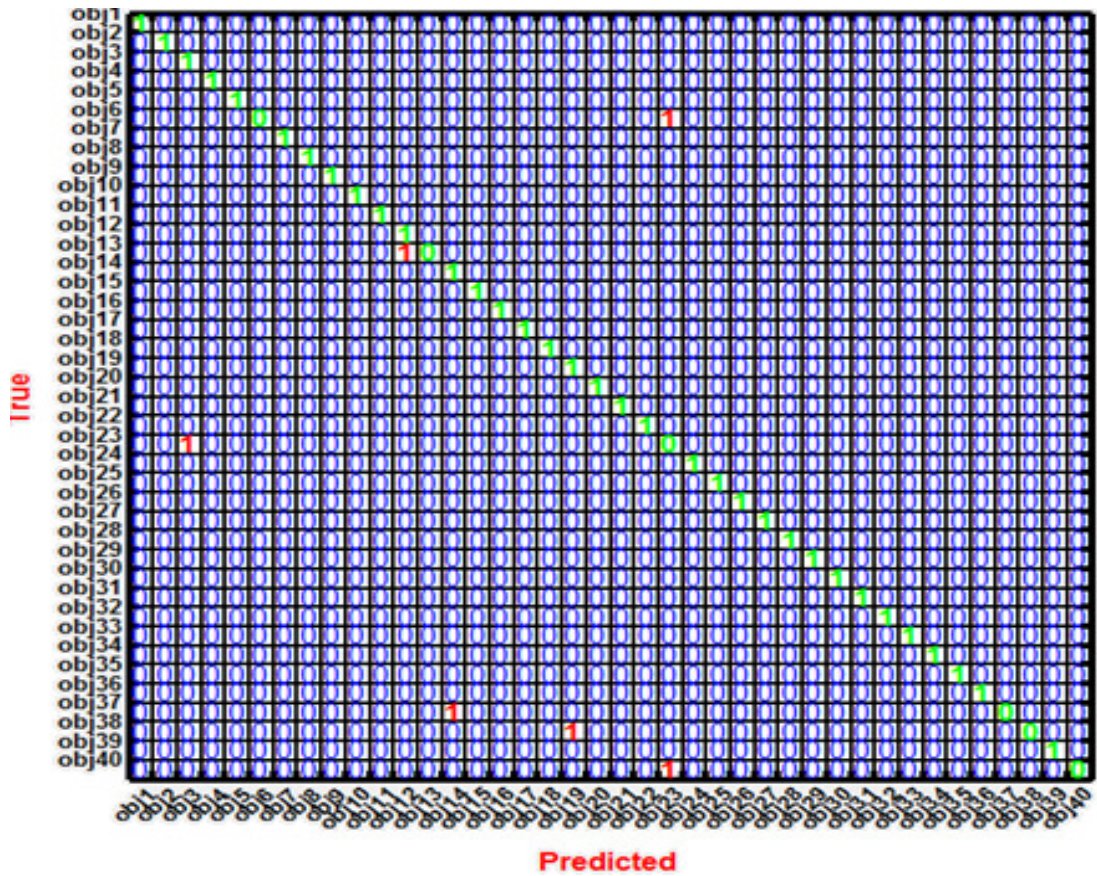


FIGURE 9: Confusion matrix for images of database COIL-100 by using Bayesian networks.

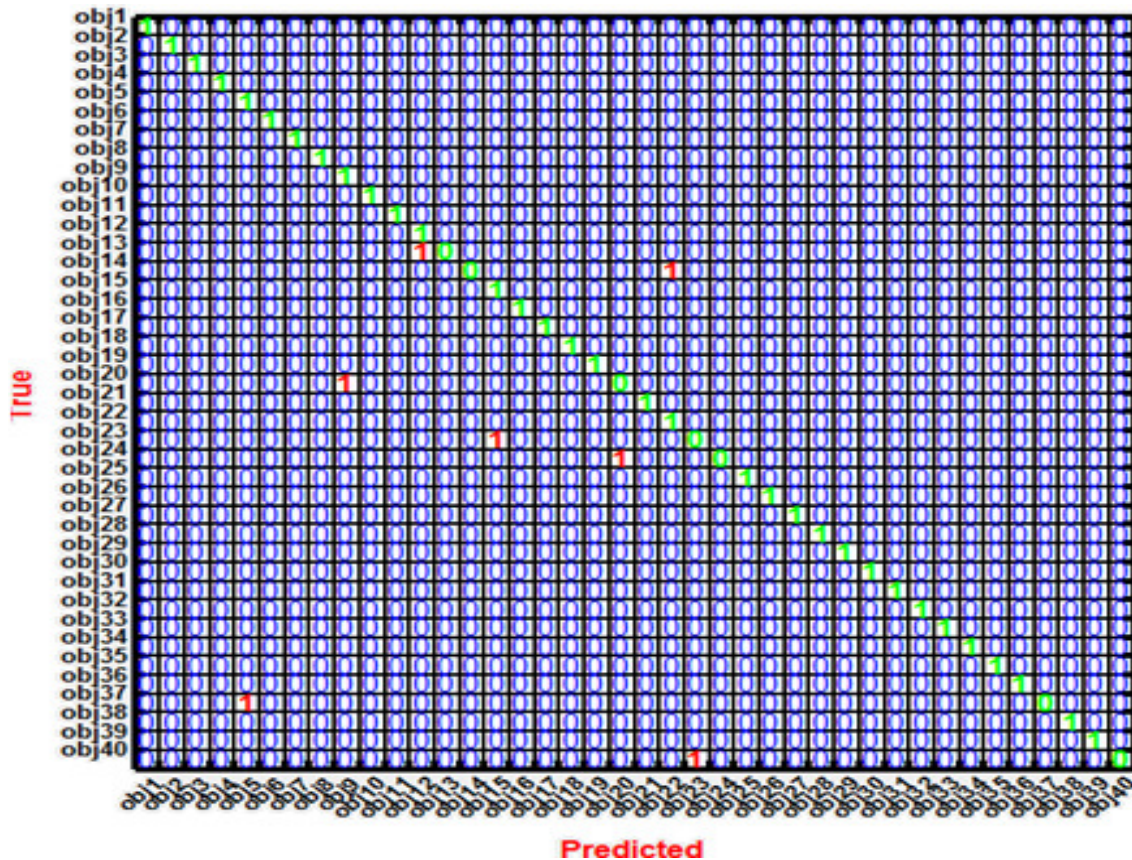


FIGURE 10: Confusion matrix for images of database COIL-100 by using neural networks.

4.2 Analysis of Results

As can be observed from Table 1, Bayesian networks produce the better average annotation rates for all the tree images databases. However, analysis of confusion matrix presented by the Figures 5,6,7,8, 9, 10 shows that the individual annotation rate obtained for some objects (cow, cup, object 6, Sahara and Gazon) with neural networks can be better than those obtained with Bayesian networks. So it appears from these remarks that the combination of these two classifiers will improve the average annotation rates. This constitutes the aim of the experiment 2.

4.3 Experiment 2

Based on the remarks released in the previous two experiments, we combined in this experiment, in addition to descriptors, neural networks and Bayesian networks in order to gain the benefit of the complementarity of these two approaches of classification (discriminative and generative). The principle of this combination is illustrated by the block diagram shown in Fig 11. Thus, with the combination of the three types of descriptors described in Section 2 and the 2 considered types of classifiers, there will be a maximum of votes equal to $3 \times 2 = 6$. Each classifier with each descriptor votes for a given keyword. The keyword with a maximum of votes will be deemed as the proper keyword for the annotation of an object contained in a query image.

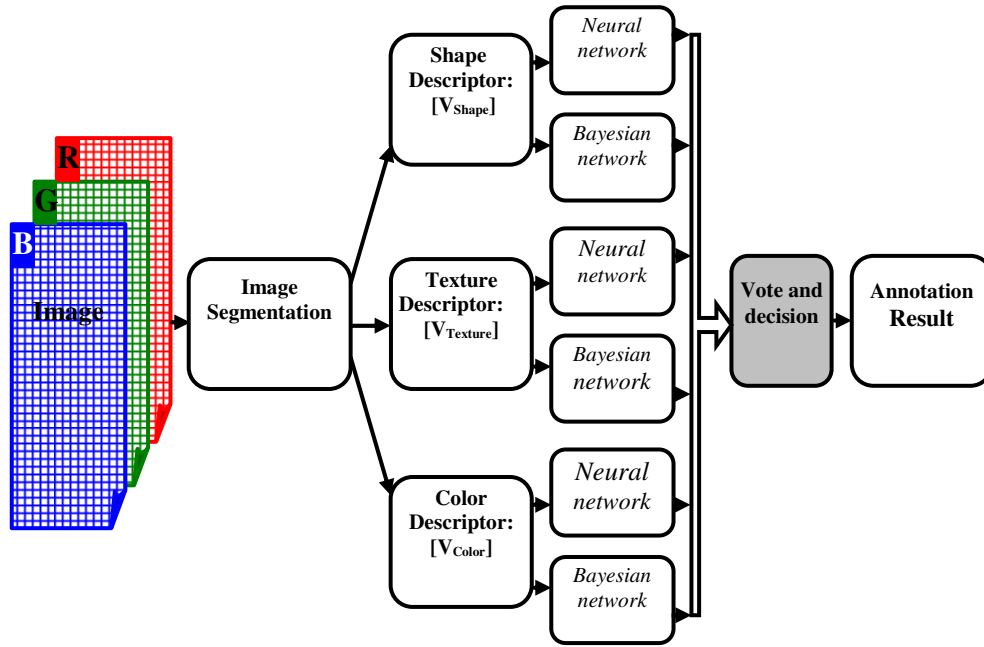


FIGURE 11: Block diagram that illustrates principle of combining discriminative and generative classifiers for automatic image annotation.

4.4 Results

Table 2 shows the average image annotation rate obtained by combining neural networks and Bayesian network classifiers and Figures 12, 13 and 14 shows the confusion matrix.

Database	Average Annotation Rate	Error Rate
ETH-80	92.50%	7.50%
COIL-100	87.50%	12.50%
NATURE	96.67%	3.33%

TABLE 2: Average annotation rate and error rate.

True	apple	5	0	0	0	0	0	0	
	car	0	4	1	0	0	0	0	
	cow	0	2	3	0	0	0	0	
	cup	0	0	0	5	0	0	0	
	dog	0	0	0	0	5	0	0	
	horse	0	0	0	0	0	5	0	
	pear	0	0	0	0	0	0	5	
	tomato	0	0	0	0	0	0	0	5
		Predicted							

FIGURE 12: Confusion matrix for images of database ETH-80.

True	forest	5	0	0	0	0	
	gazon	0	5	0	0	0	
	ground	0	0	5	0	0	
	sahara	0	0	0	5	0	
	sky	0	0	0	0	5	
	water	0	0	1	0	0	4
		for est	gazon	ground	sahara	sky	water
		Predicted					

FIGURE 13: Confusion matrix for images of database NATURE.

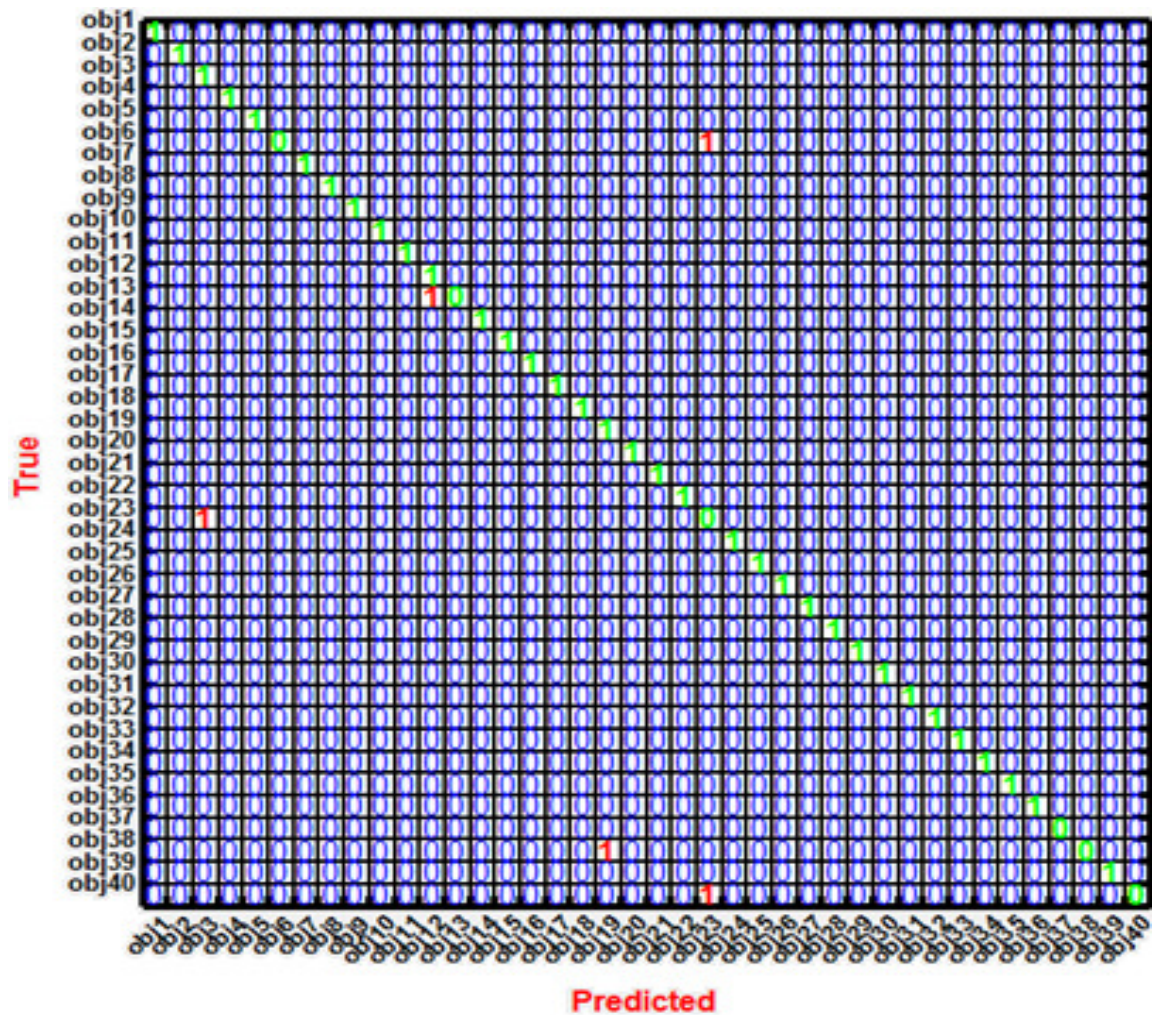
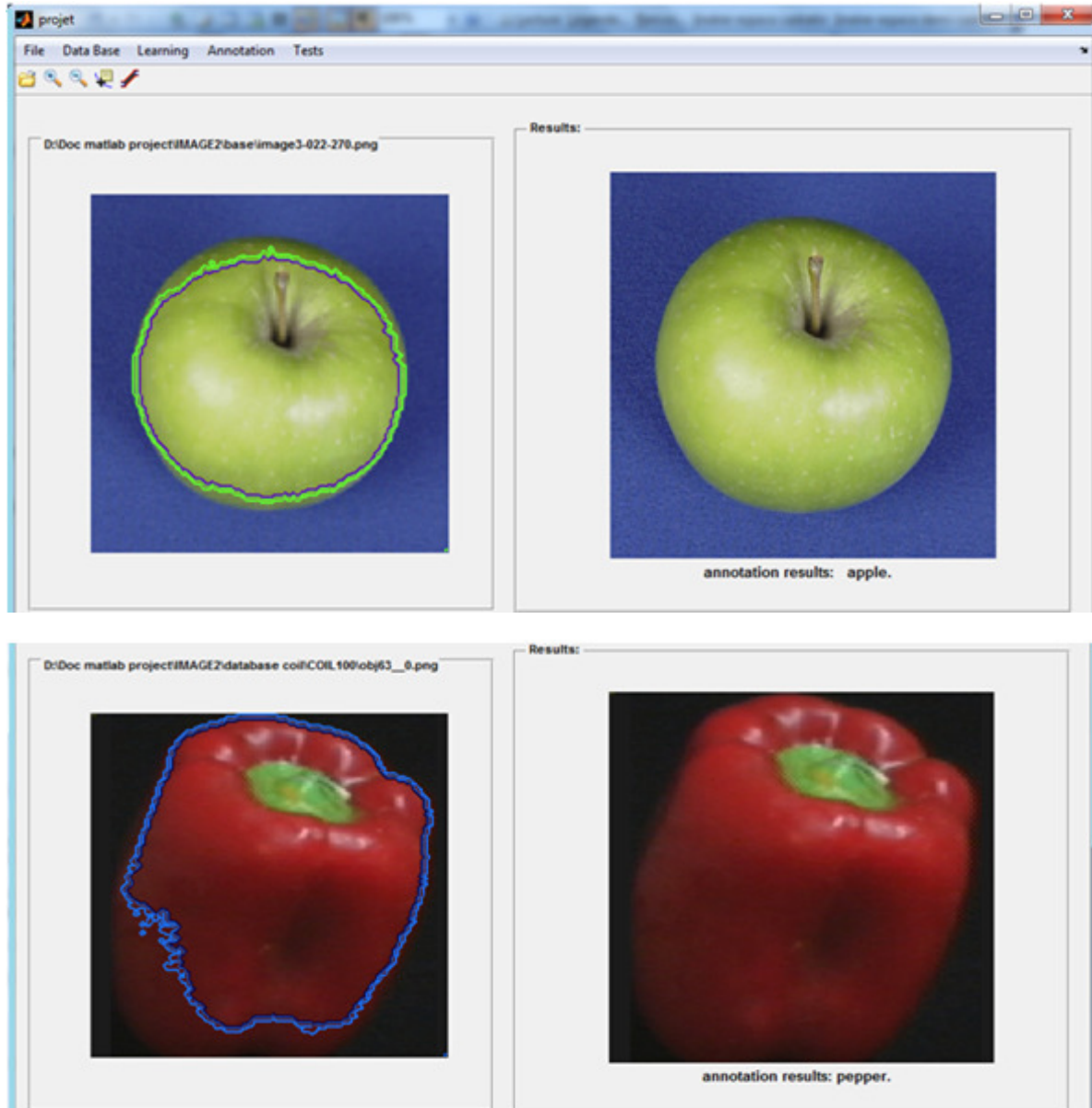


FIGURE 14: Confusion matrix for images of database COIL-100.

4.5 Analysis of Results

Analysis of the results presented in Table 2, Figures 12, 13 and 14, allows us to notice that the combination of neural networks with Bayesian networks in a parallel scheme, has significantly improved the quality of image annotation. Although, some errors are still persistent, namely in particular, the confusion between car and Cow in some times. This result is also illustrated by the examples of annotated images presented by figures 15 and 16 which shows that the exploitation of complementarities of generative and discriminative classifiers can contribute to the improvement of the image annotation. So, it would be interesting to investigate other ways to combine these two different classification approaches to possibly correct the observed annotation errors.



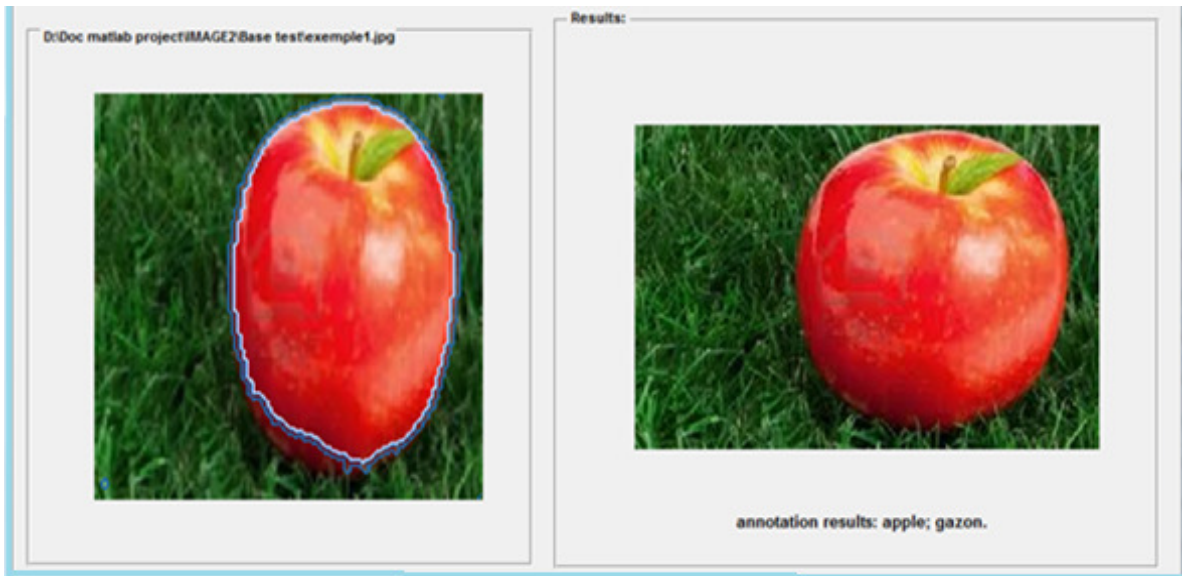


FIGURE 15: Examples of annotated images.

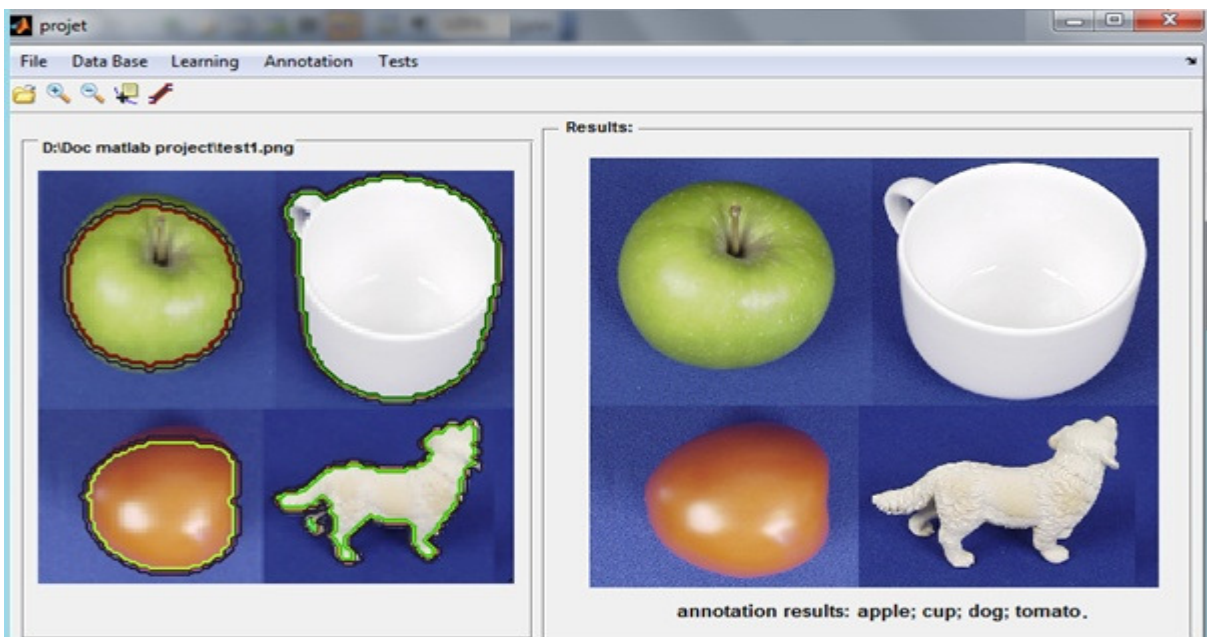




FIGURE 16: Examples of annotated images

5. CONCLUSION AND FUTURE WORK

In this work, we have proposed to build an efficient classifier for automatic image annotation via combining generative and discriminative classifiers which are respectively Bayesian networks and neural networks.

Starting with comparing these classifiers by realizing experiments on three image dataset, we have observed that neither classifier alone will be sufficient for semantic image annotation. So, we have combined the generative and discriminative classifier in parallel scheme in order to join and exploit their strengths. Experimental results show that this approach is promising for automatic image annotation because it gives better classification accuracy than either Bayesian networks or neural networks alone.

Our investigations suggest that the most fruitful approaches will involve some combination of generative and discriminative models. A principled approach to combining generative and discriminative approaches not only gives a more satisfying foundation for the development of new

models, but it also brings practical benefits, address the extreme data-ambiguity and overfitting vulnerability issues in tasks such as automatic image annotation (AIA). In future work, we would like to develop others hybrid schemes that sought to integrate the intra-class information from generative models and the complementary inter-class information from discriminative models, and to research alternative optimization techniques utilizing ideas from the multi-criteria optimization of literature.

6. REFERENCES

- [1] Li Z, Z. Shi P, Liu X and Shi Z, (2010) Automatic Image Annotation with Continuous PLSA, Proceedings of the 35th IEEE Intern Conf on Acoustics, Speech and Signal Processing, pp.: 806-809.
- [2] Carneiro G, Chan A, Moreno P, et al. (2007) Supervised Learning of Semantic Classes for Image Annotation and Retrieval, IEEE Transactions on Pattern Analysis and Machine Intelligence, 29(3), pp.:394-410.
- [3] Jianping Fan .Yuli Gao .Hangzai Luo (2007) Hierarchical Classification for Automatic Image Annotation. SIGIR Proceedings.
- [4] Zhang R, Zhang Z, Li M, *et al.*, (2005) A Probabilistic Semantic Model for Image Annotation and Multi-Model Image Retrieval, *Proc of the 10th IEEE Inter Conf on Computer Vision*, pp.:846- 851.
- [5] Lavrenko V., Manmatha R., and Jeon J (2003) A model for learning the semantics of pictures. In Proc of Advances in Neural Information Processing Systems, pp.:251–259.
- [6] Oksana Yakhnenko (2009) Learning from Text and Images: Generative and Discriminative Models for Partially Labeled Data. Thesis, Iowa State University Ames.
- [7] A. Y. Ng and A, Jordan M (2001) On discriminative vs. generative classifiers: A comparison of logistic regression and naïve Bayes. In *Neural Information Processing Systems*, pp.: 841–848.
- [8] lex Holub, Max Welling, Pietro Perona (2008) Hybrid Generative-Discriminative Visual Categorization. Inter Jour of Computer Vision, 77(3), pp.: 239-258.
- [9] Ilkay Ulusoy1 , Bishop M (2006) Comparison of Generative and Discriminative Techniques for Object Detection and Classification: toward Category-Level Object Recognition, springer, pp.: 173-195.
- [10] Lasserre J, Bishop C, Minka P. (2006) Principled hybrids of generative and discriminative models. Proc of the IEEE Computer Society Conf on Computer Vision and Pattern Recognition (CVPR), pp.:87–94.
- [11] Timothy M, Shaogang G, and Xiang T. (2013) Finding Rare Classes: Active Learning with Generative and Discriminative Models. IEEE transactions on knowledge and data engineering 25 (2), pp.: 374 – 386.
- [12] Cristani A, Castellani U, Murino V (2009) A hybrid generative/discriminative classification framework based on free energy terms. In ICCV.
- [13] Anna B, Andrew Z, Xavier M (2008) Scene Classification Using a Hybrid Generative/Discriminative Approach. IEEE transactions on pattern analysis and machine intelligence, 30 (4), pp.: 712-727.
- [14] Kelm M, Pal C, McCallum A (2006) Combining generative and discriminative methods for pixel classification with multi-conditional learning. ICPR, pp :828–832.

- [15] Guillaume Bouchard and Bill Triggs (2004) The trade-off between generative and discriminative classifiers. *proc of Computational Statistics Symposium*, Physica-Verlag, Springer.
- [16] Zhixin Li¹, Zhenjun Tang¹, Weizhong Zhao², Zhiqing Li² (2012) Combining Generative/Discriminative Learning for Automatic Image Annotation and Retrieval. *Inter J of Intelligence Science*, pp.:55-62.
- [17] Shuang Hong Yang Jiang Bian College Hongyuan Zha (2010) Hybrid Generative/Discriminative Learning for Automatic Image Annotation. *Proc Uncertainly artificial intelligence (UAI)*, pp.: 683-690.
- [18] Frank Y , Shouxian C (2005) Automatic seeded region growing for color image segmentation . *Image and Vision Computing* 23, pp.:877–886.
- [19] Ryszard S, Chora (2007) Image Feature Extraction Techniques and Their Applications for CBIR and Biometrics Systems, *Inter J of Biology And Biomedical Engineering*,1(1)1, pp.:6-16.
- [20] Chee Way Chonga, Raveendranb P, Mukundan R, (2004) Translation and scale invariants of Legendre moments, *Pattern Recognition* 37, pp.:119 – 129.
- [21] Haralick R, Shanmugan K, Dinstein I (1973) Textural features for image classification. *IEEE Transactions on SMC*, 3(6), pp. :610–621.
- [22] Yue Cao, Xiabi Liu, Jie Bing, Li Song (2011) Using Neural Network to Combine Measures of Word Semantic Similarity for Image Annotation, *IEEE International Conference on Information and Automation (ICIA)*, pp. :833 – 837.
- [23] Simard P, Steinkraus D, Platt J (2003) Best Practices for Convolutional Neural Networks Applied to Visual Document Analysis, *ICDAR*, pp. :958-962.
- [24] R. Lepage, & B. Solaiman. *Les réseaux de neurones artificiels et leurs applications en imagerie et en vision par ordinateur*, Ecole de technologie supérieure, 2003.
- [25] Ann.Becker, Patrick Naim (1999) *les réseaux bayésiens : modèles graphiques de connaissance*. Eyrolles.
- [26] Pearl J (1995) *Bayesian Networks*. UCLA Cognitive Systems Laboratory, Technical Report (R-216), MIT Press, pp.:149-153.
- [27] Sabine Barrat (2009) *Modèles graphiques probabilistes pour la reconnaissance de formes*, Thèse, Spécialité informatique, Université Nancy 2.
- [28] George H, Pat Langley (1995) Estimating continuous distributions in Bayesian classifiers. *The Eleventh Conference on Uncertainty in Artificial Intelligence*.
- [29] Philippe LERAY (2006) *Réseaux bayésiens : apprentissage et modélisation de systèmes complexes*. Habilitation à diriger les recherches, Spécialité Informatique, Automatique et Traitement du Signal, Université de Rouen, France.
- [30] Patrick Naïm, Pierre Henri Wuillemin, Philippe Leray, Olivier pourret, Anna becker, (2008) *Réseaux bayésiens*, Eyrolles, 3ème édition, Paris.
- [31] Mitchell T (2010) Generative and discriminative classifier: Naïve bayes and logistic regression. *Machine learning*.
- [32] ETH-80 database image. Online. Available: <http://www.d2.mpi-inf.mpg.de/Datasets/ETH80>.

[33] COIL-100 database image. Online. Available:
<http://www.cs.columbia.edu/CAVE/software/softlib/coil-100.php>.

Header Based Classification of Journals Using Document Image Segmentation and Extreme Learning Machine

Kalpana S

*Research Scholar
PSGR Krishnammal College for Women
Coimbatore, India.*

kalpana.msccs@gmail.com

Vijaya MS

*Associate Professor
PSGR Krishnammal College for Women
Coimbatore, India.*

msvijaya@grgsact.com

Abstract

Document image segmentation plays an important role in classification of journals, magazines, newspaper, etc., It is a process of splitting the document into distinct regions. Document layout analysis is a key process of identifying and categorizing the regions of interest in the scanned image of a text document. A reading system requires the segmentation of text zones from non-textual ones and the arrangement in their correct reading order. Detection and labelling of text zones play different logical roles inside the document such as titles, captions, footnotes, etc. This research work proposes a new approach to segment the document and classify the journals based on the header block. Documents are collected from different journals and used as input image. The image is segmented into blocks like heading, header, author name and footer using Particle Swarm optimization algorithm and features are extracted from header block using Gray Level Co-occurrences Matrix. Extreme Learning Machine has been used for classification based on the header blocks and obtained 82.3% accuracy.

Keywords: Classification, Document Segmentation, Feature Extraction, Extreme Learning Machine.

1. INTRODUCTION

In computer vision, document layout analysis is the process of identifying and categorizing the regions of interest in the scanned image of a text document. Document image segmentation is a process of subdividing the document into distinct regions or blocks. It is important process in the document analysis. Document segmentation is a fundamental step in document processing, which aims at identifying the relevant components in the document that deserve further and specialized processing. Document analysis consists of geometric and logical analysis. In geometric based segmentation, the document is segmented upon its geometric structure such as text and non-text regions. Whereas in logical segmentation the document is segmented upon its logical labels assigned to each region of the document such as title, logo, footnote, caption, etc., [1]. The geometric layout analysis is also called as physical layout analysis. The physical layout of a document refers to the physical location and boundaries of various regions in the document image.

The process of document layout analysis aims to decompose a document image into a hierarchy of homogenous regions such as figures, backgrounds, text blocks, text lines, words, characters, etc., Logical structure is the result of dividing and subdividing the content of a document into increasingly smaller parts on the basis of the human-perceptible meaning of the content [2]. A logical object is an element of the specific logical structure of a document. For logical objects no classification other than basic logical objects, composite logical objects and document logical

root. The logical objects, which are the subject of extraction in the proposed method, are roughly categorized into the following headlines, headers, footers, captions, notes, and programs, titles, paragraphs, lists, and formulas.

Document layout analysis algorithms can be categorized into three approaches namely top-down approaches, bottom-up approaches and hybrid approaches. Top-down algorithms start from the whole document image and iteratively split it into smaller ranges. Bottom-up algorithms start from document image pixels, and cluster the pixels into connected components such as characters which are then clustered into words, lines or zones. Hybrid algorithms can be regarded as a mix of the above two approaches. The Docstrum algorithm was presented in [3], the Voronoi-diagram-based algorithm was proposed in [4] the run-length smearing algorithm was implemented in [5] and the text string separation algorithm is implemented by [6] are typical bottom-up algorithms. The X – Y cut-based algorithm of [7] and the shape-directed-covers-based algorithm [8] are top-down algorithms. In [9] the author proposed a hybrid algorithm using a split-and-merge strategy. The advantage of using top-down approach is, its high speed processing and the drawback is, it cannot process table, improper layout documents and forms.

This research work proposes the document segmentation based on logical layout. The segmentation of document image is done using Particle Swarm Optimization (PSO). The document image is segmented as header, heading, footer, author name. From the segmented blocks, features are extracted using Gray Level Co-occurrence Matrix (GLCM), which is the statistical method of examining the textures that considers the spatial relationship of the pixels. Features such as Energy, Entropy Autocorrelation, Contrast, Correlation, Cluster Prominence, Cluster Shade, Dissimilarity, Homogeneity, Maximum probability, Variance, Sum average, Sum variance, Sum entropy, Difference variance, Difference entropy, Information measure of correlation1 and correlation2 are computed. Energy and entropy are renowned properties of an image. Energy identifies the uniformity of the image and entropy identifies the randomness of the image. Finally the classification is performed based on header block of the document image using Extreme Learning Machine.

2. PROPOSED MODEL FOR DOCUMENT SEGMENTATION AND CLASSIFICATION

The proposed work aims to segment the document image based on logical layout. For this the documents are collected from five different journals and they are used as the input. First the noise is removed from the given input document image using median filter. The noiseless image is used for segmenting the document using the Particle Swarm optimization (PSO) algorithm and the features are extracted. The features are extracted using Gray Level Co-occurrence Matrix (GLCM). At last, the classification of journals based on the header block is carried out by using Extreme Learning Machine and Support Vector Machine. The overview of the proposed work is shown in Fig.1.

2.1. Pre-processing

Pre-processing is a sequence of tasks performed on the image. It enhances the quality of the image for segmentation. The various tasks performed on the image in pre-processing stage are scanning of documents, binarization and noise removal.

2.1.1. Scanning of Documents

The documents are collected from various journals and only the first page of each document is scanned. They are stored in the database and used as input image.

2.1.2. Binarization

It is a process which converts the gray scale image into a binary image using the global threshold method. A binary image has only two values 0 or 1 for each pixel. 0 represents white pixel and 1 represents black.

2.1.3. Noise Removal

Filters are used to remove the noise in the image or document. The noise is removed using median filter. The segmentation is focused with the noiseless image for best result.

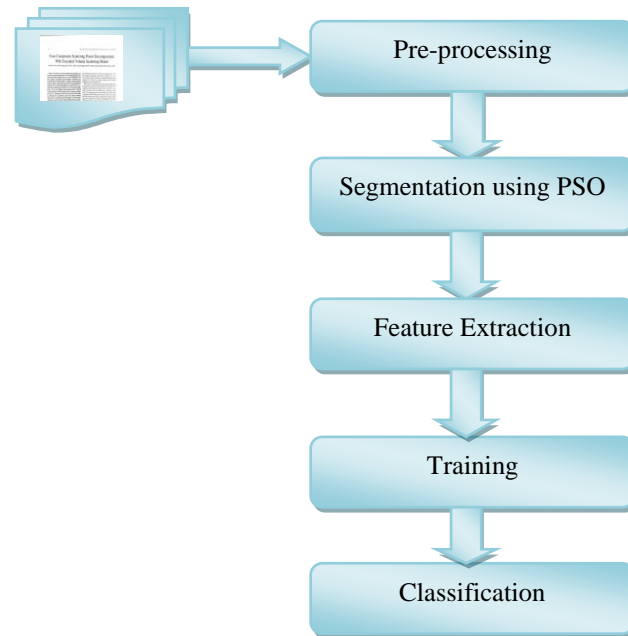


FIGURE 1: Block Diagram of Proposed methodology.

2.2. Segmentation Using Pso

The PSO algorithm is used for segmenting the document. The document is subdivided into blocks like heading, header, author name and footnote. The space between the lines is used to separate the lines. Normally the distances between two lines are larger than the distances between words, thus lines can be segmented by comparing this distance against a suitable threshold. To determine an optimal threshold, Particle Swarm Optimization technique is used. Particle Swarm Optimization (PSO) algorithm is used to solve many of difficult problems in the field of pattern recognition. Hence, PSO is used to compute an optimal value.

Let X and V denote the particle's position and its corresponding velocity in search space respectively. At iteration K , each particle i has its position defined by $X_i^k = (x_{i1}, x_{i2}, \dots, x_{in})$ and a velocity is defined by $V_i^k = (v_{i1}, v_{i2}, \dots, v_{in})$ in search space n . Velocity and position of each particle in next iterations can be calculated using following equation (1) and (2).

$$V_{ij}^{k+1} = wv_{ij}^k + C_1r_1(pbest_{ij}^k - x_{ij}^k) + C_2r_2(gbest_{ij}^k - x_{ij}^k) \quad (1)$$

$$x_{ij}^k = x_g^k + v_g^k \quad (2)$$

Where k is the current iteration number, w is inertia weight, v_{ij} is then updated velocity on the i^{th} dimension of the j^{th} particle, C_1 and C_2 is acceleration constants, C_1 and C_2 is positive constant parameters, usually $C_1 = C_2 = 2$. r_1 and r_2 , are the real numbers drawn from two uniform random sequences of $U(0, 1)$. The algorithm starts by generating randomly initial population of the PSO. Every particle is initialized with locations and velocities using the equations (1) and (2). These locations consist of the initial solutions for the optimal threshold. The procedure of the proposed PSO algorithm is described as follows.

Step 1: Initialize N particles with random positions x_1, x_2, \dots, x_n according to equation (2) and velocities V_i where $i = 1, 2 \dots N$.

Step 2: Evaluate each particle according to equation (4)

$$f(t) = w_0(t) \times w_1(t) \times (\mu_0(t) - \mu_1(t))^2$$

Where, t is a gray level between 0 and 255 which can be obtained through the particle's position

Step 3: Update individual and global best positions. If $f(pbest_i) < f(x_i)$, then $pbest_i = x_i$ search for the maximum value f_{max} among $f(pbest_i)$, if $\max f(gbest) < f_{max}$ then $gbest = x_{max}$, x_{max} is the particle associated with f_{max} .

Step 4: Update velocity: update the i^{th} particle velocity using the equation (2) restricted by maximum and minimum threshold v_{max} and v_{min} .

Step 5: Update Position: update the i^{th} particle position using equation (2) and (3).

Step 6: Repeat step 2 to 5 until a given maximum number of iterations is achieved or the optimal solution so far has not been improved for a given number of iteration.

By performing the above steps the header block is segmented for a document image for the classification of journals is shown in Figure.2.

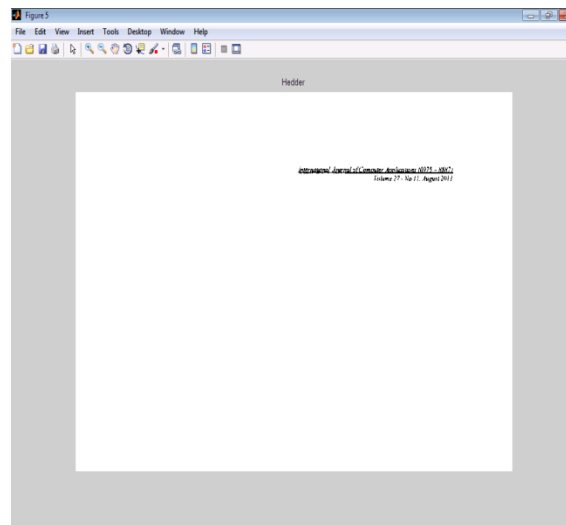


FIGURE 2: Segmented Header.

2.2.1. Labeling Connected Components

With a selection of optimal threshold value, the connected areas will form blocks of the same region. Labeling is the process of identifying the connected components in an image and assigning each component a unique label an integer number which must be same as connected black runs. Figure.3. shows the labeled connected components of document image.

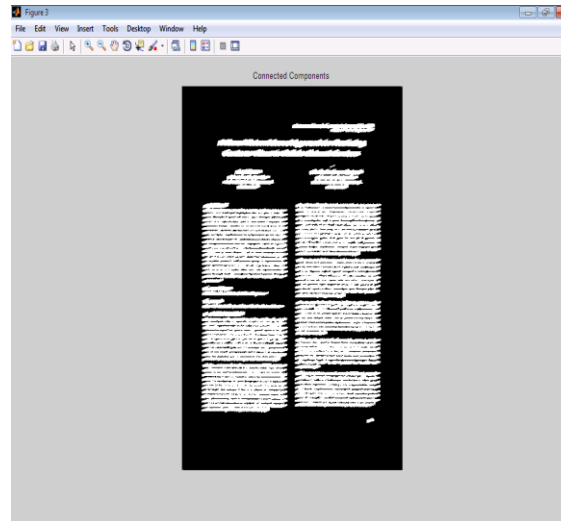


FIGURE 3: Labeling Connected Components.

2.3. Feature Extraction

Features such as Energy, Entropy Autocorrelation, Contrast, Correlation, Cluster Prominence, Cluster Shade, Dissimilarity, Homogeneity, Maximum probability, Variance, Sum average, Sum variance, Sum entropy, Difference variance, Difference entropy, Information measure of correlation1 and correlation2 are computed for classification of document region[10]. Few of the common statistics applied to co-occurrence probabilities are discussed below.

1) Energy

This is also called uniformity or angular second moment. It measures the textural uniformity that is pixel pair repetitions. It detects disorders in textures. Energy reaches a maximum value equal to one.

2) Entropy

This statistic measures the disorder or complexity of an image. The entropy is larger when the image is not texturally uniform and many GLCM elements have very small values. Complex textures tend to have high entropy.

3) Contrast

It measures the spatial frequency of an image and difference moment of GLCM. It is the difference between the highest and the lowest values of a contiguous set of pixels. It measures the amount of local variation present in the image.

4) Variance

It is a measure of heterogeneity and is strongly correlated to first order statistical variable such as standard deviation. Variance increases when the gray level values differ from their mean.

5) Homogeneity

If weights decrease away from the diagonal, the result will be larger for windows with little contrast. Homogeneity weights values by the inverse of the contrast weight, with weights decreasing exponentially away from the diagonal.

6) Correlation

The correlation feature is a measure of gray tone linear dependencies in the image. GLCM correlation is quite a different calculation from the other texture measures. It also has a more intuitive meaning to the actual calculated values: 0 is uncorrelated, 1 is perfectly correlated.

7) Autocorrelation

An autocorrelation function can be evaluated that measures the coarseness. This function evaluates the linear spatial relationships between primitives. If the primitives are large, the function decreases slowly with increasing distance whereas it decreases rapidly if texture consists of small primitives. However, if the primitives are periodic, then the autocorrelation increases and decreases periodically with distance.

The rest of the textural features are secondary and derived from those listed above.

8) Sum Variance

$$\text{sum variance (sv)} = \sum_{i=2}^{2N_g} (i - sa)^2 g_{x+y}(i)$$

9) Difference variance

$$\text{difference variance} = \text{variance of } g_{x-y}$$

10) Sum Average

$$\text{sum average (sa)} = \sum_{i=2}^{2N_g} i g_{x+y}(i)$$

11) Sum Entropy

$$\text{sum entropy (se)} = - \sum_{i=2}^{2N_g} i g_{x+y}(i) \log\{g_{x+y}(i)\}$$

12) Difference Entropy

$$\text{difference entropy} = - \sum_{i=0}^{N_g-1} g_{x-y}(i) \log\{g_{x-y}(i)\}$$

13) Information Measures of Correlation

i) Information Measures of Correlation 1 (IMC1)

$$\text{IMC1} = \frac{\text{HXY} - \text{HXY1}}{\max\{\text{HX}, \text{HY}\}}$$

ii) Information Measures of Correlation 2 (IMC2)

$$\text{IMC2} = \sqrt{(1 - \exp[-2.0(\text{HXY2} - \text{HXY})])}$$

14) Cluster shade

$$\text{Shade} = \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} \{i + j - \mu_x - \mu_y\}^3 * P(i, j)$$

15) Cluster Prominence

$$\text{Prom} = \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} \{i + j - \mu_x - \mu_y\}^4 * P(i, j)$$

16) Dissimilarity

$$\text{Diss} = \sum P_{i,j} * |i - j|$$

Dissimilarity is a measure that defines the variation of grey level pairs in an image. It is the closest to contrast with a difference in the weight.

3. EXTREME LEARNING MACHINE

Extreme Learning Machine (ELM) is a new learning algorithm for Single-hidden Layer Feed forward neural Networks (SLFNs) supervised batch learning which provides good generalization performance for both classification and regression problems at highly fast learning speed. The output function of the generalized SLFN is given by,

$$F(x) = \sum_{i=1}^L \beta_i h_i(x)$$

Where $h_i(x)$ is the output of the i^{th} hidden- node. The ELM algorithm which consists of three steps that are: Given a training set $N = \{(x_i, t_i), x_i \in R^n, t_i \in R^m, i = 1, \dots, N\}$, kernel function $f(x)$, and hidden neuron \tilde{N} . The ELM algorithm which consists of following steps:

Step 1: Select suitable activation function and number of hidden neurons \tilde{N} for the given problem.

Step 2: Assign arbitrary input weight w_i and bias $b_i, i = 1, \dots, H$

Step 3: Calculate the output matrix H at the hidden layer

$$H = f.(w. + x + b)$$

Step 4: Calculate the output weight β

$$\hat{\beta} = H^{-1}T$$

In kernel based ELM, If the hidden layer feature mapping $h(x)$ is unknown to users, users can be described a kernel function for ELM. ELM Kernel function is given by, $\text{KELM}(x_i, x_j) = 1/H f(x_i).f(x_j)$. That is, the data has feed through the ELM hidden layer to obtain the feature space vectors, and their co-variance is then calculated and scaled by the number of hidden units. The main difference is that where ELM explicitly generates the feature space vectors, but in SVM or another kernel method only similarities between feature space vectors are used. The entire above mentioned can be used to apply in regression, binary and multi-label classification applications directly. Kernel ELMs can be applied to complex space as well.

4. EXPERIMENTS AND RESULTS

The documents used for creating the dataset are collected from various journals like IEEE, IJCA, IJDKP, International Journal of Advances in Image Processing and European International Journal of Science and Technology. The dataset consists of 76 document images and in that 59 is used for training and remaining for testing. In the first phase each instance is segmented into four blocks as heading, header, footer and author name. After the segmentation, for each header

block 19 features are extracted using GLCM. The dataset is then trained using ELM based on the header block for the classification of journals and it is compared with SVM classifier for predictive accuracy.

The prediction accuracy and learning time of the ELM is observed. The function `elm_train` is used to train the model by identifying `elm` type, number of hidden neurons and activation function as parameters. The `elm_predict` function is used to test the model as `[output] = elm_predict (TestingData_File)`. To calculate the accuracy the whole testing data is used. Based on the accuracy and the learning time the performance evaluation of the proposed work is obtained. The classification result of ELM gives the list of document headings in the specific folder based on the header block of the input image. The accuracy of ELM is evaluated using the following formula and achieved 82.3% accuracy.

$$\text{Accuracy} = \frac{\text{Number of correctly recognized image}}{\text{Total number of images in test database}} * 100$$

To compare with ELM the second experiment is carried out using the same dataset and the classification algorithm SVM is trained using the same dataset to create the classifier. The accuracy of the SVM classifier is tested using the same test dataset and the classification results are obtained as 64.7% accuracy. It is observed from the results that the performance of the model built based on Extreme Learning Machine for classification of segmented document image is more accurate and fast compared to Support Vector Machine. Comparative results of Support Vector Machine and Extreme Learning Machine are summarized in Table I. The comparative results in terms of accuracy and learning time of both classifiers are shown in Figure.4 and Figure.5 respectively.

Classifiers	Learning Time (seconds)	Accuracy (%)
SVM	15.29	64.7
ELM	13.35	82.3

TABLE 1: Comparative Results of the Classifiers.

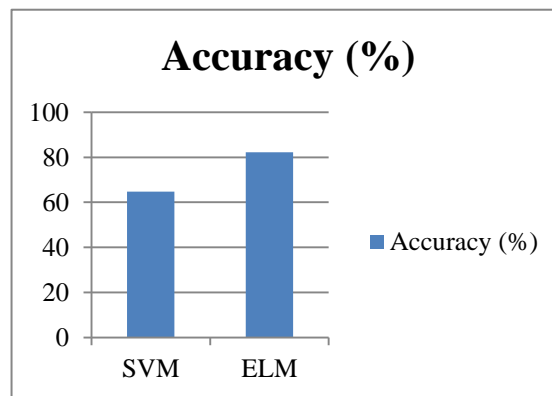


FIGURE 4: Comparison - Accuracy of Classifiers.

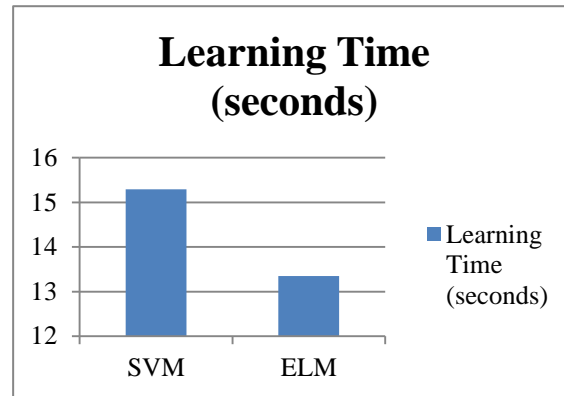


FIGURE 5: Comparison - Learning Time of Classifiers.

5. CONCLUSION

This paper demonstrates the modeling of document segmentation and classification task that describes the implementation of machine learning approach for segmenting the document into various regions. The corpus is created by collecting the documents from five different journals and stored in the database. These documents are pre-processed to remove the noise using median filter. The pre-processed documents are segmented into various blocks such as heading, header, author name and footer using Particle Swarm Optimization algorithm. From each header block the features are extracted and the training dataset is created. Finally classification based on header blocks is done using supervised classification algorithms namely ELM and SVM. The performance of both classifiers is evaluated in terms of accuracy and learning time. It has been observed that ELM technique shows better performance than SVM technique for document image classification. Future work of segmentation can be extended by detecting images, postal codes, handwritten and printed documents with more features.

6. REFERENCES

- [1] Okun O. Doermann D and M. Pietikainen. "Page segmentation and zone classification". The state of the art. In UMD, 1999.
- [2] Yuan. Y. Tang and M. Cheriet, Jiming Liu, J.N Said, "Document Analysis and recognition by computers".
- [3] L. O. Gorman, "The document spectrum for page layout analysis," IEEE Transactions on Pattern Analysis and Machine Intelligence 15, pp. 1162–1173, 1993.
- [4] K. Kise, A. Sato, and M. Iwata, "Segmentation of page images using the area Voronoi diagram," Computer Vision and Image Understanding 70, pp. 370–382, 1998.
- [5] Wahl. K. Wong, and R. Casey, "Block segmentation and text extraction in mixed text/image documents," Graphical Models and Image Processing 20, pp. 375–390, 1982.
- [6] L. A. Fletcher and R. Kasturi, "A robust algorithm for text string separation from mixed text/graphics images," IEEE Transactions on Pattern Analysis and Machine Intelligence 10, pp. 910–918, 1988.
- [7] Nagy, S. Seth, and M. Viswanathan, "A prototype document image analysis system for technical journals," Computer 25, pp. 10–22, 1992.
- [8] S. Baird, S. E. Jones, and S. J. Fortune, "Image segmentation by shape-directed covers," in Proceedings of International Conference on Pattern Recognition, pp. 820–825, (Atlantic City, NJ), June 1990.
- [9] T. Pavlidis and J. Zhou, "Page segmentation and classification," Graphical Models and Image Processing 54, pp. 484–496, 1992.

- [10] Haralick R.M., Shanmugam K., Dinstein I., "Textural Features for Image Classification", IEEE Trans. on System Man and Cybernetics, 1973, 3(6), p.610-621.
- [11] Santanu Chaudhury, Megha Jindal, and Sumantra Dutta Roy, "Model-Guided Segmentation and Layout Labeling of Document Images using a Hierarchical Conditional Random Field", New Delhi, India.
- [12] Jianying Hu, Ramanujan Kashi, Gordon Wilfong, "Document Classification using Layout Analysis", USA.
- [13] Gerd Maderlechner, Angela Schreyer and Peter Suda, "Information Extraction from Document Images using Attention Based Layout Segmentation", Germany.
- [14] Y. Ishitani. Document layout analysis based on emergent computation. Proc. 4th ICDAR, 1:45–50, 1997.
- [15] K. T. Spoehr. Visual information processing. W. H. Freeman and Company, 1982.
- [16] Robert M. Haralick,"Document image Understanding: Geometric and Logical layout", University of Washington, Seattle.
- [17] ISO: 8613: Information Processing-Text and Office Systems-Office, Document Architecture (ODA) and Interchange Format, International Organization for Standardization, 1989.
- [18] Y. Ishitani. Logical structure analysis of document images based on emergent computation. Proc. 5th ICDAR, 1999.
- [19] Esposito, F., Malerba, D., Francesca, Lisi, F.A., Ras, W.: Machine learning for intelligent processing of printed documents. Journal of Intelligent Information Systems 14 (2000) 175–198.
- [20] M. Krishnamoorthy, G. Nagy, S. Seth, and M. Viswanathan, "Syntactic segmentation and labeling of digitized pages from technical journals," IEEE Transactions on Pattern Analysis and Machine Intelligence 15, pp. 737–747, 1993.

Graph Theory Based Approach For Image Segmentation Using Wavelet Transform

Vikramsingh R. Parihar

*Department of PG Studies (Electrical and Electronics Engg.)
Prof Ram Meghe Collge of Engineering and Management
Badnera-Amravati. 444701, INDIA*

vikramparihar05@gmail.com

Nileshsingh V. Thakur

*Department of PG Studies (Electrical and Electronics Engg.)
Prof Ram Meghe Collge of Engineering and Management
Badnera-Amravati. 444701, INDIA*

thakurnisvis@rediffmail.com

Abstract

This paper presents the image segmentation approach based on graph theory and threshold. Amongst the various segmentation approaches, the graph theoretic approaches in image segmentation make the formulation of the problem more flexible and the computation more resourceful. The problem is modeled in terms of partitioning a graph into several sub-graphs; such that each of them represents a meaningful region in the image. The segmentation problem is then solved in a spatially discrete space by the well-organized tools from graph theory. After the literature review, the problem is formulated regarding graph representation of image and threshold function. The boundaries between the regions are determined as per the segmentation criteria and the segmented regions are labeled with random colors. In presented approach, the image is preprocessed by discrete wavelet transform and coherence filter before graph segmentation. The experiments are carried out on a number of natural images taken from Berkeley Image Database as well as synthetic images from online resources. The experiments are performed by using the wavelets of Haar, DB2, DB4, DB6 and DB8. The results are evaluated and compared by using the performance evaluation parameters like execution time, Performance Ratio, Peak Signal to Noise Ratio, Precision and Recall and obtained results are encouraging.

Keywords: Segmentation, Graph Theory, Threshold, Wavelet Transform.

1. INTRODUCTION

Segmentation is the process of partitioning a digital image into set of pixels or regions. Among the various existing segmentation approaches, graph theoretic approach found to have several good features in practical applications. The graph theoretic approach organizes the image elements into mathematically sound structures. It makes the formulation of the problem more flexible and the computation more resourceful. The problem is modeled in terms of partitioning a graph into several sub-graphs; such that each of them represents a meaningful object of interest in the image. The segmentation problem is then solved in a spatially discrete space by the efficient tools from graph theory [1].

All the existing graph based approaches involves the use of following terminologies. Let $G = (V, E)$ be a graph, where $V = \{v_1, v_2, \dots, v_n\}$ is a set of vertices corresponding to the image elements, which might represent pixels or components. E is a set of edges connecting pairs of neighboring vertices. Each edge $(v_i, v_j) \in E$ has a corresponding weight $w(v_i, v_j)$ which measures a quantity based on the property between the two vertices connected by that edge e.g., color, motion, location or some other local attribute (in our case the difference in intensity). For image segmentation, a segmentation S is a partition of V into components such that each component $C \in S$ corresponds to a connected component in a graph $G' = (V', E')$, where $V' \subseteq V, E' \subseteq E$.

A segmentation approach should capture perceptually important components or regions. Now three problems arise as to provide description of what is perceptually important, to specify what a developed segmentation approach does and precise definitions of the properties of a resulting segmentation, in order to better understand the method as well as to facilitate the comparison of different approaches. The segmentation approach should run at speeds similar to edge detection or other low-level visual processing techniques in order to be of practical use. Also the visual quality of segmentation is to be maintained at the same time.

This paper presents the image segmentation approach based on graph theory. The problem is modeled in terms of partitioning a graph into several sub-graphs; such that each of them represents a meaningful region in the image. The segmentation problem is then solved in a spatially discrete space by the well-organized tools from graph theory. The boundaries between the regions are determined as per the segmentation criteria and the segmented regions are labeled with random colors. In presented approach, the image is preprocessed by discrete wavelet transform and coherence filter before graph segmentation. The experiments are carried out on a number of natural images taken from Berkeley Image Database as well as synthetic images from online resources.

The organization of this paper is as follows. Section 2 includes the literature review. The section concludes with our findings from the literature review and motivation behind identified problems. Section 3 focuses on the formulation of the identified problem regarding the graph based representation of image and threshold function. Section 4 is dedicated to the proposed approach; where the working of the graph based algorithm for segmenting an image is described along with its implementation and our contribution to the work. Section 5 emphasize on the experimental results for a number of images along with comparison of the obtained results followed by the thorough discussion about the experimental results. Section 6 addresses the conclusions along with the future work.

2. LITERATURE REVIEW

The earliest graph-based approaches use fixed thresholds and local measures in computing segmentation. Later the focus was moved towards segmenting the image based on minimum spanning tree (MST) of the graph. For image segmentation, the edge weights in the graph are based on the differences between pixel intensities. The segmentation criterion is to break MST edges with large weights. The inadequacy of simply breaking large edges is that it would result in the high variability region being split into multiple regions. The splitting of such highly variable region is inappropriate.

Another class of graph based approaches is introduced [2-5] where the technique primarily focuses on finding minimum cuts in a graph. The cut criterion is designed in order to minimize the similarity between pixels that are being split. This bias is addressed with the normalized cut criterion. These cut-based approaches to segmentation capture non-local properties of the image, in contrast with the early graph-based approaches. However, they provide only a characterization of each cut rather than of the final segmentation.

The normalized cut criterion [6-8] provides a significant advance over the previous works. However, the normalized cut criterion also yields an NP-hard computational problem. In practice these approximations are still fairly hard to compute, limiting the approach to relatively small images.

Later the eigenvector-based approximations [9-10] are related to more standard spectral partitioning approaches on graphs. However, all such approaches are too slow for many practical applications. Also the eigen vector approach captures computationally important groupings or clusters and not according to human perception. Hence, our focus is moved towards another approach.

Pedro F. Felzenszwalb and Daniel P. Huttenlocher [11] using a graph-based representation of the image developed a segmentation algorithm and found that their approach satisfy global properties. The algorithm runs in time nearly linear in the number of graph edges and is also fast in practice. The specialty of the approach is that it is able to preserve detail in low-variability image regions and ignore detail in high-variability regions. Further improvements in [11] are made by Ming Zhang and Reda Alhaji [12] by re-defining the internal difference used to define the property of the components and the threshold function, which is the important factor in determining the size of the components. They claimed the efficiency and effectiveness of the adjusted approach through experimentations. However, no performance evaluation parameter is presented by both [11] and [12].

2.1 Our findings from the Literature Review

- Fixed threshold and local measures cannot be employed for good segmentation as it has many drawbacks.
- Simply breaking the MST edges or edges with high weights would result in improper segmentation.
- The eigen-vector based segmentation approaches are too slow for practical applications and the segments obtained by these approaches are computationally important but perceptually important regions are not obtained.
- The normalized cut criterion provides a significant advance over the previous works. However, the normalized cut criterion also yields an NP-hard computational problem. In practice these approximations are still fairly hard to compute, limiting the approach to relatively small images or requiring computation times of several minutes.
- Graph cuts algorithm based on iterated region merging requires lot of user interaction.
- In the image segmentation based on mean shift and normalized cuts, the spatial structure and the detailed edge information of an image are not preserved.
- If the image is treated as an undirected weighted non-planar finite graph and image segmentation is handled as graph partitioning problem, then the approach could not segment the images having high overlapping of objects or very dark images.
- If weighted Euclidean distance is used to calculate the edge weight, then the efficiency becomes less.
- When the segmentation is done based on the principle that in an Eulerian circuit, each edge is traversed only once and further segregation in open and closed sub-graphs is done by choosing critical vertices at a minimum directed distance, the algorithm itself cannot trace the boundary in images. The input of traced boundary is, thus, to be given; so more user interaction is required.
- When the objects to be segmented contain similar colors with the background, Grab Cut might fail to correctly segment them.
- The iterated region merging-based graph cuts algorithm requires a lot of user interaction.

2.2 Motivation Behind Identified Problem

From the critical analysis of the related work, we find that graph partitioning problem is categorized as NP-hard problem. Since image segmentation can be reduced to graph partitioning therefore it is also a NP-hard problem. Though, the different approaches exist to perform the color image segmentation, no particular approach produce the most efficient segmentation for the given color image. Also there is no standard basis on which an image can be segmented. Therefore, the scope of contribution exists in this area and this motivated us for problem formulation. Our goal is to develop an image segmentation approach that can be broadly useful, just like the other low-level techniques such as edge detection which are utilized in a wide range of computer vision tasks. In order to achieve such broad utility, we believe it is important that a segmentation approach should have the two properties. First is to capture perceptually important groupings or regions, which often reflect global properties of the image. And second is to run the segmentation approach at the speeds similar to edge detection or other low level process. We have developed an approach for image segmentation considering these two factors.

3. PROBLEM FORMULATION

This section presents the formulation of the identified problem, which involves the use of graph based representation of an image along with threshold function. Although, the literature consists of a various approaches to represent an image onto a graph, all the graph theoretic approaches involve the same common terminologies. The problem of graph based segmentation can be formulated as:

- The image is initially mapped on a graph $G = (V, E)$, where, $V = \{v_1, v_2, \dots, v_n\}$ is a set of vertices corresponding to the image elements, which might represent pixels or regions. E is a set of edges connecting certain pairs of neighboring vertices.
- Each edge $(v_i, v_j) \in E$ should have a corresponding weight $w(v_i, v_j)$ which measures a certain quantity based on the property between the two vertices connected by that edge.
- For image segmentation, an image should be partitioned into mutually exclusive components, such that each component C is a connected graph $G' = (V', E')$ where $V' \subseteq V$, $E' \subseteq E$ and E' contains only edges built from the nodes of V' .
- In other words, non empty sets C_1, \dots, C_k form a partition of the graph G such that $C_i \cap C_j = \phi$ ($i, j \in \{1, 2, \dots, k\}, i \neq j$) and $C_1 \cup \dots \cup C_k = G$.
- Although, there are different aspects to measure the quality of segmentation but, in general, it is believed that the elements in a component are supposed to be homogeneous and the elements in different components to be heterogeneous.
- This means that edges between two vertices in the same component should have relatively low weights, and edges between vertices in different components should have higher weights.
- A threshold function is used to manage the extent to which the difference between the components must be larger than the minimum internal difference within each component.
- An approach is to be produced which when there are more components than expected, the threshold function should "encourage" merging. When there are fewer components than expected, the threshold function should "discourage" merging.
- Before graph based segmentation, the image should pass through a filter which will remove noise. However, the edges should be preserved for proper segmentation.
- To increase the speed of computation, some preprocessing should be done on image so that the smaller insignificant regions will be merged and the computational complexity of the graph based segmentation algorithm is thus reduced.
- The segmentation results should be evaluated based on appropriate performance evaluation parameters.
- Finally, the segmentation result of the proposed approach is to be compared.

4. PROPOSED APPROACH

This section is subdivided into three parts wherein the paper presents the working of the graph based representation approach that is incorporated in our work along with our approach. Section 4.1 primarily focuses on the study and working of the graph based representation of image. Our proposed approach is introduced in Section 4.2. Section 4.3 provides detailed working of the proposed approach.

4.1 Graph-Based Segmentation [11]

- Let $G = (V, E)$ be an undirected graph
- Vertices $v_i \in V$, the set of elements to be segmented
- Edges $(v_i, v_j) \in E$ corresponding to pairs of neighboring vertices.
- Each edge $(v_i, v_j) \in E$ has a corresponding weight $w((v_i, v_j))$ which is a non-negative measure of the dissimilarity between neighboring elements v_i & v_j .
- In the case of mentioned approach, the elements in V are pixels and the weight of an edge is the difference in intensity between the two pixels connected by that edge.

The systematic working of the graph based approach is demonstrated by means of flow chart, prepared by us, as shown in Figure 1. The input image is initially mapped on a graph $G = (V, E)$ with n vertices and m edges. The output is a segmentation of V into components $S = (C_1, \dots, C_r)$. The edges E are sorted into non decreasing edge weight order $\pi = (o_1, \dots, o_m)$. The segmentation is started with S^0 , where each vertex v_i is in its own component. S^q is constructed from S^{q-1} as shown below. The following process is repeated for $q = 1, \dots, m$. Let v_i and v_j denote the vertices connected by the q^{th} edge in the ordering, i.e., $o_q = (v_i, v_j)$. If v_i and v_j are in disjoint components of S^{q-1} and $w(o_q)$ is small compared to the internal difference of both the components, then the two components are merged.

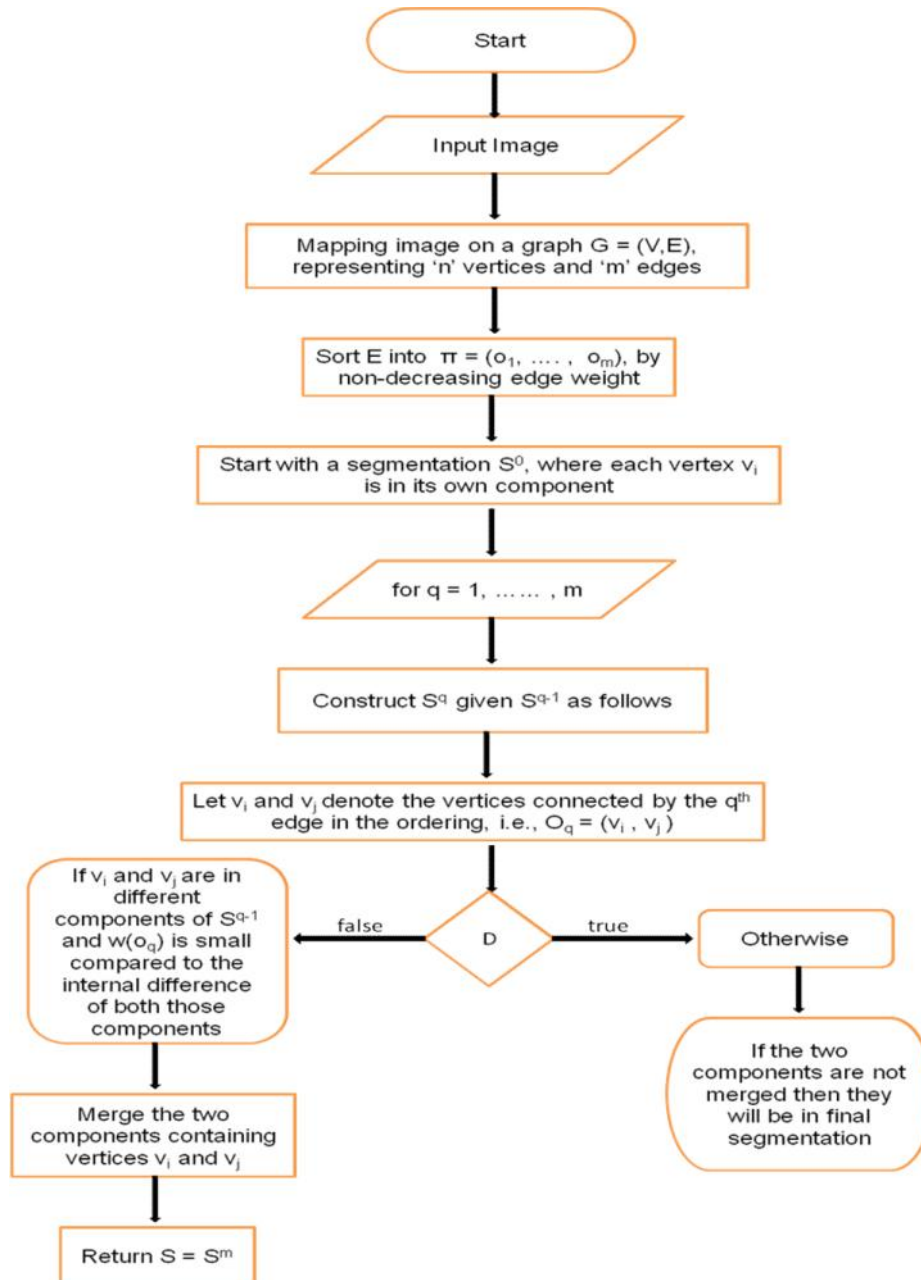


FIGURE 1: Flowchart, prepared by us, for the graph based segmentation approach [11].

In other words, the input image is considered as a graph where the pixels are vertices and the edges connecting two pixels have some weights that are the difference between the intensity values of the two pixels. These edges are initially sorted according to the non decreasing order. The segmentation process is then initialized with the consideration that each vertices belong to its own components. Now the edges connecting two vertices in the neighboring regions are evaluated. Based on the threshold value, the predicate decides whether the two regions have to be merged or to be considered as segmented. If the edges connecting two pixels of different components have less value than the threshold, then the two regions are merged together. If the edges connecting two pixels of different components have equal or larger value than the threshold then the two regions remain separated and are obtained in the final segmentation results. Similar calculation is performed for all the edges and thus the boundaries between the two pixels are determined. The regions are finally labeled with random colors so as to distinguish the adjacent regions. The above process can be interpreted with the help of the flowchart.

4.5 Proposed Approach

From the study of the graph based segmentation approach, it is found that changing the definition to use the median weight, or some other property, in order to make the computation more robust, makes the problem of finding a good segmentation NP-hard. Thus a small change to the segmentation criterion vastly changes the difficulty of the problem. So changing the segmentation criteria is not appropriate. So, while maintaining the segmentation criteria of [11], we carried out experimentations by preprocessing the input image by using wavelets transforms like Haar, DB2, DB4, DB6 and DB8 as well as filtering the image using coherence filter [13]. A number of natural images and synthetic images are used for experimentations. The evaluation of the proposed graph based segmentation approach which includes the execution time, Performance ratio (PR) [14], Precision and Recall and Peak Signal to Noise ratio [PSNR].

4.5.1 Wavelet Transform

Wavelets have the special ability to examine signals simultaneously in both time and frequency. In the DWT, an image is analyzed by passing it through an analysis filter bank. This process is followed by a decimation operation. This analysis filter bank of a low pass and a high pass filter is commonly used in image compression. A signal is split into two bands when it passes through these filters. The coarse information of the signal is extracted by low pass filter which corresponds to an averaging operation. The high pass filter extracts the detail information of the signal which corresponds to a differencing operation. The output of the filtering operations is then decimated by two.

By performing two separate one-dimensional transforms, a two-dimensional transform can be accomplished. Here, initially, the image is filtered along the x-direction using low pass and high pass analysis filters and decimated by two. On the left part of the matrix, low pass filtered coefficients are stored and on the right part of the matrix, high pass filtered coefficients are stored. Later, the same process is followed by filtering the sub-image along the y-direction and decimated by two. On the lower part of the matrix, low pass filtered coefficients are stored and on the upper part of the matrix, high pass filtered coefficients are stored. Finally, the image is split into four bands. These bands are denoted by HH, LH, HL and LL after one-level decomposition.

The following process demonstrated how reconstruction of the image is carried out. Initially, the image is upsampled by a factor of two on all the four subbands at the coarsest scale and filters the subbands in each dimension. Then the four filtered subbands are sum up to reach the low-low subband at the next finer scale. This process is repeated until the image is fully reconstructed.

Among the various wavelet transforms, we carried out experimentations by preprocessing the image by using Haar transform, DB2 transform, DB4 transform, DB6 transform and DB8 transform and found that the execution speed is marginally increased and also the visual quality of the segmentation output is maintained and even improved in many cases.

4.5.2 Coherence Filter [13]

In order to compensate for digitization artifacts and removal of the noise inculcated in the images, we used a Coherence filter to smooth the image slightly before computing the edge weights. When the image is passed through a coherence filter, the coherence filter performs Anisotropic Diffusion of the color or grayscale image. This process reduces the noise in an image while preserving the region edges. Anisotropic diffusion is a technique that aims at reducing image noise while preserving significant parts of the image details like edges, lines or other parameter that are important for the analysis of the image. As a result, the images obtained after filtering preserves linear structures while at the same time smoothing is made along these structures. A generalization of the usual diffusion equation describes both these cases where the diffusion coefficient is a function of image position and assumes a matrix value. In Anisotropic diffusion each new image in the family is computed by applying the above mentioned generalized equation to the previous image. As a result, anisotropic diffusion is an iterative process where a relatively simple set of computation are used to compute each successive image in the family and this process is continued until a sufficient degree of smoothing is obtained. Due to the above mentioned advantages, we preprocessed the image by means of the coherence filter.

4.6 Working of the Proposed Approach

Flowchart for the proposed approach is shown in Figure 2. This flowchart helps in visualization of the stepwise working of the proposed approach. Flowchart represents the process for color image segmentation.

- As a preprocessing step discrete wavelet transform is done on the images. In our experimentations we used the single-level discrete 2-D wavelet transform (DWT2) which performs single-level 2-D wavelet decomposition with respect to either a particular wavelet or particular wavelet filters specified. We used the wavelets like Haar, DB2, DB4, DB6 and DB8 for experimentations.
- Before passing the image to the coherence filter, the gray scale component image for each color plane, i.e. red, green and blue colors, is extracted by simple operation.
- The grayscale color plane image is then given to the coherence filter where the noise is removed while preserving the edges.
- The graph based segmentation is done on this filtered image. Some input parameters have to be initiated before segmentation is done. These parameters includes
 1. *neighbor_radius*: the neighborhood radius of each pixel [1 by default]
 2. *Coefficient k*: segmentation algorithm coefficient (large prefer large segmented component)
 3. *min_size*: the minimum size allowed for each segment.
- The graph segmentation is then done on the three color planes respectively depending upon the parameters provided.
- As discussed earlier, the boundaries between the two regions are determined based on the definition of predicate.
- Gradient operator help visualize the boundaries between the components. The white color indicates the presence of boundaries. The black color regions are the components separated by the boundaries.
- Morphological operations are done on the gradient image from where the contours are obtained. Finally the contours obtained are more prominent as the insignificant boundaries get eliminated.
- The image is then labeled with random intensity values for each color plane. This image is normalized for display purpose.
- Two neighboring pixels are put in the same component when they appear in the same component in all three of the color plane segmentations.
- The contours obtained from the three color planes are intersected together to form the final contours and the regions are determined based on these contours for color images.
- The regions are finally assigned random colors so that the neighboring regions can be differentiated.

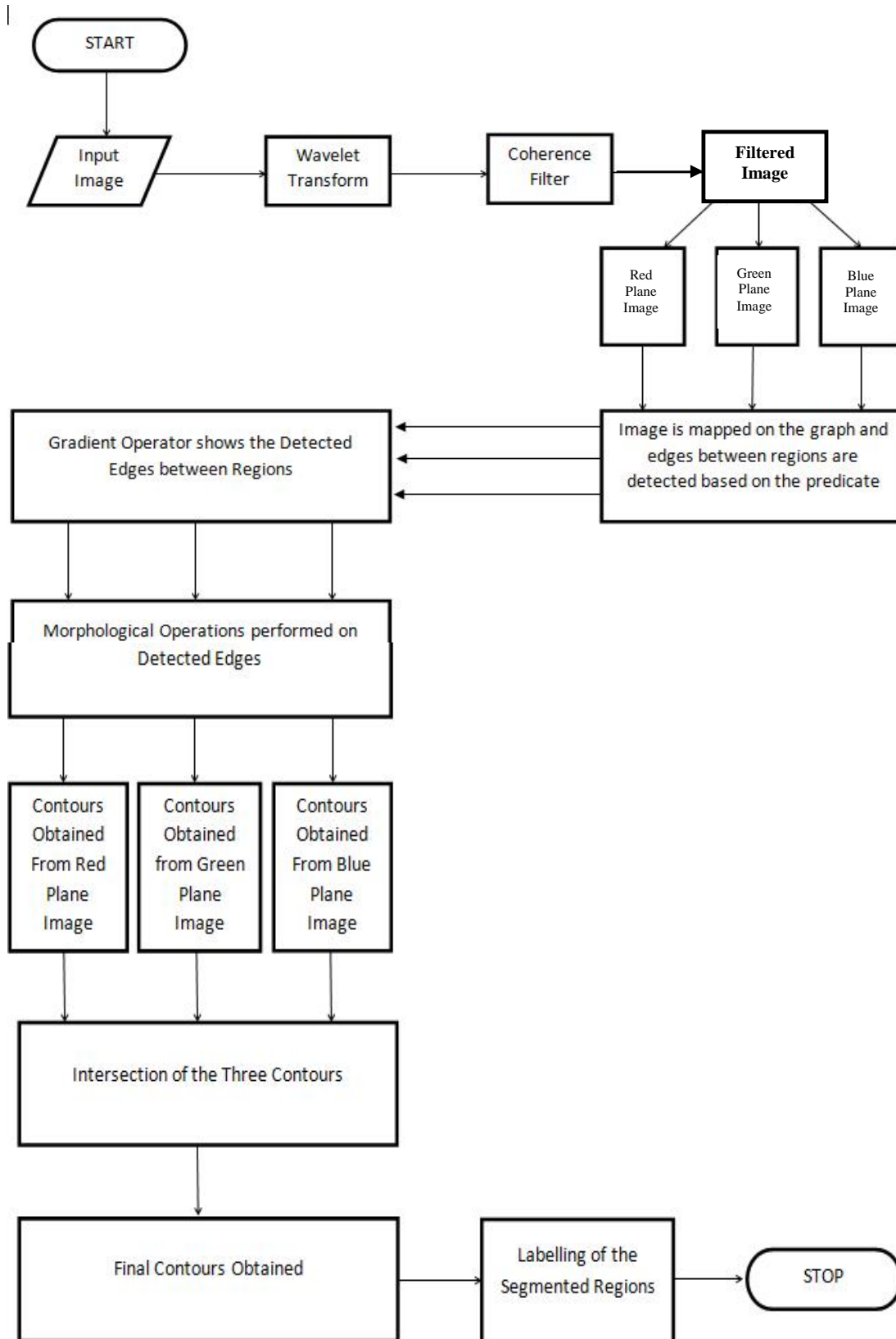


FIGURE 2: Flowchart of the proposed graph based segmentation approach for color images.

5. EXPERIMENTAL RESULTS AND DISCUSSIONS

This section presents the experimental results and discussion on obtained results. Section 5.1 demonstrates the stepwise working of the proposed approach for the different wavelets. The comparison on perceptual basis of various segmented images obtained by our approach is done in Section 5.2. The evaluation and comparison of the obtained results is mentioned in Section 5.3. Thorough discussion about the obtained results is presented in Section 5.4.

The approach is implemented using (MATLAB 8.1.0.604) (R2013a). The experimentations are carried out on Intel (R) Core (TM) 2 Duo T6570, 2.10 GHz processor. The RAM of the system used is 3GB and ROM is 300GB. The operating system is 32-bit and the processor is x64 installed on Windows 8 platform. The experimentations are carried out on natural color and grayscale images taken from Berkeley Image Database [15] as well as synthetic images [16–19] taken from online resources.

5.1 Stepwise Output of the Proposed Approach

This section presents the stepwise results obtained from our proposed approach for each of the wavelet used. The input image “296059.jpg” of size 481 x 321 taken from Berkeley Image Database [15] is shown in Figure 3. The final contours obtained and the labeled images are also displayed for each approach. A tabular representation is provided for the display of intermediate results which are obtained at the mentioned stages. Finally the screenshots of the Graphical User Interface (GUI), which is created in order to visualize the segmentation output in a more effective manner, is presented. In the GUI, the input image, final segmented image, the graph segmented images for each color planes as well as the stepwise results are displayed.

5.1.1 Stepwise results obtained when the image is not preprocessed by any wavelet transform

The final contours obtained after segmentation when the image is not preprocessed by any wavelet transform is as shown in Figure 4 (a). The segmented regions are then labeled as shown in Figure 4 (b). The intermediate results obtained at various stages are provided in Figure 5. Finally screenshot of GUI for the mentioned approach is provided in Figure 6.



FIGURE 3: The input image “296059.jpg” of size 481 x 321 taken from Berkeley Image Database.



FIGURE 4 (a): Final contours obtained after intersecting the contours of the three color planes.



FIGURE 4 (b): Final Labeled Image showing Segmented Regions.









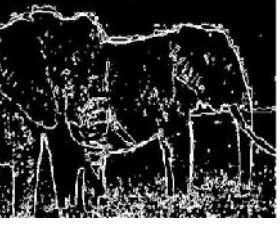
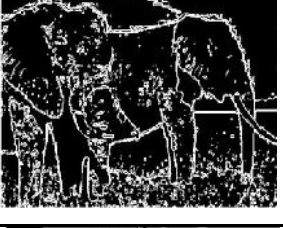
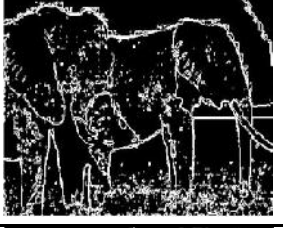
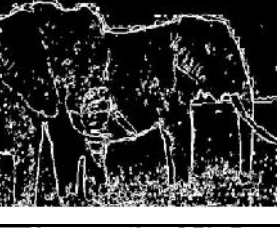






Component	Red Color Plane	Green Color Plane	Blue Color Plane
Grey scale Component of Input Image given to Coherence Filter			
Filtered Image			
Gradient after Graph Based Segmentation			
Contours obtained before morphological operation			
Contours obtained after morphological operation			
Labeled Image			

FIGURE 5: Stepwise output of the proposed segmentation approach when the input image is not preprocessed by any wavelet transforms.

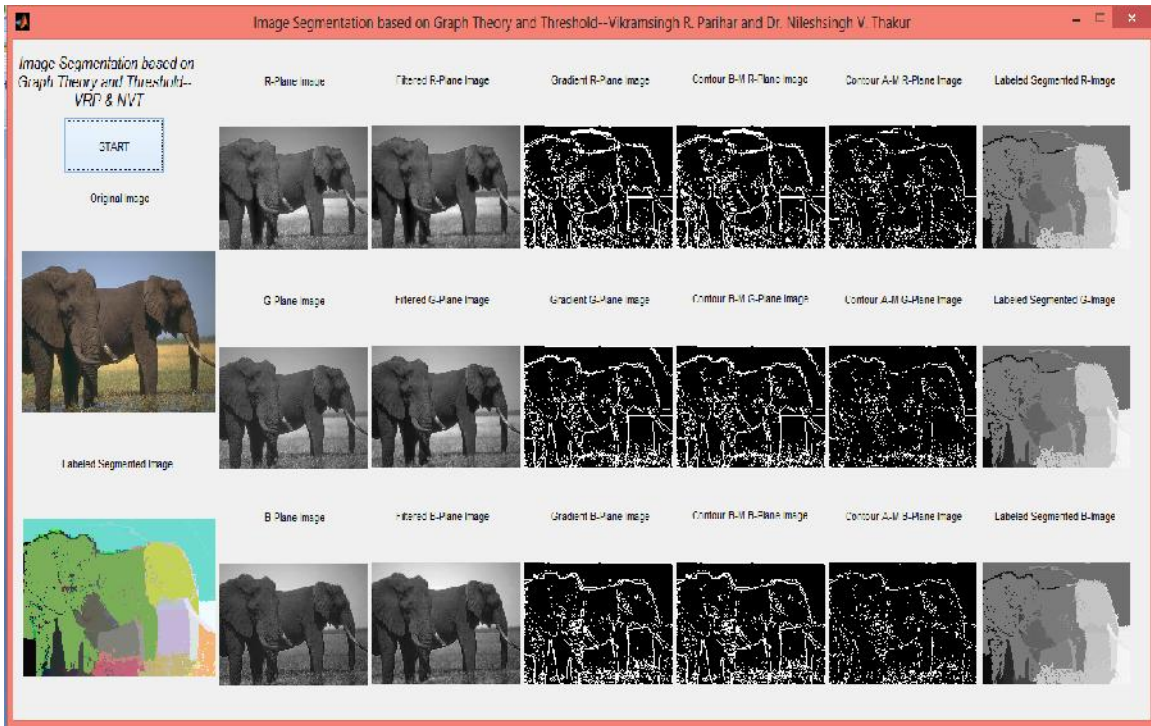


FIGURE 6: Screenshot of GUI showing the obtained results which help in visualizing the segmented output of each color plane along with the labeled image, when the image is not preprocessed by any wavelet transform.

The final contours obtained after segmentation when the image is preprocessed by discrete wavelet transform using different wavelets are as shown in Figure 7 (a) through Figure 11 (a). The segmented regions are then labeled as shown in Figure 7 (b) through Figure 11 (b).



FIGURE 7 (a): Final contours obtained after intersecting the contours of the three color planes.



FIGURE 7 (b): Segmented image obtained after preprocessing the image with Haar Transform.



FIGURE 8 (a): Final contours obtained after intersecting the contours of the three color planes.



FIGURE 8 (b): Segmented image obtained after preprocessing the image with DB2 Transform.

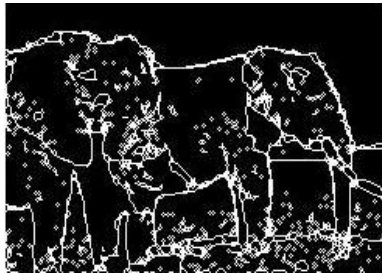


FIGURE 9 (a): Final contours obtained after intersecting the contours of the three color planes.



FIGURE 9 (b): Segmented image obtained after preprocessing the image with DB4 Transform.

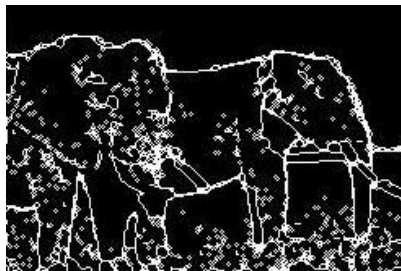


FIGURE 10 (a): Final contours obtained after intersecting the contours of the three color planes.



FIGURE 10 (b): Segmented image obtained after preprocessing the image with DB6 Transform.

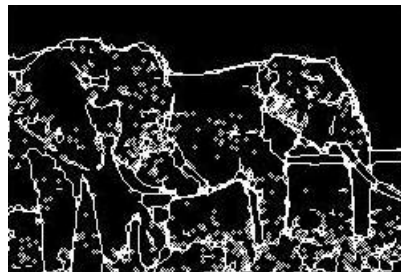


FIGURE 11 (a): Final contours obtained after intersecting the contours of the three color planes.



FIGURE 11 (b): Segmented image obtained after preprocessing the image with DB8 Transform.

5.2 Contours and Labeled Images

Based on the proposed approach, we carried out experimentations on natural as well as synthetic images and then comparative study is done based on the results obtained. The natural images are taken from Berkeley Image Database. These images include color as well as grayscale images both from the 'Test' and 'Train' datasets. We also experimented on some synthetic images taken from the online resources. All these experimentations are carried out using MATLAB (R2013a). Experimental results are shown in Figure 12 through Figure 15.

For all the experiments, we initialized the input parameters as given below:

neighbor_radius = 1 (the neighborhood radius of each pixel [1 by default])

Coefficient k = 350 (segmentation algorithm coefficient [large prefer large segmented component])

min_size = 0.01 (the minimum size allowed for each segment).



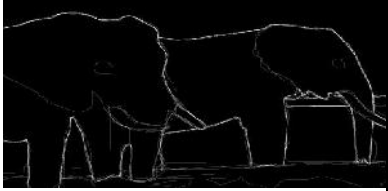

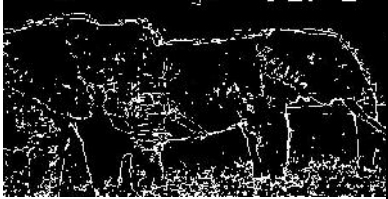




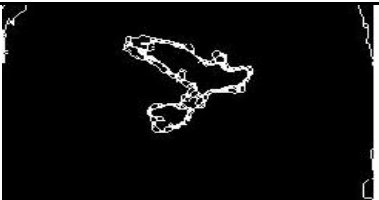

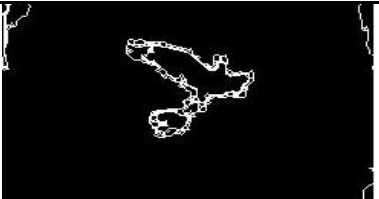

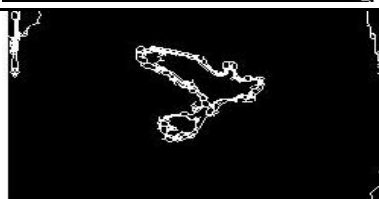


Image name	'296059.jpg'	'135069.jpg'
Image		
Ground Truth Data for Edge Detection		
Contours obtained without preprocessing by any Wavelet Transform		
Contours obtained after DWT2 using Haar Wavelet		
Contours obtained after DWT2 using DB2 Wavelet		
Contours obtained after DWT2 using DB4 Wavelet		
Contours obtained after DWT2 using DB6 Wavelet		
Contours obtained after DWT2 using DB8 Wavelet		

FIGURE 12: Demonstration of Contour Images obtained for given images.



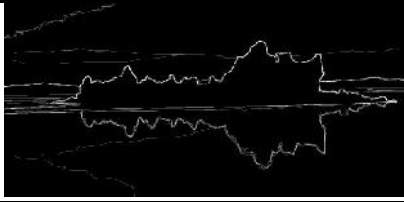


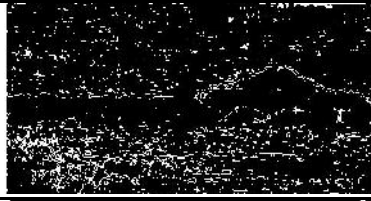
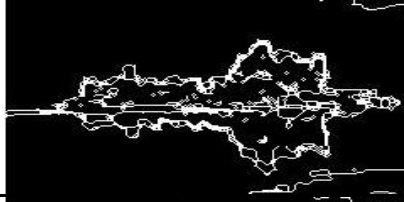

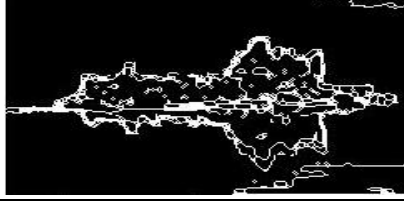
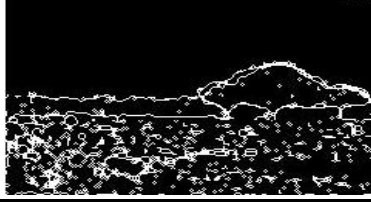
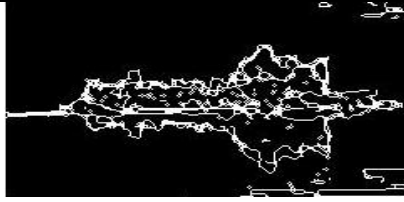

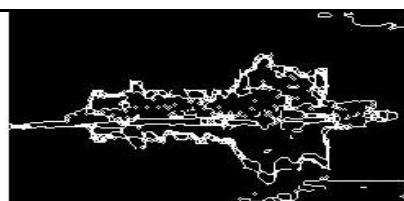
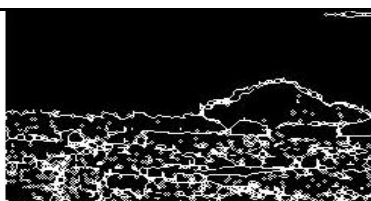
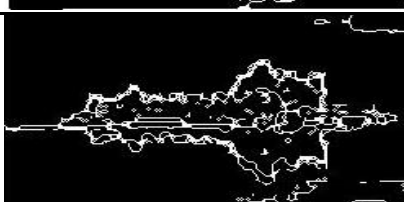
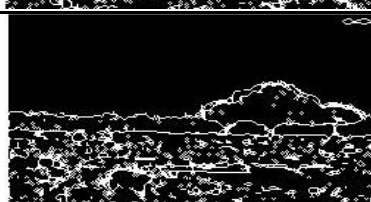
Image name	'143090.jpg'	'296007.jpg'
Image		
Ground Truth Data for Edge Detection		
Contours obtained without preprocessing by any Wavelet Transform		
Contours obtained after DWT2 using Haar Wavelet		
Contours obtained after DWT2 using DB2 Wavelet		
Contours obtained after DWT2 using DB4 Wavelet		
Contours obtained after DWT2 using DB6 Wavelet		
Contours obtained after DWT2 using DB8 Wavelet		

FIGURE 13: Demonstration of Contour Images obtained for given images.





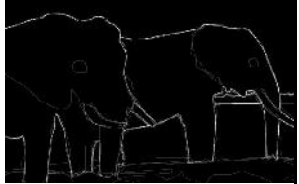
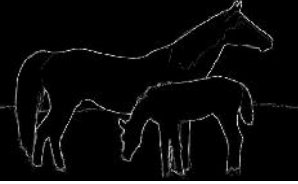








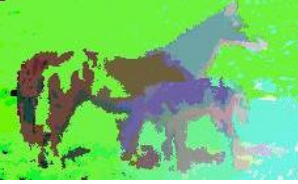


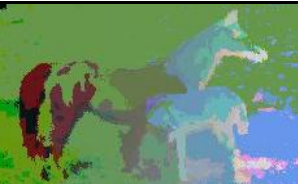






Image name	'45096.jpg'	'65019.jpg'	'113044.jpg'
Image			
Ground Truth data for Edge Detection			
Segmented Image without Wavelet Transform			
Segmented Image after Haar Transform			
Segmented Image after DB2 Transform			
Segmented Image after DB4 Transform			
Segmented Image after DB6 Transform			
Segmented Image after DB8 Transform			

FIGURE 14: Labeled images obtained after segmentation.





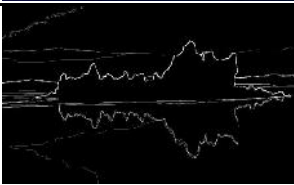

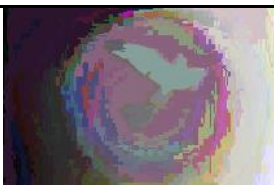
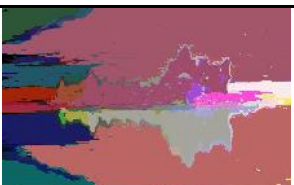




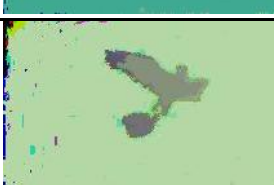











Image name	'135069.jpg'	'143090.jpg'	'296007.jpg'
Image			
Ground Truth Data for Edge Detection			
Segmented Image without Wavelet Transform			
Segmented Image after Haar Transform			
Segmented Image after DB2 Transform			
Segmented Image after DB4 Transform			
Segmented Image after DB6 Transform			
Segmented Image after DB8 Transform			

FIGURE 15: Labeled images obtained after segmentation.

5.3 Evaluation and Comparison of Obtained Experimental Results

In this section, the evaluations of the obtained results are done. The comparison of the results is also carried out. However, the comparison with the existing approach of [11] or [12] is not attained. This is due to the fact that the authors in these papers did not evaluate their results. They used the term “Human Perception” as to evaluate their result which is, in fact, a vague term. However, we find that our segmentation results also produce regions that are meaningful. We, here, used the following parameters used for evaluations of our results.

1. Time Required for Graph Based Segmentation
2. Peak Signal to Noise ratio (PSNR)
3. Performance Ratio (PR)
4. Precision and Recall.

1. Time Required for Graph Based Segmentation

The time required for graph based segmentation approach to execute after preprocessing the image by DWT2 using the wavelets like Haar, DB2, DB4, DB6 and DB8 is shown in the Table 1. Table 1 helps for comparative study of the results.

2. Peak Signal to Noise Ratio (PSNR)

The PSNR of the final contour is calculated with reference to the ground truth dataset for edge detection. The PSNR is calculated using the following formula:

$$PSNR (dB) = 10 * \log\left(\frac{255^2}{MSE}\right), \text{ where } MSE = \sum_{i=1}^x \sum_{j=1}^y \frac{x*y}{(|A_{ij}-B_{ij}|)^2}$$

3. Performance Ratio (PR) [14]

PR is the ratio of true edges to false edges. Here true edges mean the edge pixels identified as edges in the ground truth data and false edges means the non edge pixels identified as edges and the edge pixels identified as non edges. The PR is calculated from the given formula

$$PR = \frac{\text{True Edges (Edge pixels identified as Edges)}}{\text{False Edges (Non edge pixels identified as edges) + (Edge pixels identified as Non-Edge pixels)}} \times 100$$

4. Precision and Recall

Precision is the fraction of the edges that are obtained by our approach that are relevant with the edges obtained from ground truth data. Whereas recall is the fraction of all relevant instances that are retrieved. There are four cases which have to be first evaluated:

- TN / True Negative: case is negative and predicted negative
- TP / True Positive: case is positive and predicted positive
- FN / False Negative: case is positive but predicted negative
- FP / False Positive: case is negative but predicted positive

Now, Precision = $\frac{TP}{TP + FP}$ and Recall = $\frac{TP}{TP + FN}$

Experimental results are shown in Table 2 through Table 5 and Figure 16 through Figure 19.

Input Image	Haar	DB2	DB4	DB6	DB8
'3096.jpg'	0.46	0.46	0.47	0.48	0.48
'42049.jpg'	0.43	0.46	0.45	0.46	0.46
'62096.jpg'	0.45	0.46	0.46	0.47	0.48
'108082.jpg'	0.45	0.46	0.48	0.49	0.49
'167062.jpg'	0.45	0.46	0.45	0.47	0.47

TABLE 1 (a): Time required for segmentation of natural gray images in seconds

Input Image	Haar	DB2	DB4	DB6	DB8
'45096.jpg'	0.46	0.46	0.48	0.48	0.49
'65019.jpg'	0.47	0.46	0.47	0.48	0.49
'113044.jpg'	0.47	0.47	0.48	0.48	0.50
'135069.jpg'	0.40	0.41	0.42	0.42	0.44
'143090.jpg'	0.42	0.43	0.44	0.45	0.45
'296007.jpg'	0.43	0.44	0.45	0.46	0.48
'296059.jpg'	0.46	0.46	0.46	0.48	0.49
'306005.jpg'	0.46	0.47	0.47	0.48	0.50
'beach.jpg'	0.24	0.25	0.25	0.24	0.25
'rice.jpg'	0.51	0.65	0.50	0.53	0.52

TABLE 1 (b): Time required for segmentation of natural color images in seconds

Image	Haar	DB2	DB4	DB6	DB8
'im1.jpg'	0.21	0.21	0.22	0.22	0.22
'Syntheticim1.jpg'	0.12	0.13	0.13	0.13	0.14
'Syntheticim2.jpg'	0.21	0.26	0.21	0.22	0.24
'Syntheticim3.jpg'	0.23	0.22	0.23	0.38	0.23

TABLE 1 (c): Time required for segmentation of synthetic images in seconds

Input Image	Haar	DB2	DB4	DB6	DB8
'45096.jpg'	46.7611	50.0392	49.1753	51.2983	43.1365
'65019.jpg'	70.7948	71.9072	70.4551	70.5546	67.2279
'113044.jpg'	34.9189	35.0555	34.3365	35.1997	33.7685
'135069.jpg'	41.6328	42.4072	38.3769	42.1317	33.0302
'143090.jpg'	44.1998	45.2006	43.4386	42.8078	41.0001
'296007.jpg'	58.5074	59.4026	57.2390	59.1842	58.3477
'296059.jpg'	59.3160	61.5131	59.6177	60.2192	59.5920
'306005.jpg'	60.0831	61.6381	58.8643	56.7069	57.6688

TABLE 2: Performance Ratio of the Segmented Results.

Input Image	Haar	DB2	DB4	DB6	DB8
'45096.jpg'	12.2143	11.4409	12.5631	12.7120	12.2341
'65019.jpg'	9.6443	9.5434	9.1652	9.0504	9.101
'113044.jpg'	8.6669	8.8337	8.4517	8.5042	8.5383
'135069.jpg'	17.721	17.6791	16.8576	16.4634	16.7061
'143090.jpg'	12.7865	12.2663	12.0917	12.6261	12.3636
'296007.jpg'	12.0733	12.3503	12.2443	10.9233	11.7055
'296059.jpg'	10.6418	11.089	10.7377	10.9542	10.3224
'306005.jpg'	10.0494	10.3964	10.0695	9.6637	9.6162

TABLE 3: PSNR of the Segmented Results in decibels.

Input Image	HAAR	DB2	DB4	DB6	DB8
'45096.jpg'	0.0022	0.0020	0.0023	0.0035	0.0019
'65019.jpg'	0.0024	0.0024	0.0017	0.0018	0.0016
'113044.jpg'	0.0009	0.0007	0.0010	0.0011	0.0007
'135069.jpg'	0.0026	0.0025	0.0012	0.0025	0.0028
'143090.jpg'	0.0035	0.0031	0.0030	0.0019	0.0026
'296007.jpg'	0.0038	0.0041	0.0047	0.0021	0.0025
'296059.jpg'	0.0026	0.0027	0.0030	0.0024	0.0042
'306005.jpg'	0.0014	0.0015	0.0019	0.0013	0.0015

TABLE 4: Precision of the Segmented Results.

Input Image	HAAR	DB2	DB4	DB6	DB8
'45096.jpg'	0.0499	0.0420	0.0472	0.0682	0.0446
'65019.jpg'	0.0428	0.0428	0.0308	0.0317	0.0300
'113044.jpg'	0.0248	0.0189	0.0272	0.0295	0.0189
'135069.jpg'	0.0610	0.0563	0.0282	0.0563	0.0751
'143090.jpg'	0.0449	0.0402	0.0393	0.0234	0.0327
'296007.jpg'	0.0337	0.0360	0.0422	0.0186	0.0219
'296059.jpg'	0.0404	0.0404	0.0461	0.0362	0.0643
'306005.jpg'	0.0274	0.0283	0.0365	0.0256	0.0292

TABLE 5: Recall of the Segmented Results.

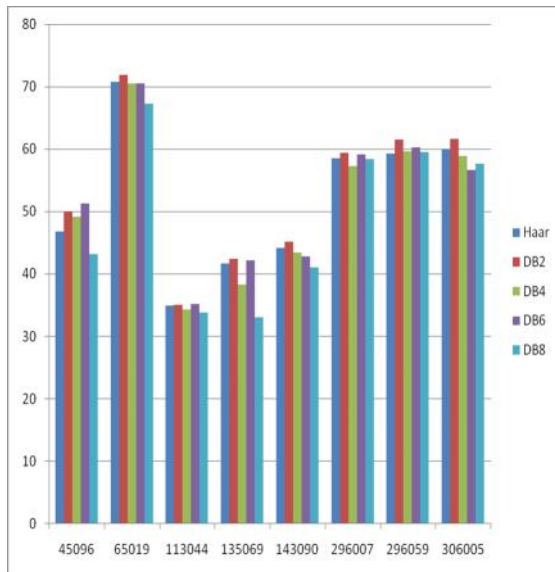


FIGURE 16: Bar Graphs showing comparison between the Performance Ratio of the results obtained for different wavelets.

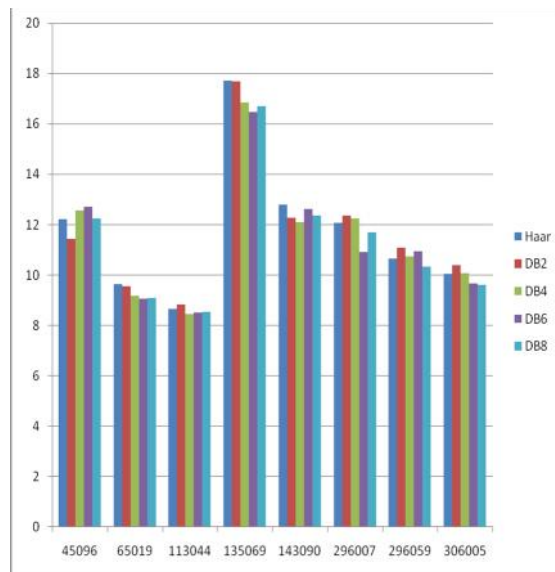


FIGURE 17: Bar Graphs showing comparison between the PSNR of the results obtained for the different wavelets.

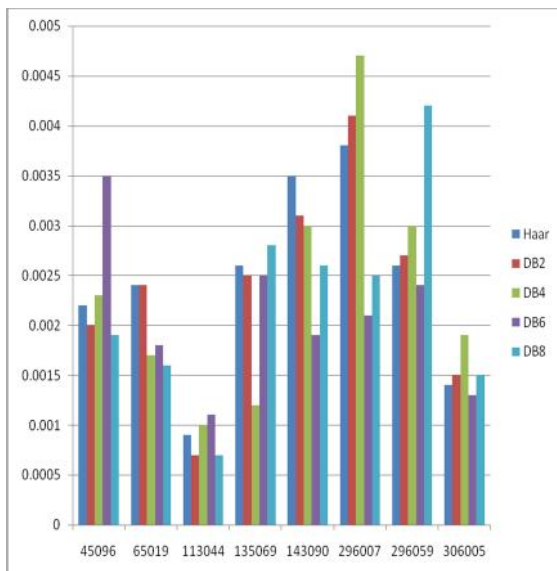


FIGURE 18: Bar Graphs showing comparison between the Precision of the results obtained for the different wavelets.

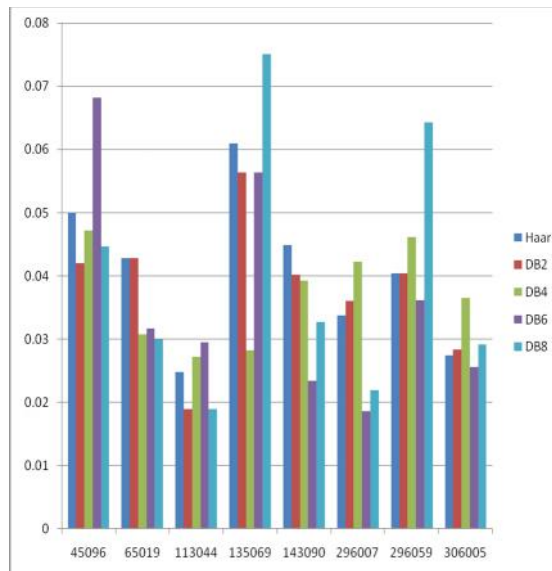


FIGURE 19: Bar Graphs showing comparison between the Recall of the results obtained for the different wavelets.

5.4 Discussion

The stepwise results obtained from each of the wavelet used for the image ‘296059.jpg’ helps us to understand the proper working of the segmentation process. It also provides us the visualization of the transformation of image at almost each and every level. The contours mentioned are nothing but the boundaries detected by the segmentation process. Through the stepwise observation, it is found that the contours obtained are inherently related with the true edges of the image when compared with ground truth data for edge detection. This is justified by analytical as well as perceptual evaluation.

Later, the segmentation output for various images are displayed which involves both the contours obtained for the various wavelets along with the labeled images. From observations, it can be identified that all the wavelets provided segmentation results which are perceptually important. It is, however, very difficult to evaluate on the basis of perception the quality of obtained images. This motivated us to use the performance evaluation parameters like Time, PR, PSNR, Precision and Recall.

The Table 1 (a) through Table 1 (c) presented the execution time after the image is preprocessed by the respective wavelets. From Table 1 it is very tough to suggest which of the approaches is more preferable as the time differences is only of a few milliseconds. However, it is found that time taken for graph segmentation after DWT2 using Haar wavelets required less time for almost all the mentioned images.

The PR, PSNR, Precision and Recall parameters are generally used for edge detection. As our approach is also incorporating the determination of edges between the regions, we applied these parameters to the detected boundaries. The comparative study shows that all the wavelet images performed in nearly an equal conduct as long as PR is concerned. The PR of DB2 is, however, found to be better when the comparison is done based on the bar graphs and is closely followed by DB6.

The PSNR is also an important parameter which can evaluate the experimental results. The comparison of the obtained results for each of the wavelet used is done with the ground truth image and the edges of both the images are compared. The higher the value of PSNR indicates

the minimum is the value of MSE and ultimately the better is the obtained results. From the study of the Table 3 and the bar graph shown in Figure 17, we found that the PSNR values of Haar wavelet is more in few cases closely followed by DB2 wavelets.

The Precision and Recall have also emerged as one of the wisely used evaluation parameters. The precision and recall are indirectly proportional to each other; wherein as the precision increases the recall decreases and vice versa. Higher precision is preferable and so does lower recall. The precision and recall parameters provide information about the relevancy between the obtained results with respect to some standard quantity.

We compared the edges obtained by our approach to the ground truth data set of edge detection. The precision and recall of each wavelet are shown in Table 4 and Table 5, plotted as bar graphs shown in Figure 18 and Figure 19 for comparison. By observing the results we found that precision of DB2 and Haar wavelets are comparatively higher. The recall of DB6 is minimum, closely followed by DB4 and DB2.

6 . CONCLUSION AND FUTURE SCOPE

This section presents the conclusions drawn from the evaluation and comparison of experimental results. The section concludes with future scope.

6.1 Conclusion

Based on the experimental results and discussion, the following conclusions are drawn:

- The contours obtained from graph segmentation are relevant to the true edges of the image. The observations regarding the edges are done in Figure 12 and Figure 13.
- The algorithm captures perceptually important regions. This can be justified from the Figure 14 and Figure 15 where the segmented results are compared with the input image as well as ground truth data.
- From the Table 1, it is found that time taken for graph segmentation after DWT2 using Haar wavelets required less time for almost all the mentioned images.
- The comparative study from the Table 2 shows that all the wavelet images performed in nearly an equal conduct as long as PR is concerned. The PR of DB2 is, however, found to be better when the comparison is done based on the bar graphs and is closely followed by DB6.
- The precision and recall of each wavelet are calculated and presented in the Table 4 and Table 5 and plotted as bar graphs in Figure 18 and Figure 19 for comparison. By observing the results, we found that precision of DB2 and Haar wavelets are comparatively higher. The recall of DB6 is minimum, closely followed by DB4 and DB2.

6.2 Future Scope

This section provides the possible future directions to extend the presented work.

- We, here, considered the wavelets of Haar, DB2, DB4, DB6 and DB8. In the future work, one can carry out experimentations considering other families of wavelets for preprocessing the image before segmentation and observe the results.
- Also, after the contours are extracted, the regions are labeled with random colors. This however may provide an improper segmentation results sometime as the neighboring regions are assigned colors that may not be distinguished. So, instead of random colors, a function can be developed that can assign largely varying colors to the neighboring regions.

- Developed approach can be used in other image processing work, in particular, image compression and image recognition.
- Presented paper deals with the segmentation of the still images, but, can be extended for the analysis of video. One can use the proposed approach as the basis for video compression.
- Self similarity check can be explored to have the better segmentation in combination with the proposed approach.
- Advanced non-classical optimization techniques, like, neural network and genetic algorithm can be used to optimize the obtained results. From this point of view, one can model the proposed approach in terms of the problem of neural network and genetic algorithm.

6. REFERENCES

- [1] B. Peng, L. Zhang and D. Zhang, "A survey of graph theoretical approaches to image segmentation", *Pattern Recognition*, Vol. 46, pp.1020-1038, (2013).
- [2] B. Peng, L. Zhang, D. Zhang and J. Yang, "Image segmentation by iterated region merging with localized graph cuts", *Pattern Recognition* Vol. 44, pp. 2527-2538, (2011).
- [3] W. Tao, F. Chang, L. Liu, H. Jin and T. Wang, " Interactively multiphase image segmentation based on variation formulation and graph cuts", *Pattern Recognition* Vol. 43, pp. 3208-3218, (2010).
- [4] J. Kim and K.Sang, "Color–texture segmentation using unsupervised graph cuts", *Pattern Recognition* Vol. 42, pp. 735-750, (2009).
- [5] M. Bleyer and M. Gelautz, "Graph-cut-based stereo matching using image segmentation with symmetrical treatment of occlusions", *Signal Processing: Image Communication*, Vol.22, pp.127-143, (2007).
- [6] J. Shi and J. Malik, "Normalized Cuts and Image Segmentation", *IEEE Transactions on pattern analysis and machine intelligence*, Vol. 22, Issue No. 8, (2000).
- [7] S. Wang and J. M. Siskind, "Image Segmentation with Ratio Cut", *IEEE Transactions on pattern and machine intelligence*, Vol. 25, Issue No. 6, (2003).
- [8] W. Tao, H. Jin, and Y. Zhang, "Color Image Segmentation Based on Mean Shift and Normalized Cuts", *IEEE Transactions on systems, man and cybernetics*, Vol. 37, Issue No. 5, (2007).
- [9] R. C. Wilson, E. R. Hancock, and B. Luo, " Pattern Vectors from Algebraic Graph Theory", *IEEE Transactions on pattern analysis and machine intelligence*, Vol. 27, Issue No 7, (2005).
- [10] Y. Yang, S. Han, T. Wang, W. Tao and X. Tai, "Multilayer graphcuts based unsupervised color–texture image segmentation using multivariate mixed student's t-distribution and regional credibility merging", *Pattern Recognition*, Vol. 46, pp. 1101-1124, (2013).
- [11] P.F.Felzenszwalb and D.P.Huttenlocher, "Efficient graph based image segmentation", *International Journal of Computer vision*, Vol.59, Issue No.2, (2004).

- [12] M. Zhang and R. Alhaji, "Improving the Graph-Based Image Segmentation Method", Proceedings of the 18th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'06), 0-7695-2728-0106, (2006).
- [13] J. Weickert, "Coherence-Enhancing Diffusion Filtering", International Journal of Computer Vision, Vol. 31, pp. 111-127, (1999).
- [14] P. A. Khaire and N. V. Thakur, "A Fuzzy Set Approach for Edge Detection", International Journal of Image Processing (IJIP), Vol. 6, Issue No. 6, (2012).
- [15] <http://www.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/> (accessed on 24/04/2014)
- [16] <http://fin6.com/2013/12/color/> (accessed on 24/04/2014)
- [17] http://clancyartpages.files.wordpress.com/2011/09/color_grid_dl.jpg (accessed on 24/04/2014)
- [18] <http://www.clker.com/clipart-grayscale-flower-decoration.html> (accessed on 24/04/2014)
- [19] <http://furbeeconsulting.com/blog/> (accessed on 24/04/2014)

An Approach For Single Object Detection In Images

Kartik Umesh Sharma

*Department of PG Studies (Computer Science and Engineering)
Prof Ram Meghe Collge of Engineering and Management
Badnera-Amravati. 444701, INDIA*

karthik8777@gmail.com

Nileshsingh V. Thakur

*Department of PG Studies (Computer Science and Engineering)
Prof Ram Meghe Collge of Engineering and Management
Badnera-Amravati. 444701, INDIA*

thakurnisvis@rediffmail.com

Abstract

This paper discusses an approach for object detection and classification. Object detection approaches find the object or objects of the real world present either in a digital image or a video, where the object can belong to any class of objects. Humans can detect the objects present in an image or video quite easily but it is not so easy to do the same by machine, for this, it is necessary to make the machine more intelligent. Presented approach is an attempt to detect the object and classify the same detected object to the matching class by using the concept of Steiner tree. A Steiner tree is a tree in a distance graph which spans a given subset of vertices (Steiner Points) with the minimal total distance on its edges. For a given graph G , a Steiner tree is a connected and acyclic sub graph of G . This problem is called as Steiner tree problem where the aim is to find a Steiner minimum tree in the given graph G . Basically the process of object detection can be divided as object recognition and object classification. A Multi Scale Boosted Detector is used in the presented approach, which is the combination of multiple single scale detectors; in order to detect the object present in the image. Presented approach makes use of the concept of Steiner tree in order to classify the objects that are present in an image. To know the class of the detected object, the Steiner tree based classifier is used. In order to reach to the class of the object, four parameters namely, Area, Eccentricity, Euler Number and Orientation of the object present in the image are evaluated and these parameters form a graph keeping each parameter on independent level of graph. This graph is explored to find the minimum Steiner tree by calculating the nearest neighbor distance. Experimentations are carried out using the standard LabelMe dataset. Obtained results are evaluated based on the performance evaluation parameters such as precision and recall.

Keywords: Object Detection, Steiner Tree, Object Classification.

1. INTRODUCTION

Object detection (OD) is a technologically challenging and practically useful problem in the field of computer vision and it has seen significant advances in the last few years [1]. Object detection deals with identifying the presence of various individual objects in an image. Humans perform object recognition effortlessly and instantaneously. Algorithmic description of this task for implementation on machines has been very difficult. Basic object detection model is shown in Figure 1. Basically an OD system can be described easily by seeing Figure 1, which shows the basic stages that are involved in the process of object detection. The basic input to the OD system can be an image or a scene in case of videos. The basic aim of this system is to detect objects that are present in the image or scene or simply in other words the system needs to categorize the various objects into respective object classes. The object detection problem can be defined as a labeling problem based on models of known objects. Given an image containing one or more objects of interest and a set of labels corresponding to a set of models known to the

system, the system is expected to assign correct labels to regions in the image. The object detection problem cannot be solved until the image is segmented and without at least a partial detection, segmentation process cannot be applied. The term detection has been used to refer to many different visual abilities including identification, categorization and discrimination.

This paper presents a object detection mechanism which not only detects the desired object in the image but also classifies the detected object. Multi scale boosted detector is used in order to detect the object of interest and a Steiner tree based classifier which uses the nearest neighbor concept in order to classify the detected objects.

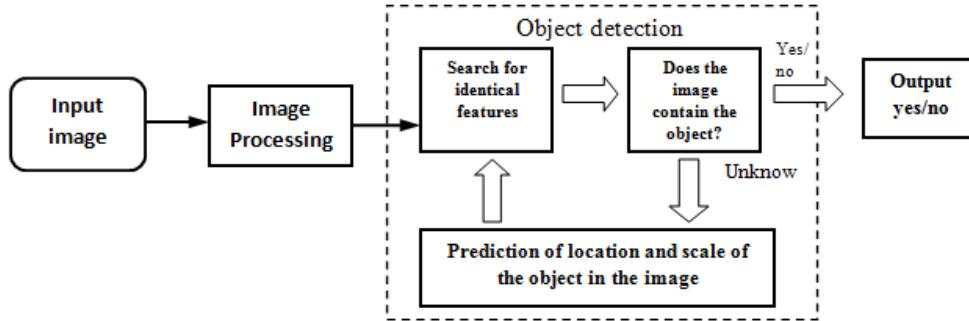


FIGURE 1: Basic Object Detection Model.

A Steiner tree is a tree in a distance graph which spans a given subset of vertices (Steiner Points) with the minimal total distance on its edges. Given a graph $G = (V, E)$, a subset $R \subseteq V$ of vertices, and a length function $d: E \rightarrow \mathbb{R}^+$ on the edges, a *Steiner tree* is a connected and acyclic sub graph of G which spans all vertices of R . The vertices in R are usually referred to as *terminals* and the vertices in $V \setminus R$ as *Steiner (or optional) vertices*.

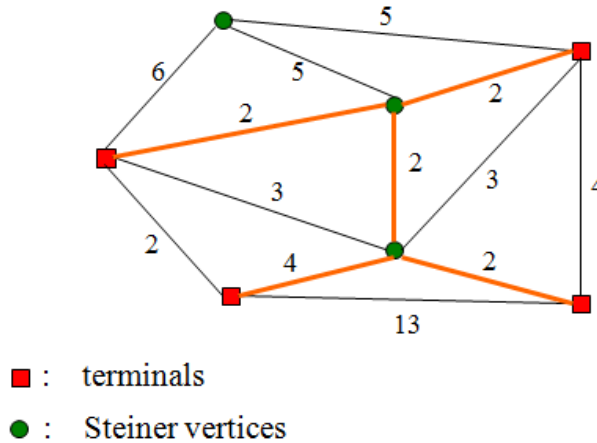


FIGURE 2: Example for Steiner Tree.

The length of this Steiner tree is $2+2+2+2+4 = 12$.

The so-called *Steiner tree problem (STP)* is an NP- hard [2], but can be approximated efficiently [3, 4]. The STP is a problem to find a Steiner minimum tree. (i.e., a Steiner tree of minimum length) in the graph G . Example for Steiner tree is shown in Figure 2. The Steiner tree problem is distinguished from the minimum spanning tree problem in that we are permitted to construct or select intermediate connection points to reduce the cost of the tree.

2. LITERATURE REVIEW

Object detection can be classified into the following types as Sliding Window based, Contour based, Graph based, Fuzzy based, Context based and some other types. Here we will review the work carried out by various authors in the field of Object Detection.

Sliding window based detection. The sliding window approach is common in object detection [5, 6, 7], and much work has been carried out to improve the running time for detecting an object. Although Segvic et al. [5] have explained how localization accuracy could be achieved by removing the need for spatial clustering of the nearby detection responses, but the use of spatial clustering can be done at the cost of localization uncertainty. Subburaman et al. [6] have presented a technique which is used to reduce the number of miss detections while increasing the grid spacing while the sliding window approach is used for object detection. Comaschi et al. [7] have proposed a sliding window approach that decides on the step size of the window at run time, which helps to apply this technique of sliding window to real time applications. They have also demonstrated that how this technique improves the performance of Viola Jones object detection [8], and also claimed to have achieved a speedup of 2.03x in frames per second without compromising the accuracy factor. The main issue being the space utilized.

Contour based detection. Contour based object detection can be well formulated as a matching problem between model contour parts and image edge fragments, and hence Yang et al. [9] have used this problem and have treated it as a problem of finding dominant sets in weighted graphs, where the nodes of the graph are pairs composed of contour parts and edge fragments and the weights between nodes are based on shape similarity. The main advantage of this system is that it can detect multiple objects present in an image in one pass. Still the question arises that can this system detect objects in an occluded image or other types of images. Amine and Farida, [10] have proposed an approach which makes use of a deformable model "Snake" which they have termed as an active contour for segmenting the range images. The process is again restricted to range images; the question still lies about the various types of images.

The system proposed by Shotton et al. [11] not only recognizes objects based on local contour features but also is capable of localizing the object in space and scale in the image. Fragments of contours could be a good idea to guess the object but here lies a question that how many fragments could be feasible.

Graph based detection. Model-based methods play a central role to solve different problems in computer vision. A particular important class of such methods relies on graph models where an object is decomposed into a number of parts, each one being represented by a graph vertex. Felzenszwalb and Huttenlocher, [12] have addressed the problem of segmenting an image into regions; this is achieved by defining a predicate in order to measure an evidence for a boundary between two regions by making use of a graph based representation of the image and by developing an efficient segmentation algorithm based on the predicate defined earlier. However finding a segmentation that is neither too coarse nor too fine is an NP-hard problem, hence there remains a huge scope in redesigning this method of image segmentation and to get good results. Dasigi and Jawahar, [13] have discussed a representation scheme for efficiently modeling parts based representation and matching them, as graphs can be used for effective representation of images for detection and retrieval of objects, the problem of finding a proper structure which can efficiently describe an image and can be matched in low computational expense remains a problem. They in their discussion have compared two graphical representations namely the Nearest-Neighbor Graphs and the Collocation Trees, for the goodness of fit and the computational expense involved in matching. A graph model based tracking algorithm which generates a model for a given frame termed as reference frame was used to track a target object in the subsequent frames.

Gunduz-Demir et al. [14] have presented a new approach to gland segmentation which decomposes the tissue image into a set of primitive objects and segments glands making use of

the organizational properties of these objects, which are quantified with the definition of object-graphs.

Fuzzy based detection. Kim et al. [15] have proposed an object recognition processor which lightens the workload by estimating the global region of interest (ROI). This estimation of ROI is performed by a neuro-fuzzy controller and this controller also manages the processors overall pipeline stages by using workload aware task scheduling. As pipelining is introduced here raises a question of parallel pipelining. Lopes et al. [16] have introduced an object tracking approach which is based on fuzzy concepts. The tracking task is performed through the fusion of these fuzzy models by means of an inference engine. Here the object properties considered are very basic, the properties like shape and textures etc. have not been considered.

Rajakumar et al. [17] have proposed a fuzzy filtering technique for contour detection; the fuzzy logic is basically applied to extract value for an image which is used for edge detection. In their approach, the threshold parameter values are obtained from the fuzzy histograms of an input image, and the fuzzy inference method selects the complete information about the border of the object. Their proposed system works for gray images, but the question whether this system is feasible under occlusion or cluttered image remains a question.

Context based detection. Perko and Leonardis, [18] have presented a framework for visual-context aware object detection; authors have tried to extract visual contextual information from images which can be used prior to the process of object detection. In addition, bottom-up saliency and object co-occurrences are used in order to define auxiliary visual context. Finally all the individual contextual cues are integrated with local appearance based object detector by using a fully probabilistic framework. This system is tested on still images, can it work on other types of images remains an issue.

Peralta et al. [19] have presented a method which learns adaptive conditional relationships that depend on the type of scene being analyzed. Basically they have proposed a model based on a conditional mixture of trees which is able to capture contextual relationships among objects using global information about an image. Relationships between objects in an image could be formed only when the image is clear enough but what if the image is occluded. Object categorization makes use of appearance information and context information in order to improve the object recognition accuracy. Galleguillos and Belongie, [20] have addressed the problem of incorporating different types of contextual information for object categorization and have also reviewed the different ways of using contextual information for object categorization. Contextual information would be accurate, once the images are labeled which will not be the case always hence efficiency of this approach could be an issue.

Other Mechanisms. Torrent et al. [21] have proposed a framework to simultaneously perform object detection and segmentation on objects of different nature, which is based on a boosting procedure which automatically decides – according to the object properties – whether it is better to give more weight to the detection or segmentation process to improve both results. Their approach allows information to be crossed from detection to segmentation and vice versa. The timing of this task may increase if initially the object detected is not the one of interest.

Hussin et al. [22] have discussed about the various techniques on how to detect the mango from a mango tree. The techniques are color processing which is used as primary filtering to eliminate the unrelated color or object in the image. Besides that, shape detection are been used where it will use the edge detection, Circular Hough Transform (CHT). Laptev, [23] presented a method for object detection that combines AdaBoost learning with local histogram features. He had introduced a weak learner for multi valued histogram features and also analyze various choices of image features. Histogram based descriptors can be feasible only when the image is natural and clear. It may not be feasible when the image is occluded or cluttered.

Steiner tree. Hambrusch and TeWinkel [24] have considered the problem of determining a minimum cost rectilinear Steiner tree when the input image is an $n \times n$ binary image I which is stored in an $n \times n$ mesh of processors. They have tried to make their work cost effective by avoiding sorting and routing operations that are expensive in practice. They have also presented parallel algorithms for the Steiner tree problem when an $n \times n$ binary image I which is stored in an $n \times n$ mesh of processors with one pixel per processor. Lin et al. [25] have developed an Obstacle Avoiding Rectilinear Steiner Minimal Tree (OARSMT) which when given a set of points and a set of obstacles on a plane, the OARSMT connects the pins, possibly through some additional points called the Steiner points and avoids running through any obstacle to construct a tree with minimal total wire length.

Liu and Sechen [26] have presented a chip-level global router based on routing model for the multilayer macro-cell technology. The routing model uses a three-dimensional mixed directed/undirected routing graph, which provides not only the topological information but also the layer information. The irregular routing graph closely models the multilayer routing problem, so the global router can give an accurate estimate of the routing resources needed. Router searching is formulated as the Steiner problem in networks (graph Steiner tree problem).

Apart from above related work, in [27] a detailed literature review for the various kinds of detection mechanisms is carried out.

3. PROPOSED APPROACH

Although for human beings, the recognition of familiar objects of any kind and in any sort of environment may be a simple task, but the process of recognition is still a huge difficulty for computers. Especially the situations where there are changes in light or there is some sort of movements in space make images of a same kind look entirely different. On the other hand, the number of instruments that are able to capture images from day to day life has increased drastically. And hence as a result, object detection has become a real challenge, in particular to be able to classify such huge amounts of data. Most of the approaches treat object detection as a complex process that requires powerful computers to run, the aim of the presented approach is to recognize the object present in the image and at the same time classify the object that is obtained through the recognition step.

Presented work basically deals with detecting and classifying the objects in images using a Steiner tree. To classify the objects in the images is modeled as Steiner tree problem. Steiner tree problem basically deals with finding the minimum path between the given set of vertices. The sole aim in the Steiner tree problem is to minimize the cost of Steiner tree. As it is an optimization problem and NP-hard problem, the scope of research contribution exists. The basic sliding window approach for object detection analyses a large number of image regions (of the order of 50,000 regions for a 640x480 pixel image) to know which of the region may contain the object of interest. And in case of many applications there is a need for recognizing multiple object classes, and hence multiple binary classifier are required to run over each region and thus if 10 object classes need to be detected, the sliding window approach may require 500,000 classifications per image.

Hence there is a need for analyzing only those regions in a particular image that have a higher probability of containing the object of interest. In the presented approach a Steiner Tree based classifier is used to classify the objects present in a particular image. Here Steiner tree is used in order to decide upon the minimum path between the nodes present at the same level of the tree, whereas a Multi Scale Boosted Detector is used for recognizing the objects in the image. The detailed internal working of the approach for the purpose of classification of an object is explained in the Figure 3, i.e. the flow diagram of the presented approach and the implementation steps.

Implementation Steps for Object Detection and Classification:

Input: Image containing an Object of Interest

Output: Detected Object of Interest and the Class to which it belongs.

● **Training Phase**

Step 1: The user is required to enter the number of images for the training purpose.

Step 2: Now the user is asked to enter some description for the object present in the selected images so that the classifier knows what exactly the class of the object in the image is.

Step 3: Now, we get the detected region of the object along with the values for the four different parameters which are: Area, Eccentricity, Euler Number and Orientation for that particular image.

Step 4: Repeat this procedure for a certain number of images of a particular class of objects so that the classifier gets to know the range of values of the 4 parameters for each object class.

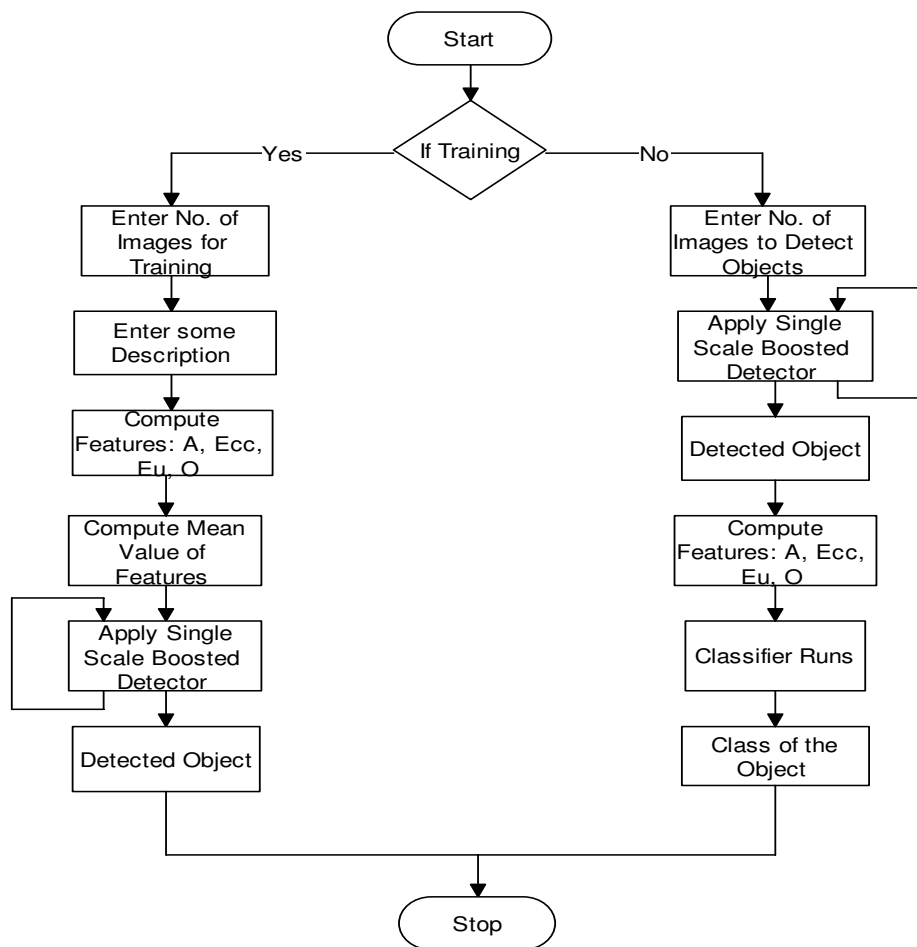


FIGURE 3: Detailed Flow Diagram of the Presented Approach and Implementation Steps.

● **Evaluation Phase**

Step 5: In this phase, the user is asked to enter the number of images in which he needs to detect the objects.

Step 6: Repeat step 3 so as to obtain the detected object of interest and to get the values for the parameters.

Step 7: The classifier makes use of the values of parameters obtained in step 6 so as to get the class of the object present in the image.

3.1 Object Detection

Distinguishing between the foregrounds objects from the stationary background image is a significant as well as a difficult research problem. Almost all the approaches for object detection or tracking have their first step as detecting the foreground objects. In order to detect the object present in the foreground, the presented approach makes use of a Gentle Boost Algorithm. Basically, in order to detect the objects present in the foreground, initially the classifier needs to be trained and only then the testing of the detector can be done. Hence the following part of this section explains the training and testing phase of the detector.

3.1.1 Detector Training

Detecting an object is a fundamental problem in computer vision: given an image, which object categories are present and where in the image are the objects located are some of the basic queries related to object detection. Almost all the best performing detection methods employ discriminative learning together with window based search, and assume that a large number of labeled training examples are available. For instance, thousands of bounding box annotations per class is a standard. More data is always advantageous and that is the reason researchers have began to explore various ways of collecting labeled data. The process of learning acts more informative for the detector in order to detect properly and improve the accuracy of detection.

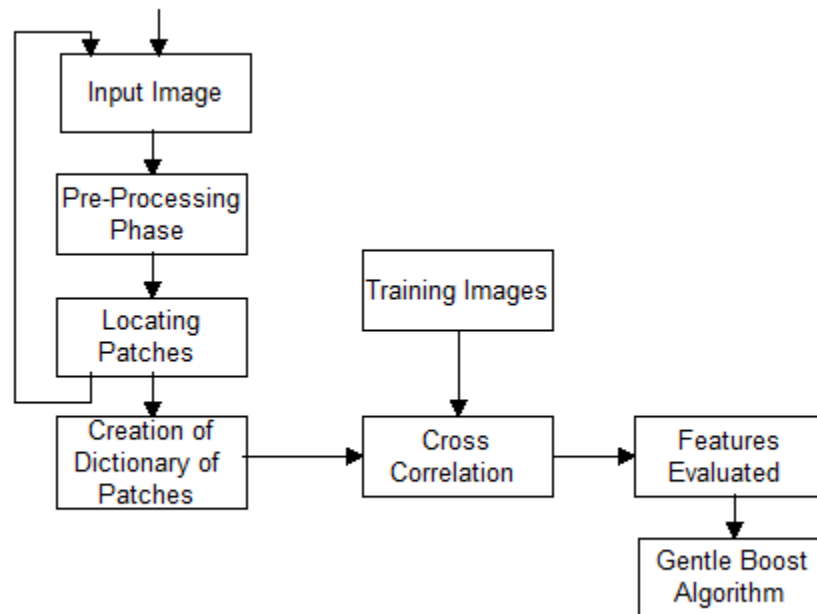


FIGURE 4: Steps for Detector Training.

Figure 4 depicts the way in which the detector is trained so as to get the accurate objects present in a particular image. During the training phase, initially in order to detect the objects of a specific class, 8 images in which one object of the class of interest appears is picked. In the pre-processing step, the x-derivative, y-derivative and Laplacian of the images are generated and are added to the original set of images. In the next step, a patch from the centre of the object is

sampled and is labeled as +1 where as the other patches from the background are labeled as -1. This process of labeling is repeated for various different patch sizes and is stored in a dictionary.

Further in the training phase, the cross correlation between the patch dictionary obtained earlier and the training images is calculated and the features at sample locations on the background region where the templates produce strong false alarms are recorded. And the positive samples are located at the centre of the object which is termed as the local maxima of the score in the object region. And finally the computed features are passed to the Gentle Boost Algorithm which in turn builds the detectors.

3.1.2 Detector Testing

During the training phase the presented approach has considered 200 images of a particular class. And hence the detector is trained on a total of 400 images belonging to two object classes and the features are computed for the images and stored which is used during the testing of the detector in order to see whether he detector is properly trained or not. Figure 5 shows the steps involved while testing the detector.

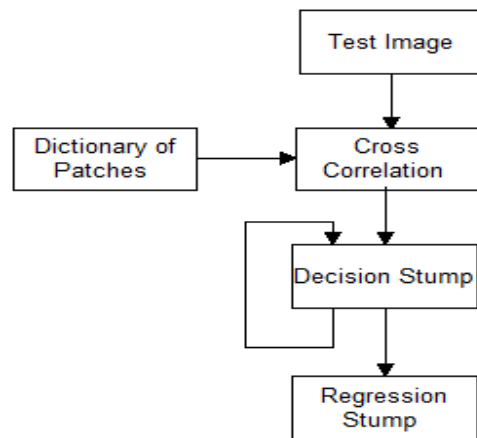


FIGURE 5: Steps for Detector Testing.

At testing time, the cross correlation between the test image and the dictionary of patches is computed first and then the regression stump is evaluated and is termed as a “weak learner”. And in order to obtain more accurate and efficient results, a number of weak detectors are combined to form a strong detector. This is basically done in order to detect an object having different viewpoints.

3.2 Object Classification

Approaches for visual classification basically proceed in two stages, firstly, features are extracted from the image and the object to be classified is represented using the obtained features. Secondly, a classifier is applied to the measured features to reach to a decision regarding the class of the obtained object.

3.2.1 Classifier Training

The presented approach makes use of the Steiner tree based classifier for the purpose of classifying the objects that are detected in the image. For the purpose of training the classifier, four features have been considered namely: Area, Eccentricity, Euler Number and Orientation. For every image, these four features are calculated and are stored. At the end of the training, the mean values for the various features are calculated.

The classifier in the training phase gets to know the range for all the four features that are calculated for every class of the object. This range will be used by the classifier to obtain the class for a particular class of the object.

3.2.2 Classifier Testing

During the testing phase, the image that is given as input has to go through the same steps that are carried out in the training phase. For an input image, the pre processing is done and the values for the four features are calculated. The values that are calculated for the different features are then matched with the mean values for each particular feature and accordingly the classifier makes use of the nearest neighbor evaluation in order to justify the class to which the detected object belongs to. The Figure 6 shows an overview of how this process is carried out and how the concept of Steiner tree comes into play.

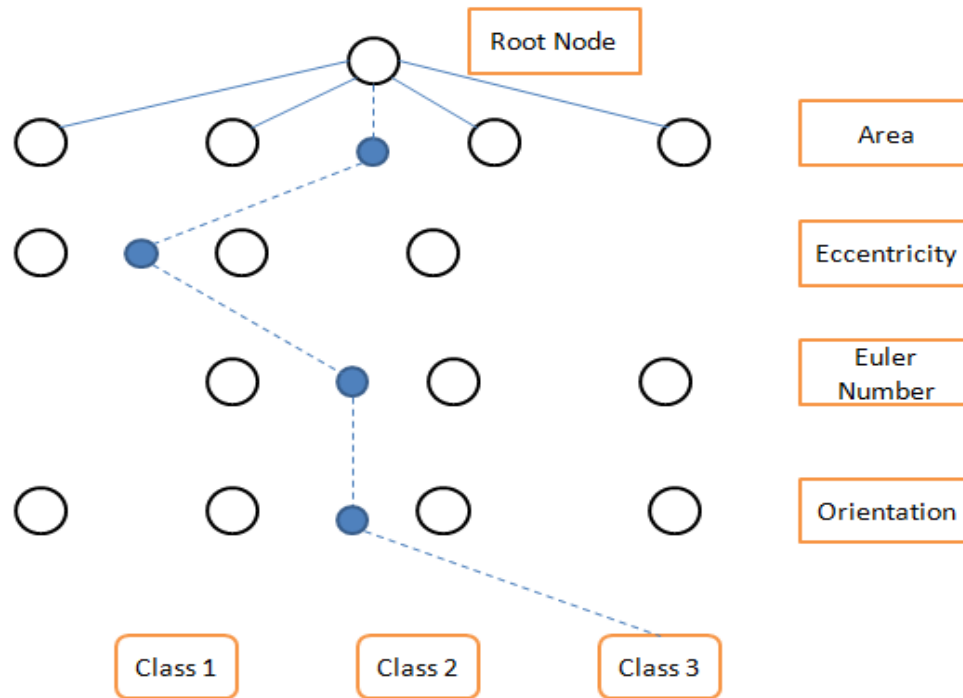


FIGURE 6: Decision Making Tree for Classifier.

Figure 6 comprises of four different levels, each corresponding to a specific feature. The sole aim of this phase is to get the class of the object that has been detected in the input image. And hence for this purpose, during the training phase of the classifier, the values of the features are calculated and the mean value is stored.

For an input image, during the testing phase, the values for the four features are calculated and are matched with those values which are obtained during the training phase. Now, if for an image, the value obtained for a feature is not present as a node, then in that case to avoid inaccurate results, the Steiner nodes are used. In the Figure 6, the colored nodes represent the Steiner nodes that would connect the different levels in the tree structure in order to reach the class of the detected object.

The presented approach makes use of the nearest neighbor evaluation to get the value of edge between the newly generated Steiner node and the MIN & MAX node of the value of each feature, like Area, on the same level of the graph. Figure 7 explains the working of the Steiner node.

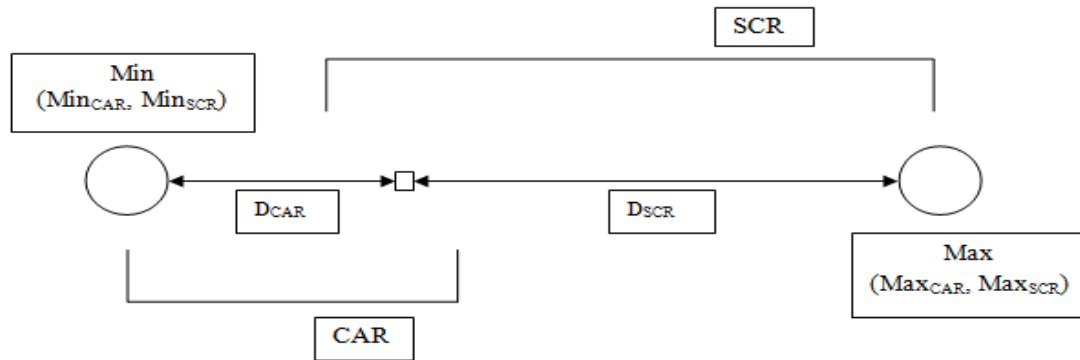


FIGURE 7: Finding the Shortest Path.

It can be seen from Figure 7, the range of classes is known a priori, but as can be seen there exists an overlapping region and if the value of the feature fall in the overlapping region then in that case the Steiner node evaluates on shortest path to the nearest node and decides the class of the object. And if the value of the feature does not fall in the overlapping region then the decision about the class of the object is done directly as the ranges are known. Based on the edge value, the decision of the class is drawn. If the edge value between the Steiner node and the MIN node is less than the edge value of Steiner node and the MAX node, then the concerned class of the Area value which is obtained for query image is 'CAR'.

In the above case, the two distances D_{CAR} and D_{SCR} are evaluated and the distance which is less will be the class to which the query object belongs. This procedure is performed at each level for the different features. And hence as the value of D_{CAR} is less, the class of the object is decided as 'CAR' and vice-versa.

4 EXPERIMENTAL RESULTS & DISCUSSION

4.1 Results

Initially in the pre-processing step, the patch from the centre of the object is sampled and is labeled as +1 while the other patches from the background are labeled -1. It can be seen from the Figure 8, the patch at the centre of the computer screen and car which are the objects in this case, are colored red representing +1 while the other patches on the screen are represented using the green color. This procedure is repeated for a sample of 8 images and is stored in the dictionary and hence is termed as dictionary of patches.

Further, during the training phase, the cross correlation between the dictionary patches and the training images are calculated along with the features at sample locations in the background region are recorded. The obtained features are then passed to a Gentle Boost Detector in order to build the detectors. After finishing off with the training of the detector, now the classifier needs to be trained, this is what is depicted in Figure 9. During the training of the classifier, the value for the four parameters namely: area, eccentricity, Euler number and orientation are calculated.

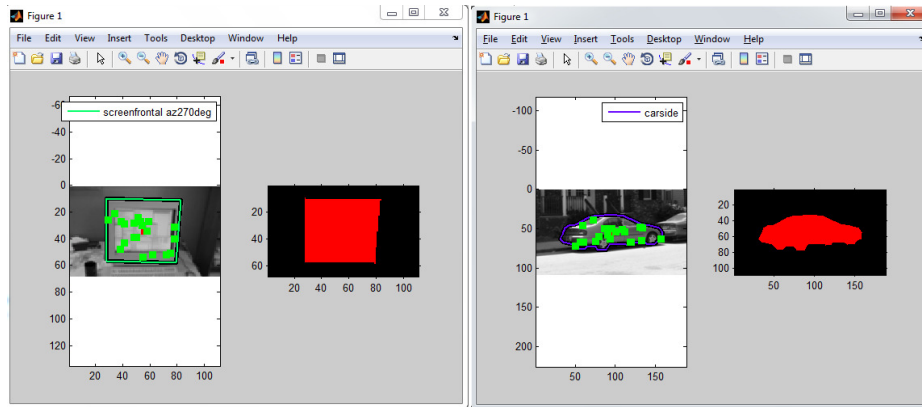


FIGURE 8: Patch Plotting.

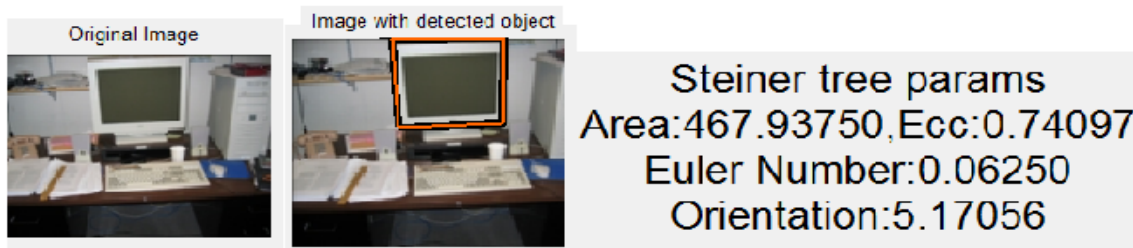


FIGURE 9: Training for the Classifier when class is Computer Screen.

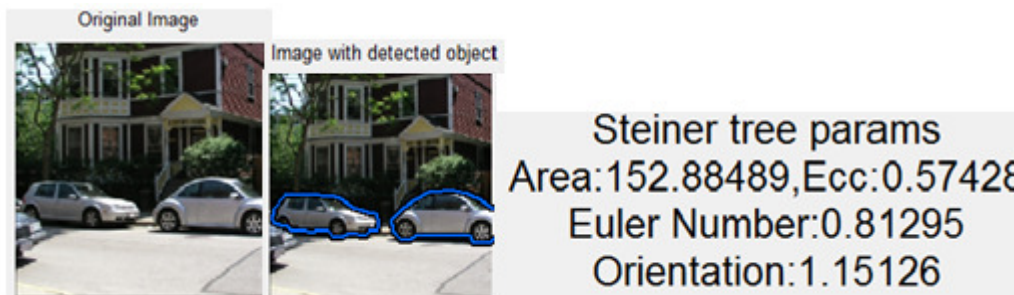


FIGURE 10: Training for the Classifier when Class is Car.

During the evaluation phase, the user enters the number of images he wants to detect and classify objects in and for each image, the values of the four features are obtained and those values are used by the classifier in order to finalize the class to which the object in the image belongs to as shown in Figure 10 and Figure 11. The classifier works on the principle of nearest neighbor evaluation (distance evaluation D_{CAR} and D_{SCR}) for each feature and in cases of a tie, the results are broken arbitrarily.

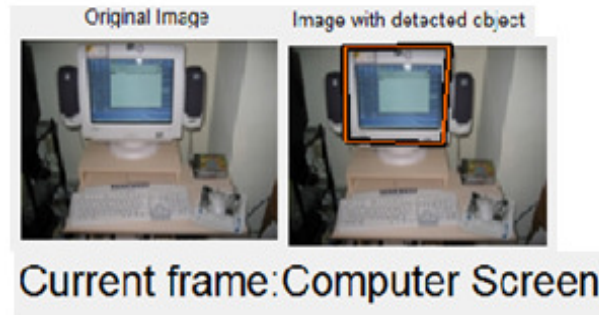


FIGURE 10: Evaluation of the Classifier for class Computer Screen.



FIGURE 11: Evaluation of the Classifier for class Car.

4.2 Discussion

In order to evaluate the performance of the approach the standard parameters: the precision, recall or F-measure (combination of both precision and recall) are considered. In information retrieval contexts, precision and recall are defined in terms of a set of retrieved documents and a set of relevant documents.

Precision:

In information retrieval contexts, precision is the fraction of retrieved documents that are relevant to the find:

$$precision = \frac{|\{relevant\} \cap \{retrieved\}|}{|\{retrieved\}|}$$

Precision takes all retrieved images into account, but it can also be evaluated at a given cut-off rank, considering only the topmost results returned by the system. This measure is called precision at n . For example, for a text search on the set of documents, precision is the number of correct results divided by the number of all returned results.

Recall:

Recall in information retrieval is the function of the documents that are relevant to the query that are successfully retrieved.

$$recall = \frac{|\{relevant\} \cap \{retrieved\}|}{|\{relevant\}|}$$

Number of Images	Recall	Precision
5	0.1	1
10	0.2	1
15	0.3	1
20	0.4	1
25	0.5	0.9
30	0.55	0.6
35	0.6	0.4
40	0.7	0.2
45	0.8	0.1

TABLE 1: Values of Precision and Recall for class Computer Screen.

Number of Images	Recall	Precision
5	0.1	1
10	0.15	1
15	0.18	0.9
20	0.2	0.8
25	0.3	0.65
30	0.55	0.4
35	0.65	0.3
40	0.8	0.2
45	0.9	0.1

TABLE 2: Values of Precision and Recall for class Car.

For the purpose of classification tasks, the terms like true positives, true negatives, false positives and false negatives. The terms positive and negative refer to the classifiers prediction, and the terms true and false refer to whether that prediction corresponds to the external judgment.

The performance of the presented approach is evaluated on the standard LabelMe dataset. The 360 images remaining after the training of the detector are used for training and evaluation of the classifier. The images are scaled down to 256x256 resolutions. Values of Precision and Recall for Computer Screen and the Car class are summarized in the Table 1 and Table 2 respectively, for the nine experimental simulation runs. Precision and Recall curves are plotted for the Computer Screen and Car classes which are shown in Figure 11. The values of precision and recall share an inverse relationship between themselves

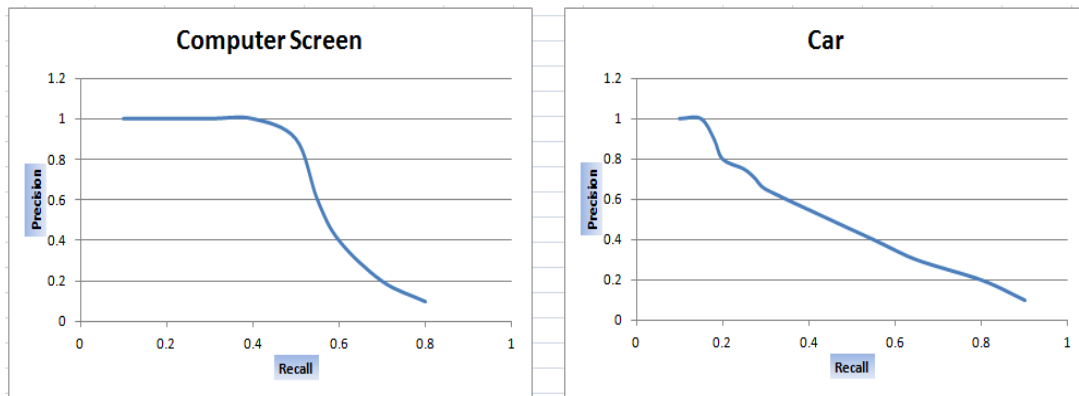


FIGURE 11: Precision and Recall Curve for Computer Screen and Car Classes.

5 CONCLUSION

The presented approach is for the detection of the objects present in the image and classification of the detected object. The presented approach makes use of Gentle Boost Detector in order to detect the object in an image and apart from this in order to classify the object in the image; Steiner tree based classifier is made used. Although there are many approaches developed for detecting objects, the presented approach uses the Steiner tree for classification purpose. The detection process is improved in accuracy by making use of multiple weak classifiers and the process of classification makes use of the concept of Steiner tree. Based on the results obtained for the precision and recall, it is observed that the presented approach is performing well for the different images. With reference to the experiments carried out for the detection and classification of the query image of the Car and Computer Screen class, the data collected through different simulations justify the accuracy of the presented approach.

6 REFERENCES

- [1] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge Results, 2007-2009.
- [2] M. R. Garey and D. S. Johnson. Computers and Intractability: A guide to the theory of NP-completeness. Freeman, San Francisco, 1978.
- [3] M. Charikar, C. Chekuri, T. Cheung, Z. Dai, A. Goel, and S. Guha. Approximation algorithms for directed Steiner problems. In Symposium on Discrete Algorithms, 1998.
- [4] L. Zosin and S. Khuller. On directed Steiner trees. In Symposium on Discrete Algorithms, 2002.
- [5] S. Segvic, Z. Kalafatić, and I. Kovačec, 'Sliding window object detection without spatial clustering of raw detection responses', Proceedings of the Computer Vision Winter Workshop, 2011.
- [6] B. Subburaman, Venkatesh and S. Marcel, 'Fast Bounding Box Estimation based Face Detection', ECCV, Workshop on Face Detection, 2010.
- [7] F. Comaschi, S. Stuijk, T. Basten, and H. Corporaal, 'RASW: a Run Time Adaptive Sliding Window to Improve Viola-Jones Object Detection', IEEE Transactions, 2012.
- [8] P. Viola and M. J. Jones. Robust real-time face detection. IJCV, Vol: 57, No: 2, 2004.
- [9] X. Yang, H. Liu, and L.J. Latecki, 'Contour Based Object Detection as Dominant Set Computation', Journal on Pattern Recognition, Vol.45 No.5, pp.1927-1936, 2012

- [10] K. Amine and M.H. Farida, 'An Active Contour for Range Image Segmentation', *Signal & Image Processing: An International Journal (SIPIJ)* Vol.3, No.3, 2012.
- [11] J. Shotton, A. Blake and R. Cipolla, 'Multi Scale Categorical Object Recognition Using Contour Fragments', *IEEE Transactions of Pattern Analysis and Machine Intelligence*, Vol.30 No.7, pp.1270-1281, 2008.
- [12] P.F.Felzenszwalb and D.P.Huttenlocher, 'Efficient Graph Based Image Segmentation', *International Journal of Computer Vision*, Vol.59, No.2, pp.167 – 181, 2004.
- [13] P. Dasigi and C.V. Jawahar, 'Efficient Graph Based Image Matching for Recognition and Retrieval', *Proceedings of National Conference on Computer Vision, Pattern recognition*, 2008.
- [14] C. Gunduz-Demir, M. Kandemir, A.B. Tosun and C. Sokmensuer, 'Automatic Segmentation of Colon Glands using Object-Graphs', *Medical Image Analysis*, Vol. 14, pp.1-12, 2010.
- [15] J. Kim, M. Kim, S. Lee, J. Oh, S. Oh and H. Yoo, 'Real-Time Object Recognition with Neuro-Fuzzy Controlled Workload-Aware Task Pipelining, Micro, IEEE, Vol. 29, No.6, pp. 28-43, 2009.
- [16] N.V. Lopes, P. Couto, A. Jurio and P. Melo-Pinto, 'Hierarchical Fuzzy Logic Based Approach for Object Tracking', *Knowledge-Based Systems*, Vol.54, pp. 255-268, 2013.
- [17] T.C. Rajakumar, S.A. Perumal and N. Krishnan, 'A Fuzzy Filtering Model for Contour Detection', *ICTACT Journal on Soft Computing*, Vol.01, No.04, 2011.
- [18] R. Perko and A. Leonardis, 'A framework for visual-context-aware object detection in still images', *Computer Vision and Image Understanding*, Vol.114, pp. 700-711, 2010.
- [19] B. Peralta, P. Espinace and A. Soto, 'Adaptive Hierarchical Contexts for object recognition with conditional mixture of trees', *Proceedings British Machine Vision Conference*, pp. 121.1-121.11, 2012.
- [20] C. Galleguillos and S. Belongie, 'Context Based Object Categorization: A critical survey', *Computer Vision and Image Understanding*, 2010.
- [21] A. Torrent, X. Lladó, J. Freixenet and A. Torralba, 'A Boosting Approach for the Simultaneous Detection and Segmentation of Generic Objects', *Journal of Pattern Recognition Letters*, Vol: 34, No: 13, pp. 1490-1498, 2013.
- [22] R. Hussin, M.R. Juhari, N.W. Kang, R.C. Ismail and A. Kamarudin, "Digital Image Processing Techniques for Object Detection from Complex Background Image". In *International Symposium on Robotics and Intelligent Sensors*, Volume: 41, pp. 340-344, 2012.
- [23] I. Laptev, 'Improving Object Detection with Boosted Histograms', *Journal of Vision Computing*, Vol: 27, No: 5, pp. 535-544, 2009.
- [24] E. Susanne, Hambrusch and L. TeWinkel, "Parallel Heuristics for Determining Steiner Trees in Images". In *Computer Science Technical Reports*, Report No. 90-1033, 1990.
- [25] C. Lin, S. Chen, C. Li, Y. Chang and C. Yang, "Obstacle-Avoiding Rectilinear Steiner Tree Construction Based on Spanning Graphs". In *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 27, Issue. 4, pp. 643-653, 2008.

- [26] L.E. Liu and C. Sechen, "Multilayer Chip-Level Global Routing Using an Efficient Graph-Based Steiner Tree Heuristic". In IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Vol. 18, Issue. 10, pp. 1442-1451, 1999.
- [27] K.U. Sharma and N.V. Thakur, "A Review and an Approach for Object Detection in Images", International Journal of Computational Vision and Robotics, Inderscience Publisher, 2014. (Submitted)

An Application of Eight Connectivity based Two-pass Connected-Component Labelling Algorithm For Double Sided Braille Dot Recognition

Shreekanth T

*Research Scholar,
JSS Research Foundation,
Mysore, India.*

speak2shree@gmail.com

V. Udayashankara

*Professor,
Department of IT,
SJCE, Mysore, India.*

v_odayashankara@yahoo.co.in

Abstract

The intrinsic noise present in the image during the acquisition phase marks the recognition of Braille dots a challenging task in Optical Braille Recognition (OBR). Further, while the Braille document is being embossed on either side in the case of Inter-Point Braille, this problem of Braille dot recognition is aggravated and it makes the differentiation between recto (convex) dots and verso (concave) dots more complex. Also, the recognition of Braille dots should be carried out by reading information recorded on both sides of paper by scanning only one side. This work proposes a novelty to circumvent this issue for distinguishing convex points from concave points even if they are adjacent to each other by using only the shadow patterns of the dots and by employing the connected component labelling using two-pass algorithm and the eight connectivity property of a pixel. Enthused by the fact that, during the acquisition phase, the reflection of light through the verso dots results in a high pixel count for them when compared to the recto dots, this technique works perfectly well with good quality Braille. Furthermore, due to the natural problems like ageing and frequent usage of the document the Braille dots tend to deteriorate resulting in the down fall of the performance of the algorithm for the Braille image. Besides to this for the recognition of the Braille cell in a Braille document with some special cases an adaptive grid construction technique has also been proposed. The results extracted reveal that the enactment of the proposed technique is much consistent and dependable and that the accuracy is very much comparable to the modern state of the art techniques.

Keywords: Braille, Connected Component Labelling, Eight-Connectivity, OBR, Recto, Verso.

1. INTRODUCTION

Braille is a form of written language for blind people, in which characters are represented by patterns of raised dots that are felt with the fingertips. The Braille system, devised in 1821 by Frenchman Louis Braille consists of patterns of raised dots arranged in cells of up to six dots, with each cell representing a letter, numeral or punctuation mark. The dots in each cell are arranged in three parallel rows having two dots each. The dot positions are identified by numbers from one through six. Sixty-four combinations are possible using one or more of these six dots.

In order to establish a bi-directional communication between the visually impaired and the sighted community workable, it is required to transliterate the Braille documents to the text document in the corresponding language. Optical Braille Recognition comprises of acquiring and processing the images of Braille documents for the purpose of converting the embossed Braille characters into their corresponding natural language characters. The need for OBR is that there are significant number of old Braille documents that need to be reproduced so that they can be

preserved and accessed by more people. Like other documents Braille documents need to be converted to digital format in order to facilitate storage, maintenance, duplication and text to speech conversion. Everyone who works with blind people and does not read Braille will benefit from using the OBR. The main reason for developing a system that can read Braille is to preserve and multiply large volumes of manually crafted books. Many books on mathematics or music are very difficult even for skilled copyist to retype due to the specific rules that apply in Braille.

As the dots on both sides of the page are visible from one side, both sides of the page can be recognised in a single scan. Printed Braille documents are very bulky. To mitigate this problem, most Braille documents are printed in inter-point with the embossing done on both sides of each page with a slight diagonal offset to prevent the dots on the two sides from interfering with each other. This makes the translation process more difficult. Depending on the presence of protrusions and depressions the Braille document can be classified as single sided and inter point Braille. If the document contains only the protrusions on single side then it is a single sided Braille document as shown in Fig.1 (a). If the document contains the protrusions and depressions on single side or if the document contains protrusions and depressions on both the sides then it is a double sided Braille document and is as shown in Fig.1 (b). Double-sided output takes less space and uses less paper as compared to single sided Braille document since the information lies on both sides of the Braille document thus volume can be reduced to half of the single sided Braille volume. All dots on a Braille page should fall on an orthogonal grid. When texts are printed double sided (Inter-point), the grid of the inter-point text is shifted so that the dots fall in between the primary side dots. During the recognition of double sided Braille document, the presence of protrusions and depressions may cause interference to recognition system. The dimensions of a Braille dot have been set according to the tactile resolution of the fingertips of person. For both the single sided and double sided Braille document the dot height, cell size and cell spacing are always uniform.

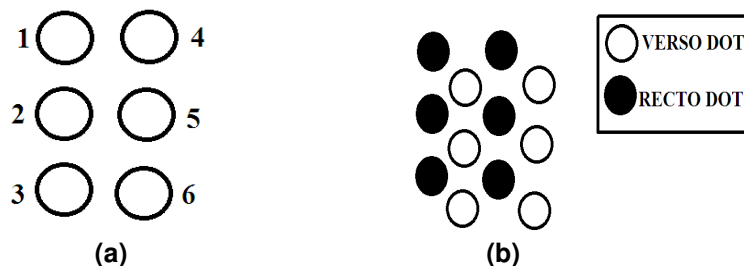


FIGURE 1: (a) The Braille Cell (b) Inter-point Braille.

Several researchers have made efforts to recognise single sided Braille documents and the recognition rates are to the acceptable level [3],[8],[9],[12]-[15],[18] and [19]. An extensive literature survey indicates that as the recognition of single sided Braille documents is easier when compared to the double sided Braille document, lesser number of works has been done on recognition of Inter-point Braille document. Several authors have as well proposed numerous techniques to differentiate the recto and verso dots in an inter-point Braille document [1], [4]-[7], [10], [11], [16], and [17].

In [1] J. Mennens et.al presented a very first approach for double sided Braille page recognition. This work throws light on the assumption as to how the dots of a digitized Braille page are represented with light and dark areas. Then the areas are classified by making a three value image. The resulting image has five values: regions with value +2(recto) and -2(verso) are called core regions, regions with value -1(recto) and +1(verso) are called side-lobes and 0 is background. Grid lines are then searched by making histograms of rows and columns.

In [4] R.T Ritchings et.al developed a prototype system that after the Braille document is scanned, the system proceeds to identify the location of the protrusions and depressions by

exploiting the differences in gray levels in the image. These arise during scanning from the reflected light and the shadows created by the protrusions and the depressions on the document surface. The protrusions first generate a dark area and then a lighter one in the scanning direction (vertical). The algorithm to identify protrusions and depressions proceeds by making an allowance for the dark regions of each dot and check is made to determine whether a light region exits above it. If it does exist within the limits of the predictable Braille character height then a depression is found, if not a check is made to govern whether a light area exits beneath the dark one. Correspondingly if one exists above within the vertical limit, a protrusion is found. If an area is found to be comprising of a protrusion or a depression, then those areas are marked as used and are not considered again and on the other hand regular spacing between the Braille dots and cells are used for Braille character recognition.

In [5] Yoshifumi oyama et.al proposed a method for distinguishing convex points from concave points by the highlight and shadow patterns generated by exposing the Braille points to LED lights. This work used the difference in the strength of the reflected light from convex and concave points to separate the convex points from the concave points. For the Braille character extraction as a substitute for conventional 2*3 point set mask here 3*3 i.e. 9 windows are used as the positioning performance of the mask that determines the character recognition accuracy.

In [6] C M Ng et.al proposed a regular feature extraction for recognition of Braille. The illumination characteristics of the recto and verso dots are efficiently made use of by this algorithm. Expending these illumination characteristics the position of the illuminated hole can be castoff as the feature to make a distinction between the front faced dots and the back faced dots. Based on the boundary co-ordinates information and the illumination characteristics, two standard templates were constructed to represent the front-face dots and the back face dots. To assess the correlation at each pixel position these templates were applied to every position of the image. The front faced and the back faced dots are then extracted depending on the correlation values attained.

In [7] Antonacopoulos et.al. Proposed an algorithm for double sided Braille dot detection which is analogous to the one debated above. At this juncture a novel technique is proposed for grid formation. The system described here constructs a relatively flexible grid by allowing variations in the position of characters between different lines. First, the grouping of dots that have the same y co-ordinates are done in order to identify the rows of dots. Having identified the rows of Braille dots, a frequency histogram of the vertical distances between adjacent rows is calculated. The histogram ought to have two main peaks, one indicating the inter-character vertical distance between dots and other the inter-line distance. i.e., vertical distance between the bottom row of a Braille character line and the top row of the next.

In [10] Abdul Malik Al-Salman et.al developed an Arabic OBR system that is competent enough to recognize both single-sided and double-sided Arabic Braille documents from a single scan. This algorithm takes into account the implication that if the dark region comes at the top and the bright one comes at the bottom then it is a verso dot, the inverse situation results in a recto dot. Also it takes into account that the average dot height is 8 pixels. Bright pixels are allotted the value +1 and dark pixels are assigned the values -1. If pixel (1) + pixel (2) < 0 and pixel (7) + pixel (8) < 0 then this is part of a recto dot. If pixel (1) + pixel (2) < 0 and pixel (7) + pixel (8) > 0 then this is part of a verso dot. At this point the Braille cell recognition is done using the horizontal and vertical projection and as well the average distance between the rows holding the dots and between the columns holding the dots.

In [11] Abdul Malik S, Al-Salman presented an innovative algorithm for Braille cells recognition using image processing technique. The Braille image segmentation is done by means of a stability thresholding method with a beta distribution. A grid containing the Braille dots is moulded to ensure accurate detection and extraction of dots composing Braille cells. Then the recto dot is identified by a light region that exits underneath a dark region. Once the recto dots and verso dots are being recognised, Braille cells are then identified based on the standard regrouping of dots.

In [16] Amany-al-soleh et.al proposed a method for dot detection of Braille images using a mixture of beta distributions. In this work it is presumed that the scanned Braille page consists of three classes of pixels; a mid-gray background, a pair of light area and dark area for each recto and verso dot. At first in order to segment the scanned Braille image into three classes, thresholding is proposed onto it. Then the initial threshold values T_1^0 and T_2^0 for stability thresholding is estimated. To do this the maximum value of the histogram is calculated first. T_1^0 will be the average gray level value of image starting from 0 to maximum value, T_2^0 will be the average gray level value of image starting from maximum value to 255. The stability thresholding procedure is repeated until the error is zero. By the culmination of this algorithm the optimal values of T_1^{New} and T_2^{New} are obtained. Then for detecting the recto and verso dots from double sided Braille documents a grid is formed first to accomplish the detection of dots for every box in the grid. Further to decide whether it holds a recto dot or a verso dot a test is being carried out. The test checks if the upper half contains a light region and the lower half contains a dark region. If this was the case then this is considered a recto dot and will be drawn on the output range in the same location.

In [17] Bhattacharya.J et.al proposed an algorithm which makes use of the regular inter-dot and inter-cell spacing. In order to detect each cell a sliding window with a fixed interval is used to slide over the entire Braille image. Each window consists of both side dots. These dots are differentiated as front or back sided through a sliding method. Firstly every window is sub divided into 3 regions R1, R2 and R3. All the dots which lie in region R1 entirely are accepted, whereas dots which lie completely in the region R3 are rejected. Dots lying in both R2 and R3 are either front or backside merged dots or front side dots. For the former case the dot centroid is modified using the merged dot centroid and bottom extreme point of the dot. Yet again dots lying in both R1 and R2 are either merged dots or backside dots. Here for the former case the dot centroid is modified using the merged dot centroid and bottom extreme point of the dot whereas for the later scan the dot is rejected.

The downsides of the above works are i) The increased average processing time introduced by the template matching procedure for differentiating the recto dots from verso dots ii) The error introduced due to the merging of Braille dots and thus causing an ambiguity and iii) The need for restructuring of templates depending on the height of the Braille character as the spatial resolution of the Braille image varies. Thus these techniques do not lead to an adaptive approach.

Driven by the above actualities, an attempt has been made in the present work to cultivate an adaptive algorithm using the concept of eight connectivity based two-pass connected-component labelling algorithm to differentiate the recto and verso dots from the double sided Braille document. The experimental results show that the proposed technique can deliver an enhanced performance when equated to the other techniques mentioned for Recto and Verso Braille dots separation. This paper is divided into four sections: wherein the section II discusses in detail the projected work. Section III presents the results and discussions. Section IV, offers conclusions and directions for the future work.

2. METHODOLOGY

The overview of the proposed OBR system is illustrated in Fig. 2. The objective here is to develop an optical Braille character recognition system which takes the different resolution of the scanned Braille document and the different Braille image quality into deliberations. The proposed system can be developed as follows

2.1 Extraction of Shadow Patterns and Median Filtering

The Braille images are scanned using the HP Scan jet 3400C A4 size scanner, with the image resolution of 200dpi, the spatial resolution of 1501 x 2121 and with the bit depth of 24 bit. The algorithm begins by converting the innate colour image to the gray scale image. The diverse gray levels result in the image which is due to the variations of the reflected light and the shadows created by the protrusions and the depression on the document surface during the scanning

process. In the scanning direction the protrusions generated dark area first and then a lighter one. It was our observation that the depression produced the opposite. The shadow produced by the depressions relatively account for a lesser number of pixel count when compared to the shadow produced by the protrusions. This motivated us to retain only the light areas of the protrusions and depressions and thus eliminating the dark regions. With a motive to preserve only the light areas of the dots and to remove any inherent noise present in the image thresholding is being performed on the gray scale image.

The thresholding function used to do this is given below:

$$Y(I, J) \begin{cases} = X(I, J); & \forall X(I, J) > \max(X(I, J)) - 10; & 1 \leq i \leq M, 1 \leq j \leq N \\ = 0; & \forall X(I, J) \leq \max(X(I, J)) - 10; & 1 \leq i \leq M, 1 \leq j \leq N \end{cases} \quad (1)$$

Those pixels having threshold value less than this are made zero and those pixels with values greater than threshold are retained as they are.

The impulse noise like components contained in a thresholded image is eliminated using the median filtering approach. Now the median filtered image contains only the recto and verso dot components. Further the morphological dilation is performed in order to increase the area of the dot and this is more effective for increasing the area of the verso dot as compared to recto dot. In order to differentiate between the recto and verso dot the eight connectivity property of the pixel relationship is employed. It is perceived that the eight connectivity pixel count for the recto dot is less compared to the verso dot. Then to differentiate between the recto and verso dots, the thresholding operation is performed on the basis of the eight connectivity based pixel count value. The scanned document sample is shown in Fig. 3 and also the portion of the recto and verso dots are shown in Fig. 4(a) and (b) respectively. It can be observed that the dots on the front side are protruded above the page and those on the back side form depressions. These concave and convex characteristics of the dots reflect the light into two different angles, creating a light region at the top half of the captured dots for front sided Braille dots and at the bottom of for those back sided dots [6].

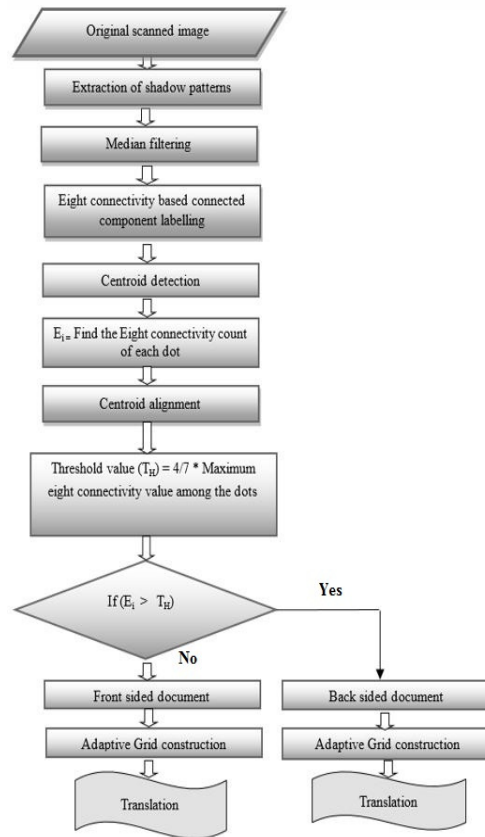


FIGURE 2: Flowchart of the Braille Recognition System.

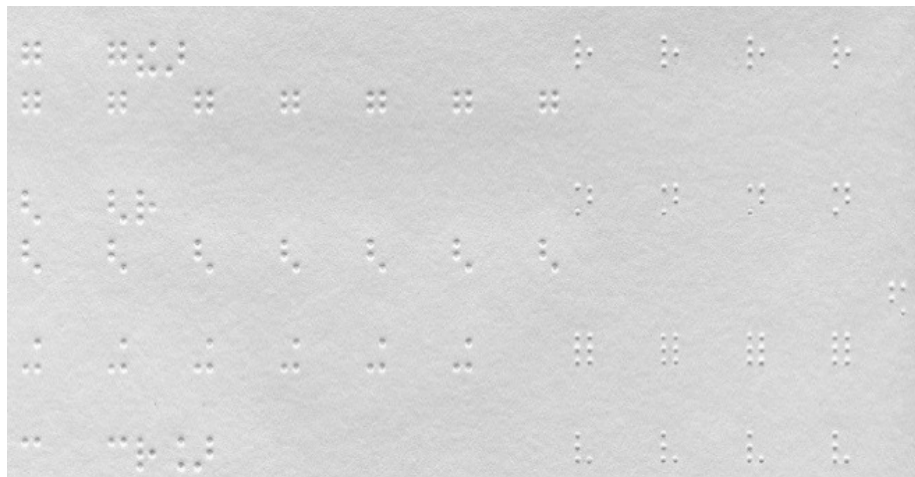


FIGURE 3: Original Scanned Braille Image.

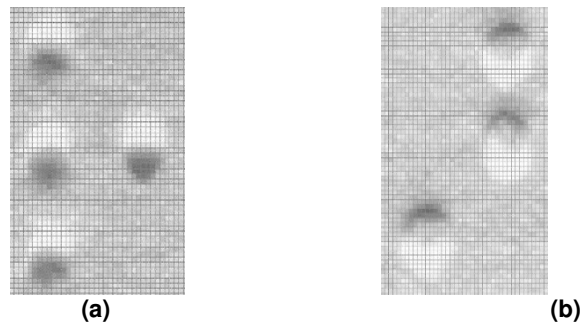


FIGURE 4: a) Recto Dot: Light area first followed by the dark area b) Verso Dot: Dark area first followed by the light area.

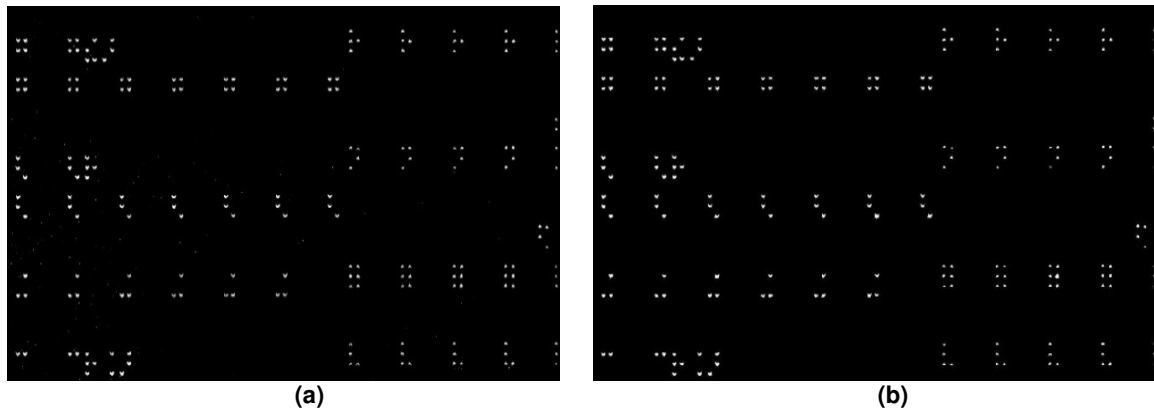


FIGURE 5: a) Thresholded image with impulse noise b) Median filtered image.

2.2 Eight connectivity based two pass connected component labelling algorithm and centroid calculation

If any pixel is connected horizontally, vertically and diagonally then it is called an eight connected pixel. For any pixel $p(x, y)$ the definition of an 8-connected component is that the pixel $p(x, y)$ should be connected with any of the pixels described below [21]:

$$(x+1,y),(x-1,y),(x,y+1),(x,y-1),(x+1,y-1),(x+1,y+1),(x-1,y-1),(x-1,y+1) \quad (2)$$

The extraction and labeling of various disjoint and connected components in an image is the most vital part in a number of automated image analysis applications. As the connected component labeling works either on binary or gray images, accordingly different measures of connectivity are possible. These images are typically the output from an alternative image-processing step, such as segmentation.

The process of grouping the *connected* pixels in an image for assigning an unmatched label to each object in an image is usually done by the *Connected Component Labeling* technique. The reason being that these labels are the key for various other analytical procedures, an indispensable part of most applications in pattern recognition and computer vision, such as character recognition. The basic approach is to scan the image and group its pixels into components based on pixel connectivity, *i.e.* all pixels in a connected component share similar pixel intensity values and are in some way connected with each other. There are two much known ways of defining connectedness for a 2D image: 4-connectedness and 8-connectedness. In this paper, we use the 8-connectedness as illustrated in expression (2). Once all groups have been determined, assign labels to each pixel until the labels for the pixels no longer change. As a result of the scan, no temporary label is attributed to the pixels belonging to different components but on the contrary different labels may be associated with the same component. Consequently,

equivalent labels are sorted into equivalence classes and a unique class identifier is designated to each class after the completion of the first scan. Then, a second scan is run over the image so as to substitute each temporary label by the class identifier of its equivalence class [20].

The number of pixels having the same label is counted and these values are used for discerning the recto and verso dots. The centroid of a labelled component is determined using the equations given below.

$$a(i,1) = 1 \quad \forall 1 \leq i \leq M \tag{3}$$

$$a(1,j) = i \quad \forall 1 \leq i \leq N \tag{4}$$

$$b(i,j) = a(i,1) * a(1,j) \tag{5}$$

$$\text{Area} = \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} z(i,j) \quad \forall 1 \leq i \leq M, 1 \leq j \leq N \tag{6}$$

$$\text{Mean } x = \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} [z(i,j) * b(i,j)] / \text{area} \tag{7}$$

$$c(i,1) = I \quad \forall 1 \leq i \leq M \tag{8}$$

$$a(1,j) = 1 \quad \forall 1 \leq i \leq N \tag{9}$$

$$d(i,j) = c(i,1) * c(1,j) \quad \forall 1 \leq i \leq M, 1 \leq j \leq N \tag{10}$$

$$\text{Mean } x = \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} [z(i,j) * b(i,j)] / \text{area} \tag{11}$$

$$\text{Mean } y = \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} [z(i,j) * d(i,j)] / \text{area} \tag{12}$$

The centroid extracted Braille image is shown in Fig.6. Either due to the little skewness of the document or due to the deteriorated Braille dots the centroids of the dots are not aligned properly. To circumvent this problem the centroids must be aligned vertically and horizontally by defining the threshold for the alignment. In this work we have used the following equation for aligning the centroids of the Braille dots.

$$x_{i+1} = x_i; \quad \forall x_i - \Delta x \leq x_{i+1} \leq x_i + \Delta x; \quad 1 \leq i \leq M \tag{13}$$

$$y_{i+1} = y_i; \quad \forall y_i - \Delta y \leq y_{i+1} \leq y_i + \Delta y; \quad 1 \leq i \leq N \tag{14}$$

This has been designed considering the very little skewness in the document as this work does not take the rotation angle into consideration. The eight connectivity component values are assigned to the respective centroid.

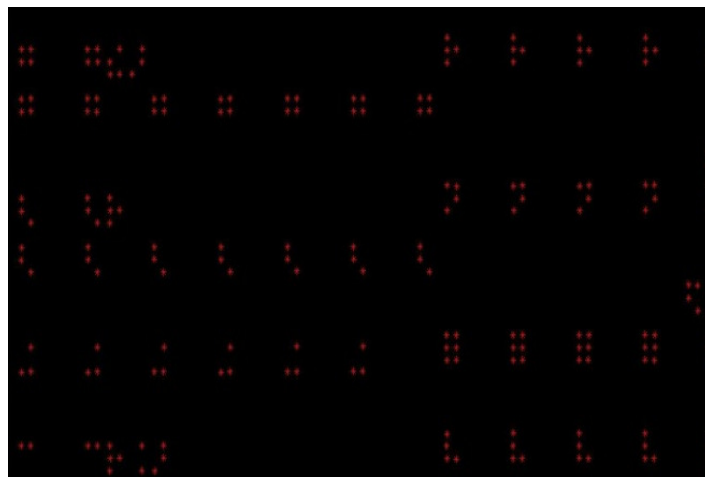


FIGURE 6: Centroid Detected Image.

2.3 Recto / Verso dot Deperation

The principal objective of this work is the separation of recto/verso dots. The incentive to execute this is the variance in the eight connectivity pixel count for the shadow region of the recto and verso dots. This is attributed due to the reflection property of the light. Also, this difference is independent of the spatial resolution of the Braille image. An in depth analysis has reviewed that the thresholding using the relation described below can be used for differentiating the recto dots from verso dots in an inter-point Braille image.

$$T_H = 4/7 * \text{Maximum eight connectivity count among the dots} \quad (15)$$

By considering the average eight connectivity pixel counts of un-deteriorated Braille dots from the developed Braille database this threshold value has been designed. Those dots with eight connectivity count value greater than the threshold T_H are considered as the verso dots and those dots for which the eight connectivity count value lesser than the threshold T_H are considered as the recto dots. The only detriment of the direct thresholding is that if the dots are deteriorated due to frequent usage and ageing then the eight connectivity count for such dots are less and thus it leads to false recognition of dots which is illustrated in Fig.7 (a) and (b). If the dots are deteriorated then the verso dots take the appearance of the recto dots and vice-versa. With the location of the dots being same as that of the original document, the recognised dots are then separated into recto and verso dots and placed in a different document. Each side of the Braille dots are transcribed into their corresponding natural text when the output of this stage is fed into the adaptive grid construction block.

2.4 Adaptive Grid Construction

After the separation of the recto and verso dots the grids are constructed discretely for the front and back sides of the document. The grid construction is essential to recognise a Braille cell in the Braille document. In former works the grids were constructed according to the standard Braille dimension and the summation of pixels within the dot frame was done to separate the recto dots from verso dots. This is a dreary and a time consuming process. Also, the constructed grids were not adaptive, as in it has not considered the different resolution of the scanned Braille document. Henceforth in this work we propose the adaptive grid construction technique which, in general can be applied to any sort of Braille document and is shown in Fig.8. This reduces the computer time and also any possible miss classification of the dot is avoided. In order to construct the adaptive grid an algorithm has been designed by considering the three factors such as, the distance calculation, horizontal projection profile and vertical projection profile of the Braille image.

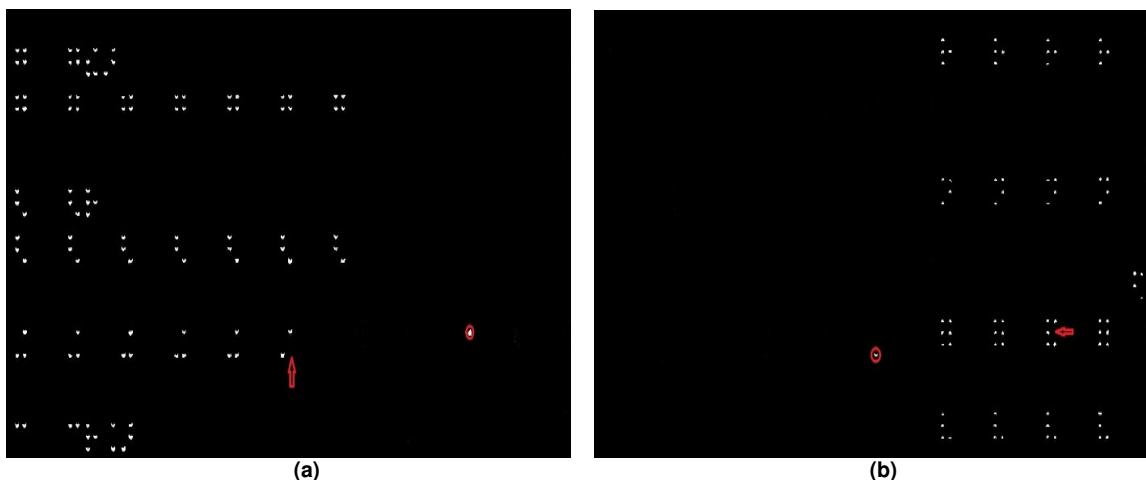


FIGURE 7: a) Recto dots extracted image. Dot enclosed by a circle is a part of verso dot and the arrow indicates the missing recto dot.

b) Verso dots extracted image. Dot enclosed by a circle is a part of recto dot and the arrow indicates the missing verso dot.

The Braille document to illustrate the process of adaptive grid construction is shown in Fig.9 (a). Here we have considered only the image consisting of recto dots, as the output from the previous step is a separate document consisting of either recto dots or verso dots only. We notice from Fig.9 (b) that, the third row of dots are missing from both line 4 and line 6. The solution to this has been incorporated and is described in the following.

It is our observation that any Braille document in general will satisfy the below characteristics

- Horizontal Inter-dot (HID) distance $> 2 \cdot \text{dot width}$ and $< 3 \cdot \text{dot width}$
- Horizontal Inter-cell (HIC) distance $> 3 \cdot \text{dot width}$ and $< 4 \cdot \text{dot width}$
- Vertical Inter-dot (VID) distance $> 2 \cdot \text{dot width}$ and $< 3 \cdot \text{dot width}$
- Vertical Inter-cell (VIC) distance $> 4 \cdot \text{dot width}$ and $< 5 \cdot \text{dot width}$

The diagram showing the Horizontal and Vertical Inter-dot and Inter-cell distances is as shown in the Fig.10. The horizontal and vertical projection profiles are drawn from the dot extracted image and are shown in Fig.11 (a) and (b). The horizontal projection profile is due to the sum of all the pixels in the row direction and vertical projection profile is due to the sum of all the pixels in the column direction. Three consecutive peaks in the horizontal projection profile indicates one complete line of a Braille document and two consecutive peaks in the vertical projection profile indicates one complete cell. From the horizontal and vertical projection profile information a grid is constructed as in Fig.12 (a). The constructed grid contains the information about the probable dot position only if the particular row or column contain the dot components. In real time all the Braille documents cannot possess all 3x2 dot information. In some cases it may be 2x2, 3x1 and so on. So as to solve this issue, distance calculation algorithm has been employed to locate the possible positions of the dots. The horizontal and vertical width of any dot in the document is calculated first and then the algorithm proceeds as follows.

It is found from the experimentation that the horizontal and vertical inter-dot distances are almost the same. However, the vertical inter-cell distance i.e., the distance between two consecutive Braille lines is slightly greater than the horizontal inter-cell distance i.e., the distance between two cells within the same Braille line.

The filled possible dot positions are shown in the Fig. 12(b). The above algorithm is used to complete the grid as shown in Fig. 12 (c). Fig.12(c) shows the concatenation of the extracted Recto/Verso dots with the adaptive constructed grid. Then the dots are scanned as shown to convert the grid to binary number followed by the decimal conversion.

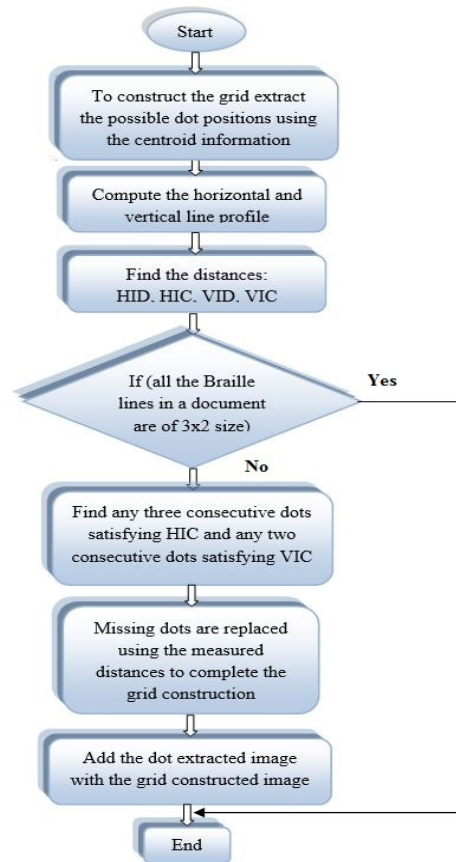


FIGURE 8: Flowchart of Adaptive Grid Construction.

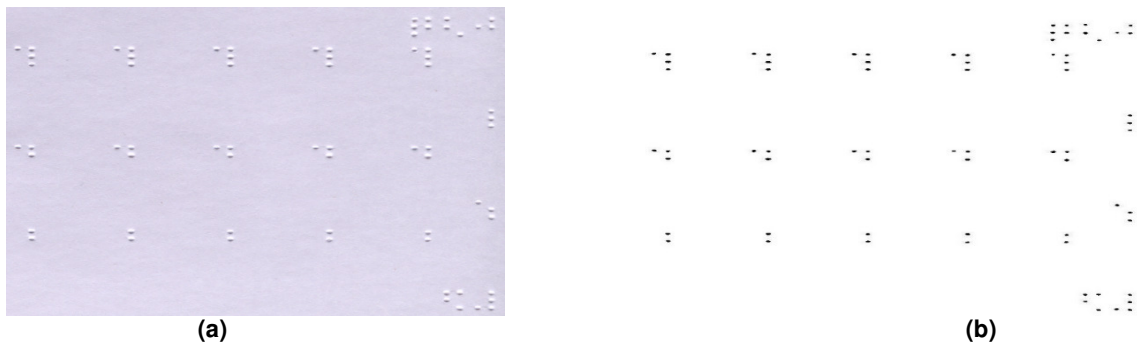


FIGURE 9: (a) Specimen document for illustrating the adaptive grid construction (b) Dot extracted image.

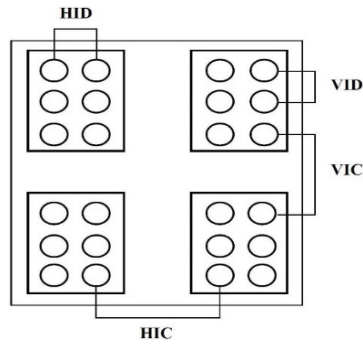


FIGURE 10: Horizontal and Vertical Inter-dot and Inter-cell distances.

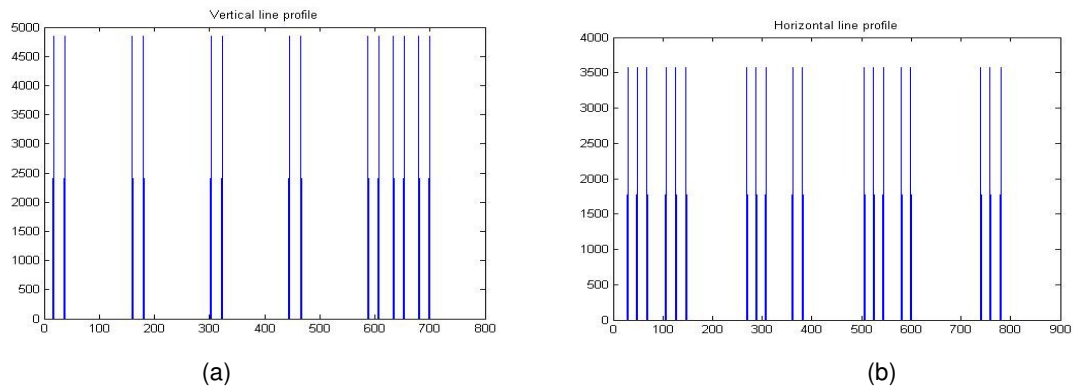


FIGURE11: (a) Vertical line profile and (b) Horizontal line profile for the image shown in Fig 11(b).

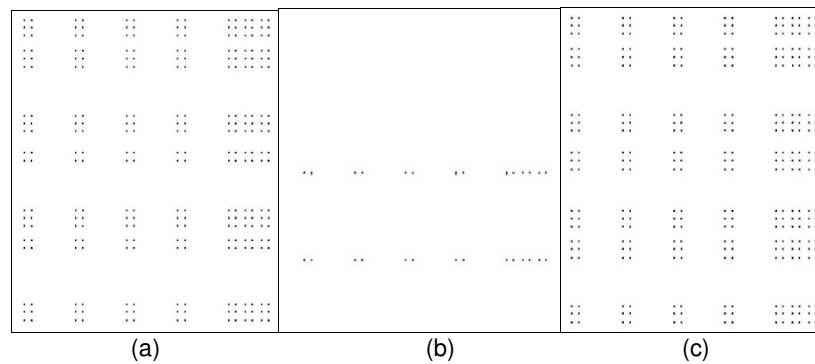


FIGURE 12: (a) Grid constructed using the possible dot positions, (b) Missing dots detected and (c) Showing the adaptive grid construction.

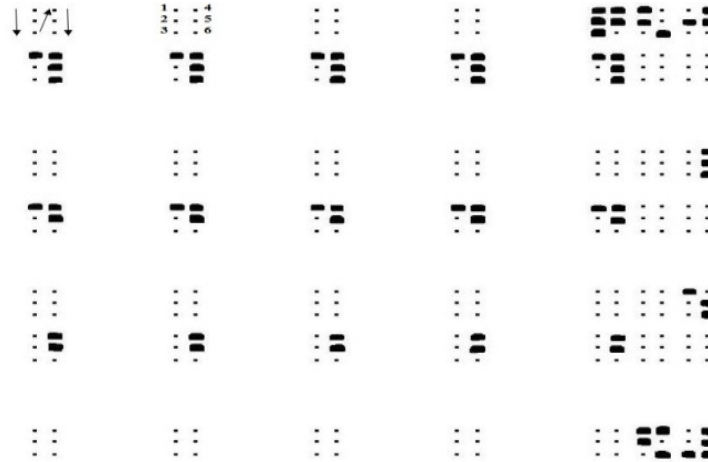


FIGURE 13: Overlapping of reconstructed grid with the recognised dots and the representation of the Braille cell scan pattern.

If there are three consecutive lines in the horizontal line profile satisfying the horizontal inter-dot distance and if there are two consecutive lines in the vertical line profile satisfying the vertical Inter-dot distance property then the algorithm does nothing. Only when Horizontal Inter-dot distance, Horizontal Inter-cell distance, Vertical Inter-dot distance and Vertical Inter-cell distance properties are violated the algorithm comes into picture. In order to address this matter an algorithm has been designed in such a way that the algorithm looks for any three dots satisfying the Horizontal Inter-cell distance and any two dots satisfying the vertical Inter-cell distance. Then if there are any missing dots of lines either in the horizontal direction or in the vertical direction it will be filled taking into considerations the distance calculation. This completes the adaptive grid construction.

Far ahead the grid constructed images and the dot extracted images are added to get an image as shown in the Fig.13. The presence or absence of all the valid Braille dots is found by multiplying the grid constructed image with the dot extracted image. If the product is true then it indicates the presence of dot and its value is indicated through 1 and if the product is false then it indicates the absence of dot and its value is indicated through 0.

During the dot recognition process all the valid Braille dots have been detected on either sides of the document and the two images are formed for each side of the document. Currently in order to convert the recognised dots into their corresponding natural characters, the scanning pattern as depicted in the Fig. is used for dividing each cell into grids consisting of six parts and corresponding code for each cell is generated according to the presence or absence of a dot in each grid. Here the binary 1 and the binary 0 represent the presence of the dot and the absence of the dot respectively. The dot positions are determined through number 1 to 6. The positions being universally numbered 1 to 3 from top to bottom on the left, and 4 to 6 from top to bottom on the right. Within each cell, the dot pattern is determined and is also represented by a bit string. These bit strings are then converted into their equivalent decimal codes by using the expression: $Decimal\ code = b_1 + b_2 * 2 + b_3 * 4 + b_4 * 8 + b_5 * 16 + b_6 * 32$ [10]. For example, Fig.14 (a) shows the Braille cell in which recognised dots are represented by black pixels and the absence of dots are represented by white pixels. Fig.14 (b) shows the dot position, bit strings and the equivalent decimal codes for the Braille cell shown in Fig.14 (a). In order to retrieve the natural characters corresponding to the Braille characters, a matching algorithm is employed in which, an input decimal code generated from the processed image could be searched against the lookup table wherein the Unicode corresponding to the Braille characters are being stored. These Unicode's are then converted into their corresponding natural text using the Matlab function `unicode2native`. The entire process has to be repeated for the other side of the Braille document too.

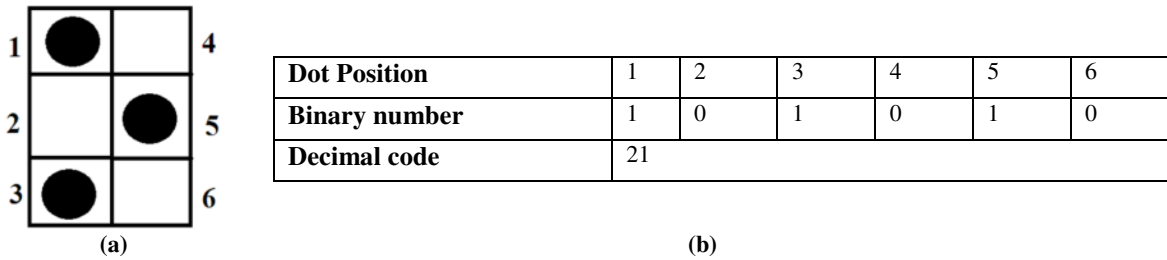


FIGURE 14: (a) Braille Cell after Dot recognition (b) Braille code generation.

3. RESULTS AND DISCUSSION

In this section, we present the materials on which the Recto and Verso Braille dot separation using Eight Connectivity based two-pass Connected-Component Labelling Algorithm and a Novel Thresholding Technique is evaluated, the performance measures used to evaluate the algorithm, and the results obtained. The methodology has been gauged quantitatively and qualitatively expending two locally established databases (DB1 and DB2). The database DB1 contains 25 colour images of the Inter-point Braille. This image set sans overlapping of Recto-Verso dots and Inter-dot deformation which is one of the vital hitches that is most commonly introduced during the Braille embossing process. Thus the Recto and Verso Braille dots from this database can be separated workably. The database DB2 contains 25 colour images of the Inter-point Braille consisting of scanned Braille documents with Inter-dot deformation and overlapping of Recto-Verso dots. The images of these two databases were captured by a HP Scan jet 3400C scanner. Table 1 gives the complete description of the databases used in this work.

Number of Braille Documents	50 (25 DB1 + 25 DB2)
Braille Document Type	Inter-Point Braille, Grade-1
Digital Format	24 bit Color Image
Resolution	300 dpi
Pixel Resolution	2300x1700
Image Format	Bit-Map (bmp)
Document size	26cm (Horizontal) , 30cm (Vertical)
Total number of dots in DB1	6359
Average number of characters per sheet in DB1	254
Total number of dots in DB2	14855
Average number of dots per sheet in DB2	594

TABLE 1: Description of the Braille Database created.

The dot separation using a novel combination of Eight Connectivity based two-pass Connected-Component Labelling algorithm and a novel thresholding technique is insensitive to majority of the noise present in the acquired image.

For evaluating the efficiency of the proposed method, we have considered four events; two classifications and two misclassifications. The classifications are the True Positive (TP) where a recto/verso dot is identified as recto/verso dot in both the ground truth and dot extracted image, and the True Negative (TN) where a recto/verso dot is classified as a non-dot in dot extracted image. The two misclassifications are the False Negative (FN) where a recto dot is classified as verso-dot in dot separated image but as a recto dot in the ground truth image, and the False

Positive (FP) where a verso dot is marked as recto dot in the dot separated image but as verso-dot in the ground truth image.

True Positive Rate (TPR) is the fraction of recto/verso dots correctly recognised as recto/verso dots respectively. True Negative Rate (TNR) is a fraction of recto/verso dots which are classified as non-dot in the dot extracted image. False Negative Rate (FNR) is the fraction of recto dots erroneously detected as verso-dots. False Positive Rate (FPR) is the fraction of verso-dots detected as recto-dots.

FNR and FPR may be attributed due to degradation of the dots which in turn is due to ageing of the Braille document. Ageing in sense as the Braille writing is read using the finger touch over the document, after multiple readings it is possible that the dots may deteriorate. Also, it may be attributed due to the surface imperfection of the Braille document and also due to the defacing of the Braille document by any means. The performance of the proposed algorithm is evaluated on manual basis with TPR, TNR, FNR and FPR. This paper also presents a new way of calculating the accuracy which differs from the previously used traditional method where in the accuracy was calculated as the ratio of the total number of correctly extracted dots to the total number of dots in the image field of view. The expression used in this paper is as follows

$$\text{Accuracy} = \text{TPR} - (\text{FNR} + \text{FPR} + \text{TNR}) * 100 \quad (16)$$

Table.2 gives the average TPR, FNR, FPR TNR and accuracy for database DB1 and DB2. Fig.15 shows the plot of accuracy for DB1 and DB2 Fig.16 and Fig.17 give the plots of FNR, FPR and TNR for the two databases respectively. Fig.18 and Fig.19 show up the accuracy plots of all the images of DB1 and DB2.

Parameters	Database DB1	Database DB2
True Positive Rate	1.00	1.00
False Negative Rate	0.006	0.009
False Positive Rate	0.002	0.003
True Negative Rate	0.001	0.002
Average Accuracy	0.991	0.986

TABLE 2: Performance Evaluation for DB1 and DB2.

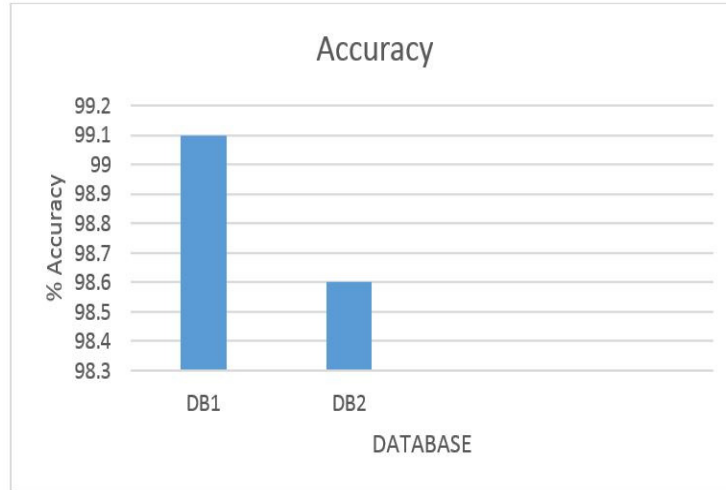


Figure 15: Graph showing Average Accuracy of DB1 and DB2.

A significant drawback of the proposed technique is the merging of recto-verso dots, which is due to the fading shade patterns in the double sided Braille document being used for dot extraction. Although this transpires hardly ever, the overall error rate of the proposed system can be ascribed to the quality of the acquired image of the Braille document.

Additionally in this work, for evaluating the performance of this technique we have considered merged dots as the error and has been assigned as FNR and error due to dirty mark and blemishes as FPR.

The proposed method fails to work if the rotation angle for the document is more.

The proposed system not only possesses the excellent detection rates up to 99.1% and 98.6% for DB1 and DB2 databases respectively but it could as well be applied over any Braille document regardless to the writing grade or language.

All the experiments were done under MATLAB environment. The average execution time on an I7 machine with 8 GB of memory for separating the Recto and Verso dots of the Braille image for DB1 database is 5.24 sec and is 5.42 sec for database DB2 respectively.

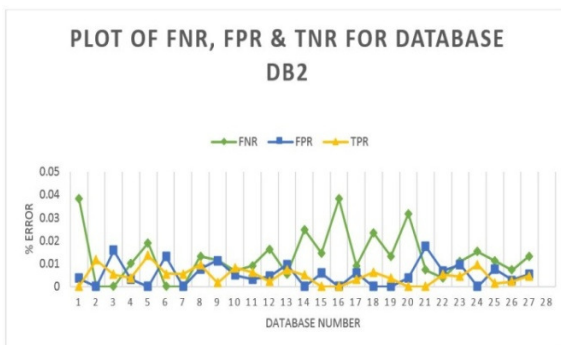


FIGURE 16: Plot of error sources for DB1.

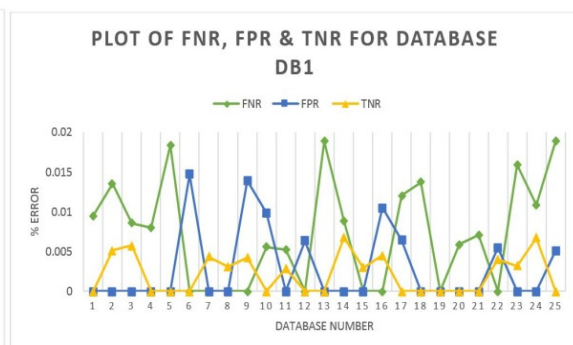


FIGURE 17: Plot of error sources for DB2.

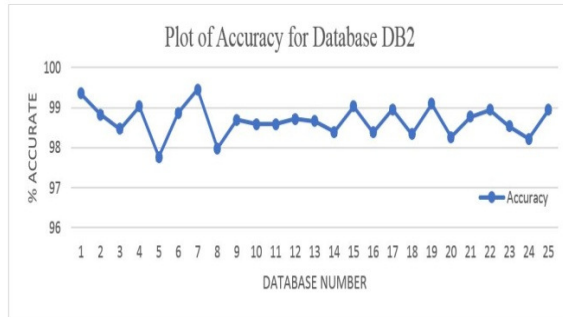


FIGURE 18: Plot of Accuracy for DB1.

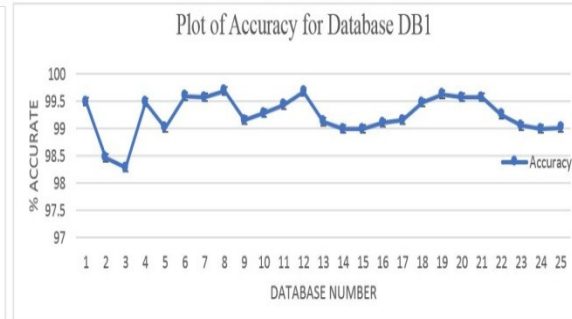


FIGURE 19: Plot of Accuracy for DB2.

4. CONCLUSION

In this projected work, the separation of Recto and Verso dots from Inter-point Braille Image using a novel combination of Eight Connectivity based two-pass Connected-Component Labelling algorithm and a novel thresholding technique is presented. With an aim to recognise the Braille cells with some special cases, an adaptive grid construction technique too has been employed. The competence of this technique is 99.1% for database DB1 and 98.6% for database DB2. The proposed technique has proved very advantageous for processing the pages of Braille documents bearing the inter-dot noise attributed by the deformation during the Braille embossing process. The precincts being merging of Recto-Verso dots, rotation angle of the Braille document. These limitations are not very severe and with some additional pre-processing on the input image these noises can be effortlessly dealt with in the future work. Furthermore our chore is to deepen the study and to come up with a novel algorithm to reduce the ambiguity in recognizing the true recto and verso dots and to upsurge the recognition rate. Irrespective of the language and the writing grade, this algorithm can be applied to any Braille document. Herein the processing was executed under MATLAB implementation environment and this can be extended in real time as well.

5. ACKNOWLEDGEMENTS

We would like to thank Council of Scientific and Industrial Research (CSIR), New Delhi, for providing the financial assistance for this work under the research scheme No: **22(0613)/12/EMR-II** and also the work is supported by JSS Research Foundation, Mysore, Karnataka.

6. REFERENCES

- [1] J. Mennens, L. Van Tichelen, G. Francosis and J. Engelen. "Optical recognition of Braille writing", *Proceedings of second international conference on Document analysis and Recognition*, IEEE Oct-1993, pp 428-431.
- [2] J. Mennens, L. Van Tichelen, G. Francosis and J. Engelen. "Optical Recognition of Braille Writing Using Standard Equipment", *IEEE transactions of rehabilitation engineering*, Vol. 2, No. 4, Dec 1994.
- [3] T. W .Hentzschel, and P. Blenkhorn. "An Optical Reading Systems for Embossed Braille Characters using a Twin Shadows Approach", *Journal of Microcomputer Applications*, 1995, pp. 341-345.
- [4] R .T. Ritchings, A. Antonacopoulos and D .Drakopoulos. "Analysis of Scanned Braille Documents", In: Dengel, A., Spitz, A.L. (eds.): *Document Analysis Systems*, World Scientific Publishing Company 1995, pp. 413-421.

- [5] Y. Oyama, T. Tajima, and H. Koga. "Character Recognition of Mixed Convex- Concave Braille Points and Legibility of Deteriorated Braille Points", *System and Computer in Japan*, Vol. 28, No. 2, 1997.
- [6] C. M. Ng, V.Ng and Y.Lau. "Regular feature extraction for recognition of Braille", *Third International conference on computational Intelligence and Multimedia Applications*, 1999, pp. 302—306.
- [7] A. Antonacopoulos and D. Bridson. "A Robust Braille Recognition System", *Document Analysis Systems VI*, A. Dengel and S. Marinai (Eds.), Springer Lecture Notes in Computer Science, LNCS 3163, 2004, pp. 533-545.
- [8] L. Wong, W. Abdulla and S. Hussmann. "A Software Algorithm Prototype for Optical Recognition of Embossed Braille", *17th Conference of the International Conference in Pattern Recognition*, Cambridge, UK, IEEE-2004, pp. 23–26.
- [9] N. Falcon, C. M. Travieso, J. B. Alonso and M. A. Ferrer. "Image Processing Techniques for Braille writing Recognitor", *EUROCAST 2005*, LNCS 3643.
- [10] A. Malik Al-Salman, Y. ALOHAI, M. Alkanhal and A. Airajith. "An Arabic Optical Braille Recognition System", *ICTA Apr 2007*, pp.12-14.
- [11] A. Malik S. Al-Salman, A. El-Zaart, Y. Al-Suhaibani, K. Al-Hokail and A. O. Al-Qabbany. "An Efficient Braille Cells Recognition", *IEEE-2010*.
- [12] J. Yin, L. Wang and J. Li. "The Research on Paper-mediated Braille Automatic Recognition Method", *Fifth International Conference on Frontier of Computer Science and Technology*, IEEE-2010, pp 619-624.
- [13] J. Li, X. Yan. "Optical Braille Character Recognition with Support-Vector Machine Classifier", *International Conference on Computer Application and System Modelling (ICCASM 2010)*.
- [14] S. D. Al-Shamma and S. Fathi. "Arabic Braille Recognition and Transcription into Text and Voice", *5th Cairo International Biomedical Engineering Conference Cairo, Egypt*, IEEE-Dec 2010.
- [15] Z. Tai, S. Cheng, P. K. Verma and Y. Zhai. "Braille document recognition using Belief Propagation", *Journal of Visual Communication and Image Representation* 21(7): 722-730 (2010)
- [16] A. Al-Saleh, A. El-Zaart and A. Malik Al-Salman. "Dot Detection of Braille Images Using A Mixture of Beta Distributions", *2011 Journal of Computer Science* ISSN 1549-3636 pp-1749-1759.
- [17] J. Bhattacharya, S.Majumder and G.Sanyal. "Automatic Inspection of Braille character: A Vision based approach", *International Journal of computer and Organization trends – volume1, Issue3 -2011*, ISSN: 2249-2593, pp. 19-26
- [18] M. Wajid, M. Waris Abdullah and O. Farooq. "Imprinted Braille-Character Pattern Recognition using Image Processing Techniques", *International Conference on Image Information Processing*, IEEE- 2011.
- [19] R. Ismail Zaghloul and T. Jameel Bani-Ata. "Braille Recognition System – With a Case Study Arabic Braille Documents", *European Journal of Scientific Research*, ISSN 1450-216X Vol.62 No.1 (2011), pp. 116-122.

- [20] L. Di Stefano and A. Bulgarelli. "A Simple and Efficient Connected Components Labeling Algorithm". Proceedings ICIAP, IEEE- 1999, Venice, Italy, pp. 322-327.
- [21] R.C.Gonzalez and R.E. Woods. "Digital Image Processing", 2nd edition, Prentice Hall, 2002.

Mixed Language Based Offline Handwritten Character Recognition Using First Stroke Based Training Sets

Magesh Kasthuri

*Research Scholar
SCSVMV University, Kanchipuram, India*

magesh.kasthuri@wipro.com

V. Shanthi

*Professor, Dept. of Computer Science
St. Joseph's College of Engineering
Chennai, India*

drvshanthi@yahoo.co.in

Venkatasubramanian Sivaprasatham

*Professor, Dept. of Information Technology,
Nizwa College of Technology,
Nizwa, Sultanate of Oman.*

mukundmeghna@gmail.com

Abstract

Artificial Neural Network is an artificial representation of the human brain that tries to simulate its learning process. To train a network and measure how well it performs, an objective function must be defined. A commonly used performance criterion function is the sum of squares error function.

Full end-to-end text recognition in natural images is a challenging problem that has recently received much attention in computer vision and machine learning. Traditional systems in this area have relied on elaborate models that incorporate carefully hand-engineered features or large amounts of prior knowledge.

Language identification and interpretation of handwritten characters is one of the challenges faced in various industries. For example, it is always a big challenge in data interpretation from cheques in banks, language identification and translated messages from ancient script in the form of manuscripts, palm scripts and stone carvings to name a few.

Handwritten character recognition using Soft computing methods like Neural networks is always a big area of research for long time and there are multiple theories and algorithms developed in the area of neural networks for handwritten character recognition

Keywords: Handwritten Character Recognition, Noise Reduction, Pre-processing Techniques In Character Recognition, Pattern Matching, Strokes, Fixed-language, Training Neural Networks, Gabor Filter.

1. INTRODUCTION

The key idea of this paper is broadly categorized as:

- Study and evaluation of various noise reduction techniques in Character recognition and establishing mechanism for properly identifying the base of noise reduction (eg: strokes, shapes, weightage, fonts etc.) to handle mixed language character recognition.
- Defining an improvised training process called self-training based on first stroke identification.

- Design an algorithm for first stroke identification and further methods of segment identification in an offline character recognition.
- Conceptualize a unified system (system and methods) utilizing above training and character recognition process as a unique and combined Character recognition and interpretation system handling mixed language content.

2. PROBLEM DESCRIPTION

A stroke is not limited to a continuous line segment. A stroke may also include a portion of a character that has a discontinuity in its representation. For example, an English alphabet 'i' may also be considered as a single stroke according to some embodiments in spite of a discontinuity in its representation because there is no sudden change in angle in any portion of this alphabet. Therefore, there is need for greater accuracy in offline handwriting recognition of such handwritten text. Hence displaying the confidence of recognition helps the user to decide if this can be taken as acceptable threshold or improvise with further noise reduction or manual correction process.

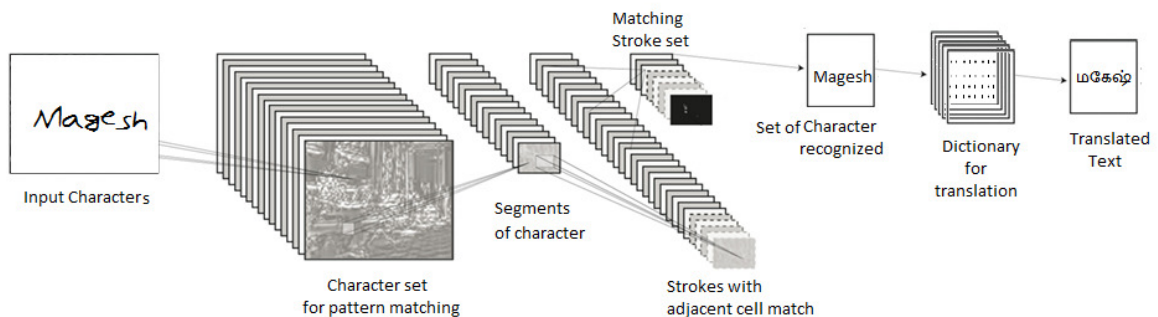


FIGURE 1: Proposed System for Handwritten Character Recognition.

One challenge associated with offline handwriting recognition is that the handwritten text cannot be edited or re-entered by the user until the entire handwritten text is recognized. A user, thus, may not be able to provide a feedback for correction after the recognition of each character or word in case of an incorrect recognition of that character or word. Thus, the errors in the offline handwriting recognition of that handwritten text may keep on accumulating in the absence of user supervision. Therefore, it is desirable to increase the accuracy of recognition of handwritten text in systems that implement offline handwriting recognition.

3. REVIEW OF EXISTING SOLUTION

The article in [1] discusses an Online character recognition designed based on Feature extraction from sample character using Gabor filter and Noise removal (size & shape normalization, imaginary strokes) is done prior to that. Pattern generation based on training with input feature vectors and Handwritten character Feature identification is done using Gabor filter.

Pattern generation and classification (statistical analysis) and Comparison of generated pattern with reference patterns to recognize the character are additional feature of the system proposed. Pattern based recognition is partially common with proposed approach whereas stroke identification is a novel part in pattern or relative cell forming in proposed system.

The article in [2] proposes a system designed for online character recognition (handwriting) suitable for noisy environments having - Improved accuracy based on self-learning specific to separate characters and users which is Alphabet independent and Low resource usage (40 KB) - mobile devices. It also uses Activity matrix based feature extraction & character comparison. The claim uses feature extraction like proposed system but the concept behind them is classical way of training the data which is completely different from our system.

Noise reduction (due to handwritten character stroke change) based on adjacent cell comparison from matrix of strokes and Binarization of strokes post feature extraction.

As per [10], Handwritten character recognition based on decomposition of characters into segments or features is designed which has Weightage based character recognition from features and Binarization of data based on feature comparison with highest score or Weightage from comparison. This comparison is between features and character model. The functionality includes decomposition the handwritten input character into one or more segments in accordance with the model specific segmentation scheme of the respective character model.

As per [11], Character Recognition for Ink based characters and uses mathematical notation for representing character shapes with Vector based recognition. Noise Reduction or feature filtering using mathematical comparison from vector representation and Repetitive process for vector normalization. The claim uses feature extraction like proposed system but the concept behind them is mathematical representation of shapes in the character system (predefined vocabulary through training or manual feed).

Method of language identification and language identifying module using short word lists and n-grams [21] does Pre-processing using n-gram technique (statistical representation) and Character Recognition based on knowledge base (training input). The Knowledge base is mapping to input source after feature extraction. This is a language identifying module and a method for identifying the language of a text string This is for providing language information to another application (eg: text-to-speech system in this case). This system uses language detection to feed the content to "text-to-speech" system whereas we use the detection for translating the content to target language.

4. PROCESSING COMPARISON WITH OTHER EXISTING SOLUTION

Performance of single-algorithm systems drops precipitously as the quality of input decreases. [3] In such situations, a human subject can continue to perform accurate recognition, showing only a gradual decrease in reliability. Collaboration between separate algorithms proves beneficial, in that such systems will allow a gradation of recognition levels expressed as probabilities or loose guesses to be passed from one level to the next. More specifically, a front-end system will perform some useful first-order basic processing. Then a second level of processing will be engaged which will judge whether to assimilate the results of the first process, extend them and proceed to the next stage with a positive recognition, or to dismiss them and reinvoke the first level again while asking for modifications.

In one embodiment[1] called Gabor filter based handwritten character recognition system, a character recognition method executed on an electronic device is disclosed, the method comprising: receiving, at the electronic device, an image representing a character including one or more central strokes; determining a set of parameters associated with each of the one or more relative (associative) strokes; comparing, for each of the one or more relative strokes, the associated set of parameters with a plurality of stored sets of adjacent parameters, wherein each of the plurality of stored adjacent strokes is associated with a stored set of relative parameters; identifying next stroke, from among the plurality of stored strokes, corresponding to each of the one or more strokes based on the comparison to identify the possible character comprising these strokes in order

5. STEPS INVOLVED IN TRAINING

The multiple-layered system which makes up any robust handwriting recognizer has progressed greatly from the days when character recognition meant reading printed numerals of a fixed-size OCR-A font. However, only in a decade have the successes within the field approached the level of a truly practical handwriting recognizer [1].

If a Neural Network mimics the input pattern it was presented with, then that network is said to be autoassociative. For example, if a neural network were presented with the pattern "0110" and the output were also "0110", then that network would be said to be autoassociative. A neural network calculates its output based in the input pattern and the neural network's internal connection weight matrix. The values for these connection weights will determine the output from the neural network, based upon input pattern.

During a pattern matching, segmented characters are taken and mapped as input neuron. All neuron nodes weights, defined as:

- i) $W_j(1), j = 1 \dots n$, are initialized randomly.
 W is the number of neurons in the output layer.*
- ii) $K = \text{Maximum}(X, y)$, for iteration step $y=1 \dots K$, get an input vector X_k from first recognized stroke*
- iii) Calculate Distance = $X_k k = 1 \dots n$ $1 \dots n$ refers to neuron nodes for all strokes in the character to match.*
- iv) Select the winner output neuron j^* with minimum distance (which is more resembling to the stroke of testing)*
- v) Update weights $W_j(k)$ to neurons j^* and its neighborhood*
- vii) If pattern is not matching, then take adjacent neuron as desired and goto (iv)*
- vi) If k has more weights from K go to step (ii).*

To find a local minimum of a function using gradient descent, one takes steps proportional to the negative of the gradient (or of the approximate gradient) of the function at the current point. If instead one takes steps proportional to the positive of the gradient, one approaches a local maximum of that function; the procedure is then known as gradient ascent.

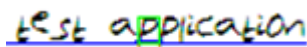
6. EXPERIMENTAL SETUP

From a hard copy document, image will be extracted for offline character recognition. There are quite a few conventions in determining the input mode of such offline character recognition viz:


- complete document read / scan
- sequential reading (word / sentence wise) from the document (scan)
- Reading character / word / sentence directly from the digital image of the document

This required offline character recognition from a handwritten, multi-lingual document with mixed-language content.

Consider a scanned text "This is a test scan" which is fed to the system for recognition as follows:



The system first learns the strokes by itself and maps to the character set (alphabet series) it stores in the knowledge base. Hence before training (including pre-processing and noise reduction) it is able to recognize some of the characters like



The confidence of this recognition is about 73.49%. After the character set is manually corrected in the system (re-trained) for recognition, there is a good improvement in the system where it recognized the characters with 91.22% confidence like as summarized below:

Character	Originally recognized	Confidence	Corrected Recognition (after self training)	Confidence
t	t	77.9	t	89
e	e	78.5	e	86.4
s	s	79.1	s	88.05
t	t	81	t	89
a	a	84.8	a	84.8
p	o	75.95	p	98.94
p	o	75.95	p	98.94
l	l	93.05	l	93.05
i	i	96.9	i	96.9
c	c	78.9	c	94.5
a	a	79.4	a	88.5
t	t	79.5	t	97.5
i	i	74.21	i	94.21
o	o	76.3	o	86.3
n	n	78.01	n	98.01
Average		80.6313333		92.2733333

TABLE 1: Training Metrics for the Input Source.

Steps involved:

- Step 1: Obtain a stroke
- Step 2: Normalize stroke
- Step 3: Generate Index
- Step 4: Obtain a stroke
- Step 5: Create index structure
- Step 6: Index retrieval
- Step 7: Grouping characters
- Step 8: Store the character set with index

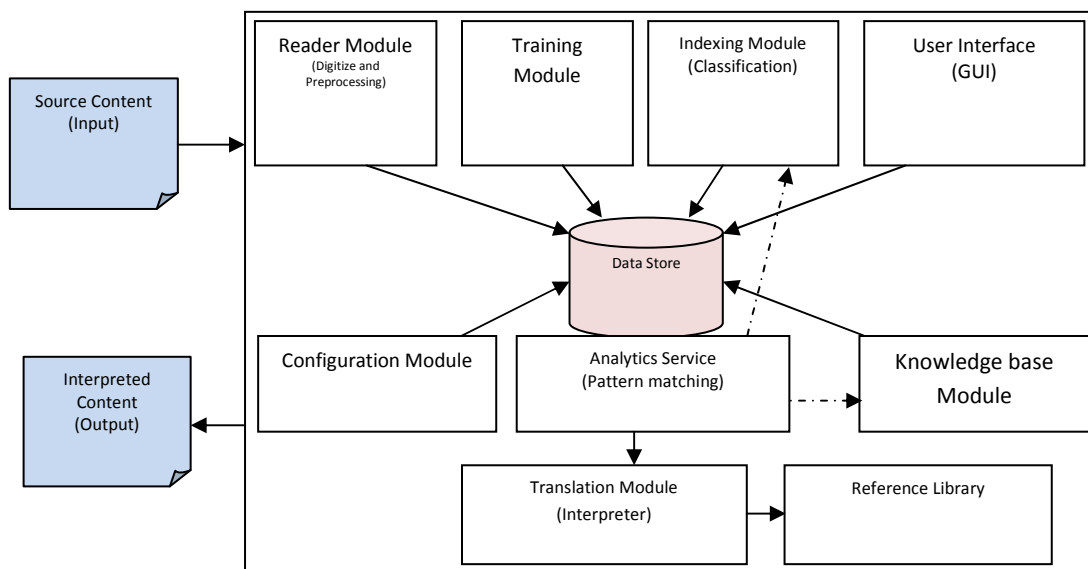


FIGURE 2: System Architecture of Proposed System.

Metrics on individual character recognition **Accuracy Ratio** is shown below:

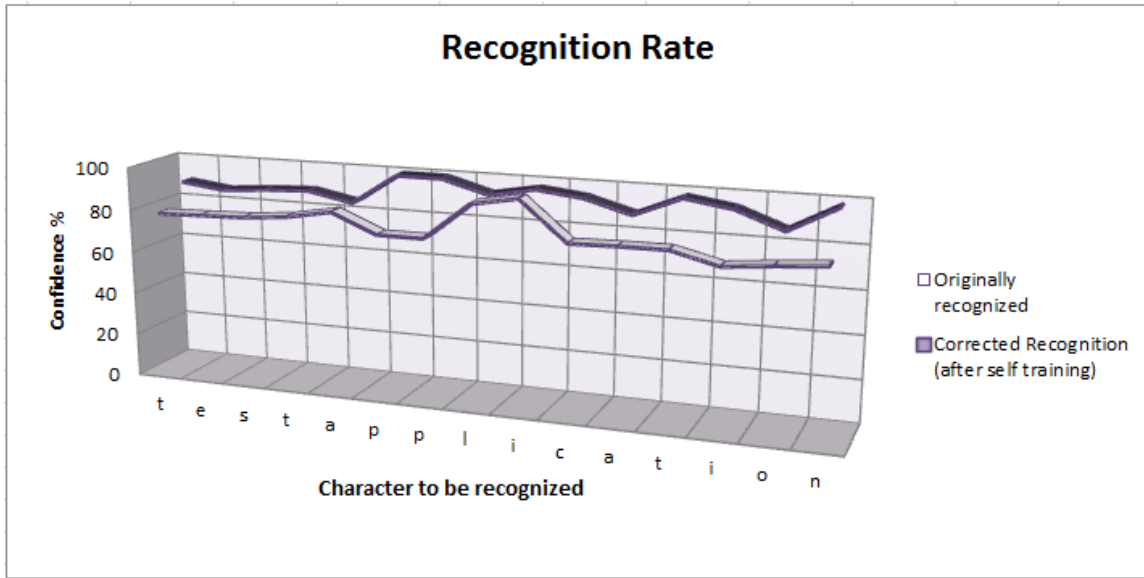


FIGURE 3: Metrics – Acceptance Ratio.

With this setup, when a distributed variance of input text are fed to the system for recognition, there would be interesting statistics based on number of lines to scan and time taken in producing the result co-related with accuracy level of the recognition.

This sample data processing detail is summarized below:

No. of Scanned Lines	Metrics on Originally recognized Image		Metrics After training and Correction	
	Execution Time (in milliseconds)	Acceptance Ratio (in Accuracy)	Execution Time (in milliseconds)	Acceptance Ratio (in Accuracy)
1	6499	35.86115	453	97.25
2	3124	90.862495	1499	98.85909
3	5101	80.980095	2397	91.194
6	9124	76.2904	8071	89.93145
16	64663	82.56244	13670	91.035355
32	88113	87.488625	78942	92.488625
97	11952	72.43023	9021	94.48054
169	9031	90.80054	9202	90.80054
1000	91345	89.5441	89173	92.4852

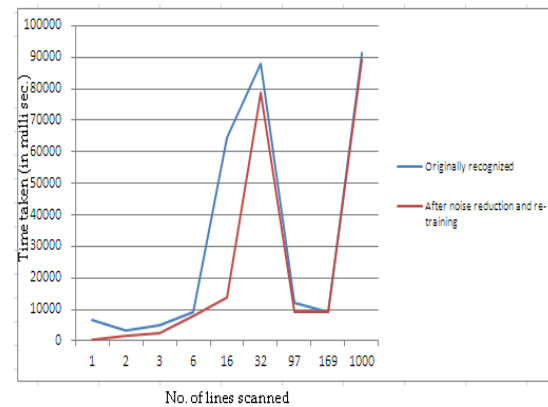


FIGURE 4: Metrics on Execution Time and Acceptance Ratio.

Various users write in varied handwriting styles and in different languages. Each character in the handwritten text may have been written in multiple handwriting styles by different users. It is desirable that a character be correctly recognized in spite of having been written in varied handwriting styles. In addition, a character in one language may be similar, but not same, to a character of a different language and is, thus, prone to be incorrectly recognized.

```

This is a test program
My first neural network test application
This is a test scan
My atione is networesh
I networ 0 test ws ationmuT teT
To scan 0 0 networoze am neural Tsa oramT networation

Overall Level of Accuracy Confidence: 85.23145%
Total Time taken: 9671 ms.
Total lines recognized: 0

Detailed stats:
Line :1 Confidence level :79.97083%
Line :2 Confidence level :76.206245%
Line :3 Confidence level :89.869225%
Line :4 Confidence level :92.2925%
Line :5 Confidence level :90.80667%
Line :6 Confidence level :83.62222%

Recognizer Summary:
-----
Image-recognizer (user-made):::61.164444
mag recognizer (fixed):::85.23145
Old style:::70.26598
Old style2:::63.78558
Printed Characters:::59.94278
Tamil Unicode:::63.52475
mag recognizer (user-made):::79.17114
-----

Best Recognizer:mag recognizer (fixed) Confidence %:85.23145
Best Recognized Text:This is a test program
My first neural network test application
This is a test scan
My atione is networesh
I networ 0 test ws ationmuT teT
To scan 0 0 networoze am neural Tsa oramT networation
    
```

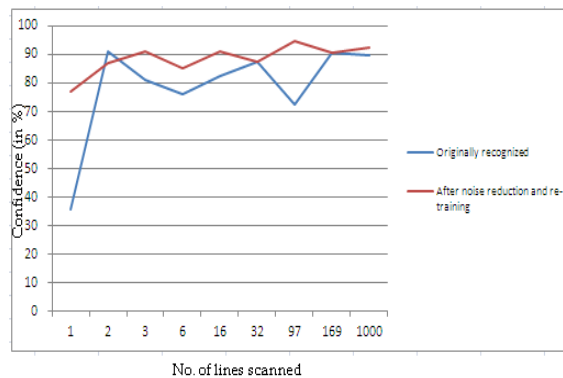


FIGURE 5: Metrics – Confidence Accuracy in Recognition.

Consider a simple example of learning a character ‘A’ based on mapping tables given above. The steps involved in this learning process are explained below:

6.1. Input

From an input source (scanner), the image representing the text may be received by the electronic device by means of scanning a handwritten document.

6.2. Pre-process

A processor of the electronic device may preprocess the received image. The preprocessing may include digitization of the received image.

The preprocessing may further include removing noise such as, but not limited to, salt and pepper noise and Gaussian noise from the digitized image using one or more noise removal techniques known in the art. In addition, the preprocessing may also include making the width of all the characters in the received image uniform by normalizing the width of each portion of the text to a predetermined value of width.

6.3. Normalization

This may include either reducing width of some portions of the handwritten text or increasing their width to the predetermined value of width. The width of a portion may be reduced by converting any undesired black pixels to white if the handwritten text is represented by black pixels.

Once the received image is preprocessed, the received image is segmented into one or more first strokes by the processor.

6.4. Segmentation

On preprocessing the image, the processor of the electronic device may segment the handwritten text in the received image into characters. The processor may distinguish one component of the text from another component based on spacing between the components.

6.5. Strokes Preparation

A sudden change in angle may be considered at a point on a character when two linear or non-linear line segments form an angle at that point that is below a predetermined threshold angle.

For example, if an angle formed by two line segments at a point is below a predetermined threshold angle of 40°, it may be considered as a sudden change in angle. Once all the points representing a sudden change in angle have been identified, the processor may split character at these points into different strokes.

6.6. Stroke Recognition

The processor may scan each of these cells that represent a portion of a stroke sequentially to determine one or more parameters associated with a portion of another stroke represented in that cell.

Language	Character	Lang ID	Char ID	Strokes per style				
English	A	1	1	A1	A2	A3		
	B	1	2	B1	B2			
	C	1	3	C1	C2			
	D	1	4	D1	D2			
	E	1	5	E1	E2	E3	E4	
	F	1	6	F1	F2	F3		
	G	1	7	G1	G2	G3	G4	G5

FIGURE 6: Indexed List of Strokes for Character Sets In Knowledge Base.

7. MIXED LANGUAGE RECOGNITION

Once the text is recognized based on character sets available in the system, then comes mixed language detection, Offline recognition API like LangDetect or Online detection API from Google can be used. They support various profiles using Unicode based character recognition with confidence level detection as well. This helps in deciding best language possibility based on higher detection confidence.

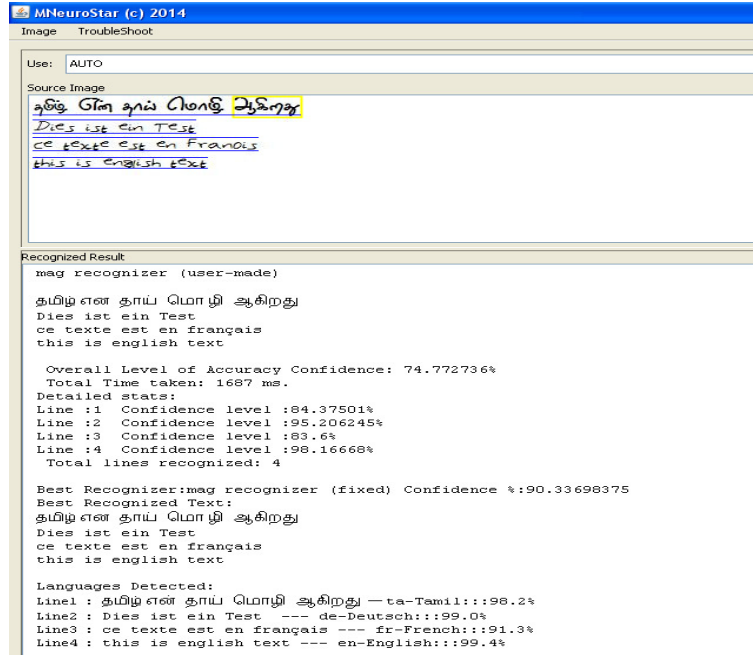


FIGURE 7: Mixed Language Detection.

In Mixed language detection, the idea is to first recognize each line/word of text using all character set available in the knowledge base and then detect the language using language detection API. The system is planned in such a way that the knowledge base will be in synchronous with the language detection profile to have equal or more set of profiles for language detection to enable all possible detections for languages.

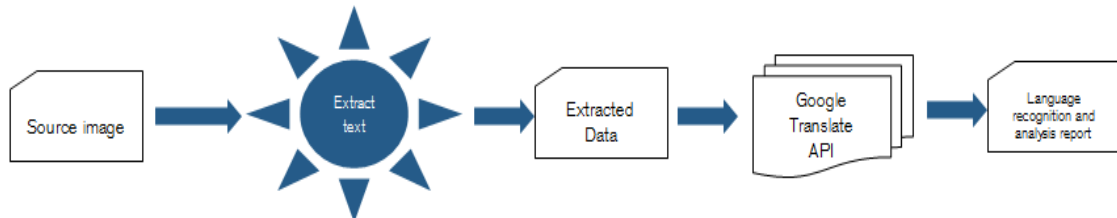


FIGURE 8: Technique in Character Extraction and Language Detection.

8. DETERMINING PROCESSING ACCURACY

When we have a document having multiple languages, then it would be always a tough process to detect languages. It is a tedious and error prone process in automated language detection or in manual process as the translator person need to understand all the languages used in the document.

Though the possibility of such a document is less as there is no real-time usecase available for such a requirement, it is always best to handle all possible alternate situations in usecases to avoid or minimize the mistakes in language detection and improvise the accuracy of the language detection process.

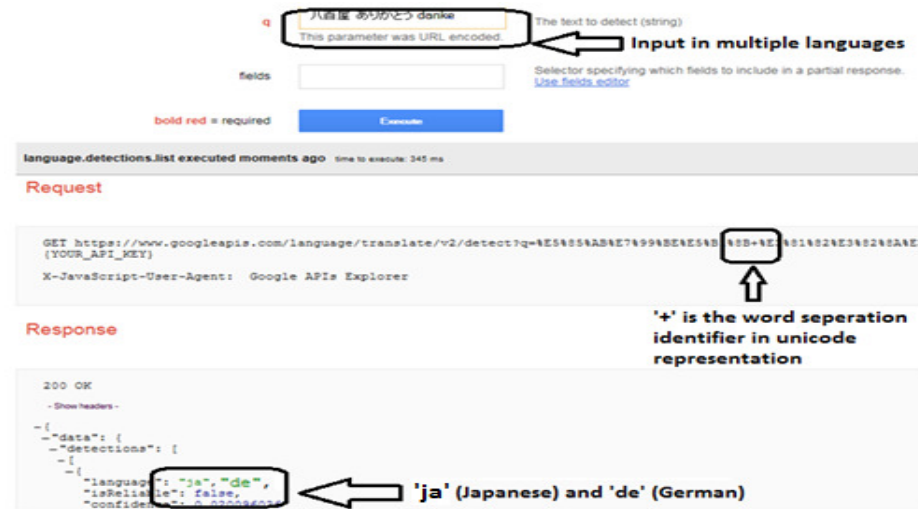


FIGURE 9: Sample of Language Detection (Fixed algorithm for Multi-lingual content).

9. FURTHER RESEARCH AND IMPROVEMENT

The idea of this research is to develop a unified system for character recognition and language detection in mixed language content. The idea is being further developed to integrate with online translators like Google Translate API or LangDetect API library to do instant translation as well along with character recognition and language detection. Language detection helps in determining the source of language to translate and this helps in a complete end-to-end processing system for recognition and translation together.

Also, this is a challenging area as it helps a lot of training data and offline dictionary elements to do mixed language content based translation to bring in all content to one single language.

10. RESULT OF EVALUATION

Working on the statistical data points on processing the characters for accuracy, processing time and MSE (mean squared error), posted below a sample test result [19].

In statistics, the mean squared error (MSE) of an estimator is one of many ways to quantify the difference between values implied by an estimator and the true values of the quantity being estimated. MSE is a risk function, corresponding to the expected value of the squared error loss or quadratic loss. MSE measures the average of the squares of the "errors." The error is the amount by which the value implied by the estimator differs from the quantity to be estimated. The difference occurs because of randomness or because the estimator doesn't account for information that could produce a more accurate estimate.

The processor may also determine a set of first parameters for other strokes such as those shown in figure 2 into which the image of character 'A' was segmented by the processor.

This is how self-learning based training is conceptualized based on first stroke identification.

11. ACKNOWLEDGEMENT

This concept is a sub-section of US/India patent filed concept and acknowledged by the patent (Indian Patent Journal Issue 47/2013 dated 22/11/2013). All content and idea are copyrighted.

12. CONCLUSION

The idea explained in this paper relates generally to text recognition, and more particularly to systems and methods for offline character recognition.

Algorithm	Training sets	Time taken in recognition for Test set (in secs)
Supervised training	43 X 10 character sets	21.240
Gabor Filter	43 X 5 character sets	9.8871
Markov Chains and Markov Filter	43 X 8 character sets	64.663
MNeustar (Self-training)	43 X 1 character sets	6.499

Algorithm	Accuracy % in Test set 1 (English Handwritten)	Accuracy % in Test set 1 (Mixed language content)
Supervised training (Coates et al.)	94.56%	53.2%
Gabor Filter	91.34%	48.2%
Unsupervised training (Neuman et al.)	95.29%	69.19%
Wang et. Al.	98.14%	91.9%
MNeustar (Self-training)	97.25%	93.25%

TABLE 2: Metrics Showing Benefit of Proposed Algorithm.

In one embodiment, a character recognition method executed on an electronic device is disclosed, the method comprising: receiving, at the electronic device, an image representing a character including one or more first strokes; determining a set of first parameters associated with each of the one or more first strokes; comparing, for each of the one or more first strokes, the associated set of first parameters with a plurality of stored sets of second parameters, wherein each of the plurality of stored second strokes is associated with a stored set of second parameters; identifying a second stroke, from among the plurality of stored second strokes, corresponding to each of the one or more first strokes based on the comparison; and identifying the character based on the identified one or more second strokes.

13. References

- [1] Wai Kin Kong, David Zhang, Wenxin Li, Palmprint feature extraction using 2-D Gabor filters, The Journal of the Pattern Recognition Society (Elsevier) Pattern Recognition 36 (2003) 2339 - 2347.
- [2] Daming Shi, Robert I. Damper And Steve R. Gunn, Offline Handwritten Chinese Character Recognition by Radical Decomposition, ACM Transactions on Asian Language Information Processing, Vol. 2, No. 1, March 2003, Pages 27-48.
- [3] Anita Pal, Dayashankar Singh, Handwritten English Character Recognition Using Neural Network, International Journal of Computer Science & Communication Vol. 1, No. 2, July-December 2010, pp. 141-144.
- [4] R.Jagadeesh Kannan And R.Prabhakar, Off-Line Cursive Handwritten Tamil Character Recognition, WSEAS Transactions On Signal Processing, Issue 6, Volume 4, June 2008, ISSN: 1790-5052 Pages: 351-360.
- [5] Lubna Badri, Development of Neural Networks for Noise Reduction, The International Arab Journal of Information Technology, Vol. 7, No. 3, July 2010 Pages: 289-294
- [6] Mansi Shah And Gordhan B Jethava, A Literature Review On Hand Written Character Recognition, Indian Streams Research Journal, Vol -3 , ISSUE 2, March.2013, ISSN:-2230-7850.

- [7] Zhiyi Zhang, Lianwen Jin, Kai Ding, Xue Gao, Character-SIFT: a novel feature for offline handwritten Chinese character recognition, 10th International Conference on Document Analysis and Recognition, 2009 Pages: 763-767.
- [8] Li Fuliang, Gao Shuangxi, Character Recognition System Based on Back-propagation Neural Network, 2010 International Conference on Machine Vision and Human-machine Interface Pages: 393-396.
- [9] Dr.J.Venkatesh and C. Sureshkumar, Tamil Handwritten Character Recognition Using Kohonon's Self Organizing Map, IJCSNS International Journal of Computer Science and Network Security, VOL.9 No.12, December 2009 Pages: 156-161.
- [10] L. D. Jackel, C. E. Stenard, H. S. Baird, B. Boser, J. Bromley, C. J. C. Burges, J. S. Denker, H. P. Graf, D. Henderson, R. E. Howard, W. Hubbard, Y. leCun, O. Matan, E. Pednault, W. Satterfield, E. Sickinger, and T. Thompson, A Neural Network Approach to Handprint Character Recognition, IEEE CH2961-1/91/0000/0472 2001 Pages: 472-475.
- [11] Seong-Whan Lee, Young- Jaon Kim, A New Type of Recurrent Neural Network for Handwritten Character Recognition, IEEE 0-8186-7128-9/95 2005 Pages: 38-41.
- [12] Ishwarya .M.V, R. Jagadeesh Kannan, An Improved Online Tamil Character Recognition Using Neural Networks, 2010 International Conference on Advances in Computer Engineering IEEE 978-0-7695-4058 Pages: 284-288.
- [13] G. Tambouratzis, Applying Logic Neural Networks to Hand-written Character Recognition Tasks, IEEE 0-8186-7686-8/9 10996 Pages : 268-271.
- [14] Anil K.Jain, Jianchang Mao, K.M.Mohiuddin, Artificial Neural Networks : A Tutorial, IEEE 0018-9162/96 March 1996 Pages: 31-44.
- [15] Neural Networks, Fuzzy Logic and Genetic Algorithms – Sythethis and Applications by S.Rajasekaran and G.A.Vijayalakshmi Pai from Eastern Economy Edition Page-31-33.
- [16] LI Guo-hong, SHI Peng-fei, An approach to offline handwritten Chinese character recognition based on segment evaluation of adaptive duration, Journal of Zhejiang University Science ISSN 1009-3095 2004 5(11):Pages: 1392-1397.
- [17] Magesh Kasthuri, Dr. V.Shanthi, Noise Reduction and Pre-processing techniques in Handwritten Character Recognition using Neural Networks, TECHNIA International Journal of Computing Science and Communication Technologies, VOL.6 NO. 2, January. 2014 (ISSN 0974-3375) Pages: 940-947.
- [18] Magesh Kasthuri, Dr.V.Shanthi Self-training Method using First strokes in Handwritten Character Recognition International Journal of Scientific Research, Vol.III, Issue. V, May 2014, ISSN No. - 2277-8179 Pages: 73-77.
- [19] Magesh Kasthuri, Dr.V.Shanthi Pre - processing and Self training techniques in Handwritten Character Recognition Indian Journal of Applied Research, Vol.IV, Issue. IV April 2014 ISSN - 2249-555X – Pages: 189-193.

Unsupervised Classification of Images: A Review

Abass Olaode

*School of Electrical Computer Telecommunication Engineering
University of Wollongong
Wollongong, 2500, Australia*

abass.olaode808@uowmail.edu.au

Golshah Naghdy

*School of Electrical Computer Telecommunication Engineering
University of Wollongong
Wollongong, 2500, Australia*

golshah@uow.edu.au

Catherine Todd

*School of Electrical Computer Telecommunication Engineering
University of Wollongong
Dubai, UAE*

CatherineTodd@uowdubai.ac.ae

Abstract

Unsupervised image classification is the process by which each image in a dataset is identified to be a member of one of the inherent categories present in the image collection without the use of labelled training samples. Unsupervised categorisation of images relies on unsupervised machine learning algorithms for its implementation. This paper identifies clustering algorithms and dimension reduction algorithms as the two main classes of unsupervised machine learning algorithms needed in unsupervised image categorisation, and then reviews how these algorithms are used in some notable implementation of unsupervised image classification algorithms.

Keywords: Image Retrieval, Image Categorisation, Unsupervised Learning, Clustering, Dimension Reduction.

1. INTRODUCTION

The advent of computers and the information age has created challenges in the storage, organisation and searching of complex data especially when the quantity is massive. Therefore, it is not surprising that application of pattern recognition techniques has been found useful in image retrieval, where it has been helpful in managing image repositories. Pattern recognition enables the learning of important patterns and trends, which can be used in the indexing of the images in a repository. The applicable learning approaches can be roughly categorised as either supervised or unsupervised [1]. The supervised classification of images based on patterns learnt from a set of training images has often been treated as a pre-processing step for speeding-up image retrieval in large databases and improving accuracy, or for performing automatic image annotation [2]. The block diagram of a typical supervised image classification process is shown in Figure 1. This training data is manually selected and annotated, which is expensive to obtain and may introduce bias information into the training stage [3].

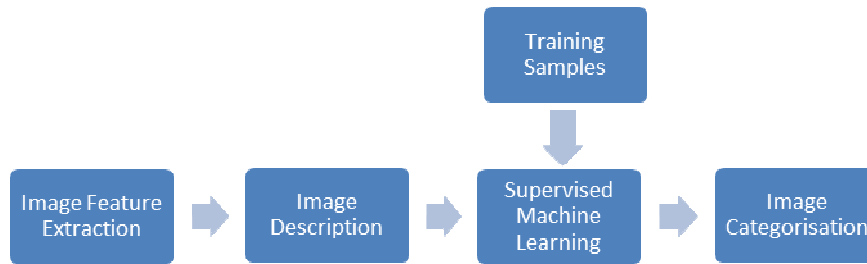


FIGURE 1: The Block diagram of a typical supervised Image categorisation process.

Alternatively, unsupervised learning approach can be applied in mining image similarities directly from the image collection, hence can identify inherent image categories naturally from the image set [3]. The block diagram of a typical unsupervised classification process is shown in Figure 2. Due to the ability to support classification without the need for training samples, unsupervised classifications are a natural fit when handling large, unstructured image repositories such as the Web [2], therefore, this study considers the unsupervised classification of images an important step in the semantic labelling of the images in a large and unstructured collection because it enables grouping of images based on semantic content for the purpose of mass semantic annotation.



FIGURE 2: The Block diagram of an unsupervised Image categorisation process.

Due to the ability of unsupervised image categorisation to support classification without the use of training samples, it has been identified as a means of improving visualisation and retrieval efficiency in image retrieval [2]. It has also been identified as a means of matching low-level features to high-level semantics especially in learning based applications [3]. These qualities makes unsupervised image categorisation a likely solution for bridging the semantic gap in image retrieval [4]. In view of these, this study provides an overview of recently developed unsupervised image classification frameworks.

In the remainder of this paper, Section 2 provides a review of image modelling process, and analyses some image modelling approaches with some recent attempts at their improvement by the research community. Section 3 provides an insight into unsupervised learning algorithms, while Section 4 examines some notable implementations of unsupervised image classification. Finally, Section 5 suggest the future application of unsupervised image classification in the automated image annotation for image retrieval purposes, while Section 6 concludes the paper with a brief overview of the reviewed implementations with focus on their suitability in the semantic labelling of images.

2. IMAGE MODELLING

Features of a digital image such as colour, texture, shapes and the locations of these features on the image represent characteristics that enable the image to be distinguished from other images. As indicated by Figure 1 and Figure2, the first step in any supervised or unsupervised image classification is the detection and extraction features present in the image, which is then followed

by the development of a description that is based on the features extracted from each image. This section examines popular algorithms for achieving these two important image modelling functions during unsupervised image categorisation.

2.1. Image Feature Extraction

For reliable recognition, it is important that the features extracted from images be detectable even under changes in image scale, noise and illumination. To satisfy this need, keypoints corresponding to high-contrast locations such as object edges and corners on the image are typically regarded as good descriptive image features are sought. The features at these keypoints are typically described using image feature descriptors. Most popular image feature extraction algorithms consist of image feature detection and description components [5, 6]. Some popular image feature extraction algorithms mentioned in recent literature are discussed below.

The Shift Invariance Feature Transform (SIFT) is an image feature extraction algorithm that ensures the detection of keypoints that are stable and scale invariant [5]. In the SIFT algorithm, the keypoints are detected via a DoG (Difference of Gaussian) pyramid created using a Gaussian filtered copy of the image. Each of the detected keypoint is then described by a 128-dimensional histogram of the gradients and orientations of pixels within a box centred on the keypoint.

Although SIFT remains one of the best descriptors in terms of accuracy, the 128-dimensions of the descriptor vector makes its feature extraction process relatively computationally expensive [7]. Khan et al. [8] explain that compared to the standard SIFT, a smaller size descriptor uses less memory and results in faster image classification. The authors [8] propose generating 96D-SIFT, 64D-SIFT and 32D-SIFT by skipping some orientation values during computation of the descriptors. Classification experiments on images from the Caltech dataset revealed that the 32D variant achieved 93% accuracy, the 64D and 96D versions recorded 95%, and the 128D achieved 97% accuracy [8]. The study reported that 128D, 96D, 64D and 32D recorded 59, 33, 18 and 11 seconds respectively to complete the classification task [8], which confirms that reducing the dimension of the descriptors reduces the amount of computation required for classification, thereby improving the speed of the process but leads to reduced accuracy. A similar result was also obtain by Ke and Sukthankar [9] when the dimension of the SIFT descriptor is reduced using Principal Component Analysis (PCA).

Rather than using DoG and image pyramid for the detection of keypoints as in SIFT, Speeded-Up Robust Features(SURF) uses the Hessian matrix in which the convolution of Gaussian second order partial derivatives with a desired image are replaced with box filters applied over image integrals (sum of grayscale pixel values), thereby reducing computation time [6]. To develop feature descriptions for the detected keypoints, each of keypoint is assigned a reproducible orientation using Haar wavelet responses in x and y directions for a set of pixels within a radius of 6σ , where σ refers to the detected keypoint scale [8]. The SURF descriptor is then computed by placing a 4×4 square window centre on the keypoint to produce a normalised 64D descriptor vectors [8]. The use of integral image representation at the keypoint detection stage of SURF ensures that the computational cost of applying box filter is independent of the size of the filter. This allows SURF to achieve much faster keypoint detection than SIFT by keeping the image size the same while varying only the filter size [8].

Although, SURF's performance is mostly similar to SIFT, it is unstable to rotation and illumination changes [10]. Liu et al. [11] noted that although SURF is capable of representing most image patterns, it is not equipped to handle more complicated ones. However, Khan et al [29] implemented classification experiments on images from David Nister, Indoor, Hogween and Caltech datasets to yield results that confirms that SURF's performance is as good as that of SIFT, with both recording 97% accuracies on Caltech dataset. The study however indicates that SURF's image matching time is higher at 80s compared to SIFT's 59s. Therefore this research finds SURF adequate enough to be considered for the purpose of feature extraction during image classification.

SIFT and SURF feature extraction algorithms can be regarded as sparse feature extraction algorithms because they only detect and describe features at chosen locations on an image. Extracting features from locations covering the entire area of an image rather than few selected location provides additional information which may improve the accuracy of image retrieval result [12]. Dalal and Triggs [13] proposed the Histogram of Oriented gradients (HOG) also known as Dense-SIFT which extracts and describes local image features from each of the uniformly spaced cells placed on an image.

A HOG description is developed for a cell by counting the occurrence of gradient orientations for the pixels within the cell in a manner similar to SIFT[14] The algorithm achieves more accurate description than SIFT by using overlapping local contrast normalization to make the result less variant to changes in illumination and shadowing [14]. This is achieved by calculating a measure of the intensity across a larger region of the image, called a block, and then using the value obtained to normalize all cells within the block [14]. Since the HOG descriptor operates on localised cells, it is invariant to geometric transformation of the image [14]. HOG was originally designed for the problem of pedestrian detection in static images but the use has been extended to other applications such as scene and object classification in static imagery [14, 13].

Due to recent increase in the use of computer vision application on mobile phones and other low power devices, research efforts are heading in the direction of development of feature extraction algorithms with minimum computation requirement and low power consumption. Examples of such algorithm include Oriented-FAST and Rotation-Aware BRIEF (ORB) [15] and Fast Retina Keypoint (FREAK) [16].

2.2. Image Description

After the extraction of features present in an image, there is a need for mathematical description of the image before supervised or unsupervised classification can be possible. Thus, the performance of image annotation is dependent on the reliability of the image feature representation (image mathematical model) [17].The most common approaches discussed in recent literatures use a normalised histogram or a vector to represent the number of times quantised features occurs on an image. The most popular of these methods is the Bag-of-Visual words (BOV) image model.

The BOV model is a popular image representation for classification purposes, which uses a visual-words histogram to effectively represent an image for image annotations and retrieval tasks [18, 19]. An important stage during BOV representation of images is the visual codebook development; a process that requires the use of K-means clustering to quantise the vectors representing image features into visual-words [17, 19, 20]. The computational requirement of this stage is very high and is therefore regarded as the most expensive part of the BOV modelling process, and attempts at reducing the computation time often lead to noisy visual-words [19].This study considers the limiting of visual-words to the unique vectors available in the set of image features extracted as a likely solution to this problem.

Currently, there are no proven methods for determining the number of visual-words to quantise image feature vectors into, during codebook development. Tsai [17] explained that although most implementations of the BOV modelling are based on 1000 visual-words, the number of visual-words is dependent on the dataset. Bosch et al. [12] used an arbitrary value of 1500 as the number of visual-words developed from SIFT features vectors of sample images in all experimentation involving BOV, while Verbeek and Triggs [21] quantise the SIFT descriptors used in their work into 1000 bins. The use of these approaches exposes the classification process to limited distinctiveness due to a small number of visual-words in the codebook, and high processing overhead when a codebook with too many visual-words is used [18]. Therefore, a research into the determination of the number of visual-words needed during BOV modelling will provide a means of eliminating some unnecessary computation overhead.

Another problem with BOV modelling of images is the loss of classification accuracy due to the disregard for the spatial location of the visual words during the modelling process [21, 20, 22].

Verbeek et al. [21] used Random Field theory to provide spatial information along with the BOV models for Probabilistic Latent Semantic Analysis (PLSA) classification of image regions, the same approach was also adopted by Xu et al. [38] for image classification via Latent Dirichlet Allocation (LDA). Zhang et al. [20] proposed the Geometry-preserving visual phrase that encodes more spatial information into the BOV models at the searching step of a retrieval system, thereby representing local and long-range spatial interactions between the visual words. However, this approach only seeks to improve search results by providing additional information to the image BOV models but does not improve the BOV modelling therefore may not be suitable for semantic-based image retrieval purposes. Lazebnik et al. [22] proposed the Spatial Pyramid in which histograms are computed for multi-level regions of an image, and then concatenated to form a single spatial histogram. This method achieved 64.6% during the classification of images from Caltech-101 [22]. While this study considers the work of Lazebnik et al. [22] as an intuitive attempt at solving the spatial coherency problem in BOV modelling especially for semantic purposes; it however suggests that the classification accuracy needs to be improved further.

Bosch et al. [23] extend the concept of spatial pyramid to the development of an image signature known as Pyramid Histogram of Oriented Gradient (PHOG). This image representation combines pyramid representation with HOG and has been found to be effective in object classification [24], facial emotion recognition [25], and facial component based bag of words [26]. How some of the techniques discussed in this Section have been used in recent works on unsupervised image classification is examined in Section 4.

3. UNSUPERVISED LEARNING

This section reviews existing literatures in which unsupervised learning approaches have been successfully applied to image categorisation. In general, unsupervised learning attempts to base grouping decisions directly on the properties of the given set of samples without the use of training samples. Although Datta et al. [2] identified three main categories of unsupervised classification algorithms: clustering based on overall minimisation of objective function, pairwise distance clustering, and statistical modelling; such classification has limited the scope of unsupervised learning to clustering algorithms. Therefore, based on the extensive list of unsupervised learning algorithms provided by Hastie et al. [1], this paper recognises Dimension reduction algorithms and clustering algorithms as the two main unsupervised machine learning algorithms needed in unsupervised image categorisation. The recognition of these groups of algorithms provides a reasonable lead way into the diverse world of the application of unsupervised machine learning to image classification.

3.1. Dimension Reduction Algorithms

It is often necessary to reduce the dimension of samples in a dataset before the patterns can be recognised. For example, the application of BOV modelling may produce 1000 dimensioned image representations which can make the categorisation of a set image BOV representation computationally inefficient especially when handling a large collection (1000 samples and above). This challenge can be minimised by estimating a rather crude global models that characterise the samples in the collection using descriptive statistics [1]. These descriptive statistics reduces the data dimensions by using a new set of variables based on properties observed at regions of high probability density on the sample space. Common unsupervised learning methods by which a descriptive statistics can be obtained include Principal Component Analysis (PCA), Non-negative matrix factorisation, and Independent component analysis (ICA) [1]. These are linear approaches which are not desirable as the means of achieving dimension reduction of images because image data possesses complicated structures which may not be conveniently represented in a linear subspace [27], therefore mapping them to a linear low dimensioned features space may incur significant loss of categorisation accuracy.

Several methods have been recently proposed for nonlinear dimension reduction. These methods are all based on the idea that the data lie close to an intrinsically low-dimensional nonlinear feature space embedded in a high-dimensional space [1]. Scholkopf et al. [28] proposed Kernel PCA as a mean of achieving non-linear dimension reduction [28]. Using non-linear functions,

Kernel PCA generates a kernel matrix for a given dataset, and then identifies a chosen number column on the kernel matrix with the largest eigenvalues [1]. Other popular non-linear dimension reduction methods include Isometric Feature Mapping (ISOMAP), Local Linear Embedding, and Local Multi-Dimensional Scaling (Local MDS) [1]. In general, these methods produce low-dimensional model of each sample in a collection by describing the sample in the terms of its approximate distances from a chosen number of nearest neighbours [1].

3.2. Clustering Algorithms

Clustering algorithms groups the samples of a set such that two samples in the same cluster are more similar to one another than two samples from different clusters, Clustering methods can be categorised into two broad classes: non-parametric and parametric methods. Non-parametric clustering involves finding natural groupings (clusters) in a dataset using an assessment of the degree of difference (such as Euclidean distance) between the samples of the dataset. It requires the defining a measure of (dis)similarity between samples, defining a criterion function for clustering, and defining an algorithm to minimise (or maximise) the criterion function. Popular non-parametric clustering algorithms include k-means, Hierarchical clustering algorithms and Spectral clustering.

3.2.1. Non-Parametric Clustering

K-mean clustering is the most widely used nonparametric technique for data clustering [2]. It represents each category in a given dataset with a centre obtained after repeated optimisation of an overall measure of cluster quality known as the objective function. The result of K-means clustering algorithm is sensitive to the initial centres used in the clustering process [29]. For example, randomly picking the initial centres may lead to accidentally picking too many centres that attracts few or no members while most of the samples allocated to a few of the centres. El Agha and Ashour [29] demonstrated that classification results can be improved when the overall shape of the dataset is considered during the initialisation phase of the K-means algorithm.

The K-means algorithm is very similar to the unsupervised Artificial Neural Network Algorithm known as Self Organising Map (SOM). The ability of all ANNs to process complex or high dimensional data at high speed makes SOM desirable for image classification [30]. In a SOM, the hidden layer consists of a matrix or a vector of neurons arranged in a grid, hexagonal or random pattern. In response to an input pattern, the neurons compete to be activated and the neuron whose weight has the smallest Euclidean distance from the input pattern is selected. The network updates the weight of the chosen neuron and its neighbours using Kohonen learning rule pattern and re-arranged its topology such that it correlates with the input vector space, thereby ensuring the same neuron will be chosen in response to subsequent input pattern similar to the current input [31]. Hastie et al. [1] considers SOM to be a constrained version of K-means algorithm in which the performance depends on the learning rate and the distance threshold, and stated under the same condition, SOM will outperform K-means clustering, if the constraints are adequate [1]. Therefore the determination of these constraints and the number of clusters are important for maximum accuracy to be achieved when using SOM for data clustering. Decision trees and Associative rules also provide simple rules by which each samples of an image dataset can be labelled.

Although the K-mean algorithm is very effective when the centres are positioned to capture the distribution of the category [1] and for the process to be credible, the number of clusters specified at the beginning of the process must closely match the number of categories present in the dataset. In contrast, hierarchical clustering methods do not require such specifications, but requires the user to specify a measure of dissimilarity between (disjoint) groups of observations. Hierarchical clustering creates a nested sequence of partitions in which the entire dataset is considered to be a single, all inclusive cluster at the highest level of the hierarchy, while each cluster at the lowest level contains a single sample. Hierarchical clustering can be implemented using either Agglomerative or Divisive approach in grouping samples into clusters [1].

In the Agglomerative approach, the clustering process starts at the lowest level and proceeds to the top, merging any two clusters whose members are considered to be similar. In the Divisive approach, the process starts from the all-inclusive clusters and repeatedly splits the dataset into smaller groups until the process attains a level where the members of each cluster are considered to be different from any other [1]. Hierarchical clustering based on the Agglomerative approach determines the affinity between samples using either single linkage, complete linkage or average linkage [1, 32]. Zhang et al. [32] explained that the Agglomerative approach is susceptible to noise and outliers because in calculating the link between two clusters to be merged, it does not consider the global similarities of the entire dataset, therefore it is not adequate for high-dimensional data such as images [32].

The use of K-means and hierarchical clustering in the unsupervised learning from an image dataset is often faced with the high dimensionality of the samples. Spectral clustering is a popular non-parametric [33] algorithm that achieves clustering through a combination of non-linear dimension reduction and K-means clustering, therefore it is preferred when the clusters are non-convex [1]. It achieves non-linear dimension reduction through the use of an undirected similarity graph to represent the pairwise distances between each sample and every other sample in the dataset, from which the normalised eigenvectors of each dimension is obtained and the desired number of columns is chosen based on their eigenvalues. The dataset samples represented by the normalised eigenvectors are then clustered using the k-means algorithm [12].

3.2.2. Parametric Clustering Methods

While on-parametric methods infer the underlying pattern structure from the dataset, parametric approaches impose a structure on the dataset. Parametric learning assumes the samples of the dataset can be represented by a probability function made up of several components [1]. In parametric clustering methods, each sample in a set is described as a combination of a finite number of functions and samples with similar combinations as assumed to be in the same cluster. The use of probabilistic parametric clustering method such as Gaussian Mixture Model (GMM) [2] and Topic-based model [34] has been shown to be successful in a wide variety of applications concerning the analysis of continuous and discrete data, respectively [33].

Given a dataset, GMM fits a single probability density function to the entire set [35]. This function is assumed to be a mixture of a finite number of Gaussian functions as shown by Equation 1 and Equation 2[36]:

$$f(X, \theta) = \sum_{k=1}^k p_k g(X; m_k, \sigma_k) \tag{1}$$

Where

$$g(X; m_k, \sigma_k) = \frac{1}{(\sqrt{2\pi}\sigma_k)^D} e^{-\frac{1}{2} \left(\frac{\|X - m_k\|_2}{\sigma_k} \right)^2} \tag{2}$$

In Equation 1, P_k is the mixing probability for the Gaussian density function k in the mixture, while m_k and σ_k are its mean and standard deviation respectively. These parameters are estimated through model fitting using Expectation-Maximisation (EM) process [36].

In GMM, knowledge of the probability density function parameters for a dataset enables the representation of each of its samples with a vector whose dimension is the same as the number of Gaussians in the mixture. While K-means clustering is regarded as hard clustering model because it exclusively maps each sample to a cluster, the GMM is considered a soft clustering method because it does not exclusively place a sample into any of the available clusters but describes the probability of its membership of each of the clusters.

Since each data sample is represented with a vector at the end of a GMM process, it is possible to represent these vectors in a multi-dimensional Euclidean space. Liu et al. [35] explained that this representation may reveal naturally occurring data patterns on, or close to subgroups within the data set and proposed the Locally Consistent Gaussian Mixture Model (LCGMM) which exploit these patterns to improve the learning performance of GMM. Experimentation conducted by the authors on Yale face and Breast cancer datasets revealed accuracies of 54.3 % and 95.5% respectively which is better than the 29.1% and 94.7% recorded by the conventional GMM [35].

Topic-based models such as PLSA and LDA are soft clustering techniques that are similar to GMM. Hoffman [34] presented the PLSA (also known as the Aspect model) for categorising collections of textual documents. Given $D = d_1, \dots, d_N$ is a set of BOV representations of images and a corresponding $W = w_1, \dots, w_V$, a set of visual vocabularies. In the PLSA modelling a joint probability model over $D \times W$ with a set of unobserved variables $Z = z_1, \dots, z_k$ is defined by the mixture in Equation 3 and Equation 4[34, 12]:

$$P(d, w) = P(d)P(w|d) \tag{3}$$

Where

$$P(w|d) = \sum_{z \in Z} P(w|z)P(z|d) \tag{4}$$

$P(w|z)$ and $P(z|d)$ are the topic specific distributions for the entire set and topic mixtures for each image respectively. The model is parameterised as shown in Equation 5[34].

$$P(d, w) = \sum_{z \in Z} P(z)P(d|z)P(w|z) \tag{5}$$

Similar to GMM, the model parameters are estimated using the EM algorithm, at the end of which each image in the dataset is represented by the topic mixture $P(z|d)$ and images with similar topic mixtures are considered to belong to the same cluster. The advantage of PLSA lies in its use of generative models obtained from the BOV representation of images for image modelling, rather than directly using the BOV representations; a step which enables the discovery of latent topics from the image data [37]. Since a group of related words is mapped to one latent topic at the end of Topic-based modelling, the resulting image representation has a reduced dimension compared to the BOV representation [34, 38].

Blei et al. [37] noted that PLSA does not provide a proper probabilistic model at the document level because the number of latent topics used to model each document grows linearly with the

size of the dataset which may lead to over-fitting. The authors therefore proposed Latent Dirichlet Analysis (LDA) which provides additional regularisation by encouraging the topic mixtures to be sparse [37]. Verbeek et al. [21] noted that LDA only outperforms PLSA when classifying small number of documents with many topics, therefore, considered PLSA to be computationally more efficient than LDA [21].

In general, Parametric models are advantageous since they provide principled ways to address issues like the number of clusters, missing feature values etc.[33]. They also combine dimension reduction capability with soft clustering [34, 12, 37, 38]. The unfortunate drawback is that they are only effective when the underlying distribution of the data is either known, or can be closely approximated by the distribution assumed by the model [33]. However similar shortcomings can be attributed to squared error based clustering algorithms such as K-means[33].

4. COMPARISON OF RELATED WORK

Unsupervised image classification is useful in the annotation of images in a large repository. In such a scenario, it can enable images to be grouped into a manageable number of clusters such that semantic labelling can be applied conveniently and efficiently. This section review literatures in which unsupervised machine learning have been applied to image categorisation.

Work	Year	Feature extraction	Image Model	Unsupervised learning approach
Xu et al. [23]	2013	SIFT	BOV	LDA / Markov Random Field / Bayesian Information criterion (BIC)
Zhang et al. [33]	2012	LBP, pixel intensity	Image Feature Histograms	Hierarchical clustering (GDL-U, AGDL)
Huang et al. [3]	2011	Dense SURF / PHOG	KNN/Hyper-graph	Spectral clustering
Mole and Ganesan [39]	2010	Local Binary Pattern	Texture histogram	K-Means
Duong et al. [40]	2008	HSV and Canny Edge Orientation histogram	Hierarchical tree	Tree matching
Kim et al. [41]	2007	Harris-Affine detector / SIFT descriptor	Page-Rank/Visual Similarity Network	Spectral clustering
Bosch et al. [12]	2006	SIFT / HOG	BOV	PLSA / KNN
Graum and Darrell [42]	2006	SIFT	Multi-resolution histogram/Similarity Graph	Spectral clustering
Todorovic and Ahuja [43]	2006	Pixel gray levels	Multiscale segmentation tree	Decision trees

Lee and Lewicki [44]	2002	Image texture	Pixel patches	ICA
Le Saux and Boujemaa [45]	2002	Pixel gray level, Fourier power spectrum and edge orientation	Feature histogram	PCA / Adaptive Robust Competition Clustering

TABLE 1: A comparison of some notable implementations of unsupervised image categorisation.

4.1. Methodology of Image Modelling

From Table 1, it can be observed that SIFT and SIFT-based algorithms are the most popular image feature extraction algorithm for the implementation of unsupervised image classification, while images are mostly described in terms of feature histograms. Bosch et al. [12] shows that dense descriptors (HOG) outperform the sparse ones, especially in the classification of images of scenes, where they enable the capturing more information than sparse features, and produce improved classification accuracy when image colour information is captured during feature extraction. This superiority was confirmed in the supervised classification by Verbeek et al. [21] the average accuracy of 61.3 % recorded by HOG descriptors was improved to 75.2 % by combining HOG with colour descriptors.

Table 1 also reveals that unsupervised image classification is yet to exploit the advantage of BOV modelling, especially its potential in supporting Semantic-Based image retrieval. The work of Huang et al. [3] is of particular interest in this study not only because of its use of the combination of PHOG and Dense SURF features to develop representation for each image, but mainly because of its the use of Regions of Interest (ROI) of each image rather than the entire image, an approach with the potential to capture spatial information when applied along with BOV modelling.

4.2. Unsupervised Learning

As mentioned in the last section, the use of non-parametric clustering such as K-means, Hierarchical or SOM on high dimensional data samples such as histograms representing images is often a computationally inefficient process. This is perhaps the reason why none of these methods is popular as the unsupervised learning approach on Table 1. In Le Saux and Boujemaa [45], reduction of image signature dimensions was achieved using Principal Component Analysis (PCA). However, the use of PCA in the reduction of the dimensionality of image representations may lead to significant loss of pattern information due to its linear approach which may be inadequate for images [27]. Zhang et al. [32] also proposed an improved hierarchical clustering: Graph Degree Linkage (GDL), which replaces the high dimensioned representation of each image with a K-Nearest Neighbour (KNN) graph developed by analysing the analysing the indegree and outdegree affinity between clusters using thus achieving non-linear dimension reduction before the Agglomerative clustering [32].

In the works of Kim et al. [41] and Huang et al. [3] spectral clustering was adopted as a means of providing the non-linear dimension reduction needed in the categorisation of images. While Kim et al. achieved non-linear dimension reduction via simple graph partition and link analysis, Huang et al. [3] introduced the application of hyper-graph partitioning to representing both local and global similarities among unlabelled images before the application of spectral clustering. Although spectral clustering method is capable of categorising any given image set irrespective of the complexity of the image signature, it favours compact and coherent groups over heterogeneous data collections and produce highly intricate clusters, which do not delineate separation between clusters [2]. It also require the calculation of an n^2 order Pair-wise distances (where n is the size of the dataset) making the computation required for the procedure very high [2], which can make its application in the classification of a large image dataset set difficult. Another problem is its

reliance on the K-means algorithm, which also means that there is the need for prior knowledge of the number of categories present in the dataset.

Alternatively, Topic-based model can be used for the same purpose. Recently, the application of Topic-based model in semantic labelling has been generating some research interest due to its ability to capture image semantic contents while achieving dimension reduction [38, 46]. Although Topic-based model clustering is rated above centre-based techniques such as K-means clustering for unsupervised image categorisation [3], its classification accuracy is affected by the use of order-less BOV image representation [19, 21, 38, 22]. Topic-model based clustering can be improved through the inclusion of spatial information of visual words during BOV modelling [3, 21, 38]. Verbeek et al. [21] improved PLSA classification accuracy from 78.5% to 80.2% using local Markov Random Field (MRF). The same approach was used by Xu et al. in the unsupervised image classification using LDA. The spatial pyramid pool by Lazebnik [22] is another alternative for the reduction of spatial incoherency during the use of Topic modelling for image categorisation.

In general, all Mixture modelling clustering approaches are memory-based methods, the model is often built using the entire dataset, and the fitting is done at evaluation or prediction time, which makes this unsupervised approach naturally unsuitable for many real-time applications [33]. The PLSA/KNN approach of Bosch et al. [12] in which the authors built a PLSA Simplex using a fraction of the image collection as training images, after which any image to be classified is fitted to the simplex using Kullback-Leibler divergence is a likely solution to this drawback of mixture models. However, a disadvantage of this approach is that its use of KNN approach introduces the need for labelled training samples.

In general, the applicability of topic modelling to unsupervised classification can be further enhanced through the development of a method for determining the number of latent topics required for an efficient implementation, and the establishment of the relationship between latent topics and semantic contents (objects and locations).

4.3. Experimentation Protocol / Results

The accuracy of a typical unsupervised classification is determined by counting the number of image classification that matches the ground truth [3]. The performance of an unsupervised image categorisation process can also be displayed in details using a confusion table, where the overall performance is determined by the average value of the diagonal entries of the confusion table [3, 12, 41]. In general, the main goal of an unsupervised image categorisation process is the allocation of each image of a dataset to one of a number of categories. However, Datta et al. [2] identified the unknown number of categories and the unknown nature of the categories in an image collection to be the two main challenges to the implementation of unsupervised image classification [2]. This sub-section examines how various implementations of unsupervised classifications have responded to these challenges.

Kim et al. [41] adopted the experimental protocol developed by Graum and Darrell [42] in evaluating their proposed unsupervised image categorisation. The experiment involves running ten iteration of the proposed algorithm on six object classes from Caltech101 (Airplanes, Cars, Faces, Motorbikes, Watches and Ketches) and three object classes from TUD/ETHZ (Giraffes, Motorbikes, Cars). The samples of images in the 6-chosen Caltech-101 classes are shown in Figure 3. For each iteration, the algorithm randomly picked 100 and 75 images from each object category in the Caltech101 and TUD/ETHZ datasets respectively. The experiment recorded an average of 98.55%, 97.30% and 95.42% for 4, 5 and 6 object Caltech101 classes respectively, and recorded 95.47% over the TUD/ETHZ dataset. In a comparative experiment, Huang et al. [3] recorded 98.55%, 97.38%, and 96.05% under similar condition, which confirms the superiority of the ROI/hyper-graph partitioning technique over the simple graph partitioning.

Huang et al. [3] extended their experiment to 4, 8, and 12 randomly selected object classes from the entire caltech101 and caltech256 dataset. 100 iterations over Caltech101 revealed 95.8%,

86.2% and 71.5%, while in the case of caltech256 87.7%, 77.1% and 64.3% were recorded. While this result confirms the superiority of the proposed hypergraph based algorithm over chosen baseline unsupervised image classification based on affinity propagation, normalised cut, and K-centre, it also demonstrates a reduction in classification accuracy as the size of the dataset increases, and the increase in complexity of the images in the collection. Although Huang et al. [3] experimented further using PASCAL VOC2008 which recorded 81.3%, 77.2% and 69.3%, a result similar to what was recorded with caltech256 [3], this study recognises the increasing popularity of Caltech-101 and Caltech-256 datasets, and considers them to be a means of providing adequate challenge to all object recognition based experiments.

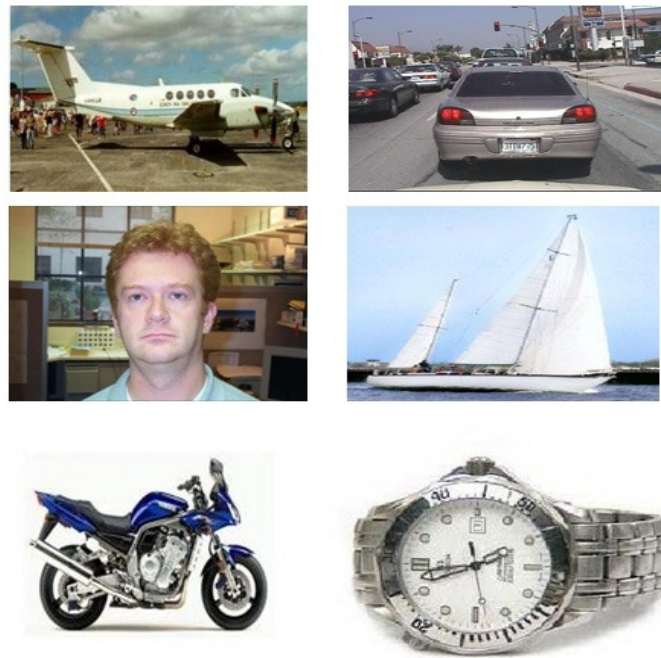


FIGURE 3: Sample images from the 6-categories chosen from Caltech-101 by Kim et al. [41] for the evaluation of the proposed unsupervised classification framework.

Zhang et al. [32] experimented with the proposed algorithm on object image databases (COIL-20 and COIL-100), hand-written digit databases (MNIST and USPS), and facial image databases (Extended Yale-B, FRGC ver2.0), and compared the GDL's performance to k-medoids (k-med), average linkage (Link), graph-based average linkage (GLink), normalized cuts (NCuts), NJW spectral clustering (NJW-SC), directed graph spectral clustering (DGSC), self-tuning spectral clustering (STSC) and Zell. The proposed algorithm demonstrated better robustness to noise and higher speed than the other algorithms. It also successfully implements unsupervised classification without the need for prior knowledge of the number of inherent categories while basing its categorisation on object matching thus accounting for the nature of each cluster [32]. This paper uses Figure 4 to illustrate the result of an evaluation of GDL carried out as part of its study. Although the average accuracy is approximately 85%, the graph indicates that the accuracy of the algorithm's classification reduces as the size of the experimental dataset increases, which may discourage its application to large image datasets. This paper considers further experiments on the object matching capability of GDL (and AGDL) using Caltech-101 and Caltech-256 to be necessary so as to establish a common evaluation environment with other recent works.

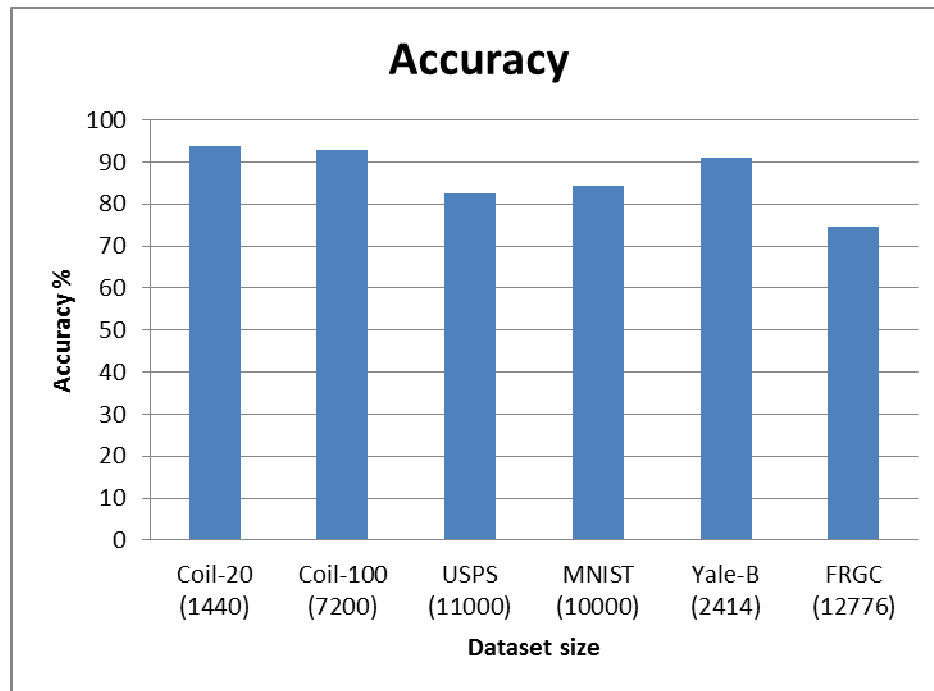


FIGURE 4: A summary of GDL classification accuracy showing variation in accuracy with the size of dataset.

Bosch et al. [12] evaluated their classification algorithm on three different datasets used in the supervised classification of Vogel and Schiele [47] and Oliva and Torralba [48], and the semisupervised classification of FeiFei and Perona [49]. The image collection used by Oliva and Torralba (OT) [48] has 8 categories, Vogel and Schiele (VS) [47] has 6 categories, while FeiFei and Perona (FP) [49] has 13 categories. Using 100 randomly selected images from the OT dataset, the authors determined that the optimum parameters for the experimentation are $V = 1500$, $Z = 25$, $K=10$ and $M = 10$, (where V = number of visual words Z = the number of topics, K = the number of neighbours for KNN and M = the number of pixels between cell locations during Dense-SIFT extraction), and the mean accuracy and standard deviation on the OT dataset (images of Natural and Man-made scenes) are of 84.78% and 1.93% respectively when dense colour SIFT was used as the feature descriptor.

The authors used the same parameters values on the OT, VS and FP datasets. For the OT dataset, they used approximately 200 images per category for both training and testing. In the case of the VS, they used 350 images per category for both training and testing, while a total of 1344 images were used for the FP dataset for training [12]. Despite the fact that Bosch et al. [12] training is unsupervised, the proposed algorithm outperforms all of the previous methods with 85.7% (against the previously recorded 74.1%) and 73.4% (against the previously recorded 65.2%) accuracies over VS and FP respectively, with the best classified scenes being highway and forest with 95.61% and 94.86% respectively.

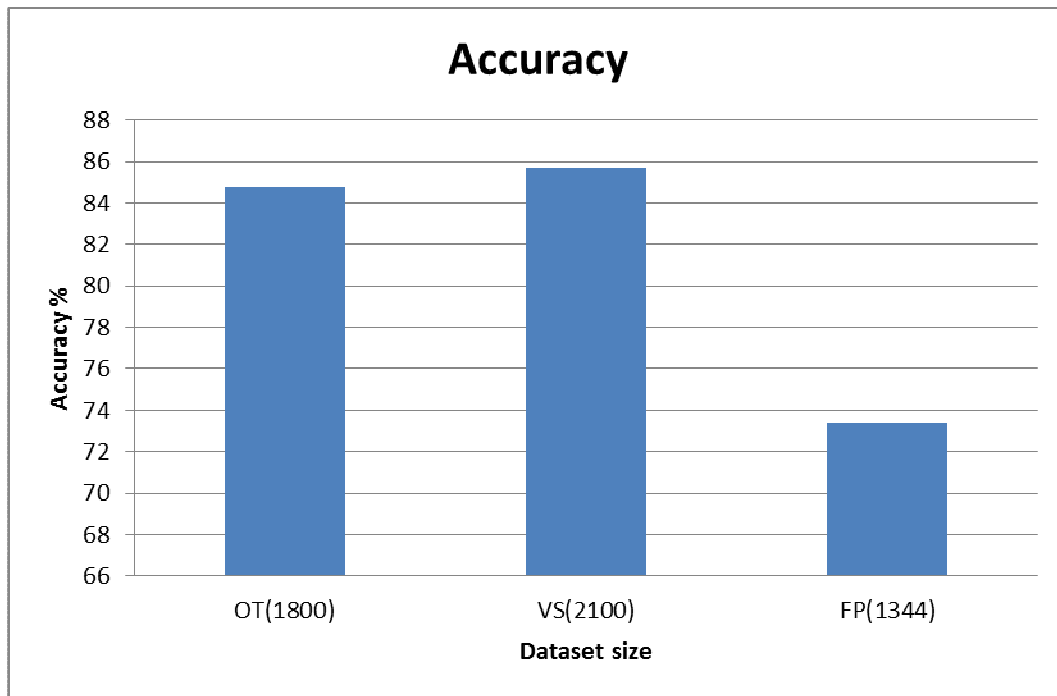


FIGURE 5: A summary of PLSA/KNN classification accuracy showing variation in accuracy with the size of dataset.

Bosch et al. [12] attributes the improved performance to the use of better features during scene categorisation, especially those features representing objects. This study recognises the ability of the combination of PLSA/KNN to implement unsupervised classification without prior knowledge of the number of inherent semantic group. This study also uses Figure 5 to demonstrate that the accuracies recorded through the use of PLSA/KNN combination in the unsupervised classification of images may not be responsive to the size of the dataset, making it a more suitable option than GDL for the categorisation of a large image collection (1000 and above). However, Figure 4 and Figure 5 do not offer a conclusive proof of the PLSA/KNN's superiority over GDL, therefore, there is a need to compare the two algorithms using Caltech-101 and Caltech-256 datasets, with emphasis on object recognition based unsupervised classification. This study also suggests an investigation into the effect of spatial incoherency (due to BOV modelling) on the PLSA/KNN combination.

5. FUTURE WORKS

The unsupervised classification of images offers a variety of opportunities as a solution to some problems in artificial intelligence. One of these problems is the elimination of the semantic gap present in CBIR via automatic annotation of images of a collection [2, 4, 50]. Wang et al. [4] explained that image retrieval researches are currently moving towards semantic-based image retrieval due to this presence of semantic gap in CBIR which has rendered its performance unsatisfactory.

Jeon et al. [51] proposed an automatic approach for the annotation of images with the aim of achieving convenient content based image retrieval. This approach depends on a supervised learning process which involves identifying common blobs from a set of labelled training images. However, like all categorisation based on supervised learning, obtaining adequate quality and quantity of labelled training images is a major challenge for this approach [51]. Therefore there is the need to look in the direction of unsupervised image categorisation.

The categorisation technique proposed by Bosch et al. [24] present a viable option for automated image annotation with the aim of eliminating semantic gap from an image retrieval process due to its use of PLSA which attempts to identify latent topics. However, the use of this technique requires labelled training samples due to the inclusion of KNN in the model, therefore there is a need for a research into a completely unsupervised but related technique. There is also a need for a research that clearly establishes the relationship between PLSA latent topics and semantic objects present in the image collection, these researches will enhance the use of unsupervised categorisation technique based on PLSA in the elimination of semantic gap from image retrieval processes.

6. CONCLUSION

Until recently, most research attention in image retrieval has been focused on feature extraction and similarity computation [2]. More recently, the need to minimise or totally eradicate the semantic gap from image retrieval systems has directed research efforts towards Semantic Image Retrieval in which the semantic gap is minimised through semantic labelling [4]. Due to its ability to categorise images without the need for training samples, unsupervised image categorisation has the potential to facilitate convenient annotation of images in a large collection. Although, non-parametric clustering techniques are simple and intuitive, their direct application to a large image database is limited because they are not very suitable for clustering high-dimensional data [1, 33].

The use of image descriptive statistics via parametric clustering enables the capturing of important information about a given dataset [1]. This is especially so for the Topic-based model such as PLSA, that captures the relationship between visual-words and the frequency of their appearance on images. Hence it can be instrumental in matching low-level features to high-level semantics; thereby supporting Semantic labelling of images. This paper also recognises the ability of PLSA/KNN combination proposed by Bosch et al. [12] and the hierarchical clustering-based GDL proposed by Zhang et al. [33] to implement unsupervised image categorisation based on the nature of inherent groups within the image collection without the need for prior knowledge of the number of categories within the collection, therefore recommends a detailed investigation and comparison of their object recognition-based categorisation abilities using the increasingly popular Caltech-101 and Caltech-256 datasets. Such research may provide a more suitable means of mapping low-level features to high level semantics than existing methods for the elimination of the semantic gap in image retrieval processes.

7. REFERENCES

- [1] T. Hastie, R. Tibshirani and J. Friedman, *The Elements of Statistical Learning-Data Mining, Inference and Prediction*, 2nd Edition ed., vol. II, Stanford: Springer, 2008, pp. 465-576.
- [2] R. Datta, D. Joshi, j. Li and J. Z. Wang, "Image Retrieval: Ideas, Influences, and Trends of the New Age," *ACM Computing Surveys*, vol. 40, no. No. 2,, p. Article 5, April 2008.
- [3] Y. Huang, Q. Liu, F. Lv, Y. Gong and D. N. Metaxas, "Unsupervised Image Categorization by Hypergraph Partition," *IEEE Transactions On Pattern Analysis And Machine Intelligence*, vol. 33, no. 6, June 2011.
- [4] H. H. Wang, D. Mohamad and N. Ismail, "Semantic Gap in CBIR: Automatic Objects Spatial Relationships Semantic Extraction and Representation," *International Journal Of Image Processing (IJIP)*, vol. 4, no. 3, 2010.
- [5] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *International Journal of Computer Vision*, January 2004.

- [6] H. Bay, T. Tuytelaars and L. V. Gool, "SURF: Speeded Up Robust Features," ETH Zurich, Zurich, 2005.
- [7] M. Guerrero, "A Comparative Study of Three Image Matching Algorithms: Sift, Surf, and Fast," Utah State University, Utah, 2011.
- [8] N. Khan, B. McCane and G. Wyvill, "SIFT and SURF Performance Evaluation against Various Image Deformations on Benchmark Dataset," in International Conference on Digital Image Computing: Techniques and Applications, Noosa, 2011.
- [9] Y. Ke and R. Sukthankar, "PCA-SIFT: A More Distinctive Representation for Local Image Descriptors," in Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference, Washington, 2004.
- [10] L. Juan and O. Gwun, "A Comparison of SIFT, PCA-SIFT and SURF," International Journal of Image Processing, vol. 3, no. 4, pp. 143-152, 2008.
- [11] C.-X. Liu, J. Yang and H. Huang, "P-SURF: A Robust Local Image Descriptor," Journal of Information Science and Engineering, vol. 27, pp. 2001-2015, January 2011.
- [12] A. Bosch, A. Zisserman and X. Munoz, "Scene Classification via PLSA," Computer Vision and Robotics Group, University of Girona, Girona, 2006.
- [13] N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection," INRIA, Montbonnot, 2004.
- [14] J. Brookshire, "Person Following using Histograms of Oriented Gradients," iRobot Corporation, Bedford, 2009.
- [15] E. Rublee, V. Rabaud, K. Konolige and G. Bradski, "ORB: an efficient alternative to SIFT or SURF," in Computer Vision (ICCV), 2011 IEEE International Conference, Barcelona, 2011.
- [16] A. Alahi, R. Ortiz and P. Vandergheynst, "Fast Retina Keypoint (FREAK)," in Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference , Providence, RI, 2012.
- [17] C.-F. Tsai, "Bag-Of-Words Representation in Image Annotation: A Review," International Scholarly Research Network, vol. 2012, pp. 1-19, 2012.
- [18] A. G. Faheema and S. Rakshit, "Feature Selection using Bag-Of-Visual-Words Representation," in Advance Computing Conference (IACC), 2010 IEEE 2nd International , Patiala, 2010.
- [19] P. Tirilly, V. Claveau and P. Gros, "Language Modelling for Bag-of-Visual Words Image Categorization," IRISA, Rennes, 2008.
- [20] Y. Zhang, Z. Jia and T. Chen, "Image Retrieval with Geometry-Preserving Visual Phrases," in Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference, Providence, RI, 2010.
- [21] J. Verbeek and B. Triggs, "Region Classification with Markov Field Aspect Models," in Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference, Minneapolis, MN, 2007.

- [22] S. Lazebnik, C. Schmid and J. Ponce, "Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories," in Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference, Illinois, 2006.
- [23] K. Xu, W. Yang, G. Liu and H. Sun, "Unsupervised Satellite Image Classification Using Markov Field Topic Model"," IEEE Geoscience And Remote Sensing Letters, vol. 10, no. 1, pp. 130-134, January 2013.
- [24] A. Bosch, A. Zisserman and X. Munoz, "Representing shape with a spatial pyramid kernel," in CIVR, Amsterdam, 2007.
- [25] Y. Bai, L. Guo, L. Jin and Q. Huang, "A novel feature extraction method using Pyramid Histogram of Orientation Gradients for smile recognition," in Image Processing (ICIP), 2009 16th IEEE International Conference, Cairo, 2009.
- [26] Z. Zhong and G. Shen, "Facial Emotion Recognition Using PHOG and a Hierarchical Expression Model," in Intelligent Networking and Collaborative Systems (INCoS), 2013 5th International Conference, Xi'an, 2013.
- [27] L. Zisheng, J. Imai and M. Kaneko, "Facial-component-based bag of words and PHOG descriptor for facial expression recognition," in Systems, Man and Cybernetics, 2009. SMC 2009. IEEE International Conference on, San Antonio, 2009.
- [28] Q. Wang, "Kernel PCA and its Applications in Face Recognition and Active Shape Models," Rensselaer Polytechnic Institute, New York, 2011.
- [29] B. Scholkopf, A. Smola and K.-R. Muller, "Kernel Principal Component Analysis," Max-Planck-Institute, Tubingen, 1999.
- [30] M. El Agha and W. M. Ashour, "Efficient and Fast Initialization Algorithm for Kmeans Clustering," I.J. Intelligent Systems and Applications, vol. 1, pp. 21-31, 2012.
- [31] M. Seetha, I. V. Muralikrishna, B. L. Deekshatulu, B. L. Malleswari, Nagaratna and P. Hedge, "Artificial Neural Networks and Other Methods Of Image Classification," Journal of Theoretical and Applied Information Technology, pp. 1039-1053, 2008.
- [32] M. Beale and D. Howard, Neural Network Toolbox, Natick: The Mathworks, 2002.
- [33] W. Zhang, X. Wang, D. Zhao and X. Tang, "Graph Degree Linkage: Agglomerative Clustering on a Directed Graph," Department of Information Engineering, The Chinese University of Hong Kong, Hong Kong, 2012.
- [34] P. K. Mallapragada, R. Jin and A. Jain, "Non-parametric Mixture Models for Clustering," Michigan State University, East Lansing, 2010.
- [35] T. Hoffman, " "Probabilistic Latent Semantic Analysis"," in Uncertainty in Artificial Intelligence, Stockholm , 1999.
- [36] J. Liu, D. Cai and X. He, "Gaussian Mixture Model with Local Consistency," in Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence (AAAI-10), 2010.
- [37] C. Tomasi, "Estimating Gaussian Mixture Densities with EM," 2004.

- [38] D. M. Blei, Y. N. Andrew and M. I. Jordan, "Latent Dirichlet Allocation," *Journal of Machine Learning Research*, vol. 3, pp. 993-1022, 2003.
- [39] S. S. S. Mole and L. Ganesan, "Unsupervised Hybrid Classification for Texture Analysis Using Fixed and Optimal Window Size," *International Journal on Computer Science and Engineering*, vol. 2, no. 9, pp. 2910-2915, 2010.
- [40] T. T. Duong, J. H. Lim, H. Q. Vu and J. P. Chevallet, "Unsupervised Learning for Image Classification based on Distribution of Hierarchical Feature Tree," in *Research, Innovation and Vision for the Future, 2008. RIVF 2008. IEEE International Conference*, Ho Chi Minh, 2008.
- [41] G. Kim, C. Faloutsos and M. Hebert, "Unsupervised Modeling of Object Categories Using Link Analysis Techniques," *Carnegie Mellon University*, Pittsburgh, 2007.
- [42] K. Grauman and T. Darrell, "The Pyramid Match Kernel: Discriminative Classification with Sets of Image Features," in *IEEE International Conference on Computer Vision*, Beijing, 2005.
- [43] S. Todorovic and N. Ahuja, "Extracting Subimages of an Unknown Category from a Set of Images," in *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference*, New York, 2006.
- [44] T. W. Lee and M. S. Lewicki, "Unsupervised Image Classification, Segmentation, and Enhancement Using ICA Mixture Models," *IEEE TRANSACTIONS ON IMAGE PROCESSING*, vol. 11, no. 3, pp. 270-279, 2002.
- [45] B. Le Saux and N. Boujerna, "Unsupervised Robust Clustering for Image Database Categorization," in *IEEE Pattern Recognition 2002 Proceedings*, 2002.
- [46] G. Passino, I. Patras and E. Izquierdo, "Aspect coherence for graph-based semantic image labelling," *IET Computer Vision*, vol. IV, no. 3, p. 183-194, 2010.
- [47] J. Vogel and B. Schiele, "Semantic Modeling of Natural Scenes for Content-Based Image Retrieval," *International Journal of Computer Vision*, vol. 72, no. 2, pp. 133-157, 2007.
- [48] A. Oliva and T. Antonio, "Modelling the shape of the scene: A holistic representation of the spatial envelope," *International Journal of Computer Vision*, vol. 3, no. 42, pp. 145-175, 2001.
- [49] L. Fei-Fei and P. Perona, "A Bayesian Hierarchical Model for Learning Natural Scene Categories," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference*, San Diego, 2005.
- [50] D. Zhang, M. Islam and G. Lu, "A review on automatic image annotation techniques," *Pattern Recognition*, vol. 45, no. 1, pp. 346-362, 2012.
- [51] J. Jeon, V. Lavrenko and R. Manmatha, "Automatic Image Annotation and Retrieval using CrossMedia," in *ACM Special Interest Group on Information Retrieval (SIGIR)*, Toronto, 2003.

Connotative Feature Extraction For Movie Recommendation

N. G. Meshram

*PG Department of Computer Science and Engineering
Prof Ram Meghe College of Engineering and Management
Badnera, 444 701, India*

meshram.naina1@gmail.com

A. P. Bhagat

*PG Department of Computer Science and Engineering
Prof Ram Meghe College of Engineering and Management
Badnera, 444 701, India*

amol.bhagat84@gmail.com

Abstract

It is difficult to assess the emotions subject to the emotional responses to the content of the film by exploring the film connotative properties. Connotation is used to represent the emotions described by the audiovisual descriptors so that it predicts the emotional reaction of user. The connotative features can be used for the recommendation of movies. There are various methodologies for the recommendation of movies. This paper gives comparative analysis of some of these methods. This paper introduces some of the audio features that can be useful in the analysis of the emotions represented in the movie scenes. The video features can be mapped with emotions. This paper provides methodology for mapping audio features with some emotional states such as happiness, sleepiness, excitement, sadness, relaxation, anger, distress, fear, tension, boredom, comedy and fight. In this paper movie's audio is used for connotative feature extraction which is extended to recognize emotions. This paper also provides comparative analysis of some of the methods that can be used for the recommendation of movies based on user's emotions.

Keywords: Audio Features, Connotative Features, Emotion Recognition, Movie Recommendation, Video Features.

1. INTRODUCTION

Due to advancement in multimedia devices, it is easy to access the private content of multimedia such as how the people enjoy the movies. Decision of which movie to watch is most of the time taken from friends suggestions. Nowadays, one can take the benefit of media recommender system [1, 2] which has the ability to suggest the video content on the basis of person's current affective state, social experiences and profile. The psychologist has investigated on the emotional properties of the film media in terms of empathy with situation and characters and also in terms of director's establishment of film making techniques which provides the emotional cues. The empathy is not with the characters which provide the affective cues with film media while film makers make the use of techniques such as editing, musical scores, lighting so as to emphasize the particular emotional interpretation by the viewer [3]. This is referred as connotation which gives the path of communication and influences how the meaning is transmitted to the audience which is conveyed by the director.

An expert system MovieGEN [4] is used for the recommendation of movies. The user's information is taken as input and their movie preferences are predicted using the support vector machine and on the basis of prediction movies are selected from the dataset and generate some questions to the users. On the user's answer it the movie is recommended for the user. Recommendation systems are the expert system where the knowledge of expert is combined with user's preferences to filter the information and provide the users with the information. There

are two main approaches for filtering collaborative and content based approach. Most of the recommendation systems use the hybrid approach combination of these two approaches. The model using SVM based learning techniques [4, 5] is used in MovieGEN. With the help of this model one can predict genres and period of movies that user prefers. The implementation of content based approach is provided in [4, 5] such that it takes into consideration the user choice which is not based on the user's past history but on the answers the user gives to the question. Collaborative approach, knowledge based approach, hybrid approach, etc. based recommendation systems have been developed for different types of domains [6]. In online movie recommendation system MovieLens [7] the first time when user logins, the system ask to rate certain movies to the user which the user has seen and these ratings are recommended with the other movies that the user has not seen. This type of filtering is based on rating which uses collaborative approach.

Recommendation system separates the relevant content from the non-relevant content which is based on the individual user's preferences are presented in [8]. The content items are described with the metadata in the content based recommender system and are stored in the item profile. There are two approaches for affective labeling that is explicit and implicit. The explicit approach provides the unambiguous labels and the drawback of this approach is the intrusiveness of the process. The implicit affective labeling is unobtrusive and it is not affected by the user's personal motives. Movies were used as the core label for building the user preferences and recommending the movie user-centric approach for labeling the content and building the user profile were used in recommender system. A framework for affective labels in recommender system where three stages were distinguish in interaction with the multimedia content such as context, induced emotion and implicit rating is proposed in [9].

In implicit labeling the most popular approach is to expose the user's to stimuli and record their responses [10, 11]. By using the emotion detection techniques the affective labels can be detected. Such that the [12] used the 2-minute video clips to expose the user's and recorded the different types of physiological signals and also the ground truth emotive response in valence-arousal plane explicitly. The features which are used commonly are the geometric features like active appearance model features and facial points and the various classification techniques includes support vector machine [12, 13]. The [14] took the advantage of facial expression, audio features, and shoulder tracking to predict the affect in valence arousal space. However, there is a need to model the response of complex approach which is the future challenge.

According to [15] emotions are personal and everyone reacts to the events or the media content that depends on cultural, personal, short term and subjective factors. For example, when two people watches the same horror movie and reacts differently or they may share similar view on movie from their individual affective response. Connotations are linked to the emotions. Emerging theories of filmic emotions elicit the mechanism that inform mapping between video features and emotional models. Filmic emotions are less character oriented giving a greater prominence to style and argues that moods generate the expectations of particular emotional cues and concluded that emotional associations provided by music encourage to relate the video features to emotional responses is suggested in [16]. The way to assess the affective dimension of media is by the use of expected mood is proposed in [17]. Film maker tries to communicate the set of emotions when they produce the movie for the audience. The emotional clustering of films for different genres is described in [16, 17]. This clustering approach may be used to target the user emotions. The audio and visual low level features in a high dimensional space to extract the meaningful patterns by SVM inference engine is proposed in [15]. It states audio cues are more informative than visual with respect to the affective content.

During human to human interaction changes in the person's affective state plays an important role. In affective computing one of the applications can be human computer interaction. The learning system called "Multimodal Human Computer Interaction: Towards a Proactive Computer" is discussed in [13]. In this type of learning environment the user is able to explore the games by interacting with the computer avatar. In this environment multiple sensors were used to detect

and track the behavioral cues of user, camera was used to record the facial expression of user, to track the eye movements, to monitor the task progress and a microphone was used to record the speech signals. And based on this over all information, the avatar offers an appropriate strategy in this type of learning environment. The psychological studies [1], [16] indicated that while judging someone's affective state, people rely on the facial expression and the vocal intonations. The motivation for the audio-visual fusion is the improved reliability. Current techniques which are used for the detection and tracking of facial expression are very sensitive to the clutter, head pose and variations in lighting condition and the current techniques which are used for the speech processing are very sensitive to auditory noise. The table 1 shows the comparative analysis of available movie recommendation methodologies.

Title	Methodology	Feature Extraction	Data Used	Experiment	System	Measures
Affective Recommendation of Movies Based on Selected Connotative Features	Scene ratings by users	Visual, audio, film grammar, color	Movie scenes	SVR	SVR model	Connotative space, precision For example, precision for Top -3 close and far scenes at different emotional distance is given by (Top-3 Close) $d=0$, (precision@3) 0.30, (Top-3 Far) $d=4$, (precision@3) 0.22.
	Feature extraction					
	Feature selection					
	Regression					
	Scene recommendation					
MovieGEN: A Movie Recommendation System	Machine learning based preference prediction	Vector format (sample is quantified into vector of integer number)	Movie preferences	SVM Correlation analysis using SVM regression	Machine learning model	Training data, preference vector, clustering
Affective Labeling in a Content-Based Recommender System for Images	Emotion detection evaluation methodology	Active appearance model feature, facial feature, audio features	images	SVM, NaiveBayes, AdaBoost, C4.5	CBR system	Accuracy, confusion matrix, classifier For example, for explicit affective labeling classifier SVM (Precision) 0.68, (Recall) 0.54, (F)measure 0.60
	Affective CBR system evaluation methodology					
A Connotative Space for Supporting Movie Affective Recommendation	Span the semantic space	Video features, emotional responses, audio features, visual features	Movie segments, video frames movie scenes	Osgood's semantic space	SVM inference engine	Connotative space, emotional wheel
	Validate by inter-rater agreement					
	Support affective recommendation					
Audio-Visual Affective Expression Recognition Through Multistream Fused HMM	MFHMM	Facial features, audio features, multistream fused HMM	images	Face only HMM, pitch only HMM, energy only HMM, independent only HMM, MFHMM	Motion units	Expression, performance For example, MFHMM for (happy) 0.70
	HCI					
Affective Video Content Representation and Modeling	Affective level	Movie facts, feelings or emotions	Movie scenes, video clips	Video content	Arousal, valence	Arousal and valence
	Cognitive level					
Robust Face-Name Graph Matching for Movie Character Identification	Character identification	To identify faces of characters	movie	Face-Name graph matching	Error correcting graph matching	Face track Detection Accuracy clip 1 (# Face track) 372, (# Track detected) 352, (Accuracy) 95.2%
	Face-name Graph matching					
SMERS: Music Emotion Recognition Using Support Vector Regression	Feature extraction	Pitch, tempo, loudness, tonality, key, rhythm and harmonics	Music and emotion	SVM,SVR and GMM	Thayer's two-dimensional emotion	SVR training parameters and obtained optimums in polar representation Name of parameters:
	Mapping					

	Training				model	Distance Angle Nu (u) : 2^{-8} , 2^{-8} , Gamma of RBF (g) : 2^{-10} , 2^{-4} Cost (C) : 2^8 , 2^6 mean squared error: 0.02498 ,0.09834
--	----------	--	--	--	-------	--

TABLE 1: Comparative analysis of available movie recommendation methods.

The presence of entrainment at the emotion level in cross-modality settings and its implications on multimodal emotion recognition systems is investigated in [18]. The relationship between acoustic features of the speaker and facial expressions of the interlocutor during dyadic interactions are explored. More than 72 % speakers displayed similar emotions, indicating strong mutual influence in their expressive behaviors. The cross-modality, cross-speaker dependency, using mutual information framework is also investigated. A strong relation between facial and acoustic features of one subject with the emotional state of the other subject is revealed. It has been suggested that the expressive behaviors from one dialog partner provide complementary information to recognize the emotional state of the other dialog partner. Classification experiments exploited cross-modality, cross-speaker information. The emotion recognition experimentations are demonstrated using the IEMOCAP [19] and SEMAINE [20] databases.

In this paper the audio features CEP and SPEC are used to extract the emotions from the audiovisual datasets. These extracted features are used to affection based movie recommendations. An experimental result shows that the proposed methodology can be utilized to provide precise and effective recommendation results in faster manner. The total time required for recommendation can be reduced due to use of CEP and SPEC features.

2. CONNOTATIVE FEATURE EXTRACTION

The audio features such as sound, voice and music play an important role of expression in shaping the scene affection of the audience. The algorithm called as Piecewise Bezier Volume Deformation (PBVD) [5] tracking can be use to extract the facial features. It uses the 3-D facial mesh model that is embedded in multiple Bezier volumes. With the help of the movements of control points the shape of mesh can be changed in Bezier volumes which guarantees that the surface patch to be smooth and continuous.

Entropic Signal Processing system is used as a software package for the audio feature extraction. It implements the algorithm using cross correlation function and dynamic programming. In the experimental results [3], [5] it is suggested that pitch and energy are the important factors in affect classification. The pitch varies from person to person. Males speak with lower pitch than the females.

Multistream Fused Harmonic Markov Model can be used for integrating audio and visual feature which is used to construct new structure for linking the multiple components HMM according to maximum entropy principle and maximum mutual information. It is the generalization of two-stream fused HMM. Such that MFHMM is used for recognition problem with more than the two feature streams.

The problem of emotions which is connected to the use of other people’s affective annotations while the connotative properties agreed by the people’s emotional reaction provides accurate recommendation methods and to establish the method for performing research on emotions such as users behavior, users self reporting and learning method shows how to translate the low level and mid-level properties of videos into inter-subjective for affective analysis of film. The emotional characters of videos are used to study the narrow set of situations such as for the sporting events or the horror movies. The advantage of ranking is based on similarities between items which are close to human emotions instead of using absolute labeling.

3. AUDIO ANALYSIS USING CEPSTRUM (CEP) AND SPECTRUM (SPEC)

A Cepstrum (Cep) is the result of taking the Inverse Fourier transform (IFT) of the logarithm of the estimated spectrum of a signal. There is a complex cepstrum, areal cepstrum, a power cepstrum, and phase cepstrum. The power cepstrum in particular finds applications in the analysis of human speech. The name "cepstrum" was derived by reversing the first four letters of "spectrum". Operations on cepstra are labelled quefrency analysis, liftering, or cepstral analysis.

The power cepstrum of a signal is defined as the squared magnitude of the inverse Fourier transform of the logarithm of the squared magnitude of the Fourier transform of a signal. Mathematically represented as

$$|F^{-1}\{\log(|F\{f(t)\}|^2)\}|^2$$

The complex cepstrum holds information about magnitude and phase of the initial spectrum, allowing the reconstruction of the signal. The real cepstrum uses only the information of the magnitude of the spectrum. The process is defined as $FT \rightarrow \text{abs}() \rightarrow \log \rightarrow IFT$, i.e., that the cepstrum is the "inverse Fourier transform of the log-magnitude Fourier spectrum". The cepstrum is a representation used in homomorphic signal processing, to convert signals (such as a source and filter) combined by convolution into sums of their cepstra, for linear separation. In particular, the power cepstrum is often used as a feature vector for representing the human voice and musical signals. For these applications, the spectrum is usually first transformed using the mel scale. The result is called the mel-frequency cepstrum or MFC (its coefficients are called mel-frequency cepstral coefficients, or MFCCs). It is used for voice identification, pitch detection and much more. The cepstrum is useful in these applications because the low-frequency periodic excitation from the vocal cords and the formant filtering of the vocal tract, which convolve in the time domain and multiply in the frequency domain, are additive and in different regions in the quefrency domain.

The frequency spectrum of a time-domain signal is a representation of that signal in the frequency domain. The frequency spectrum can be generated via a Fourier transform of the signal, and the resulting values are usually presented as amplitude and phase, both plotted versus frequency. A source of sound can have many different frequencies mixed. A musical tone's timbre is characterized by its harmonic spectrum. Sound spectrum is one of the determinants of the timbre or quality of a sound or note.

Spectrum (Spec) analysis, also referred to as frequency domain analysis or spectral density estimation, is the technical process of decomposing a complex signal into simpler parts. As described above, many physical processes are best described as a sum of many individual frequency components. Any process that quantifies the various amounts (e.g. amplitudes, powers, intensities, or phases), versus frequency can be called spectrum analysis. Spectrum analysis can be performed on the entire signal. Alternatively, a signal can be broken into short segments (sometimes called frames), and spectrum analysis may be applied to these individual segments. Periodic functions (such as $\sin(t)$) are particularly well-suited for this sub-division. General mathematical techniques for analyzing non-periodic functions fall into the category of Fourier analysis. The Fourier transform of a function produces a frequency spectrum which contains all of the information about the original signal, but in a different form. This means that the original function can be completely reconstructed (synthesized) by an inverse Fourier transform. For perfect reconstruction, the spectrum analyzer must preserve both the amplitude and phase of each frequency component. These two pieces of information can be represented as a 2-dimensional vector, as a complex number, or as magnitude (amplitude) and phase in polar coordinates. A common technique in signal processing is to consider the squared amplitude, or power; in this case the resulting plot is referred to as a power spectrum.

In practice, nearly all software and electronic devices that generate frequency spectra apply a fast Fourier transform (FFT), which is a specific mathematical approximation to the full integral solution. Formally stated, the FFT is a method for computing the discrete Fourier transform of

a sampled signal. Because of reversibility, the Fourier transform is called a representation of the function, in terms of frequency instead of time; thus, it is a frequency domain representation. Linear operations that could be performed in the time domain have counterparts that can often be performed more easily in the frequency domain. Frequency analysis also simplifies the understanding and interpretation of the effects of various time-domain operations, both linear and non-linear. Some kind of averaging is required in order to create a clear picture of the underlying frequency content (frequency distribution). Typically, the data is divided into time-segments of a chosen duration, and transforms are performed on each one. Then the magnitude or (usually) squared-magnitude components of the transforms are summed into an average transform. This is a very common operation performed on digitally sampled time-domain data, using the discrete Fourier transform. Such processing techniques often reveal spectral content even among data which appears noisy in the time domain.

4. PROPOSED CONNOTATIVE FEATURE EXTRACTIONS FOR MOVIE RECOMMENDATION

Figure 1 shows the overall system flow which includes the extraction of audio features. There are different types of movies such as Horror, Action, Thriller, Comedy, Animation etc. So, all these different types of movies or movie scenes will be displayed to the multiple users. From these movies/movie scenes the audio will be extracted. That is, to extract the connotative features from the extracted audio using support vector regression (SVR). These extracted audio features will be mapped with the connotative features. SVR will rate/rank according to the connotative features. Then SVR will ask or query to the users. After asking questions SVR will compare the result with the actual user preferences. Then the user will modify the rate/rank of the connotative features. At last, SVR will recommend the movie to the user.

One of the first decisions in any pattern recognition system is the choice of what features to use: How exactly to represent the basic signal that is to be classified, in order to make the classification algorithm's job easiest. Speech recognition is a typical example. The most popular feature representation currently used is the Mel-frequency Cepstral Coefficients or MFCC. Another popular speech feature representation is known as RASTA-PLP, an acronym for Relative Spectral Transform - Perceptual Linear Prediction. PLP was originally proposed by Hynek Hermansky [10] as a way of warping spectra to minimize the differences between speakers while preserving the important speech information. RASTA is a separate technique that applies a band-pass filter to the energy in each frequency sub-band in order to smooth over short-term noise variations and to remove any constant offset resulting from static spectral coloration in the speech channel e.g. from a telephone line. Machine learning focuses on the prediction based on the known properties learned from the training data and utilized in data mining and knowledge discovery domains. The data sets for any machine learning model are parts such as input and output. The support vector machine based on statistical learning can be used to overcome the problems such as over-fitting, local minimum and sufficient for high generalization. The introduction of fuzzy logic into SVM created a new multi-layer SVM which can be applied later into the numerical regression problem.

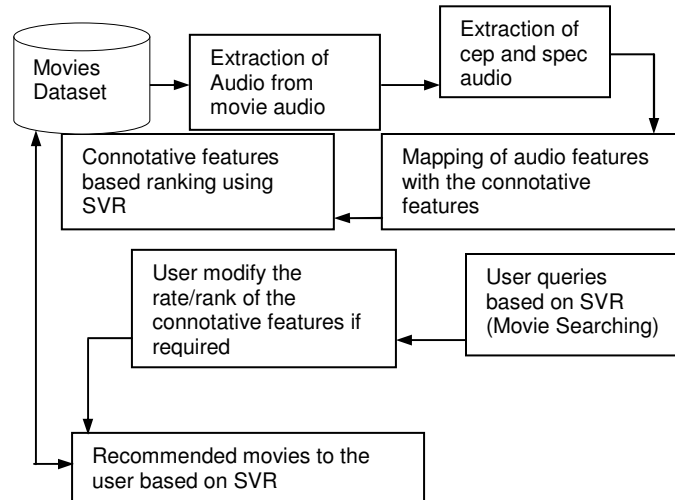


FIGURE 1: Proposed Connotative Feature Based Movie Recommendation System Model.

The proposed approach for recommendation of movies works as explained here. Initially for the extraction audio features from the movie scenes the audio is extracted from the movies. After the extraction of audio the cep and spec features are extracted from the audio. The set of movies which belong to the popular films is considered and asked users to rate each of these scenes on three connotative dimensions. The distances are computed using the Earth mover's distance on rate histogram axis (N, T, E) as:

$$\Delta_{i,j}^x = EMD(H_i^x, H_j^x) \quad x \in \{N, T, E\}$$

Support vector regression is used to relate connotative distances on the user's rates. The connotative distances are predicted between the movie scenes when the model is validated. With the help of SVR model the connotative distances are predicted. The user has to choose the query item and the scenes which have small connotative distance from query are then recommended to the users.

5. EXPERIMENTAL RESULTS

The connotative features happiness, sleepiness, excitement, fight, etc are mapped with the extracted audio features from the movie scenes. Some of the sample extracted audio features and their cep and spec values are presented below. The extracted happiness audio feature is shown in figure 2 and its cep and spec values are shown in table 2. The extracted sleepiness audio feature is shown in figure 3 and its cep and spec values are shown in table 3. The extracted excitement audio feature is shown in figure 4 and its cep and spec values are shown in table 4. The extracted fight audio feature is shown in figure 5 and its cep and spec values are shown in table 5.

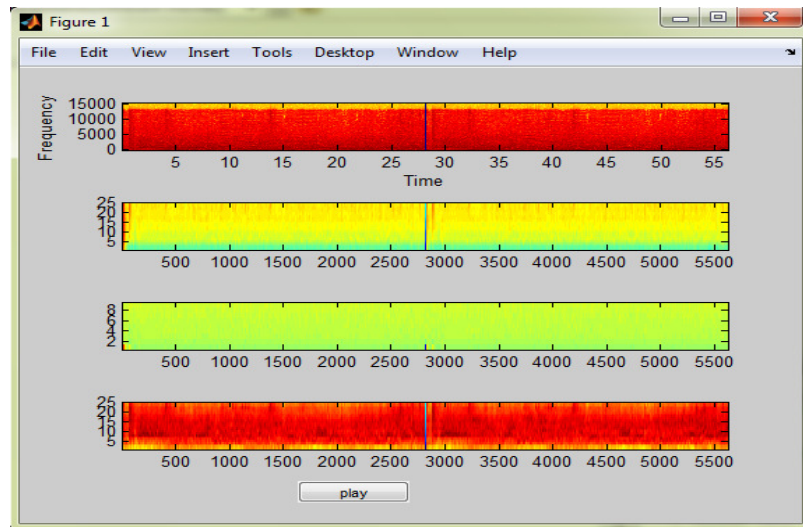


FIGURE 2: Snapshot of Happiness Audio Features Extracted From Movie Scenes.

Audio Feature	Min Value	Max Value
Cep1	-5.7452	4.2331
Cep2	-0.8349	8.6520
Spec1	3.031	10.03
Spec2	1.2945	1.609

TABLE 2: Cep and Spec Features Extracted for Happiness Movie Scenes.

After getting the audio feature values these values are mapped with the connotative features as shown in the table 6. For getting the exact mapping of the audio features with the connotative features 25 videos of each category i.e. happy, sleepy, excitement, sad, etc. are used for extracting audio features. Then by identifying the similarity between the extracted features as per movie category it has been mapped with the connotative features.

6. CONCLUSION AND FUTURE SCOPE

Feature selection algorithms are popular in different disciplines such as array analysis and multimedia analysis. The main advantage is the reduction of number of features processed and better understanding of problem. Automatic emotion recognition from the movie scenes by using the audio features by using different machine learning classification algorithms such as SVM, SVR and HMM are analyzed in this paper. SVR is the best option for the selection of the connotative feature selection and mapping for the recommendation of movie scenes. From the experimental results it has been cleared that the audio features can be successfully utilized for the affection based recommendation of movies.

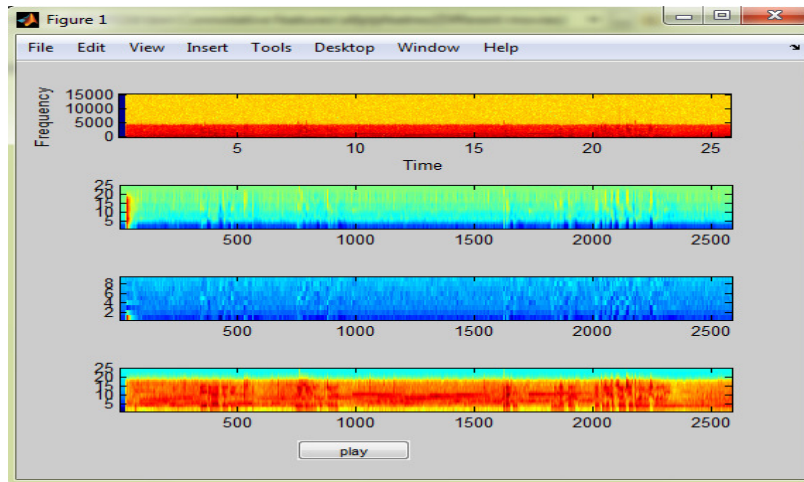


FIGURE 3: Snapshot of Sleepiness Audio Features Extracted From Movie Scenes.

Audio Feature	Min Value	Max Value
Cep1	-1.1378	2.3878
Cep2	-1.4271	7.3225
Spec1	0.0305	35.1272
Spec2	1.2945	1.1354

TABLE 3: Cep and Spec Features Extracted for Sleepiness Movie Scenes.

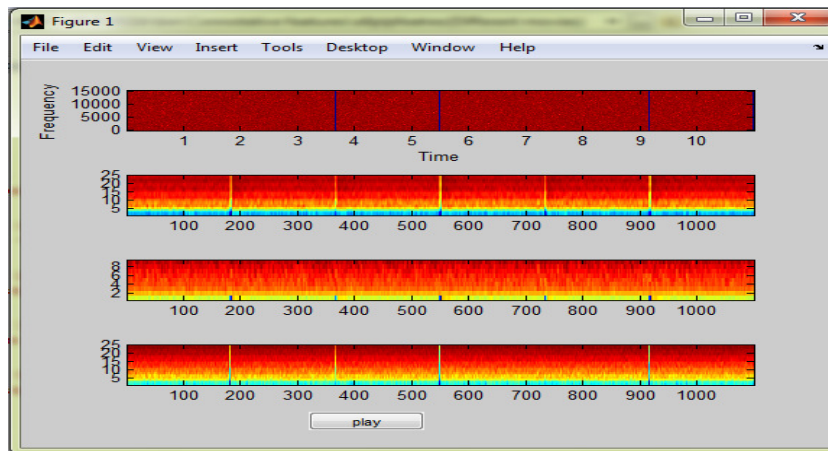


FIGURE 4: Snapshot of Excitement Audio Features Extracted From Movie Scenes.

Audio Feature	Min Value	Max Value
Cep1	-1.3713	0.0692
Cep2	-0.7832	8.9013
Spec1	0.0409	1.2208
Spec2	54.6322	2.3649

TABLE 4: Cep and Spec Features Extracted for Excitement Movie Scenes.

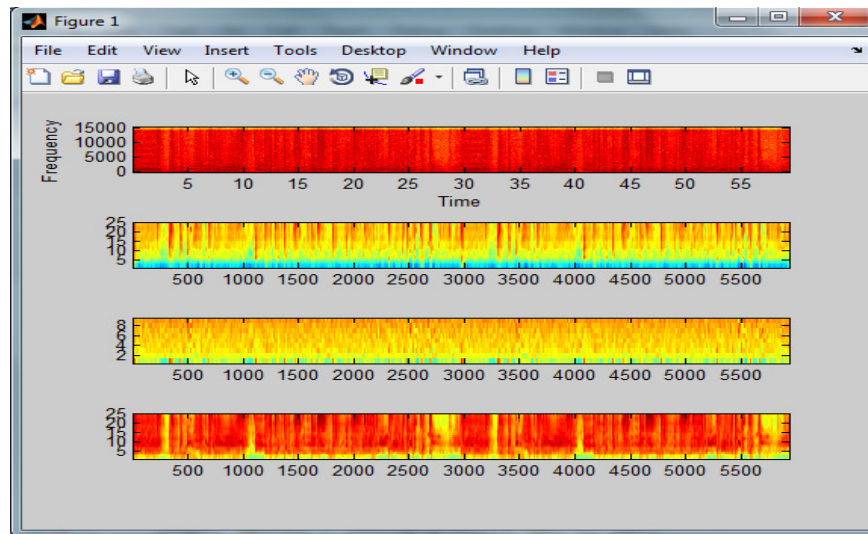


FIGURE 5: Snapshot of Fight Audio Features Extracted From Movie Scenes.

Audio Feature	Min Value	Max Value
Cep1	-2.3014	0.8861
Cep2	-0.8789	8.1108
Spec1	0.0144	6.4190
Spec2	1.3094	1.0660

TABLE 5: Cep and Spec Features Extracted for Fight Movie Scenes.

Connotative Features	Audio Features			
	Cep1	Spec1	Cep2	Spec2
Happiness	-0.75605	51.7108	3.90855	1.45185
Sleepiness	0.625	17.57885	2.9477	1.21495
Excitement	-0.65105	0.63085	4.05905	28.49855
Sadness	-1.10775	6.641	3.2043	1.5155
Relaxation	-0.5584	13.91215	3.47175	1.25855
Anger	-0.0658	11.6543	2.59565	1.92725
Distress	-0.64005	6.40145	3.1438	1.25915
Fear	-0.7565	11.5238	3.20855	1.1733
Tension	-0.77315	81.0031	3.5785	1.4281
Boredom	-0.7664	9.45755	2.85995	3.2319
Comedy	-1.1063	16.264	3.5623	1.3652
Fight	-0.70765	3.2167	3.61595	1.1877

TABLE 6: Mapping of Audio Features with Connotative Features.

Other classification algorithms such as fuzzy and KNN can be considered for the further research. The future plans are to compare the results of machine learning based emotion recognition with human performed arousal and valence data. This paper compares various strategies for the connotative feature based movie recommendations. This initial comparison is helpful for the researchers to get basics of affective recommendations of movies. This paper also presents how the audio features can be mapped with the connotative features so that these features can be

utilized for the emotion recognition. The proposed strategy can be extended so that it can support the web based interface to provide the relevant movies as per user query.

7. REFERENCES

- [1] G. M. Smith, "Film Structure and the Emotion System". *Cambridge, U.K.: Cambridge Univ. Press, 2003.*
- [2] E. A. Eyjolfsson, G. Tilak, N. Li, "MovieGEN: A Movie Rec System," *IEEE Trans on Mult.*, Vol 15, no 5, Aug 2010.
- [3] J. Kim and E. Andre, "Emotion Recognition Based on Physiological Changes in Music Listening," *IEEE Trans on Pattern Analysis and Machine Intelligence*, Vol. 30, no. 12, Dec 2008.
- [4] L. Canini, S. Benini, and R. Leonardi, "Affective Recommendation of Movies Based on Selected Connotative Features," *IEEE Trans on circuits and systems for video technology*, vol. 23, no. 4, April 2013.
- [5] A. Tawari and M. Trivedi, "Face Expression Recognition by Cross Modal Data Association," *IEEE Trans on Multimedia*, Vol. 15, no. 7, Nov 2013.
- [6] A. Smola and B. Schölkopf, "A Tutorial on Support Vector Regression", Sep 2003.
- [7] L. Lu, D. Liu and H. Zhang, "Automatic Mood Detection and Tracking of Music Audio Signals," *IEEE Trans on audio, speech and language processing*, Vol. 14, no. 1, Jan 2006.
- [8] A. Metallinou, A. Katsamanis, F. Eyben and S. Narayanan, "Context-Sensitive Learning for Enhanced Audiovisual Emotion Classification" *IEEE Trans on affective computing*, Vol. 3, no. 2, Apr-Jun 2012.
- [9] Z. Deng, U. Neumann, T. Kim, M. Bulut, and S. Narayanan, "Expressive Facial Animation Synthesis by Learning Speech Coarticulation and Expression Spaces," *IEEE Trans on visualization and computer graphics* Vol. 12, No. 6 Dec 2006.
- [10] M. Aeberhard, S. Schlichthärle, N. Kaempchen, and T. Bertram, "Track-to-Track Fusion with Asynchronous Sensors Using Information Matrix Fusion for Surround to-Track Fusion with Asynchronous Sensors Using Information Matrix Fusion for Surround Environment Perception," *IEEE Trans on intelligent transportation system*, Vol. 13, no. 4, Dec 2012.
- [11] W. Xu, C. Chang, Y. S. Hung and P. Fung, "Asymptotic Properties of Order Statistics Correlation Coefficient in the Normal Cases," *IEEE Trans on signal processing*, Vol. 56, no. 6, Jun 2008.
- [12] C. Tsai, L. Kang, C. Lin and W. Lin, "Scene-Based Movie Summarization via Role-Community Networks," *IEEE Trans On Circuits and Systems for Video Technology*, 2013.
- [13] X. Zhang, W. Hu, H. Bao, and S. Maybank, "Robust Head Tracking Based on Multiple Cues Fusion in the Kernel- Bayesian Framework" *IEEE Trans On Circuits and Systems for Video Technology*, Vol. 23, No. 7, July 2013.
- [14] J. Kim and E. Andre, "Emotion Recognition Based on Physiological Changes in Music Listening" *IEEE Trans On Pattern Analysis and Machine Intelligence* , Vol. 30, No. 12, Dec 2008.
- [15] A. Hanjalic and L. Xu, "Affective video content representation and modeling," *IEEE Trans. Multimedia*, vol. 7, no. 1, pp. 143–154, Feb. 2005.

- [16] C. Liu, D. Wang, J. Zhu, and B. Zhang, "Learning a Contextual Multi-Thread Model for Movie/TV Scene Segmentation," *IEEE Trans on Multimedia*, Vol 15, no.4, June 2013.
- [17] D. Lottridge, M. Chignell, and M. Yasumura, "Identifying Emotion through Implicit and Explicit Measures: Cultural Differences, Cognitive Load, and Immersion," *IEEE Transaction on Affective Computing*, Vol 3, no. 2, April-June 2012.
- [18] Soroosh Mariooryad and Carlos Busso, "Exploring Cross-Modality Affective Reactions for Audiovisual Emotion Recognition," *IEEE Transaction on affective computing*, Vol. 4, no. 2, April-June 2013.
- [19] C. Busso, M. Bulut, C. Lee, A. Kazemzadeh, E. Mower, S. Kim, J. Chang, S. Lee, and S. Narayanan, "IEMOCAP: Interactive emotional dyadic motion capture database," *Journal of Language Resources and Evaluation*, vol. 42, no. 4, pp. 335–359, December 2008.
- [20] G. McKeown, M. Valstar, R. Cowie, M. Pantic, and M. Schröder, "The SEMAINE database: Annotated multimodal records of emotionally colored conversations between a person and a limited agent," *IEEE Transactions on Affective Computing*, vol. 3, no. 1, pp. 5–17, January-March 2012.

Vision-Based Localization and Scanning of 1D UPC and EAN Barcodes with Relaxed Pitch, Roll, and Yaw Camera Alignment Constraints

Vladimir Kulyukin

*Department of Computer Science
Utah State University
Logan, UT, USA*

vladimir.kulyukin@usu.edu

Tanwir Zaman

*Department of Computer Science
Utah State University
Logan, UT, USA*

tanwir.zaman@aggiemail.usu.edu

Abstract

Two algorithms are presented for vision-based localization of 1D UPC and EAN barcodes with relaxed pitch, roll, and yaw camera alignment constraints. The first algorithm localizes barcodes in images by computing dominant orientations of gradients (DOGs) of image segments and grouping smaller segments with similar DOGs into larger connected components. Connected components that pass given morphological criteria are marked as potential barcodes. The second algorithm localizes barcodes by growing edge alignment trees (EATs) on binary images with detected edges. EATs of certain sizes mark regions as potential barcodes. The algorithms are implemented in a distributed, cloud-based system. The system's front end is a smartphone application that runs on Android smartphones with Android 4.2 or higher. The system's back end is deployed on a five node Linux cluster where images are processed. Both algorithms were evaluated on a corpus of 7,545 images extracted from 506 videos of bags, bottles, boxes, and cans in a supermarket. All videos were recorded with an Android 4.2 Google Galaxy Nexus smartphone. The DOG algorithm was experimentally found to outperform the EAT algorithm and was subsequently coupled to our in-place scanner for 1D UPC and EAN barcodes. The scanner receives from the DOG algorithm the rectangular planar dimensions of a connected component and the component's dominant gradient orientation angle referred to as the skew angle. The scanner draws several scanlines at that skew angle within the component to recognize the barcode in place without any rotations. The scanner coupled to the localizer was tested on the same corpus of 7,545 images. Laboratory experiments indicate that the system can localize and scan barcodes of any orientation in the yaw plane, of up to 73.28 degrees in the pitch plane, and of up to 55.5 degrees in the roll plane. The videos have been made public for all interested research communities to replicate our findings or to use them in their own research. The front end Android application is available for free download at Google Play under the title of NutriGlass.

Keywords: Skewed Barcode Localization & Scanning, Image Gradients, Mobile Computing, Eyes-free Computing, Cloud Computing.

1. INTRODUCTION

A common weakness of many 1D barcode scanners, both free and commercial, is the camera alignment requirement: the smartphone camera must be horizontally or vertically aligned with barcodes to obtain at least one complete scanline for successful barcode recognition (e.g., [1], [2]). This requirement is acceptable for sighted users but presents a serious accessibility barrier to visually impaired (VI) users or to users who may not have adequate dexterity for satisfactory camera alignment. One approach that addresses the needs of these two user groups is 1D barcode localization and scanning with relaxed pitch, roll, and yaw constraints. Figure 1 shows

the roll, pitch, and yaw planes of a smartphone. Such barcode processing is also beneficial for sighted smartphone users, because the camera alignment requirement no longer needs to be satisfied.

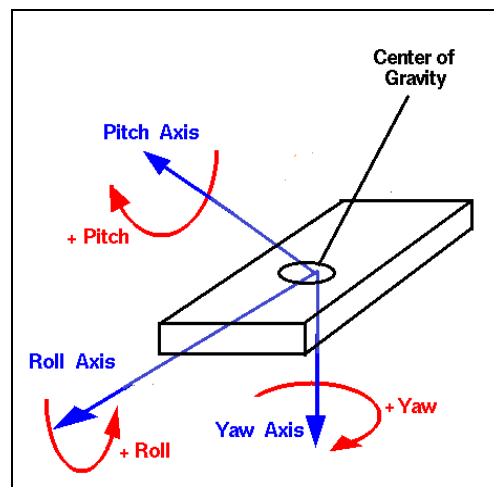


FIGURE 1: Pitch, roll, and yaw planes of a smartphone.

In our previous research, we designed and implemented an eyes-free algorithm for vision-based localization and decoding of aligned 1D UPC barcodes by augmenting computer vision techniques with interactive haptic feedback loops to ensure that the smartphone camera is horizontally and vertically aligned with the surface on which a barcode is sought [3]. We later developed another algorithm for localizing 1D UPC barcodes skewed in the yaw plane [4]. In this article, two new algorithms are presented that relax the alignment constraints to localize and scan 1D UPC and EAN barcodes in frames captured by the smartphone's camera misaligned with product surfaces in the pitch, roll, and yaw planes.

Our article is organized as follows. Section 2 covers related work. Section 3 presents the first localization algorithm that uses dominant gradient orientations (DOGs) in image segments. Section 4 presents the second barcode localization algorithm that localizes barcodes by growing edge alignment trees (EATs) on binary images with detected edges. Section 5 presents our 1D barcode localization experiments on a corpus of 7,545 images extracted from 506 videos of bags, boxes, bottles, and cans in a supermarket. Section 6 describes our 1D barcode scanner for UPC and EAN barcodes and how it is coupled to the DOG localization algorithm. Section 7 gives technical details of our five node Linux cluster for image processing. Section 8 presents our barcode scanning experiments on the same sample of 7,545 images as well as our experiments to assess the ability of the system to scan barcodes skewed in the pitch, roll, and yaw planes. These experiments were conducted on ten products (two cans, two bottles, and six boxes) in our laboratory. Section 9 discusses our findings and outlines several directions for future work.

2. RELATED WORK

The use of smartphones to detect barcodes has been the focus of many research and development projects for a long time. Given the ubiquity and ever increasing computing power of smartphones, they have emerged as a preferred device for many researchers to implement and test new techniques to localize and scan barcodes. Open source and commercial smartphone applications, such as RedLaser (redlaser.com) and ZXing (code.google.com/p/zxing), have been developed. However, these systems are intended for sighted users and require them to center barcodes in images.

To the best of our knowledge, the research effort most closely related to the research presented in this article is the research by Tekin and Coughlan at the Smith-Kettlewell Eye Research

Institute [1, 2, 5]. Tekin and Coughlan have designed a vision-based algorithm to guide VI smartphone users to center target barcodes in the camera frame via audio instructions and cues. However, the smartphone cameras must be aligned with barcode surfaces and the users must undergo training before they can use the mobile application in which the algorithm is implemented.

Wachenfeld et al. [6] present another vision-based algorithm that detects barcodes on a mobile phone via image analysis and pattern recognition methods. The algorithm overcomes typical distortions, such as inhomogeneous illumination, reflections, or blurriness due to camera movement. However, a barcode is assumed to be present in the image. Nor does the algorithm appear to address the localization and scanning of barcodes misaligned with the surface in the pitch, roll, and yaw planes.

Adelmann et al. [7] have developed a randomized vision-based algorithm for scanning barcodes on mobile phones. The algorithm relies on the fact that, if multiple scanlines are drawn across the barcode in various arbitrary orientations, one of them might cover the whole length of the barcode and result in successful barcode scans. This recognition scheme does not appear to handle distorted or misaligned images.

Lin et al. [8] have developed an automatic barcode detection and recognition algorithm for multiple and rotation invariant barcode decoding. However, the system requires custom hardware. In particular, the proposed system is implemented and optimized on a DM6437 DSP EVM board, a custom embedded system built specifically for barcode scanning.

Galo and Manduchi [9] present an algorithm for 1D barcode reading in blurred, noisy, and low resolution images. However, the algorithm detects barcodes only if they are slanted by less than 45 degrees in the yaw plane. The researchers appear to make no claims on the ability of their algorithm to handle barcodes misaligned in the pitch and roll planes.

Peng et al. [10] present a smartphone application that helps blind users locate EAN barcodes and expiration dates on product packages. It is claimed that, once barcodes are localized, existing barcode decoding techniques and OCR algorithms can be utilized to obtain the required information. The system provides voice feedback to guide the user to point the camera to the barcode of the product, and then guide the user the point the camera to the expiration date for OCR. The system requires user training and does not appear to handle misaligned barcodes.

3. BARCODE LOCALIZATION ALGORITHM I

3.1 Dominant orientation of gradients

The first algorithm is based on the observation that barcodes characteristically exhibit closely spaced aligned edges with the same angle, which sets them apart from text and graphics. Let I be an RGB image and let f be a linear relative luminance function computed from a pixel's RGB components:

$$f(R, G, B) = 0.2126R + 0.7152G + 0.0722B. \quad (1)$$

The gradient of f and the gradient's orientation θ can then be computed as follows:

$$\nabla f = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right]; \theta = \tan^{-1} \left(\frac{\partial f}{\partial x} / \frac{\partial f}{\partial y} \right). \quad (2)$$

Let M be an $n \times n$ mask, $n > 0$, convolved with I . Let the dominant orientation of gradients of M , $DOG(M)$, be the most frequent discrete gradient orientation of all pixels covered by M . Let (c, r) be the column and row coordinates of the top left pixel of M . The regional gradient orientation

table of M , $RGOT(c, r)$, is a map of discrete gradient orientations to their frequencies in the region of I covered by M . The global gradient orientation table ($GGOT$) of I is a map of the top left coordinates of image regions covered by M to their RGOTs. In our implementation, both GGOTs and RGOTs are implemented as hash tables.

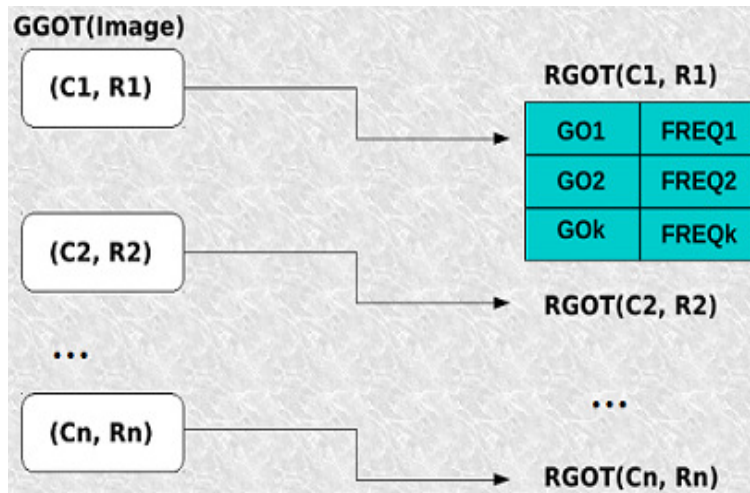


FIGURE 2: Logical structure of global gradient orientation table.

Figure 2 shows the logical organization of an image's GGOT. Each GGOT maps (c, r) 2-tuples to RGOT tables that, in turn, map discrete gradient orientations (i.e., GO_1, GO_2, \dots, GO_k in Figure 2) to their frequencies (i.e., $FREQ_1, FREQ_2, \dots, FREQ_k$ in Figure 2) in the corresponding image regions. Each RGOT represents the region whose top left coordinates are specified by the corresponding (c, r) 2-tuple and whose size is the size of M . Each RGOT is subsequently converted into a single real number called the most frequent gradient orientation. This number, denoted by $DOG(M)$, is the region's dominant orientation of gradients, also known as its DOG.



FIGURE 3: UPC-A barcode skewed in the yaw plane.

Consider an example of a barcode skewed in the yaw plane in Figure 3. Figure 4 gives the DOGs for a 20×20 mask convolved with the image in Figure 3. Each green square is a 20×20 image region. The top number in each square is the region's DOG, in degrees, whereas the bottom



FIGURE 6: D-neighborhood found in GGOT in Figure 4.

3.2 D-Neighborhoods

Let an RGOT 3-tuple (c_k, r_k, DOG_k) consist of the coordinates of the top left corner, (c_k, r_k) , of the subimage covered by an $n \times n$ mask M whose dominant gradient orientation is DOG_k . We define DOG-neighborhood (D-neighborhood) is a non-empty set of RGOT 3-tuples (c_k, r_k, DOG_k) such that for any such 3-tuple (c_k, r_k, DOG_k) there exists at least one other 3-tuple (c_j, r_j, DOG_j) such that $(c_j, r_j, DOG_j) \neq (c_k, r_k, DOG_k)$ and $sim((c_j, r_j, DOG_j), (c_k, r_k, DOG_k)) = True$, where sim is a Boolean similarity metric. Such similarity metrics define various morphological criteria for D-neighborhoods. In our implementation, the similarity metric returns true when the square regions specified by the top left coordinates (i.e., (c_k, r_k) and (c_j, r_j)) and the mask size n are horizontal, vertical, or diagonal neighbors and the absolute difference of their DOGs does not exceed a small threshold.



FIGURE 7: D-neighborhood detected in Figure 6.



FIGURE 8: Multiple d-neighborhoods.

An image may have several D-neighborhoods. The D-neighborhoods are computed simultaneously with the computation of the image's GGOT. As each RGOT 3-tuple becomes available during the computation of RGOTs, it is placed into another hash table for D-neighborhoods. The computed D-neighborhoods are filtered by the ratio of the total area of their component RGOTs to the image area. For example, Figure 6 shows RGOTs marked as blue rectangles that are grouped into a D-neighborhood by the similarity metric defined above, because they are horizontal, vertical, and diagonal neighbors and the absolute difference of their DOGs does not exceed a small threshold. This resultant D-neighborhood is shown in Figure 7. This neighborhood is computed in parallel with the computation of the GGOT in Figure 4.

Detected D-neighborhoods are enclosed by minimal rectangles that contain all of their RGOT 3-tuples, as shown in Figure 7, where the number in the center of the white rectangle denotes the neighborhood's DOG. A minimal rectangle is the smallest rectangle that encloses all RGOTs of the same connected component. All detected D-neighborhoods are barcode region candidates. There can be multiple D-neighborhoods detected in an image. For example, Figure 8 shows all detected D-neighborhoods when the threshold is set to 0.01, which is too low. Figure 7 exemplifies an interesting and recurring fact that multiple D-neighborhoods tend to intersect over a barcode.

The DOG algorithm is given in Appendix A. Its asymptotic complexity is $O(k^2)$, where k is the number of masks that can be placed on the image. This is because, in the worst case, each RGOT constitutes its own D-neighborhood, which makes each subsequent call to the function *FindNeighbourhoodForRGOT()*, which finds the home D-neighborhood for each newly computed RGOT, to unsuccessfully inspect all the D-neighborhoods computed so far. A similar worst-case scenario happens when there is one D-neighborhood that absorbs all computed RGOT, which takes place when the similarity metric is too permissive. Both of these scenarios, while theoretically possible, rarely occur in practice.

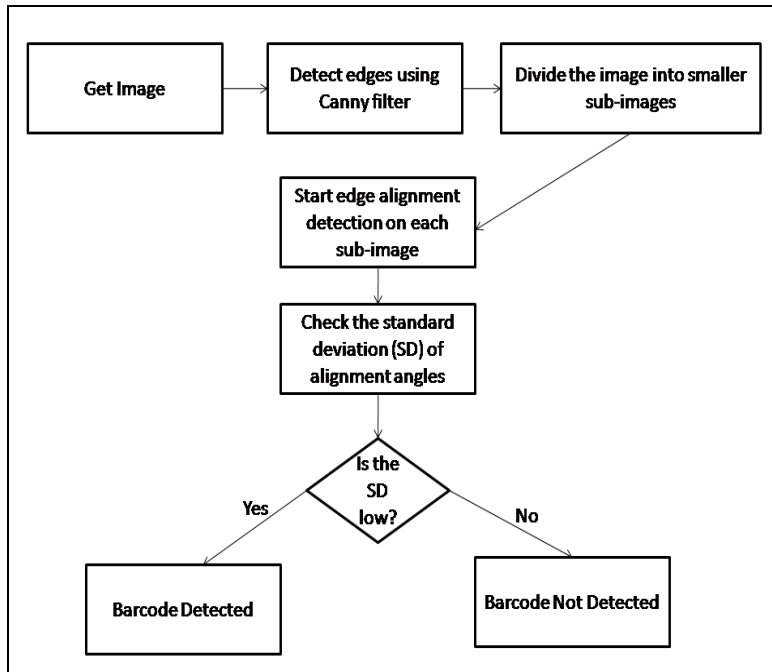


FIGURE 9: EAT algorithm.

4. BARCODE LOCALIZATION ALGORITHM II

The second algorithm for skewed barcode localization with relaxed pitch, roll, and yaw constraints is based on the idea that binarized image areas with potential barcodes exhibit continuous patterns (e.g., sequences of 1's or 0's) with dynamically detectable alignments. We call the data structure for detecting such patterns the edge alignment tree (EAT). Figure 10 shows several examples of EATs marked with different colors. In principle, each node in an EAT can have multiple children. In practice, since barcodes have straight lines or bars aligned side by side, most dynamically computed EATs tend to be linked lists, as shown in Figure 10.

The EAT algorithm is given in Appendix B. Each captured frame is put through the Canny edge detection filter and binarized [11] as one of the most reliable edge detection techniques [12]. The binarized image is divided into smaller rectangular subimages. The size of subimages is specified by the variable *maskSize* in the function *ComputeEATs()* in Appendix B. Each mask is a square. The subimages are scanned row by row and column by column. For each subimage, the EATs are computed to detect the dominant skew angle of the edges.

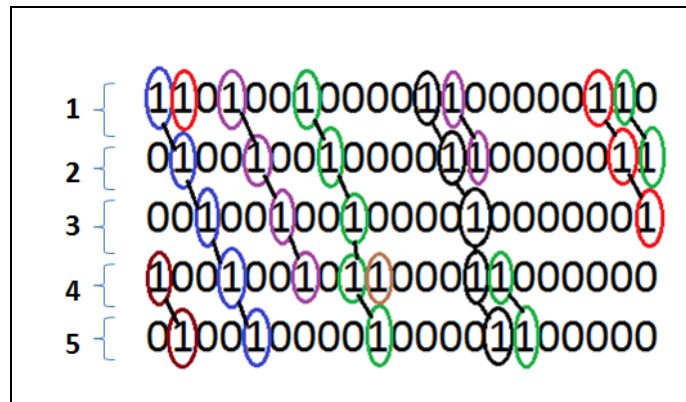


FIGURE 10: EATs from binarized edges.

The algorithm starts from the first row of each subimage and moves to the right column by column until the end of the row is reached. If a pixel's value is 255 (white), it is treated as a 1, marked as the root of an EAT, and stored in the list of nodes. If the current row is the subimage's first row, the node list contains only the root nodes. In the second row (and all subsequent rows), whenever a white pixel is found, it is checked against the current node list to see if any of the nodes can become its parent. The test of parenthood is done by checking the angle between the potential parent's pixel and the current pixel with Equation 3.

$$\theta = \tan^{-1} \left(\frac{\text{current pixel's row} - \text{root's row}}{\text{root's column} - \text{current pixel's column}} \right) \quad (3)$$

If the angle is between 45 and 135 degrees, the current pixel is added to the list of children of the found parent. Once a child node is claimed by a parent, it cannot become the child of any other parent. This parent-child matching is repeated for all nodes in the current node list. If none of the nodes satisfies the parent condition, the orphan pixel becomes a EAT root that can be matched with a child on the next row. Once all the EATs are computed, as shown in Figure 10, the dominant angle is computed for each EAT as the average of the angles between each parent and its children, all the way down to the leaves. For each subimage, the standard deviation of the angles is computed for all EATs. If the subimage's standard deviation is low (less than 5 in our current implementation), the subimage is classified as a potential barcode region.

The asymptotic complexity of the EAT algorithm, which is presented in Appendix B, is as follows. Let I be an image of side n . Let the mask's size be m . Recall that each mask is a square. Let k denote the number of masks, where $k = n^2/m^2$. The time complexity of the Canny edge detector is $\Theta(n)$ [11]. In the worst case, for each subimage, m trees are grown of the height equal to the number of rows in the subimage, which takes km^3 computations, because each grown EAT must be linearly scanned to compute its orientation angle. Thus, the overall time complexity can be calculated as $\Theta(n) + O(k * m^3) = O(n^2 m)$.

5. LINUX CLUSTER

We built a Linux cluster out of five nodes for cloud-based computer vision and data storage. Each node is a PC with an Intel Core i5-650 3.2 GHz dual-core processor that supports 64-bit computing. The processors have 3MB of cache memory. The nodes are equipped with 6GB DDR3 SDRAM and have Intel integrated GMA 4500 Dynamic Video Memory Technology 5.0. All nodes have 320 GB of hard disk space. Ubuntu 12.04 LTS was installed on each node. We installed JDK 7 in each node.

We used JBoss (<http://www.jboss.org>) to build and configure the cluster and the Apache mod_cluster module (http://www.jboss.org/mod_cluster) to configure the cluster for load balancing. The cluster has one master node and four slaves. The master node is the domain controller that runs mod_cluster and httpd. All nodes are part of a local area network and have hi-speed Internet connectivity.

The JBoss Application Server (JBoss AS) is a free open-source Java EE-based application server. In addition to providing a full implementation of a Java application server, it also implements the Java EE part of Java. The JBoss AS is maintained by [jboss.org](http://www.jboss.org), a community that provides free support for the server. JBoss is licensed under the GNU Lesser General Public License (LGPL).

The Apache mod_cluster module is an httpd-based load balancer. The module is implemented with httpd as a set of modules for httpd with mod_proxy enabled. This module uses a communication channel to send requests from httpd to a set of designated application server nodes. An additional communication channel is established between the server nodes and httpd. The nodes use the additional channel to transmit server-side load balance factors and lifecycle

events back to httpd via a custom set of HTTP methods collectively referred to as the Mod-Cluster Management Protocol (MCMP).

The `mod_cluster` module provides dynamic configuration of httpd workers. The proxy's configuration is on the application servers. The application server sends lifecycle events to the proxies, which enables the proxies to auto-configure themselves. The `mod_cluster` module provides accurate load metrics, because the load balance factors are calculated by the application servers, not the proxies.

All nodes in our cluster run JBoss AS 7. Jboss AS 7.1.1 is the version of the application server installed on the cluster. Apache httpd runs on the master node with the `mod_cluster-1.2.0` module enabled. The Jboss AS 7.1.1 on the master and the slaves are discovered by httpd. A Java servlet for image recognition is deployed on the master node as a web archive file. The servlet's URL is hardcoded in every front end smartphone. The servlet receives images uploaded with HTTP POST requests, recognizes barcodes, and sends an HTML response back to front end smartphones. No data caching is done on the servlet or the front end smartphones.

6. BARCODE LOCALIZATION EXPERIMENTS

6.1 Experimental Design

The DOG and EAT algorithms were tested on images extracted from 506 video recordings of common grocery products. Each video recorded one specific product from various sides. The videos had a 1280 x 720 resolution and were recorded on an Android 4.2.2 Galaxy Nexus smartphone in a supermarket in Logan, Utah. All videos were recorded by a user who held a grocery product in one hand and a smartphone in the other. The videos covered four different categories of products: bags, bottles, boxes, and cans. The average video duration is fifteen seconds. There were 130 box videos, 127 bag videos, 125 box videos, and 124 can videos. Images were extracted from each video at the rate of 1 frame per second, which resulted in a total of 7,545 images, of which 1950 images were boxes, 1905 images were bags, 1875 images were bottles, and 1860 images were cans. These images were used in the experiments and the outputs of both algorithms, i.e., enclosed barcode regions (see Figures 7 and 8), were manually evaluated by the two authors independently.

A frame was classified as a *complete true positive* if there was a D-neighborhood where at least one straight line across all bars of a localized barcode. A frame was classified as a *partial true positive* if there was a D-neighborhood where a straight line could be drawn across some, but not all, bars of a barcode. An image was classified as a *false positive* if there was a D-neighborhood that covered an image area with no barcode and no D-neighborhood detected in the same image covered a barcode either partially or completely. For example, in Figure 8, the D-neighborhood, with a DOG of 100, in the upper left corner of the image, covers an area with no barcode. However, the entire frame in Figure 8 is classified as a complete true positive, because there is another D-neighborhood, with a DOG of 47, in the center of the frame that covers a barcode completely. A frame was classified as a *false negative* when it contained a barcode but no D-neighborhoods covered that barcode either completely or partially and no D-neighborhood covered an area with no barcode, because in the latter case, the frame was classified as a false positive. A frame was classified as a true negative when the frame contained no barcode and could not be classified as a false positive.

6.2 DOG Localization Experiments

The DOG algorithm was implemented in Java with OpenCV2.4 bindings for Android 4.2 and ran on Galaxy Nexus and Samsung Galaxy S2. In our previous research [4], the best threshold values for each mask size were determined. These values are given in Table 1. We used these values to run the experiments for each category of products. The performance analysis for the DOG algorithm is presented in the pie charts in Figures 11-14 for each category of products.

Product Type	Mask Size	Threshold
Bag	20 x 20	0.02
Bottle	40 x 40	0.02
Box	20 x 20	0.02
Can	20 x 20	0.01

TABLE 1: Optimal mask sizes and threshold.

As can be seen in Figures 11 – 14, the DOG algorithm produces very few false positives or false negatives and performs well even on unfocussed and blurry images. The large percentages of true negatives show that the algorithm is conservative. This is done by design, because it is more important, especially for blind and visually impaired users, to avoid false positives. Moreover, at a rate of two frames per second, eventually there will be a frame where a barcode is successfully and quickly localized. The algorithm produces very few false negatives, which indicates, that, if a frame contains a barcode, it will likely be localized.

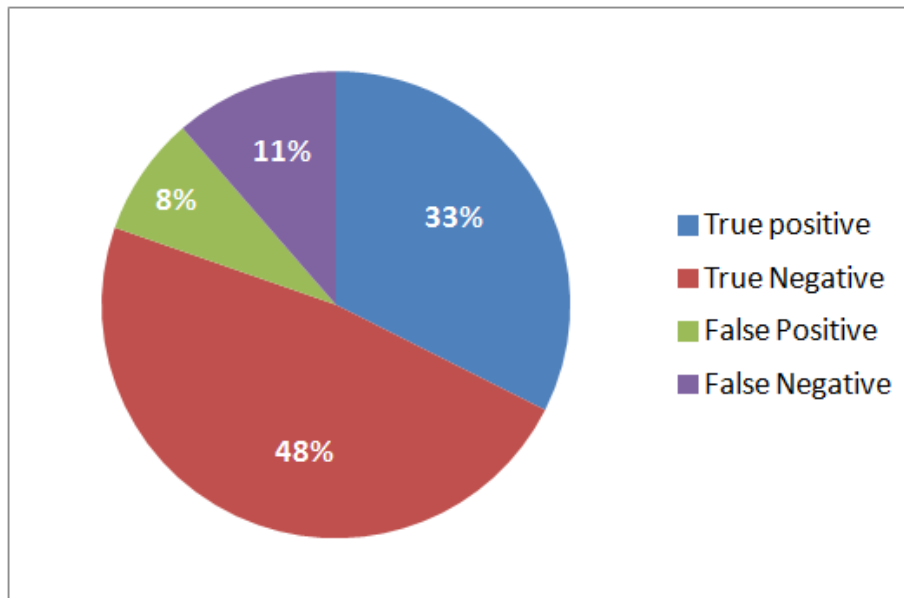


FIGURE 11: DOG performance on bags.

Figure 15 gives the DOG precision, recall, accuracy, and specificity values for different categories of products. The graph shows that the algorithm produced the best results for boxes, which can be attributed to the clear edges and smooth surfaces of boxes that result in images without major distortions. The largest percentages of false positives were on bags (8 percent) and bottles (7 percent). Many surfaces of these two product categories had shiny materials that produced multiple glares and reflections.

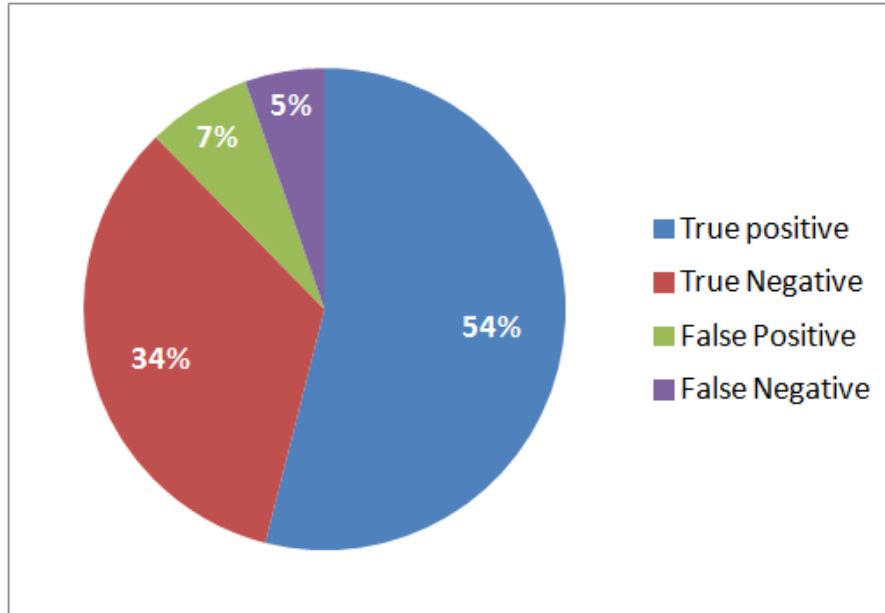


FIGURE 12: DOG performance on bottles.

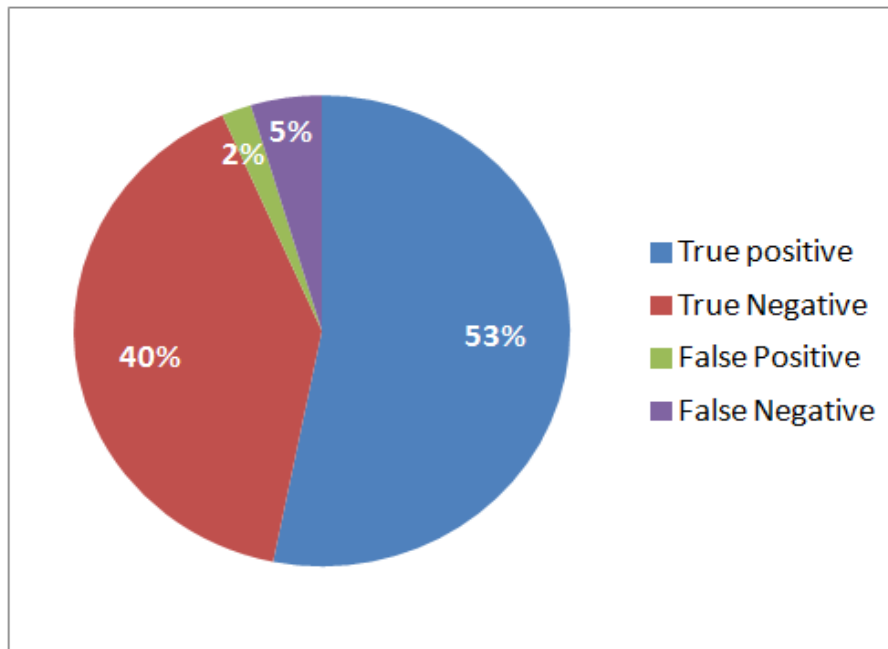


FIGURE 13: DOG performance on boxes.

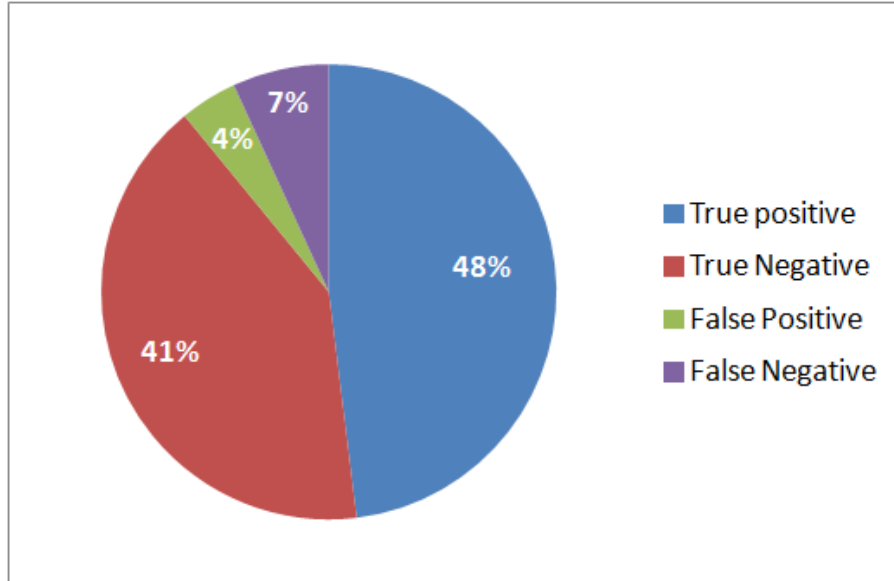


FIGURE 14: DOG performance on cans.

6.3 DOG Localization Experiments

The EAT algorithm was also implemented in Java with OpenCV2.4 bindings for Android 4.2 and ran on Galaxy Nexus and Samsung Galaxy S2. The algorithm's pseudocode is given in Appendix B. The EAT algorithm gave best results for bags with a Canny threshold of (300,400). For bottles, boxes and cans, it performed best with a threshold of (400,500). The algorithm gave the most accurate results for window size of 10 for all categories of products.

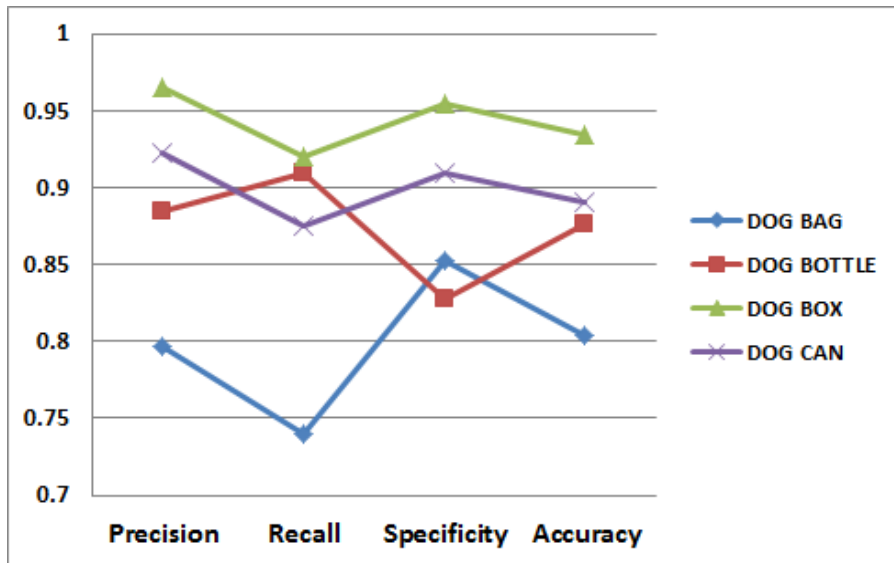


FIGURE 15: DOG precision, recall, specificity, and accuracy.

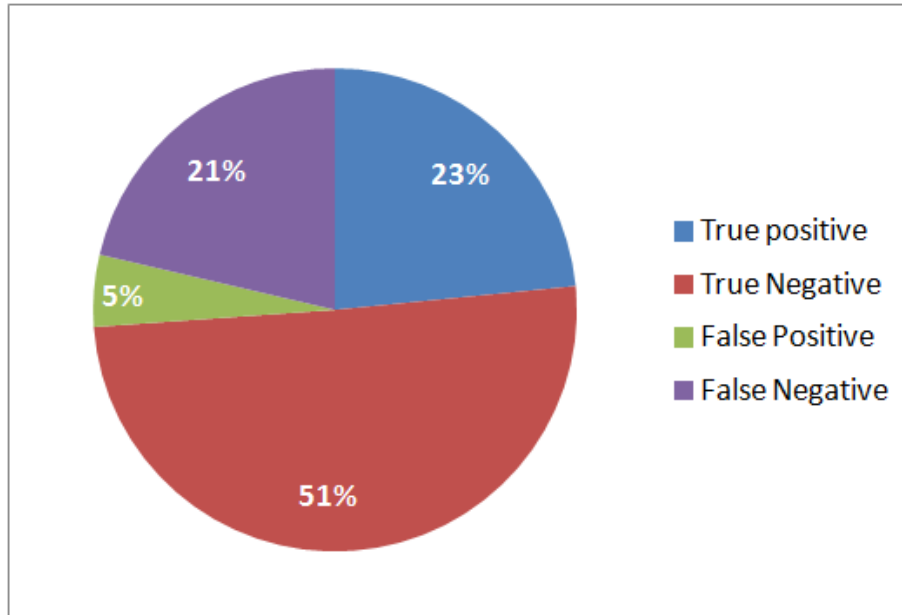


FIGURE 16: EAT performance on bags.

The charts in Figures 16 - 19 summarize the performance of the EAT algorithm for each product category. The experiments were performed on the same sample of 509 videos. The video frames in which detected barcode regions had single lines crossing all bars of a barcode were considered as *complete true positives*. Frames where such lines covered some of the bars were classified as *partial true positives*. Figure 20 shows examples of complete and partial true positives. Frames, where detected barcode regions did not have any barcodes were classified as *false positives*. *True negatives* were the frames with no barcodes where the algorithm did not detect anything. *False negatives* were the frames where the algorithm failed to detect a barcode in spite of its presence.

The experiments showed that the false negative rates of the EAT algorithm were substantially higher than those of the DOG algorithm. The true positives rates of the EAT algorithm were also lower. Figure 21 shows the statistical analysis for each category of products. The best results were produced for boxes and the worst for cans. These results were similar to the results of the DOG algorithm.

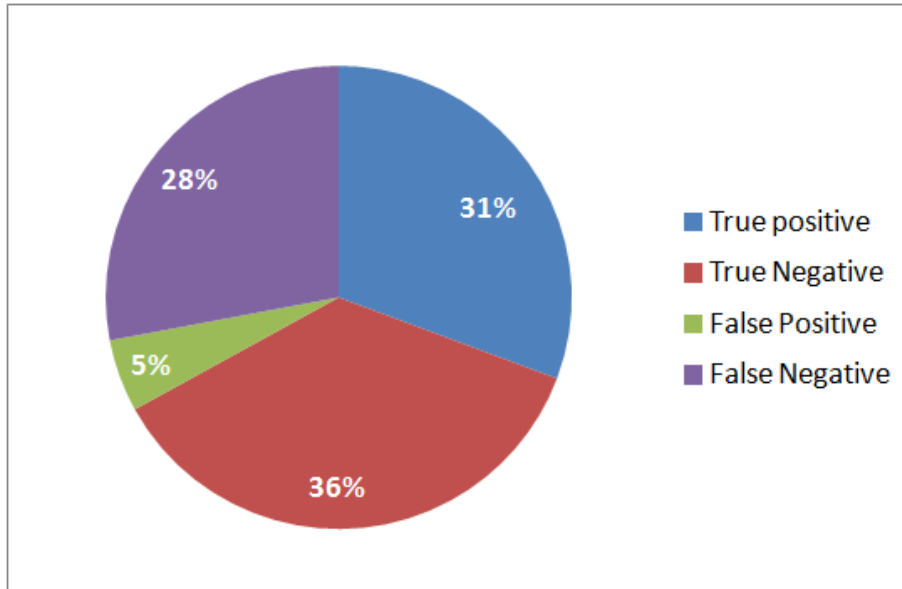


FIGURE 17: EAT performance on bottles.

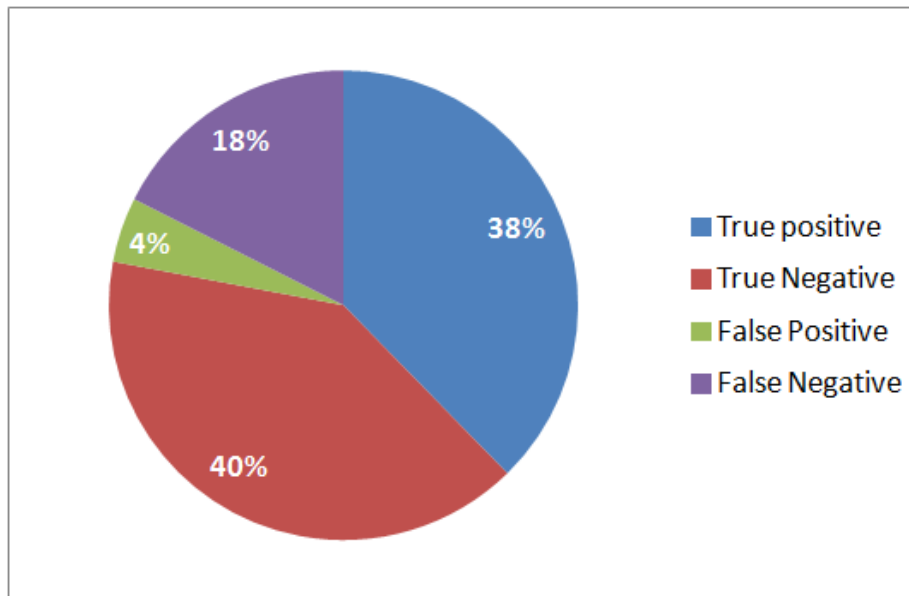


FIGURE 18: EAT performance on boxes.

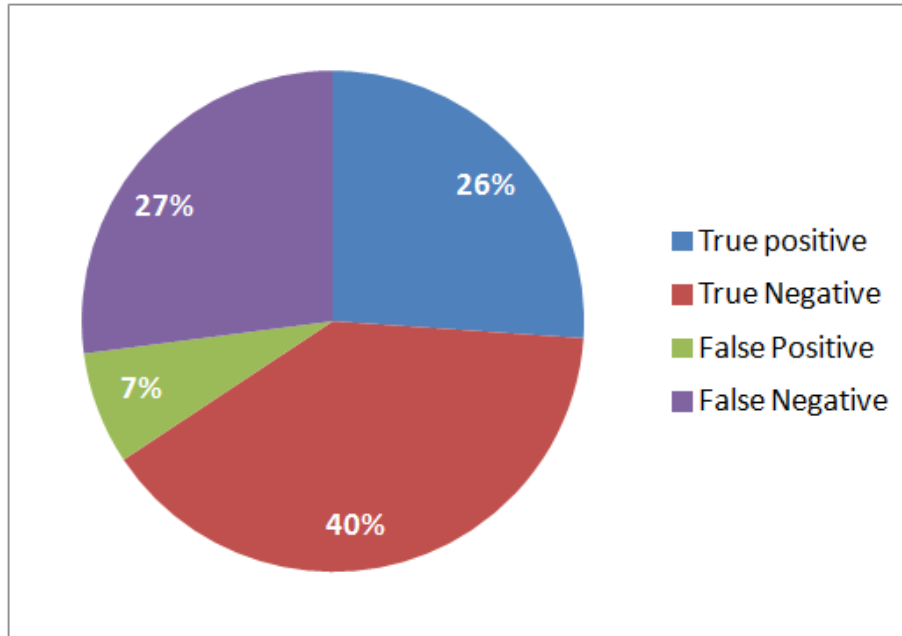


FIGURE 19: EAT performance on cans.

6.4 Comparison of Algorithms

It is evident from the experiments that the DOG algorithm has a better performance than the EAT algorithm. The bar graph in Figure 22 gives the DOG and EAT localization probabilities. These probabilities were calculated using Equation 4, where $n(\text{True positive})$ and $n(\text{False negative})$ are the numbers of frames were classified as true positives and true negatives, respectively.

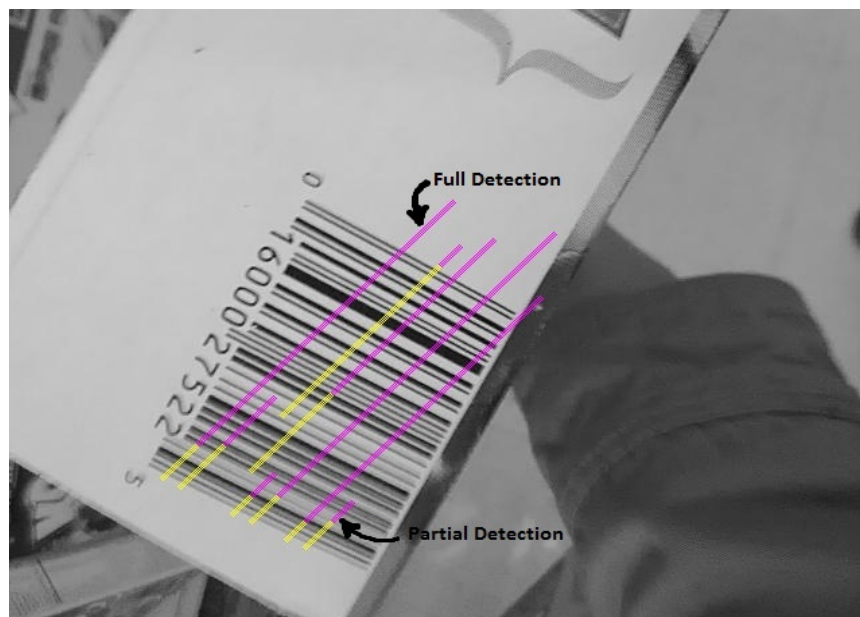


FIGURE 20: Complete and partial EAT detection.

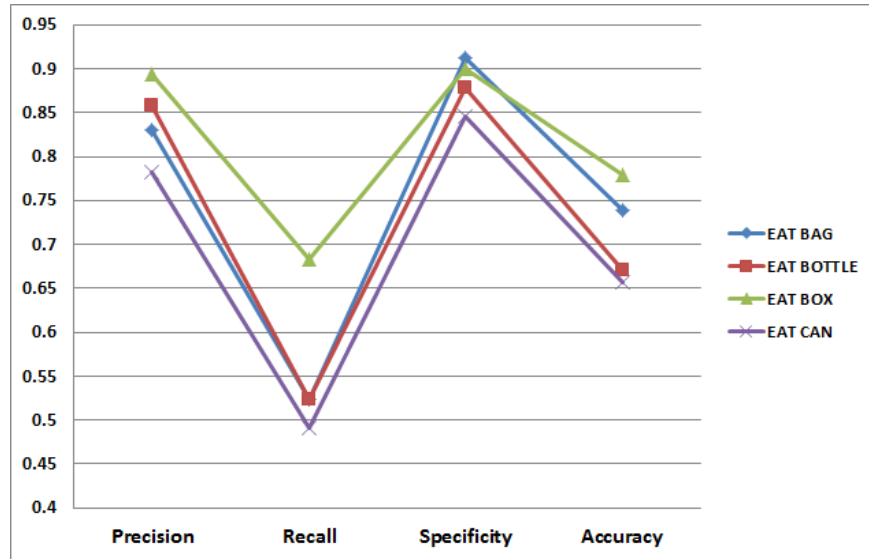


FIGURE 21: EAT precision, recall, specificity, and accuracy.

$$Probability = \frac{n(True\ positive)}{n(True\ positive) + n(False\ negative)} \quad (4)$$

It can be seen that the DOG algorithm gives better results than the EAT algorithm for all four types of products. Our analysis of the poorer performance of the EAT algorithm indicates that it may be attributed to the algorithm's dependency on the Canny edge detector. The edge detector found reliable edges only in the frames that were properly focused and free of distortions, which negatively affected the subsequent barcode localization. In that respect, the DOG algorithm is more robust, because it does not depend on any other edge detection.

It should also be noted that the DOG algorithm is in-place, because it does not allocate any additional memory structures for image processing. Based upon this observation we decided to choose the DOG algorithm as the localization algorithm for our barcode scanning experiments covered in Section 8.

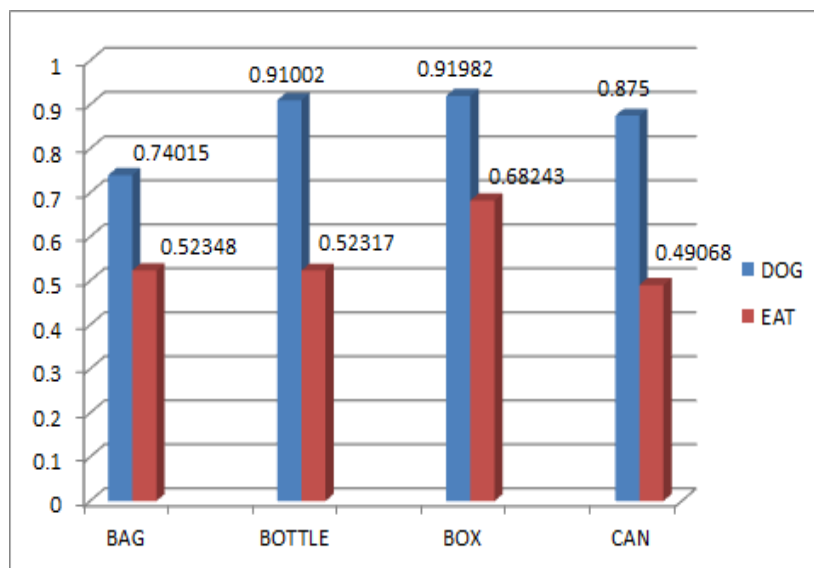


FIGURE 22: DOG and EAT localization probabilities.

7. 1D BARCODE SCANNING ALGORITHM

Our 1D algorithm for UPC and EAN barcode scanning works on frames with localized barcodes. In Figure 23, the output of the DOG localization algorithm is shown with a blue rectangle around the localized barcode. As was discussed above, the barcodes are localized in captured frames by computing dominant orientations of gradients (DOGs) of image segments and collected into larger connected components on the basis of their DOG similarity and geometrical proximity.



FIGURE 23: Barcode localization with DOG algorithm.

Figure 24 shows the control flow of the 1D barcode scanning algorithm. The algorithm takes as input an image captured from the smartphone camera's video stream. This image is processed by the DOG algorithm. If the barcode is not localized, another frame is grabbed from the video stream. If the DOG algorithm localizes a barcode, as shown in Figure 23, the coordinates of the detected region is passed to the line grower component. The line grower component selects the center of the localized region, which is always a rectangle, and starts growing scanlines.

For an example of how the line growing component works, consider Figure 25. The horizontal and vertical white lines intersect in the center of the localized region. The skew angle of the localized barcode, computed by the DOG algorithm, is 120 degrees. The line that passes the localized region's center at the skew angle detected by the DOG algorithm is referred to as the skew line. In Figure 35, the skew line is shown as a solid black line running from north-west to south-east.

After the center of the region and the skew angle are determined, the line growing module begins to grow scanlines orthogonal to the skew line. A scanline is grown on both sides of the skew line. In Figure 25, the upper half of the scanline is shown as a red arrow and the lower half of the scanline is shown as a blue arrow. Each half-line is extended until it reaches the portion of the image where the barcode lines are no longer detectable. A five pixel buffer region is added after the scanline's end to improve subsequent scanning.

The number of scanlines grown on both sides of the skew line is controlled through an input parameter. In the current implementation of the algorithm, the value of this parameter is set to 10. The scanlines are arrays of luminosity values for each pixel in their growth path. It should be noted that the scanlines are grown and scanned in place without any image or line rotation. For each grown scanline, the Line Widths (LW) for the barcode are then computed by finding two points that are on the intensity curve but lie on the opposite sides of the mean intensity. By modelling the curve between these points as a straight line the intersection points are obtained

between the intensity curve and the mean intensity. Interested readers are referred to our previous publication on 1D barcode scanning for technical details of this procedure [3, 4].

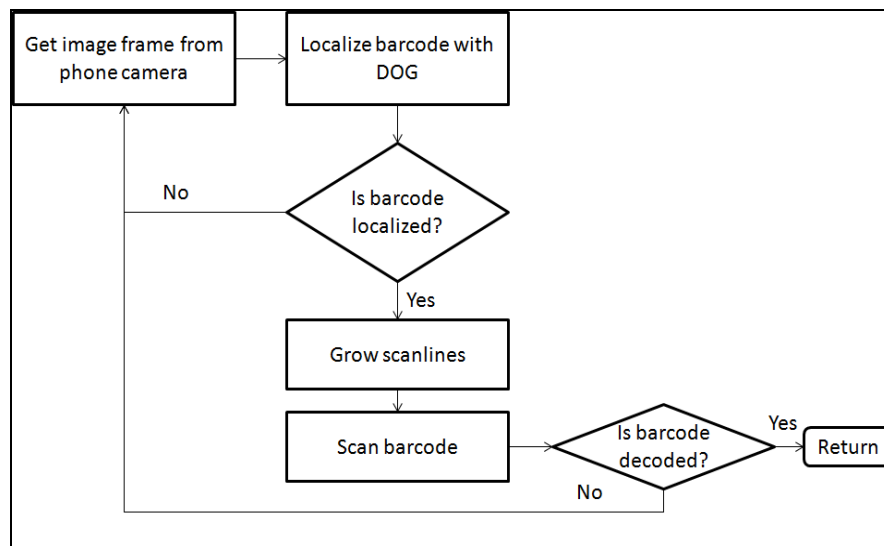


FIGURE 24: Barcode localization with DOG algorithm.

Figure 26 shows a sequence of images that gives a visual demonstration of how the algorithm works on a captured frame. The top image in Figure 26 is a frame captured from the smartphone camera’s video stream. The second image from the top in Figure 26 shows the result of the clustering stage of the DOG algorithm that clusters small subimages with similar dominant gradient orientations and close geometric proximity. The third image shows a localized barcode enclosed in a white rectangle. The bottom image in Figure 26 shows ten scanlines, one of which results in a successful barcode scan.

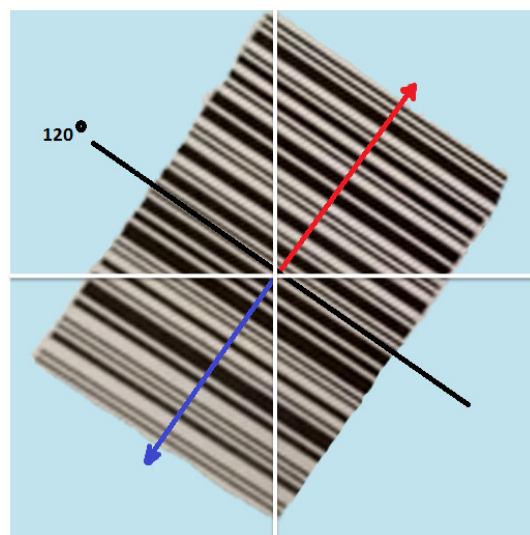


FIGURE 25: Growing a scanline orthogonal to skew angle.



FIGURE 26: 1D barcode scanning.

8. BARCODE SCANNING EXPERIMENTS

Our 1D algorithm for UPC and EAN barcode scanning works on frames with localized barcodes. In Figure 23, the output of the DOG localization algorithm is shown with a blue rectangle around the localized barcode. As was discussed above, the barcodes are localized in captured frames by computing dominant orientations of gradients (DOGs) of image segments and collected into larger connected components on the basis of their DOG similarity and geometrical proximity.

8.1 Experiments in a Supermarket

We conducted our first set of barcode scanning experiments in a local supermarket to assess the feasibility of our system. A user, who was not part of this research project, was given a Galaxy Nexus 4 smartphone with an AT&T 4G connection. Our front end application was installed on the smartphone. The user was asked to scan ten products of his choice in each of the four categories: box, can, bottle, and bag. The user was told that he can choose any products to scan so long as each product was in one of the above four categories. A research assistant accompanied the user and recorded the scan times for each product. Each scan time started

from the moment the user began scanning and ended when the response was received from the server.

Figure 27 denotes the average times in seconds for each category. The cans showed the longest scanning average due to glares and reflections. Bags showed the second longest scanning average due to some crumpled barcodes. As we discovered during these experiments, another cause for the slower scan times on individual products in each product category is the availability of Internet connectivity at various locations in the supermarket. During the experiments in the supermarket, we noticed that at some areas of the supermarket the Internet connection did not exist, which caused delays in barcode scanning. For several products, a 10- or 15-step change in location within a supermarket resulted in a successful barcode scan.

8.2 Impact of Blurriness

The second set of barcode scanning experiments was conducted to estimate the impact of blurriness on skewed barcode localization and scanning. These experiments were conducted on the same set of 506 videos of boxes, bags, bottles, and cans that we used for our barcode localization experiments described in Section 6. The average video duration is fifteen seconds. There are 130 box videos, 127 bag videos, 125 box videos, and 124 can videos. Images were extracted from the videos at the rate of 1 frame per second, which resulted in a total of 7,545 images, of which 1950 images were boxes, 1905 images were bags, 1875 images were bottles, and 1860 images were cans.

Each frame was automatically classified as blurred or sharp by the blur detection scheme using Haar wavelet transforms [13, 14] that we implemented in Python. Each frame was also manually classified as having a barcode or not and labeled with the type of grocery product: bag, bottle, box, can. There were a total of sixteen categories.

Figure 28 shows the results of the experiments. Images that contained barcodes for all four product categories had no false positives. In each product category, the sharp images had a significantly better true positive percentage than the blurred images. A comparison of the bar charts in Figure 28 reveals that the true positive percentage of the sharp images is more than double that of the blurry ones. Images without any barcode for all categories produced 100% accurate results with all true negatives, irrespective of the blurriness. In other words, the algorithm is highly specific in that it does not detect barcodes in images that do not contain them.

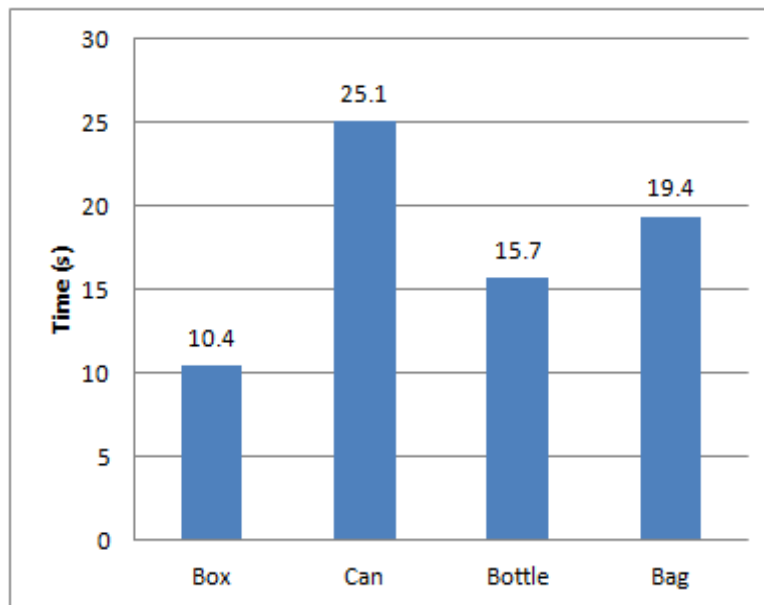


FIGURE 27: Average scan times.

Another observation on Figure 28 is that the algorithm showed its best performance on boxes. The algorithm's performance on bags, bottles, and cans was worse because of some crumpled, curved, or shiny surfaces. These surfaces caused many light reflections, which hindered performance of both barcode localization and barcode scanning. The percentages of the skewed barcode localization and scanning were better on boxes due to smoother surfaces. Quite expectedly, the sharpness of images made a positive difference in that the scanning algorithm performed much better on sharp images in each product category. Specifically, on sharp images, the algorithm performed best on boxes with a detection rate of 54.41%, followed by bags at 44%, cans at 42.55%, and bottles at 32.22%.

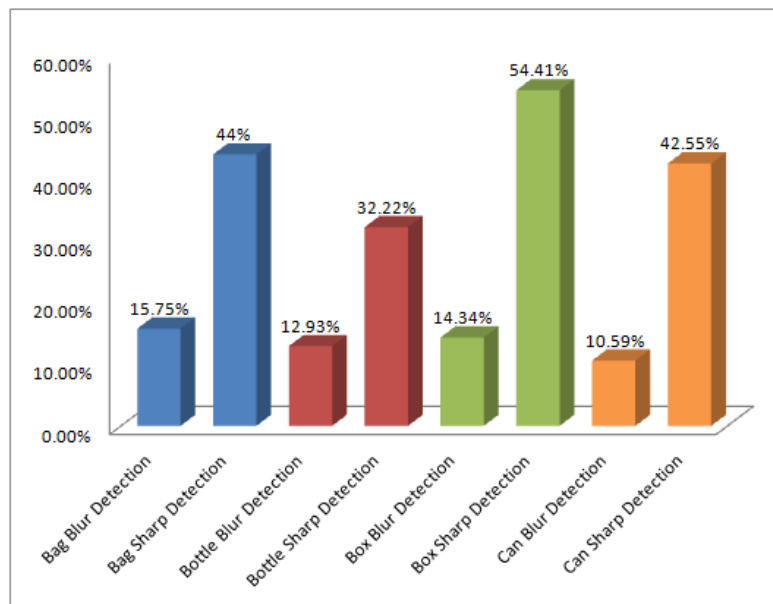


FIGURE 28: Blurred vs. non-blurred images.

8.3 Robustness and Speed of Linux Cluster

The third set of experiments was conducted to assess the robustness and speed of our five Linux cluster for image processing described in Section 5. After all classifications were completed (blurred vs. sharp; barcode vs. no barcode; type of grocery product), the classified frames were stored in the smartphone's sdcard. An Android service was implemented and installed on a Galaxy Nexus 4 smartphone. The service took one frame at a time and sent it to the node cluster via an http POST request over a local Wi-Fi network with a download speed of 72.31 Mbps and an upload speed of 29.64 Mbps.

The service recorded the start time before uploading each image and the finish time once a response was received from the cluster. The difference between the finish and start times was logged as a total request-response time. The service was run with one image from each of the sixteen categories described in Section 8.2 for 3000 times, and the average request-response time for each session.

Each image sent by the service was processed on the cluster as follows. The DOG localization algorithm was executed and, if a barcode was successfully localized, the barcode was scanned in place within the localized region with ten scanlines, as described in Section 7. The detection result was sent back to the smartphone and recorded on the smartphone's sdcard. Figure 29 gives the graph of the node cluster's request-response times. The lowest average request-response time was 712 milliseconds; the highest average was 1813 milliseconds.

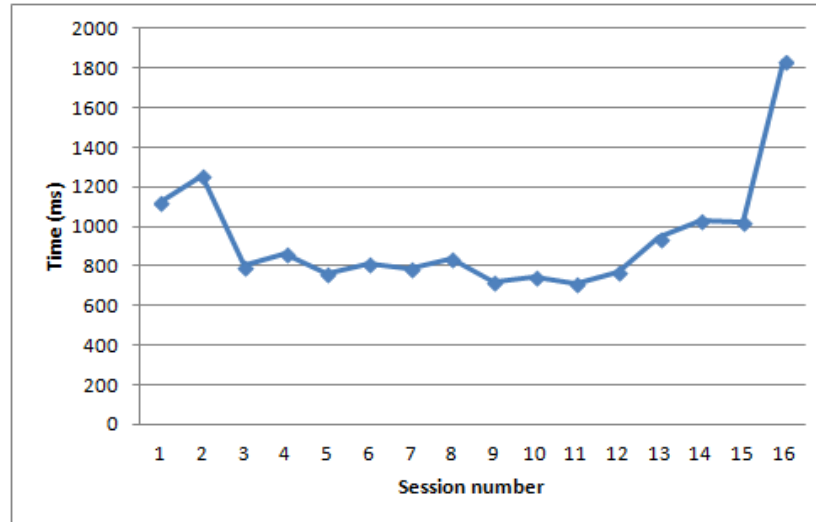


FIGURE 29: Average request-response times.

8.4 Pitch, Roll, and Yaw Constraints

In the fourth set of experiments, we assessed the ability of the system to scan barcodes skewed in the pitch, roll, and yaw planes. These experiments were conducted on ten products (two cans, two bottles, and six boxes) in our laboratory. Each product was scanned by a user with a Galaxy Nexus 4 Android 4.2 smartphone with the front end application installed on it. The user was asked to move the smartphone in the pitch, roll, and yaw planes while scanning each barcode. The pitch, roll, and yaw values were captured through the Android orientation sensor and logged for each successful scan.

It was discovered that the barcodes were successfully scanned at any orientation in the yaw plane. Figure 30 gives the maximum readings for each product scanning session. In the pitch plane, the maximum orientation at which a barcode was successfully scanned was 83.98 degrees while the minimum orientation was 68.19 degrees. The average pitch orientation at which barcodes were successfully scanned was 73.28 degrees. In the roll plane, the maximum orientation at which a barcode was successfully scanned was 59.73 degrees whereas the minimum orientation was 50.71 degrees. The average roll orientation at which barcodes were successfully scanned was 55.5 degrees.

9. DISCUSSION

Two algorithms were presented for vision-based localization of 1D UPC and EAN barcodes with relaxed roll, pitch, and yaw camera alignment constraints. The first algorithm (DOG) localizes barcodes in images by computing dominant orientations of gradients of image segments and grouping smaller segments with similar dominant gradient orientations into larger connected components. Connected components that pass specific morphological criteria are marked as potential barcodes and enclosed with minimal rectangular areas. The second algorithm (EAT) localizes barcodes by growing edge alignment trees (EATs) on binary images with detected edges. Trees of certain sizes mark regions as potential barcodes. Both algorithms were implemented in a distributed, cloud-based system. The system's front end is a smartphone application that runs on Android smartphones with Android 4.2 or higher. The system's back end was deployed on a five node Linux cluster where images are processed. Both algorithms were evaluated on a sample of 506 videos of bags, boxes, bottles, and cans in a supermarket. All videos were recorded with an Android 4.2 Google Galaxy Nexus smartphone. The videos have been made public for all interested research communities to replicate our findings or to use them in their own research [15]. The front end Android application is available for free download at Google Play under the title of NutriGlass [16].

The DOG algorithm was found to outperform the EAT algorithm on the image sample and was generally faster because it does not require edge detection. In other words, the algorithm is highly specific, where specificity is the percentage of true negative matches out of all possible negative matches. In all product categories, the true negative and false positive percentages were 0, which means that the algorithm is accurate not to recognize barcodes in images that do not contain them. The algorithm is designed to be conservative in that it rejects the frames on the slightest chance that it does not contain any barcode. While this increases false negatives, it keeps both true negatives and false positives close to zero. The DOG algorithm was subsequently coupled to our 1D UPC and EAN barcode scanner. The scanner receives a localized barcode region from the DOG algorithm along with the region's skew angles and uses a maximum of ten scanlines drawn at the skew angle to scan the barcode in place without any rotation of the scanlines or the localized barcode region.

After the DOG algorithm was coupled to our 1D barcode scanner, four sets of barcode scanning experiments were conducted with the system. The first set of barcode scanning experiments was conducted in a local supermarket to assess the feasibility of our system by a user with a Galaxy Nexus 4 smartphone with an AT&T 4G connection. The user was asked to scan ten products of his choice in each of the four categories: box, can, bottle, and bag. The cans showed the longest scanning average due to glares and reflections. Bags showed the second longest scanning average due to some crumpled barcodes. Another cause for the slower scan times on individual products was the availability of Internet connectivity at various locations in the supermarket. At some areas of the supermarket the Internet connection did not exist, which caused delays in barcode scanning.

The second set of barcode scanning experiments was conducted to estimate the impact of blurriness on skewed barcode localization and scanning. These experiments were conducted on the same set of 506 videos of boxes, bags, bottles, and cans. Images were extracted from the videos at the rate of 1 frame per second, which resulted in 1950 box images, 1905 bag images, 1875 bottle images, and 1860 can images. Images for all four product categories had no false positives. In each product category, the sharp images had a significantly better true positive percentage than the blurred images. The true positive percentage of the sharp images was more than double that of the blurry ones. Images without any barcode for all categories produced 100% accurate results with all true negatives, irrespective of the blurriness. The sharpness of images made a positive difference in that the scanning algorithm performed much better on sharp images in each product category.

The third set of experiments was conducted to assess the robustness and speed of our five Linux cluster for image processing. An Android service was implemented and installed on a Galaxy Nexus 4 smartphone. The service took one frame at a time and sent it to the node cluster via an http POST request over a local Wi-Fi network with a download speed of 72.31 Mbps and an upload speed of 29.64 Mbps. Sixteen sessions were conducted during each of which 3,000 images were sent to the cluster. The cluster did not experience any failures. The lowest average request-response time was 712 milliseconds; the highest average was 1813 milliseconds.

In the fourth set of experiments, the system's ability to scan barcodes skewed in the pitch, roll, and yaw planes. These experiments were conducted on ten products (two cans, two bottles, and six boxes) in our laboratory. The user was asked to move the smartphone in the pitch, roll, and yaw planes while scanning each barcode. The pitch, roll, and yaw values were captured through the Android orientation sensor and logged for each successful scan. The barcodes were successfully scanned at any orientation in the yaw plane. The average pitch orientation at which barcodes were successfully scanned was 73.28 degrees. The average roll orientation at which barcodes were successfully scanned was 55.5 degrees. Thus, the system can scan barcodes skewed at any orientation in the yaw plane, at 73.28 degrees in the pitch plane, and at 55.5 degrees in the roll plane.

One limitation of the current front end implementation is that it does not compute the blurriness of the captured frame before sending it to the back end node cluster where barcode localization and scanning are performed. As the experiments described in Section 8.2 indicate, the scanning results are substantially higher on sharp images than on blurred images. This limitation points to a potential improvement that we plan to implement in the future. When a frame is captured, its blurriness coefficient can be computed on the smartphone and, if it is high, the frame should not even be sent to the cluster. This improvement will reduce the load on the cluster and increase its responsiveness.

Another approach to handling blurred inputs is to improve camera focus and stability, both of which are outside the scope of our research agenda, because it is, technically speaking, a hardware problem. It is likely to work better in later models of smartphones. The current implementation on the Android 4.2 platform attempts to force the camera focus at the image center through the existing API. Over time, as device cameras improve and more devices run newer versions of Android, this limitation will likely have a smaller impact on the system's performance.

10. REFERENCES

- [1] E. Tekin and J. Coughlan. "An algorithm enabling blind users to find and read barcodes," in Proc. of Workshop on Applications of Computer Vision, pp. 1-8, Snowbird, UT, December, 2009. IEEE Computer Society.
- [2] E. Tekin and J. Coughlan. "A mobile phone application enabling visually impaired users to find and read product barcodes," in Proc. of the 12th international conference on Computers helping people with special needs, pp. 290-295, Vienna, Austria, 2010. Klaus Miesenberger, Joachim Klaus, Wolfgang Zagler, and Arthur Karshmer (Eds.), Springer-Verlag, Berlin, Heidelberg.
- [3] V. Kulyukin, A. Kutiyawala, and T. Zaman. "Eyes-free barcode detection on smartphones with Niblack's binarization and support vector machines," in Proc. of the 16-th International Conference on Image Processing, Computer Vision, and Pattern Recognition, vol. 1, pp. 284-290, Las Vegas, NV, 2012. CSREA Press.
- [4] V. Kulyukin and T. Zaman. "Vision-based localization of skewed upc barcodes on smartphones," in Proc. of the International Conference on Image Processing, Computer Vision, & Pattern Recognition, pp. 344-350, Las Vegas, NV, 2013. CSREA Press.
- [5] E. Tekin, D. Vásquez, and J. Coughlan. "S-K smartphone barcode reader for the blind." Journal on Technology and Persons with Disabilities, to appear.
- [6] S. Wachenfeld, S. Terlunen, J. Xiaoyi. "Robust recognition of 1-D barcodes using camera phones," in Proc. of the 19th International Conference on Pattern Recognition, pp. 1-4, Tampa, FL, 2008. IEEE Computer Society.
- [7] R. Adelman, M. Langheinrich, and C. Floerkemeier. "A Toolkit for barcode recognition and Resolving on Camera Phones - Jump Starting the Internet of Things," in Proc. of workshop on mobile and embedded information systems (MEIS'06) at informatik, Dresden, Germany, 2006.
- [8] D.T. Lin, M.C. Lin, and K. Y. Huang. "Real-time automatic recognition of omnidirectional multiple barcodes and DSP implementation." Appl. Mach. Vision, 22, vol. 2, pp. 409-419, 2011.

- [9] O. Gallo and R. Manduchi. "Reading 1D barcodes with mobile phones using deformable templates." IEEE Transactions on Pattern Analysis and Machine Intelligence, 33, vol. 9, pp. 1834-1843, 2011.
- [10] E. Peng, P. Peursum, and L. Li. "Product barcode and expiry date detection for the visually impaired using a smartphone," in Proc. of international conference on digital image computing techniques and applications, pp. 3-5, Perth Western Australia, Australia, 2012. Curtin University.
- [11] J. A. Canny. "A computational approach to edge detection." IEEE Transactions on Pattern Analysis and Machine Intelligence, 8, vol. 6, pp. 679-698, 1986.
- [12] R. Maini and H. Aggarwal. "Study and comparison of various image edge detection techniques." International Journal of Image Processing. 3, vol. 1, pp. 1-11, 2009.
- [13] H. Tong, M. Li, H. Zhang, and C. Zhang. "Blur detection for digital images using wavelet transform," in Proc. of the IEEE international conference on multimedia and expo, pp. 27-30, Taipei, 2004. IEEE Computer Society.
- [14] Y. Nievergelt. Wavelets Made Easy. Birkhäuser, Boston, 1999.
- [15] T. Zaman and V. Kulyukin. Videos of common bags, bottles, boxes, and cans. Available from <https://www.dropbox.com/sh/q6u70wcg1luxwdh/LPtUBdwdY1>. [May 9, 2014].
- [16] NutriGlass: An android application for scanning skewed barcodes. Available from <https://play.google.com/store/apps/details?id=org.vkedco.mobappdev.nutriglass>. [May 8, 2014].

11. APPENDIX A: DOMINANT ORIENTATION OF GRADIENTS

1. **FUNCTION ComputeDOGs(Image, MaskSize)**
2. ThetaThresh = 360; MagnThresh = 20.0;
3. FreqThresh = 0.02; ListOfNeighborhoods = [];
4. GGOT = new HashTable();
5. **Foreach** mask of MaskSize in Image **Do**
6. SubImage = subimage currently covered by mask;
7. RGOT = **ComputeRGOT**(SubImage, ThetaThresh, MagnThresh);
8. GGOT[coordinates of masks' top left corner] = RGOT;
9. **If** RGOT \neq NULL **Then**
10. RGOT.row = mask.row;
11. RGOT.column = mask.column;
12. **If** (RGOT(freq)*1.0/(SubImage.cols * subImage.rows) \geq FreqThresh)
13. Neighbourhood = FindNeighbourhoodForRGOT(RGOT, ListOfNeighborhoods);
14. **If** (Neighbourhood \neq NULL) **Then** Neighbourhood.add(RGOT);
15. **Else**
16. NewNeighbourhood=Neighbourhood(RGOT.dtheta, ListOfNeighborhoods.size+1);
17. NewNeighbourhood.add(RGOT)
18. ListOfNeighborhoods.add(newNeighbourhood);
19. **EndIf**
20. **EndIf**
21. **EndIf**
22. **EndForeach**

```

1. FUNCTION ComputeRGOT(Image, THETA_THRESH, MAGN_THRESH)
2. Height = Image.height; Width = Image.width;
3. RGOT = new HashTable();
4. For row = 1 to Height Do
5.   For column = 1 to Width Do
6.     DX = Image(row, column+1)[0]-Image(row, column-1)[0];
7.     DY = Image(row -1, column)[0]- Image(row +1, column)[0];
8.     GradientMagn = sqrt(DX^2+DY^2);
9.     GradTheta = arctan(DY/DX)*180/PI;
10.    If (|GradTheta|≤THETA_THRESH) AND (|GradMagn|≥MAGN_THRESH) Then
11.      If (RGOT contains GradTheta) Then
12.        RGOT[GradTheta] += 1;
13.      Else
14.        RGOT[GradTheta] = 1;
15.      EndIf
16.    EndIf
17.  EndFor
18. EndFor
19. Return RGOT;

```

```

1. FUNCTION FindNeighbourhoodForRGOT(RGOT, ListOfNeighborhoods)
2. ThetaThresh = 5.0;
3. Foreach neighborhood in LisOfNeighborhoods Do
4.   If (|neighborhood.dtheta - RGOT.theta| < ThetaThresh) Then
5.     If (HasNeighborMask(neighborhood, RGOT)) Then
6.       Return neighborhood;
7.     EndIf
8.   EndIf
9. EndForeach

```

```

1. FUNCTION HasNeighborMask(neighborhood, RGOT)
2. Foreach RGOTMember in Neighborhood.members Do
3.   If ( RGOT.row = RGOTMember.row ) Then
4.     If ( |RGOT.column - RGOTMember.column| = maskSize ) Then
5.       Return True;
6.     EndIf
7.   EndIf
8.   If ( RGOT.column = RGOTMember.column ) Then
9.     If ( |RGOT.row - RGOTMember.row| = maskSize ) Then
10.    Return True;
11.    EndIf
12.   EndIf
13.   If ( |RGOT.column - RGOTMember.column| = maskSize ) Then
14.     If ( |RGOT.row - RGOTMember.row| = maskSize ) Then
15.       Return True;
16.     EndIf
17.   EndIf
18. EndForeach

```

12. APPENDIX B: EDGE ALIGNMENT TREE

1. FUNCTION ComputeEATs(Image, MaskSize)

```

2. AngleList = []
3. IsBarcodeRegion = false;
4. Foreach subimage of MaskSize in Image Do
5.   AngleList = DetectSkewAngle(subimage);
6.   If ( IsBarcodeRegion(AngleList) ) Then
7.     Barcode detected;
8.   Endif
9. EndForeach

```

1. FUNCTION DetectSkewAngle(SubImage)

```

2. ListOfAngles = [];
3. ListOfRoots = [];
4. CannyEdgeDetector(400, 500);
5. //Initialize all 1's in first row as roots
6. J = 0;
7. For I=0 to SubImage Width Do
8.   If ( SubImage(J, I)[0]==255 ) Do
9.     Node = TreeNode( I , J );
10.    ListOfRoots = ListOfRoots U { Node };
11.   Endif
12. EndFor
13. FormTreeDetectAngle(ListOfRoots, SubImage, J);
14. If ( ListOfRoots.size ≠ 0 ) Then
15.   Foreach root in ListOfRoots Do
16.     ListOfAngles = ListOfAngles U FindAnglesForTree(root);
17.   EndForEach
18. Endif

```

1. FUNCTION IsBarcodeRegion(ListOfAngles)

```

2. STD = StandardDeviation(ListOfAngles);
3. If STD < 5 Then
4.   Return True;
5. Else
6.   Return False;
7. Endif

```

1. FUNCTION FormTreeDetectAngle(ListOfRoots, Image, rowLevel)

```

2. Theta = 0;
3. ParentExists = False;
4. K = rowLevel;
5. While K=rowLevel to Image.rows Do
6.   For I=0 to Image.columns Do
7.     If ( Image[K,I] = 255 ) Then
8.       Node = Treenode(K,I);
9.       NextRowNodeList = NextRowNodeList U { node };
10.    Endif
11.   EndFor
12. EndWhile
13. //Check if a next row node can form a child of a root node, otherwise form a new root

```

```
15. Foreach Node in NextRowNodeList Do
16.   Foreach ParentNode in ListOfRoots Do
17.     Theta = arctan((Node.row – ParentNode.row)/(ParentNode.column – Node.column));
18.     If ( Theta >=45 AND Theta <= 135 ) Then
19.       ParentNode = ParentNode U { Node };
20.       ParentExists = True;
21.       break;
22.     EndIf
23.   EndForeach
24.   If ( ParentExist = False ) Then
25.     ListOfRoots = ListOfRoots U { Node };
26.   EndIf
27. EndForeach
28. FormTreeDetectAngle(NextRowNodeList, Image, K);
```

```
1. FUNCTION FindAnglesForTtree(RootNode)
2.  Theta = 0;
3.  RunningAvgTheta = 0;
4.  ChildList = RootNode.children;
5.  While ChildList.size ≠ 0 Do
6.    TempChild = new Node();
7.    Foreach Child in ChildList Do
8.      If Child.children.size ≠ 0 Do
9.        Theta = arctan((Child.row-RootNode.row)/(RootNode.column – Child.column));
10.       TempChild = Child;
11.      EndIf
12.    EndForeach
13.    ChildList = TempChild.children;
14.    RunningAvgTheta = (Theta+ RunningAvgTheta)/2;
15.  EndWhile
16. Return RunningAvgTheta;
```

Lip Reading by Using 3-D Discrete Wavelet Transform with Dmey Wavelet

Sunil S. Morade

*PhD Student,
Electronics Engineering Dept,
SVNIT, Surat.*

ssm.eltx@gmail.com

Suprava Patnaik

*Professor, Department of E and TC Engineering,
Xavier Institute of Engineering, Mumbai, India.*

suprava_patnaik@yahoo.com

Abstract

Lip movement is an useful way to communicate with machines and it is extremely helpful in noisy environments. However, the recognition of lip motion is a difficult task since the region of interest (ROI) is nonlinear and noisy. In the proposed lip reading method we have used two stage feature extraction mechanism which is précised, discriminative and computation efficient. The first stage is to convert video frame data into 3 dimension space and the second stage-trims down the raw information space by using 3 Dimension Discrete Wavelet Transform (DWT). These features are smaller in size to give rise a novel lip reading system. In addition to the novel feature extraction technique, we have also compared the performance of Back Propagation Neural Network (BPNN) and Support Vector Machine(SVM) classifier. CUAVE database and Tulips database are used for experimentation. Experimental results show that 3-D DWT feature mining is better than 2-D DWT. 3-D DWT with Dmey wavelet results are better than 3-D DWT Db4. Results of experimentation show that 3-D DWT-Dmey along with BNNN classifier outperforms SVM.

Keywords: 2-D DWT, 3-D DWT, Dmey Wavelet, BPNN, SVM, Lip Reading.

1. INTRODUCTION

Lip reading is a technique by which seeing the lip movement one can recognize the speech and is helpful for hearing impaired person. Potential uses of lip reading are communication during disaster like earthquake, noisy factory areas and IVR system. Two fundamental steps involed in lip reading system are: 1) feature extraction and 2) feature classification. Lip features are extracted either by a geometrical model or by an image transform model. Lip geometrical model depends on extraction of lip contour. Inaccuracy in extraction of lip contour affects the different geometrical parameters such as width, height and area. Because of the associated risk of inaccuracy and complexity geometrical model is not suitable for real time application. Also in this model cavity information is not taken into account. In this paper the focus is on image transform model which is also known as appearance model. On the other side image transform model extracts feature by using gray scale intensity transformation and is weak in preserving minute geometrical variations. State of the art literatures deal with 2D-DCT or 2D-DWT as the foremost step of appearance model. Important constraints of the image transform techniques is the feature vector size.

State of art literatures on appearance model are many, out of which few noteworthy literatures are cited here for basic understanding of challenges in lip reading paradigm. E. Petajan [1] experimented on lip-reading to enhance speech recognition by using visual information. The speech reading system proposed by Bregler et al. [3] used Eigen lips as feature vectors. Potamianos et al. [4] compared three linear image transforms namely PCA, DWT and DCT transform techniques. R. Seymour et al. [5] used comparison of image transform features in

visual speech recognition of clean and corrupted videos. They evaluated Fast Discrete Curvelet Transform (FDCT), DCT, PCA and Linear Discriminant Analysis (LDA) methods. Wang et al. [6] used different region of interest (ROI) as a visual features in lip reading process and discussed about impact of different ROI processing methods recognition accuracy. N. Puviarasan et al. [7] used DCT and DWT methods for visual feature extraction. They generated data base of hearing impaired persons and observed that DWT with HMM gives better result. A. Shaikh et al. [8] used optical flow information as a feature vector for lip reading. The vocabulary used in their experiment was visemes. Visemes are the basic visual movements associated with phonemes. They tested the result of lip reading using SVM classifier with Gaussian Radial Basis kernel function. The classification performance parameters such as specificity, sensitivity and accuracy are used to test classifiers. Meyor et al. [9] used DCT transform technique for pixel information of continuous digit recognition and proposed different fusion techniques for audio and video feature data. They found that Word Error Rate (WER) is more for continuous digit recognition. L. Rothkrantz et al. [10] presented a lip geometry estimation (LGE) method and it was compared with geometry and image intensity based techniques such as geometrical model of lip, specific points on mouth contour and raw image. Authors found LGE method competitive with some strong points on its favor. However to our knowledge in none of the publications attempt has been made towards discriminative feature mining from volume information of video, by using 3D transforms.

Selecting predefined number of coefficients from sequence of frames gives feature vectors of defined size for all classes however can't guarantee for efficient feature. Efficient feature is a set of coefficients with interclass variation as maximum as and with the class variation as minimum as possible. While a digit is uttered by M people and by each one for N times, the feature vector is required to be more or less similar however for different digits the expectation is to deal with the feature vectors as different as conceivable. 2D transforms would work well for frames with uniform activity and variations. In lip reading framework depending on dynamism or speed of utterance many times the trailing frames seems to have silence and hence non-informative. This can be ruled out by use of 3D transforms.

2. PROPOSED LIP READING FRAMEWORK

A typical lip reading system consists of four major stages: video frame normalization, Face and lip detection, feature extraction, and the finally the classifier. Fig. 1 shows the major steps used in the proposed lip reading process. One major challenge in a complete English language lip reading system is the need to train whole of the English language words in the dictionary or to train (at least) the distinct ones. However same can be effective if it is trained on a specific domain of words, e.g. digits, names etc. Present experimentation is limited to digit utterance.

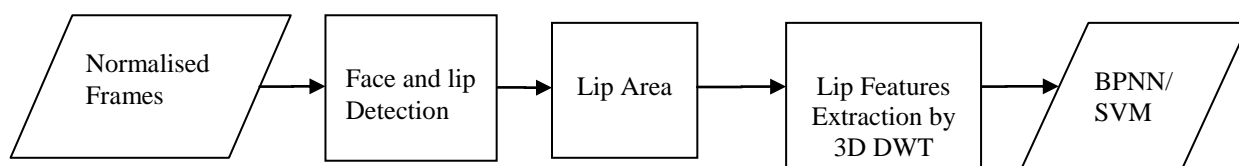


FIGURE 1: Lip reading process.

2.1 Video Segmentation and Lip Contour Localization

There are large inter and intra subject variations in speed of utterance and this results in difference in the number of frames for each utterance. We have used audio analysis, using Pratt software to segment the time duration and the associated video frames of each digit which is uttered. On an average 16 frames are sufficient for utterance of any digit between 0-9. Out of 16 frames we have selected 10 significant frames. Mean square difference σ_i , which is defined in (1), is computed for all the frames. These are arranged in decreasing order and initial 10-frames are selected for feature extraction. This step resembles the dynamic time warping operation of

speech analysis. Outcome is an optimal alignment of utterances. The number of frames for each utterance is made same such that the feature vectors size remains same for each utterance.

$$\sigma_i = \left[\frac{1}{M*N} \sum_0^M \sum_0^N \{I_i(x, y) - I_{i+2}(x, y)\} \right]^2 \quad (1)$$

where, $I_i(x, y)$ stands for the (x, y) spatial location of i^{th} video frame and each frame is of size $M*N$. Lip detection or segmentation is very difficult problem due to the low gray scale variation around the mouth. Chromatic or color pixel based features, especially red domination of lips, have been adopted by most researchers to segment lips from the primarily skin background. Viola and Jones, invented this algorithm in 2004 based on Adaboost classifier to rapidly detect any object including human face. They presented a face detector which uses a holistic approach and is much faster than any contemporaries. Adaboost classifier cascades the Haar like features and not pixels features, hence fast and work accurately for human face detection [12]. Using Adaboost algorithm for face and mouth detection result is shown in Fig. 2 (a and b).

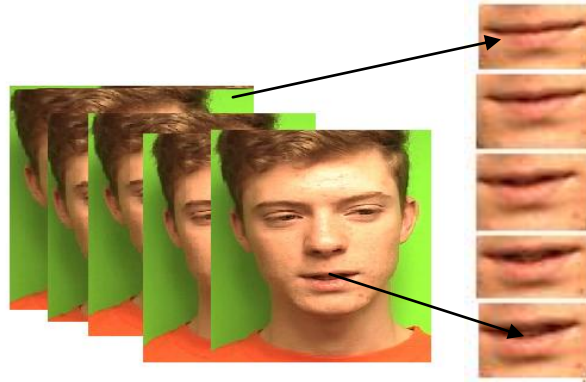


FIGURE 2: (a) Detection of face and lip area for CUAVE database s02m (b) Lip portion.

2.2 3-D Discrete Wavelet Transform (3-D DWT)

1-D transform is used in speech and music, while 2-D is used in image processing. According to Lee, et al.[12], the 3-D DCT is more efficient than the 2-D DCT for image compression application. The DWT has many advantages over the DCT. First, DCT difference coding is computationally expensive. Second, wavelets do not cause blocking artifacts. The 3-D DWT is separable, which means that the 3-D transform is able by applying 1-D DWT in each dimension. DWT considers correlation of images, which translates to better feature vector. Wang and Huang [13] showed the 3-D DWT outperforming the 2-D DWT by 40-90% in image compression. Likewise, one would expect the 3-D DWT outperform the 2-D DWT for lip reading application. Initially for lip reading application we used two dimensional data i.e. Z-axis along the frame was not considered. This axis gives the variation in lip movement so it is important to use 3-D DWT. The results get improved by using 3-D DWT as compared to 2-D DWT.

While 2-D DWT is used for computing the feature vector. Goal is to select only those coefficients which play the dominant role in the representation of lip motion. In standard 2-D wavelet decomposition based approach, each level of filtering splits the input image into four parts via pair of low-pass and high-pass filters with respect to column vectors and row vectors of the image array. Then the low-spatial frequency sub-image is selected for further decomposition. After few levels of decomposition the lowest spatial-frequency approximation sub-image, is extracted as the feature vector. The 3-D DWT is like a 1-D DWT in three directions. Lip reading is a video processing application. To use the wavelet transform for volume and video processing, a 3-D version of filter banks are implemented. In 3-D DWT, the 1D analysis filter bank is applied in turn to each of the three dimensions [2]. This is shown in Fig. 3.

DWT computations, the input are multiplied by the shifts (translation in time) and scales (dilations or contractions) of the wavelet. Below are variables commonly used in wavelet architecture. The outputs of low-pass and high-pass filters are given by equation (2) and (3) respectively.

$$W_l(n, j) = \sum_{m=0}^{2n} w(m, j - 1) * h(2n - m) \quad (2)$$

$$W_h(n, j) = \sum_{m=0}^{2n} w(m, j - 1) * g(2n - m) \quad (3)$$

where $W(n, j)$ is wavelet output. $h(n)$ and $g(n)$ are the filter impulse response of low pass and high pass filter, j is the current level, n is the current input index and $w(n, j - 1)$ is the input signal. V. Long and L. Gang [14] proposed a new method for choosing the best wavelet base for speech signals. They have compared Haar, Daubechies, Dmey, Biorthogonal, Coiflets, and Symlet and concluded that Dmey wavelets outperforms for speech signal synthesis. The results from [14] motivated us to select Dmey wavelet, as in lip reading application also the speech information is extracted from visual information.

Fig. 4(a) shows the number of frames of lip in each direction. Fig. 4 (b) shows that first the process transforms the data in the x-direction. Next, the low and high pass outputs both feed to other filter pairs, which transform the data in the y-direction. These four output streams go to four more filter pairs, performing the final transform in the z-direction. The process results in 8 data streams. In our experiment approximation component (LLL) is important so only low pass filter outputs are shown in Fig. 4(b).

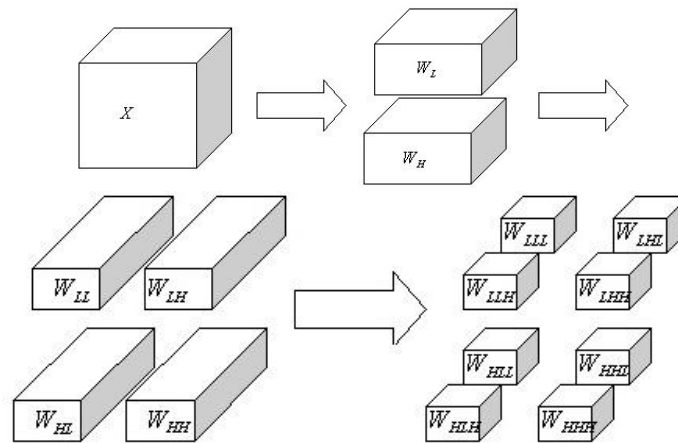
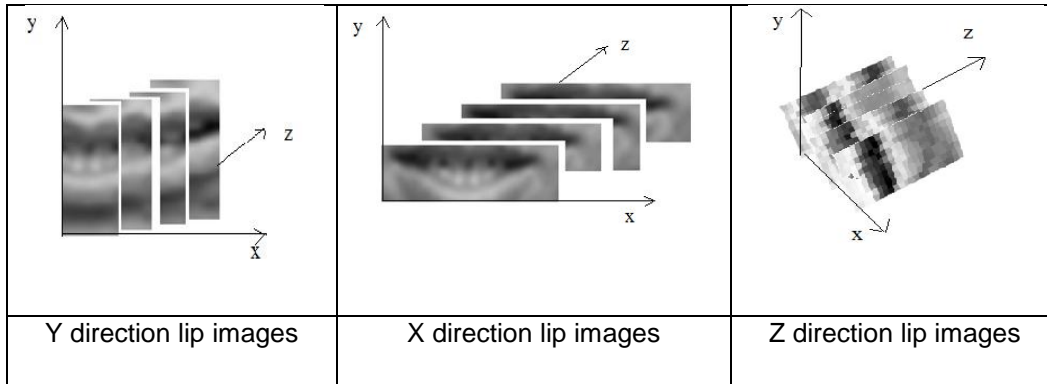
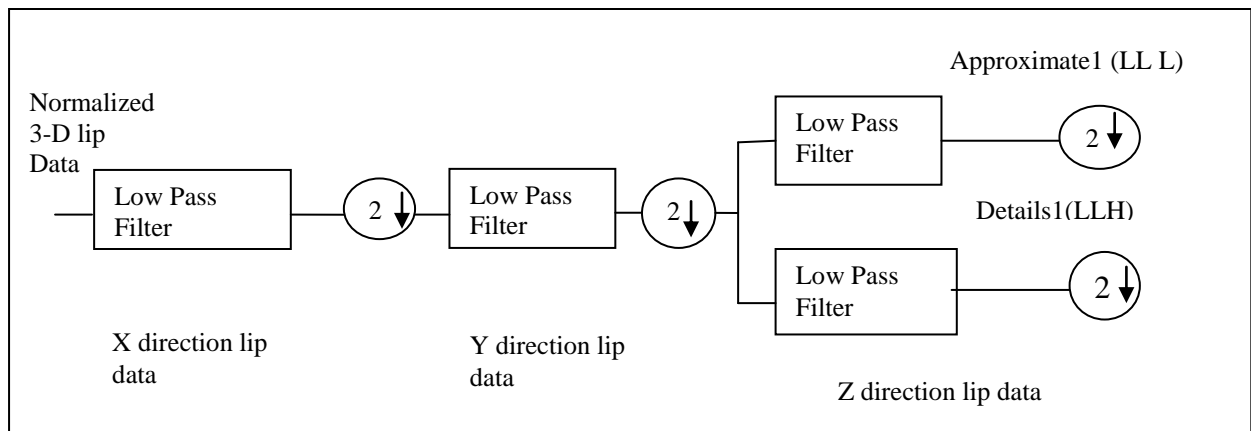


FIGURE 3: The resolution of a 3-D signal is reduced in each dimension.



4(a)



4(b)

FIGURE 4(a): Lip images for a digit in three directions (X, Y, Z) **(b)** Single level decomposition for 3-D DWT only for LLL.

3. CLASSIFICATION

The last processing stage of lip reading is feature classification. Researchers have used Naïve Bayes, KNN, ANN, SVM, and HMM as classifier for lip reading. Out of this classifier HMM is mostly used classifier but it is complex and required large training data. Also after doing experimentation it is observed that the performance of Naïve Bayes and KNN is poor as compared to SVM and ANN for lip reading application. A.K. Jain et al. [15] in their review paper of pattern recognition compared different classifier and found that ANN and SVM are most suitable for lip reading application. So in this paper experimentation results for SVM and ANN are compared.

3.1 BPNN

Artificial Neural Network (ANN) models are useful to solve pattern recognition problems due to their low dependency on domain-specific knowledge and due to the availability of efficient learning algorithms. It is important to note that neural networks themselves can lead to in any different classifiers depending on how they are trained. Layers in multilayer perceptron network architecture allow to identify nonlinear decision boundaries.

ANN is mathematical model consists of number of highly interconnected processing elements organized into layers, geometry and functionality of which have been resembled to that human brain. Bregler and Y. Konig [16] in their experiment used a TDNN for speech and visual speech data. The ANN may be regarded as possessing learning capabilities in as much as it has natural. The first layer consists of input element in accordance with feature vectors. The most frequently used training algorithm in classification problem is the back propagation algorithm. The neural network has been trained to adjust the connection weights and biases in order to produce desired mapping.

We have used a BPNN classifier. The nodes of this network are sigmoid. Number of hidden layers tried varies in the range 10 to 30. Best performance of the neural network for feature vector of DWT is obtained with 20 hidden layers. Learning rate and moment coefficients are set at 0.3 and 0.001 respectively. Input vector X is a feature vector of size n which is the length of feature vector. The ten neurons in the output layer were used to represent the digit.

3.2 Support Vector Machines (SVM)

One of the most interesting and recent developments in classifier design paradigm is the introduction of support vector machine classifier by Vapnik [17]. It is a two class classifier. The optimization criteria is the width of the margin between the classes i.e. empty area around the decision boundary defined by the distance to the nearest training patterns. These patterns are called support vectors and finally used to define the classification function.

The important advantage of the support vector classifiers is that it offers possibility to train generalizable, nonlinear classifiers in high dimensional spaces using small training set. The support vectors replace the prototypes with the main difference being that they characterize the classes by decision boundary. Moreover this decision boundary is not defined by minimum distance function but by a more nonlinear combination of these distances [15].

Data separation is completely possible by using nonlinear separation but it is not using linear separation. For nonlinear separation between classes mapping function Φ is used. Using Φ lower dimension input space is transferred to higher dimension. Mapping is projecting the original set of variables x in higher dimensional feature space Φ . Kernel functions are expressed in terms of Φ by (9). Applying kernels we do not even have to know what the actual mapping. A kernel is a function k such that the learning algorithm is a hyperplane in a feature space. Thus by choosing kernel $k(x, x_i)$, we can construct an SVM that operates in an infinite dimension space.

$$x \in R^d \Rightarrow \Phi(x)$$

$$\Phi(x) \equiv (\Phi_1(x), \Phi_2(x), \dots, \dots, \Phi_n(x)) \in R^n$$

$$k(x_i, l_j) = \Phi^T(x_i)\Phi(l_j) \quad (4)$$

Kechman in his literature discussed the mapping and different kernel function [18]. SVM maximizes the distance of separating plane from the closest training data point. Linear kernel is defined by (5a) while polynomial kernel is defined by (5b) where d is the degree of polynomial.

$$k(x, l_i) = (x^T l_i) \quad (5a)$$

$$k(x, l_i) = (x^T l_i + 1)^d \quad (5b)$$

For classification, the decision function for a two class problem derived by a support vector can be written by (6) using a kernel function $k(x, l_i)$ of a new pattern x and a training pattern l_i .

$$f(x) = \sum_{i=1}^N y_i \alpha_i k(x, l_i) + b \quad (6)$$

Where k kernel function, b scalar bias, α Lagrange's multiplier and $y_i = \pm 1$ is the label of object x_i and l_i support vector obtained from training data. In equation (7) we need to find suitable Lagrange multipliers α to get the following function reach its maximum value.

$$L_d(\alpha) = \sum_1^l \alpha_i - \frac{1}{2} \sum_1^l y_i \alpha_i \alpha_j \Phi_i^T \Phi_j \quad (7)$$

Where $k(x_i, l_j) = \alpha_j \Phi_i^T \Phi_j$

Sequential Minimal Optimization (SMO) is a SVM learning algorithm which is conceptually simple, easy to implement, and have faster and better scaling properties than a standard SVM algorithm. John Platt proposed this algorithm for efficiently solving the optimization problem which arises during the training of SVM. Though SVMs are popular, two major weaknesses made their use limited. First the training of SVM is slow, especially for large problems. Second, SVM training algorithms are complex, subtle and sometimes difficult to implement [19]. E. Osuna et al. has suggested two modifications in Platt's SMO algorithm so that the SMO algorithm speeds up to train SVM in many situations [20].

Large SVM training data can fit inside of the memory of an ordinary personal computer. Because no matrix algorithms are used in SMO, it is less susceptible to numerical precision problems. For the real-world test sets, SMO can be a approximately thousand times faster for linear SVMs and ten times faster for non-linear SVMs [20]. Because of its ease of use and better scaling with training set size, SMO is the standard SVM training algorithm. In this experiment SMO is used for training SVM with 2nd degree polynomial kernel function.

4. PROPOSED LIP READING METHODOLOGY

A Flow chart of the steps involved in our simulation technique is shown in Fig. 5. Three major execution steps of the algorithm are: 1) pre-processing, 2) feature extraction and dimension reduction, and 3) feature classification. The two step salient feature extraction step is the core contribution of our work. After applying 3-D DWT, low frequency components (LLL) of the image are taken as a feature vector for classification. 3-D DWT or 2-D DWT attempts to transform image pixels of significant lip frames into a new space which separates redundant information and provides better discrimination. Then the final feature vectors of all the train lip frames are stored in the training database along with class level.

5. CORPUS AND RESULT

5.1 CUAVE Database

CUAVE [21] (Clemson University Audio Visual Experiments) was recorded by E.K. Pattererson of Department of Electrical and Computer Engineering, Clemson University, US. The database was recorded in an isolated sound booth at a resolution of 720 x 480 with the NTSC standard of 29.97 fps using 1 Megapixel-CCD camera. This database is a speaker-independent database consisting of connected and continuous digits spoken in different situations. The database consists of two major sections: one of speaker pairs and the other one of individuals.

It contains mixture of speaker with white and black skin. Database digits are continuous and with pause. Data is recorded with sequential and random manner. Some videos are taken from side view. Total 36 videos are in data base, out of which, 19 are for male speaker and 17 are for female speaker. Disruptive mistakes are removed, but occasional vocalized pauses and mistakes in speech are kept for realistic test purposes. The data was then compressed into individual MPEG-2 files for each individual speaker and group of two speakers. It has been shown that this does not significantly affect the collection of visual features for lip reading. The object of the video captured for the presence of two speakers speaking simultaneously does not affect significant features for lip reading.

Each individual speaker was asked to move side-to-side, back-and-forth, or tilt the head while speaking 30 isolated digits. In addition to this isolated section, there is also a connected-digit

section with movement as well. So far, much research has been limited to low resolution, pre-segmented video of only the lip region.

5.2 TULIPS Database

Tulips1.0 is a small Audiovisual database of 12 subjects saying the first 4 digits in English. Subjects are undergraduate students from the Cognitive Science Program at UCSD. The database was compiled at R. Movellan's laboratory at the Department of Cognitive Science, UCSD.

Figure 6 shows the 6 frames for utterance of digit 0 using TULIPS database. The "Raw Data" directory contains two directories: Tulips1.A and Tulips1.V. The "Preprocessed Data" directory contains representations used by different researchers on this database. Tulips1.V contains the video files in 100 x 75 pixel 8 bit gray level, .pgm format. Each frame corresponds to 1/30 of a second. R. Movellan presents work on speaker independent visual speech recognition system and used simple HMM as a classifier used Tulips database of 1 to 4 digits for testing result [22].

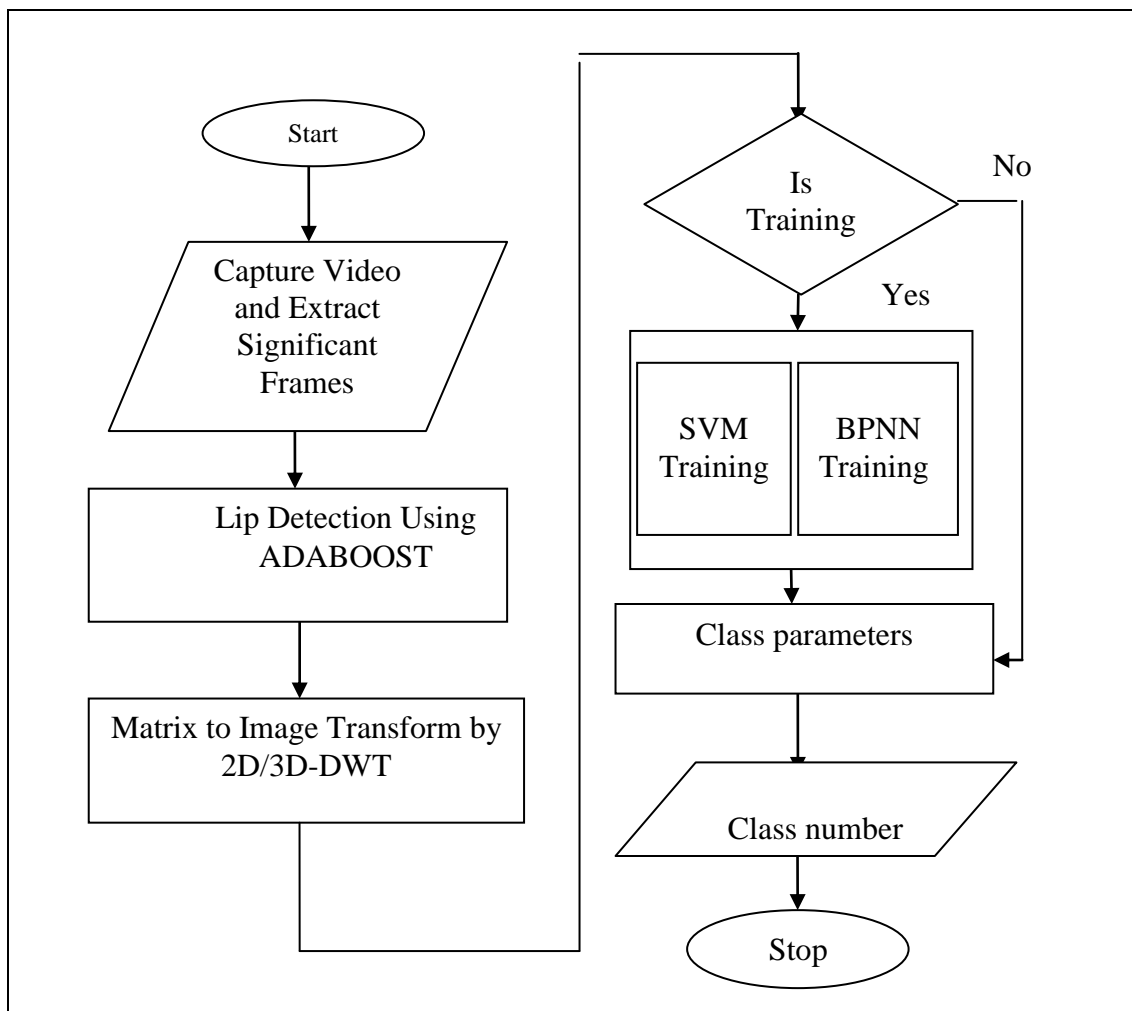


FIGURE 5: Flowchart for lip reading system.

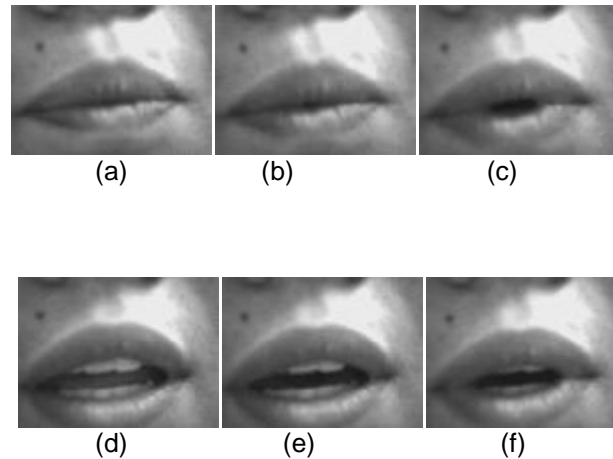


FIGURE 6 (a): to (f) Number of frames of zero utterance using TULIPS database.

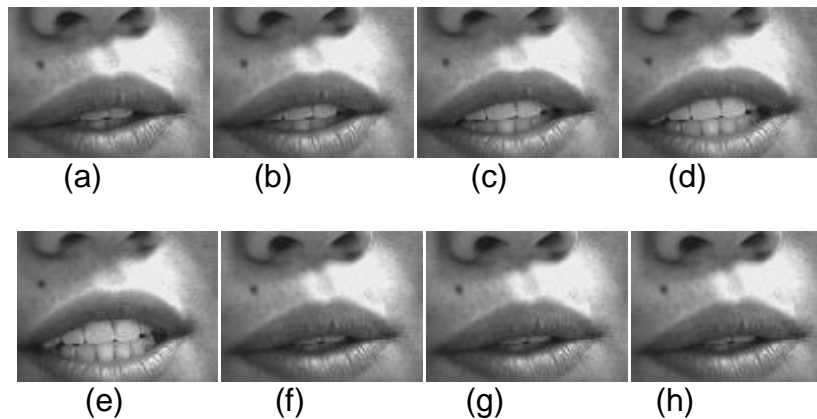


FIGURE 7: (a) to (h) Number of frames for utterance of three using TULIPS database.

6. FEATURE EXTRACTION

Before applying transformation on lip ROI it is rotated for orientation alignment with respect to a static reference frame, lip area localized to size 22 x 33 and passed through an LPF to remove high frequency noise. In proposed experimentation 3-D DWT with three decomposition levels are applied to lip area. DWT with Db2, Db4 and Dmey wavelets are used and respective coefficients are generated per frame. DWT Coefficients are calculated for 10 normalized frames in CUAVE database. This results in a feature vector of size 162 x 1. Seven users and each one uttering each digit 5 times produces 350 x 162 dimensional training dataset Feature vectors are levelled with 10 different classes each class corresponding to a digit. For 10 frames 162 coefficients are generated after 3D DWT as compared to 300 required for 2-D DWT.

7. EXPERIMENTAL RESULTS

7.1 Results for CUAVE Database

This section deals with the results. Table 1 shows recognition rate of lip reading for individual candidate with DWT and BPNN classifier. It shows that as the Recognition Rate (R.R.) of persons is more they provide more visual speech information. Table 2 indicate that Confusion Matrix for 2-D DWT with Dmey wavelet with BPNN classifier with $M=0.001$. Table 3 shows that Confusion Matrix for 3-D DWT with Dmey wavelet with BPNN classifier with $M=0.001$. Table 4 shows that recognition results of 3-D DWT are better than 2-D DWT for Db2 and Dmey wavelets. DWT with Db4 wavelet gives better result than Db2. DWT with Dmey wavelet shows highest recognition

rate with BPNN classifier. BPNN classifier outperforms SVM classifier. Table 5 shows that the performance improvement of each digit for 3-D DWT compared to 2-D DWT with BPNN and SVM classifier. Average % improvement is more in BPNN as compared to SVM. For 3-D DWT 0 digit has greater performance improvement.

TABLE 1: Lip reading results using 2-D DWT and BPNN classifier for individual candidate.

Candidate Number	Reco . Rate(%)
1	94
2	81
3	81
4	72
5	84
6	76
7	72

TABLE 2: Confusion Matrix for 2-D DWT with Dmey wavelet with BPNN classifier with M=0.001.

Digits	0	1	2	3	4	5	6	7	8	9	% R.R.(BPNN)
0	18	0	3	0	1	1	7	3	1	1	51.42
1	0	25	2	2	2	1	2	0	1	0	71.42
2	2	0	32	0	0	0	0	1	0	0	91.42
3	0	1	0	28	1	1	4	0	0	0	80
4	0	2	0	0	32	1	0	0	0	0	91.42
5	0	2	0	3	0	26	0	1	0	3	74.32
6	1	0	2	2	0	0	26	2	1	1	74.32
7	2	0	0	0	0	0	7	19	1		54.28
8	2	1	0	0	0	0	1	0	26	5	74.28
9	0	0	0	0	0	0	2	2	6	25	71.42
Average Result											73.43

TABLE 3: Confusion Matrix for 3-D DWT with Dmey wavelet with BPNN classifier with M=0.001.

Digits	0	1	2	3	4	5	6	7	8	9	% R.R.(BPNN)
0	24	0	3	1	1	0	3	2	1	0	68
1	1	26	0	2	2	1	1	1	1	0	74.28
2	1	0	34	0	0	0	0	0	0	0	97.1
3	0	0	1	26	1	2	4	1	0	0	74.28
4	0	0	1	0	33	1	0	0	0	0	94.28
5	0	1	0	0	1	30	0	2	0	1	85.71
6	2	2	0	2	0	0	29	0	0	0	82.85
7	1	0	0	1	0	0	3	27	0	3	77.14
8	0	0	0	0	0	1	0	0	31	3	88.57
9	0	0	0	0	0	0	0	1	5	29	82.85
Average Result											82.50

TABLE 4: Reco. Result (R.R.) for 3-D DWT using BPNN and SVM.

Type of Transformation	SVM R.R.%	BPNN R.R, %
2-D DWT (Db2)	67.14	71.14
2-D DWT (Db4)	55.14	59.71
2-D DWT (Dmey)	70.85	73.71
3-D DWT (Db2)	73	78
3-D DWT (Db4)	75.23	80
3-D DWT (Dmey)	78.56	82.50

TABLE 5: % improvement in R.R. for feature vector using 2-D DWT and 3-D DWT for BPNN and SVM.

Digits	BPNN	SVM
	%R.R. Improvement	%R.R. Improvement
0	16.58	17.15
1	2.86	2.85
2	5.68	0
3	-5.72	8.58
4	2.86	8.58
5	11.39	20
6	8.53	0
7	22.86	8.57
8	14.29	8.54
9	11.43	2.86
Avg	9.07	7.7

7.2 Results for TULIPS Database

SVM and BPNN are trained for feature classification. Table 6 shows the confusion matrix for feature vector using 2-D DWT with BPNN classifier with TULIPS database. From confusion matrix 1 and 4 are found to be most recognized digits and 3 is less recognized. From Table 7, 3-D DWT with Dmey wavelet performance found to be better with, as compare to 2-D DWT. Tulips database results for digit 3 are less as compare to CUAVE database because orientation of lip is not proper and nose portion also appear in lip image as shown in Fig. 7.

TABLE 6: Confusion matrix for 4 digits using 2-D DWT Dmey wavelet feature vectors with BPNN.

		Digit Presented				Recognition Rate in %
		1	2	3	4	
Subject Response	1	21	0	2	1	87.5
	2	0	18	4	2	75
	3	3	2	16	3	66.7
	4	1	1	1	21	87.5
Average Result						79.2

TABLE 7: Confusion matrix for 4 digits using 3-D DWT Dmey with BPNN.

		Digit Presented				Recognition Rate in %
		1	2	3	4	
Subject Response	1	22	0	2	1	91.67
	2	0	18	4	2	75
	3	3	2	16	3	66.7
	4	1	1	1	22	91.67
Average Result						81.25

8. CONCLUSION

In this paper, we have compared 2-D DWT with 3-D DWT features for lip reading. Naturally 3-D DWT performance is better as compare to 2-D DWT. SVM and BPNN are trained for feature classification. BPNN (classifier), performance found to be better with 3-D DWT with Dmey wavelet, as compare to the feature vector from other transform techniques. So BPNN is the most appropriate classifier with 3-D DWT. Among the digits, '4' is found as most discriminative and has been always acknowledged. '0' has less recognition rate as compared to other numbers. As using 3-D DWT length of feature vector is small, to build the training model SVM and BPNN required less computation time. Performance of lip reading system using both 2-D and 3-D DWT is less for Tulips database because of lip orientation. Further experimentation may reduce the 3-D DWT feature vector size by using proper discrimination technique and the performance of lip reading can be improve for real time application.

9. REFERENCES

- [1] E. D. Petajan, "Automatic lip-reading to enhance speech recognition", Ph.D. Thesis University of Illinois, 1984.
- [2] M. C.Weeks "Architectures For The 3-D Discrete Wavelet Transform" Ph.D. Thesis University of Southwestern Louisiana, 1998.
- [3] Bergler and Y. Konig, ""Eigenlips" For robust speech recognition," in Proc. IEEE Int. Conf. on Acustics , Speech and signal processing, 1994.
- [4] Potamianos, H. Graf, and E. Cosatto, "An image transform approach for HMM based automatic lip reading," Int. Conf. on Image Processing, 173–177, 1998.
- [5] R. Seymour, D. Stewart, and Ji Ming, "Comparison of image transform-based features for visual speech recognition in clean and corrupted videos," EURASIP Journal on Video Processing, Vol. 2008, 1-9, 2008.
- [6] X. Wang, Y. Hao, D. Fu, and C. Yuan "ROI processing for visual features extraction in lip-reading," IEEE Int. Conf. Neural Networks & Signal Processing, 178-181, 2008.
- [7] N. Puviarasan, S. Palanivel, "Lip reading of hearing impaired persons using HMM," Elsevier Journal on Expert Systems with Applications, 1-5, 2010.
- [8] A. Shaikh and J. Gubbi, "Lip reading using optical flow and support vector machines", CISP 2010, 327-310, 2010.
- [9] G. F. Meyor, J. B. Mulligan and S. M. Wuerger, "Continuous audio-visual using N test decision Fusion", Elsevier Journal on Information Fusion, 91-100, 2004.
- [10] L. Rothkrantz, J. Wojdel, and P. Wiggers, "Comparison between different feature extraction techniques in lipreading applications," SPECOM- 2006, 25-29, 2006.

- [11] P. Viola, M. Jones, "Rapid Object Detection using a Boosted Cascade of Simple features", IEEE Int. Conf., 511-517, 2001.
- [12] H. Lee, Y. Kim, A. Rowberg, and E. Riskin, "Statistical Distributions of DCT Coefficients and their Application to an Inter frame Compression Algorithm for 3-D Medical Images," IEEE Transactions of Medical Imaging, Vol. 12, 478-485, 1993.
- [13] J. Wang and H. Huang, "Three-dimensional Medical Image Compression using a Wavelet Transform with Parallel Computing," SPIE Imaging Physics Vol. 2431, 16-26,1995,
- [14] V. Long and L. Gang "Selection of the best wavelet base for speech signal" IEEE. Intelligent multimedia, video and speech processing, 2004.
- [15] A. K. Jain, R. P. Duin, and J. Mao, "Statistical Pattern Recognition: A Review" IEEE Transactions On Pattern Analysis And Machine Intelligence, 22, 1, 2000.
- [16] C. Bregler and Y. Konig, "Eigenlips" For Robust Speech Recognition", IEEE conf. Acoustics, Speech, and Signal Processing, 1-4, 1994.
- [17] V.N. Vapnik, "stastical learning theory" New York John Wiley & Suns, 1998.
- [18] V. Kechman, "Learning and soft computing, support vector machines, Neural Networks and Fuzzy logic models", MIT Press Cambridge, 1-58, 2001.
- [19] J. C. Platt, "Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines", Microsoft research reports, 1-21, 1998.
- [20] E. Osuna, R.Freund and F.Girosi, An Improved Training Algorithm for Support Vector Machines, Neural networks for signal processing", Proc. of IEEE 1997, 276-285, 1997
- [21] E. Patterson, S. Gurbuz, Z. Tufekci, and J. Gowdy, "CUAVE: a new audio-visual database for multimodal human computer- interface research", Proceedings of IEEE Int. conf. on Acoustics, speech and Signal Processing, 2017-2020, 2002.
- [22] J. R. Movellan "Visual Speech Recognition with Stochastic Networks", Advances in Neural Information Processing Systems, MIT Pess, Cambridge, 1995.

INSTRUCTIONS TO CONTRIBUTORS

The *International Journal of Image Processing (IJIP)* aims to be an effective forum for interchange of high quality theoretical and applied research in the Image Processing domain from basic research to application development. It emphasizes on efficient and effective image technologies, and provides a central forum for a deeper understanding in the discipline by encouraging the quantitative comparison and performance evaluation of the emerging components of image processing.

We welcome scientists, researchers, engineers and vendors from different disciplines to exchange ideas, identify problems, investigate relevant issues, share common interests, explore new approaches, and initiate possible collaborative research and system development.

To build its International reputation, we are disseminating the publication information through Google Books, Google Scholar, Directory of Open Access Journals (DOAJ), Open J Gate, ScientificCommons, Docstoc and many more. Our International Editors are working on establishing ISI listing and a good impact factor for IJIP.

The initial efforts helped to shape the editorial policy and to sharpen the focus of the journal. Starting with Volume 9, 2015, IJIP will be appearing with more focused issues. Besides normal publications, IJIP intends to organize special issues on more focused topics. Each special issue will have a designated editor (editors) – either member of the editorial board or another recognized specialist in the respective field.

We are open to contributions, proposals for any topic as well as for editors and reviewers. We understand that it is through the effort of volunteers that CSC Journals continues to grow and flourish.

LIST OF TOPICS

The realm of International Journal of Image Processing (IJIP) extends, but not limited, to the following:

- Architecture of imaging and vision systems
- Character and handwritten text recognition
- Chemistry of photosensitive materials
- Coding and transmission
- Color imaging
- Data fusion from multiple sensor inputs
- Document image understanding
- Holography
- Image capturing, databases
- Image processing applications
- Image representation, sensing
- Implementation and architectures
- Materials for electro-photography
- New visual services over ATM/packet network
- Object modeling and knowledge acquisition
- Photographic emulsions
- Autonomous vehicles
- Chemical and spectral sensitization
- Coating technologies
- Cognitive aspects of image understanding
- Communication of visual data
- Display and printing
- Generation and display
- Image analysis and interpretation
- Image generation, manipulation, permanence
- Image processing: coding analysis and recognition
- Imaging systems and image scanning
- Latent image
- Network architecture for real-time video transport
- Non-impact printing technologies
- Photoconductors
- Photopolymers

- Prepress and printing technologies
- Remote image sensing
- Storage and transmission

- Protocols for packet video
- Retrieval and multimedia
- Video coding algorithms and technologies for ATM/p

CALL FOR PAPERS

Volume: 9 - Issue: 1

i. Submission Deadline : November 30, 2014 **ii. Author Notification:** December 31, 2014

iii. Issue Publication: January 2015

CONTACT INFORMATION

Computer Science Journals Sdn Bhd

B-5-8 Plaza Mont Kiara, Mont Kiara

50480, Kuala Lumpur, MALAYSIA

Phone: 006 03 6204 5627

Fax: 006 03 6204 5628

Email: cscpress@cscjournals.org

CSC PUBLISHERS © 2014
COMPUTER SCIENCE JOURNALS SDN BHD
B-5-8 PLAZA MONT KIARA
MONT KIARA
50480, KUALA LUMPUR
MALAYSIA

PHONE: 006 03 6204 5627

FAX: 006 03 6204 5628

EMAIL: cscpress@cscjournals.org