An Experimental Approach For Evaluating Superpixel's Consistency Over 2D Gaussian Blur and Impulse Noise Using Jaccard Similarity Coefficient

Brekhna Brekhna *brekhna87@gmail.com*
School of Computer Science and Technology, Shandong University, Jinan 250101, P. R. China

Caiming Zhang *czhang@sdu.edu.cn*
School of Computer Science and Technology, Shandong University, Jinan 250101, P. R. China

Yuanfeng Zhou *yfzhou@sdu.edu.cn*
School of Computer Science and Technology, Shandong University, Jinan 250101, P. R. China

Abstract

This article proposes a rigorous method to assess the consistency of superpixels for different superpixel segmentation algorithms. The proposed method extracts the superpixels that remain unchanged over certain levels of noise by adopting the Jaccard Similarity Coefficient (JSC). Technically, we developed a measure of Jaccard similarity for superpixel segmentation algorithms to compare the similarity between sets of superpixels (original and noisy). The algorithm calls the superpixel segmentation algorithm to generate the superpixel results of the original images and saves their boundary masks and labels. It then applies varying degrees of noise to the images and produces the superpixels results, and the process is repeated for four levels with increased noise value at each iteration. We chose 2D Gaussian Blur, Impulse Noise and a combination of both to corrupt the images. The proposed algorithm generates similarity indices of superpixels (original and noisy) using Jaccard Similarity (JS). To be categorized as a consistent superpixel, the similarity index must meet a predefined coefficient threshold (*τ)* of JSC. The superpixels consistency of four different superpixel segmentation algorithms including Bilateral geodesic distance (BGD), Flooding based superpixels generation (FBS), superpixels via geodesic distance (GDS), and Turbopixel (TP) are evaluated. Precisely, the experimental results demonstrated that no single algorithm was able to yield an optimal outcome and failed to maintain consistent superpixels at each level of noise. Conclusively, more robust superpixel algorithms must be developed to solve such problems effectively.

Keywords: Superpixels, Jaccard Similarity Coefficient, 2D Gaussian Blur, Impulse Noise, Threshold, Consistency.

1. INTRODUCTION

A superpixel is a polygonal fragment of an image, more substantial than a typical pixel in size and concentrated with even color and brightness [1]. Compared to the old pixel representation methods, superpixels dramatically moderate the number of image segments and progress the representative effectiveness [2]. The key benefit of superpixel is to deliver a more natural and perceptually significant depiction of the input image. Concerning pixels, the superpixel contains more local information and can adhere to most of the object's boundaries in the images. Superpixel segmentation belongs to the class of over-segmentation algorithms that are widely used as a pre-processing step in different applications of computer vision such as object recognition [3], image/video segmentation [4, 5, 6, 7], image classification [8, 9, 10, 11] visual tracking [12], dense image matching [13] and saliency detection [14, 15] etc. So far, many algorithms have been introduced to generate superpixels. Some of them are based on k-means clustering such as SLIC[16], VCell [17], manifold SLIC [18], DBSCAN clustering algorithm [19], SEEDS [20], whereas, some are graph-based techniques such as normalized cuts(N-Cuts) [21, 22], lazy random walk [2]. Some of them are Geometric Flow-Based Techniques [2], such as Turbopixel [24], superpixels based on the bilateral geodesic distance [25] and flooding-based superpixel algorithm [26]. For ideal superpixels segmentation, different properties such as the uniform intensity of pixels in a particular region, consistent sizes, regular shape and the region boundary adherence to image boundary are significant. Most of these superpixel segmentation algorithms have shown remarkable results on clean and sharp images devoid of noise. None of the algorithms is applied to the set of images that suffers from noise. Many evaluation metrics are used by these algorithms to evaluate the superpixels concerning the different superpixel properties. In this artical, we present a new method of evaluation that aims to assess the consistency of superpixels and unify the comparison process of different superpixel segmentation algorithms over common types of noise using the Jaccard similarity coefficient.

An experimental assessment process of superpixels that evaluate the consistency of superpixels of different superpixel segmentation algorithms over certain levels/intensities of 2D Gaussian blur, Impulse noise and a combined effect of both. This study not only presents the assessment process of superpixels but also gives a complete performance analysis and comparison of some of the previously presented superpixel segmentation algorithms. For more quantitative assessment of the effect of noise over superpixel segmentation algorithms performance and results, we used the performance parameters including Achievable Segmentation Accuracy (ASA), Under Segmentation Error (USE) and Boundary Recall (BR). The primary purpose of the study is to evaluate each superpixel for its consistency generated by an algorithm to show that noise effect on the algorithm's performance and observe the algorithm to produce the same superpixel on the noisy image. In superpixels segmentation algorithms consistency is the pattern of production of superpixels, agreement, compatibility, especially correspondence or uniformity among the parts of the complex and noisy image data. Based on our algorithm, perceptually we defined "*a consistent superpixel as a unit of uniform pixels, covering the same object over different noise levels*." The more robust a superpixel segmentation algorithm, the more consistent superpixels it will generate over noise levels.

We performed three different experiments. For experiment one, we applied the 2D Gaussian blur filter to the set of images that smoothens out the high frequencies of an image and reduces its specific detail. For experiment two, we applied impulse noise (salt and pepper) to the set of images that is an independent and uncorrelated noise randomly distributed over the image. For experiment three, we corrupted the images with both 2D Gaussian blur and the impulse noise. The blur filter disturbs some of the specific details of the image where the impulse noise randomly spread the bright pixels in dark regions and dark pixels in the bright regions on an image. The effect of both disturbs the uniform intensity of pixels in a particular region.

The new algorithm calls a superpixel segmentation algorithm to generate the superpixels results of the original images and save their boundary mask and labels. Then the selected noise is applied to the images, and superpixels results are generated again. The process is repeated for four levels, applying various degrees of a blur filter for experiment one, impulse noise for experiment two and a combination of both for experiment three, increasing the value of noise at each level. For each experiment, the Jaccard Similarity Coefficient is used to compute the similarity indices by comparing the superpixels results of all four levels to the original superpixels results. The similarity index is the value of similar pixel labels between the original and noisy superpixel. The algorithm searched for the similarity indices with higher similar value and then comparing these indices to the coefficient threshold range from {0 to 1}. This threshold is used as a base condition to get the final consistent superpixels among all noise levels. In the range of {0 to 1} three different thresholds values are chosen, i.e., {0.3, 0.5, and 0.8}. Figure 1 shows the visual effect of 2D Gaussian blur, impulse noise and combined (blur + impulse noise). However, the last two columns depict only the consistent superpixels by our algorithm. Thus, this paper contributes to:

- The understanding of the evaluation of superpixel consistency by applying the Jaccard Similarity Coefficient to it for the first time.
- A new and rigorous evaluation process of superpixel segmentation algorithms.

The rest of the paper is arranged as follows: Section 2 explains the algorithms details and essential terminologies. Section 3 describes our proposed algorithm, and Section 4 provides a detailed discussion of the experiment results. The paper is concluded in Section 5.

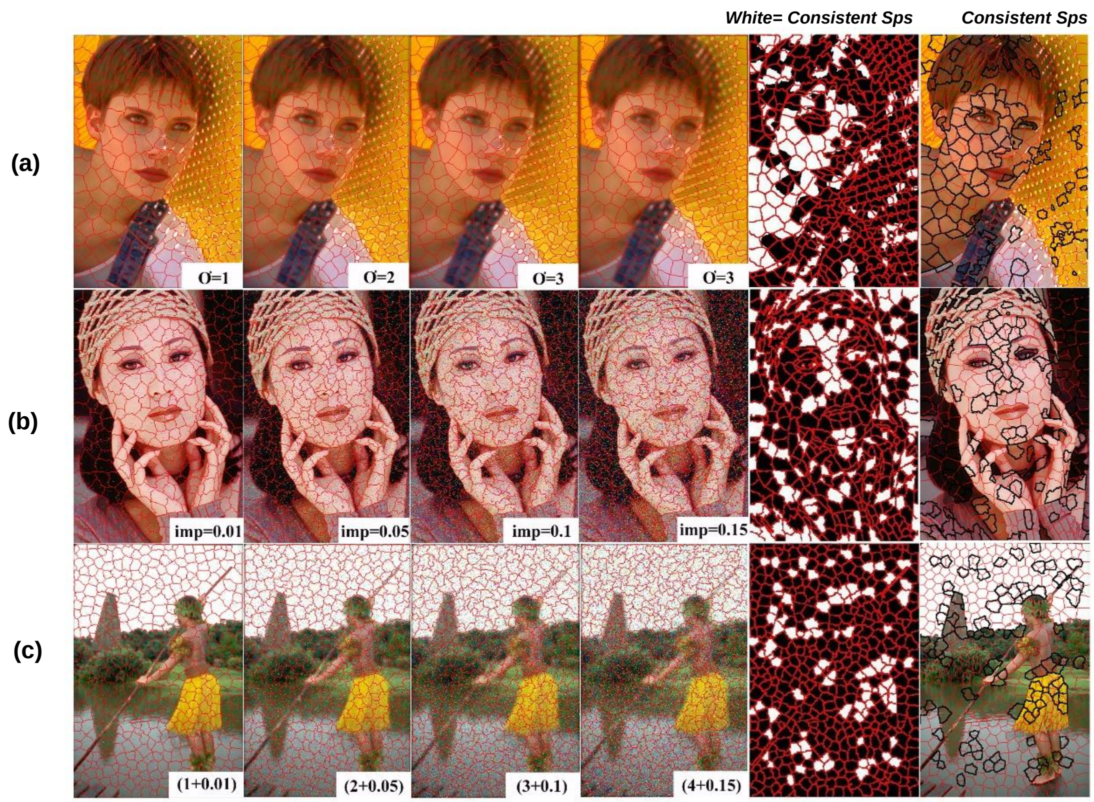


FIGURE 1: The effects of 2D Gaussian blur, Impulse noise and both (Impulse noise + blur).The last two columns present the consistent superpixels extracted for the four levels for noise. (a) 2D Gaussian blur, (b) Impulse noise and (c) Blur + impulse noise.

2. ALGORITHM DETAILS

2.1 Jaccard Similarity Coefficient (JSC)

The Jaccard similarity of data sets is the fraction of components two data sets have in common [27]. It is a measure of similarity for the two data sets, with a range from 0% to 100%. In other words, it is the intersection over the union of two sets. Figure 2 demonstrates the complete concept of Jaccard similarity. This JSC measure is suitable for several applications such as textual similarity of documents [28], the similarity of buying habits of customers [29], map reduction for entity pairs in Wikipedia [30], etc. In this study, the JSC measure is being used to evaluate the consistency of superpixel, i.e., original and noisy by comparing their labels. In general, for any two sets, the JSC is defined as:

$$sim_{jaccard}(Set_i\ ,Set_j) = \frac{|the\ intersection\ of\ Set_i\ and\ Set_j|}{the\ union\ of\ Set_i\ and\ Set_j} \tag{1}$$

In our case, the sets are the superpixels segmentation results of an original image **I** and noisy image ***Io***. For each superpixel, suppose (***si***) represents a superpixel that contains pixels of the same label in the original image I and (***sj***) represents the superpixel of the noisy image ***Io*** that contains pixels of the same label. For ***n*** number of superpixels in an image ***si*** and ***sj*** are ***si***= {*s1*, s2 ...., *sn*} and ***sj***={*s1*, *s2* ...., *sn*}. For such data, the JSC is defined as:

$$sim_{jaccard}(I\ ,Io) = \frac{I(s_i)\cap Io(s_j)}{|I(s_i)|+|Io(s_j)|-I(s_i)\cap Io(s_j)} = \frac{|I(s_i)\cap Io(s_j)|}{|I(s_i)\cup Io(s_j)|} \tag{2}$$

The general form to extract the average number of consistent superpixels over the entire set of images *Cs* is defined as:

$$C_s = \sum sim_{jaccard} \frac{|I(s_i)\cap Io(s_j)|}{|I(s_i)\cup Io(s_j)|} \tag{3}$$

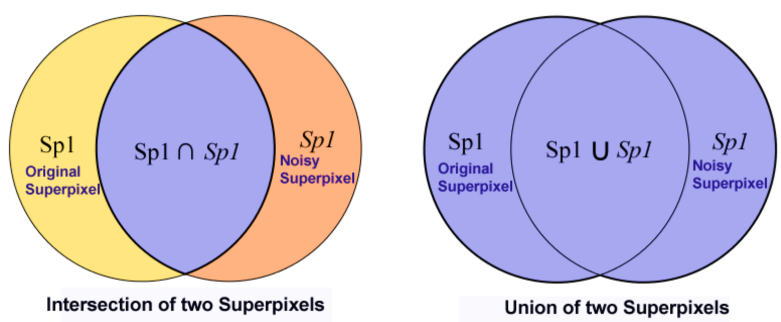


FIGURE 2: The process of Jaccard similarity criteria, i.e., intersection over the union, If two objects are represented in superpixels form as Sp1 (original superpixel) and S*p1* (noisy superpixel).

2.2 Two Dimensional Gaussian Blur

A two-dimensional Gaussian blur filter smoothens out the high frequencies of an image and reduces its specific details. The gradient magnitudes at the image boundaries dropped sharply after blurring [31]. The Gaussian function is defined as:

$$G(x,y) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\left(\frac{x^2+y^2}{2\sigma^2}\right)} \tag{4}$$

Where *x* and *y* embody the distances from the origin in the horizontal and vertical axis respectively, σ is the standard deviation which controls the degree of smoothness. The application of this filter to an image produces a surface whose contours are concentric circles with a Gaussian distribution from the center point. These distributions build a convolution matrix which is applied to the image, and each pixel receives a new value. Each pixel value is set to the weighted average of the neighborhood pixels. The original pixel's value attains the most substantial weight, and adjacent pixels receive lesser weights as their distance to the original pixel increases.

2.3 Impulse Noise

Impulse noise [32, 33] is one of the very common noises in digital images. It is always an independent and uncorrelated to the image pixels and is randomly distributed over the image plane. In such type of noise, the noisy pixels take either the salt value (grey level -225) or the pepper value (grey level -0). The random distribution of salt and pepper noise on the image plane spread bright pixels in the dark parts and dark pixels in the bright parts. It changed pixels values in both complex and flat regions of an image and appeared as black and white spots on the images.

2.4 Superpixel Segmentation Algorithms

We append our algorithm with four different superpixel segmentation algorithms to compute different results. We chose these four algorithms on the bases of their performance evaluation in the presence of different noises from our previous study presented in Ref [23]. The four different latest algorithms are:

- Superpixel generation algorithm based on the bilateral geodesic distance (BGD) [25], which focuses on a new geodesic distance, based on the color and seed location of each superpixel. The algorithm uses a dynamic seed-sensitive image gradient operator that computes the color differences between the seeds and pixels enhancing the final superpixels.
- Structure-sensitive superpixels based on geodesic distance (GDS) [34] algorithm generates superpixels using the geodesic distance by computing the density of an image structure. Its density function computes content sensitive superpixels, i.e., small superpixels in high-color and large superpixels in low-color variation regions.
- Flooding-based superpixel generation algorithm (FBS) [26] generate the superpixel using essential constraints such as color, compactness, and smoothness.
- Turbopixel (TP) [24] algorithm uses the level set method for compact and regular size superpixels generation. It creates a lattice-like structure of an image, and the level set method develops the curves of the lattice to obtain the superpixel boundaries.

3. OUR APPROACH

The algorithm is designed to extract the consistent superpixels over a set of original and noisy images by affixing different superpixels algorithms. For this purpose, the Jaccard Similarity Coefficient is used to compare the original results of a superpixel segmentation algorithm to its noisy version. Various degrees of noise is applied to the set of images, and the comparison is made for each noise level with the original results. Those superpixels that remained unchanged over certain levels of noise are extracted and categorized as consistent superpixels. In this algorithm, the computation of original results is essential because they are treated as ground truth for the comparison purpose to other noise levels results to work out the consistent superpixels. The original superpixels segmentation result is considered as a training set for training the final consistent superpixels, whereas the noisy superpixel segmentation result is testing set for evaluating the final consistent superpixels. The algorithm details are given below.

3.1 Computation of Superpixels

The first essential part is the computation of the superpixel segmentation results. The algorithm 1, first calls a superpixel segmentation algorithm and figures out the results for the original images, then applies noise to the set of images and calculates the superpixels segmentation results again. It repeats the process up to four levels of noise. We used four different algorithms including a bilateral geodesic algorithm (BGD), flooding-based superpixel generation (FBS), structure sensitive superpixels via geodesic distance (GDS) and Turbopixel (TP) for the superpixel segmentation results.

3.2 Computation of Similarity Indices

The superpixels segmentation results of the original images are treated as ground truth for the computation of consistent superpixels. The Jaccard Similarity Coefficient defined in equation 2 is used to compute the similarity index. Hence, we know that each superpixel is a set of pixels having the same labels. So the similarity index is turning the value of intersection over the union of the pixels of two superpixels, i.e., original and noisy. After computing the superpixel results for the original and noisy images (4 levels), the algorithm calculates the similarity indices between them based upon the JS criteria. For each noise level, the noisy superpixels are compared with the original superpixels to compute their similarity indices.

3.3 Determination of Coefficient Threshold

The coefficient threshold is the value of similarity of consistent superpixels. It is the degree of similarity and arrangement of the consistent superpixels between the original results and their noisy version. The value of the threshold is in the range of {0 and 1}. The value of 0 specifies that there is no similarity between the two superpixels, whereas the value of 1 determines a similarity. For our experiments, we vary the values of the coefficient threshold to acquire different results.

We chose three different τ values, i.e., {0.3, 0.5 and 0.8} to compute different results. The value {0.3} selects the similarity index as consistent superpixels if there is a 30 percent similarity between the pixels of two superpixels. Similarly, for threshold value {0.5 and 0.8}, the similarity should be 50 and 80 percent. Once the algorithm 1 generates the similarity indices; it compares these indices values to the coefficient threshold and arranges them in a descending order keeping the maximum similar value (consistent superpixel) on top in an array. First, the coefficient threshold value is set to {0.3}, which means the similarity index value is greater or equal to the {0.3} of the threshold; then the superpixel is selected as consistent superpixel. Other similarity index values that do not match this criterion can be interpreted as not selected. This process is repeated for each noise level. The same condition is applied for the coefficient threshold value {0.5 and 0.8}. For each value of the threshold, we repeated the experiment and the highest value of the threshold stern the condition of similarity, whereas the lowest value makes it lenient to extract more consistent superpixels. In figure 3 we explained how our algorithm computes the consistent superpixel based on the coefficient threshold values. Starting from τ = {0.3 to 0.8} the condition is getting strict, and it computes strong consistent superpixels.

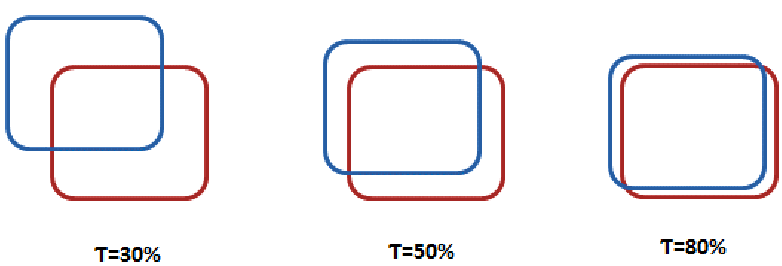


FIGURE 3: The three conditions of similarity of superpixels to define a superpixel consistent. The empty spaces show the pixels with similar labels under the boundary of a single superpixel.

3.4 Final Grouping (Output)

In algorithm 1, presented in section 3.5, steps 4(b) and 4(c) are used to do the final grouping and count the consistent superpixels for each noise level. This step is dependent on the value of the coefficient threshold. After the computation of similarity indices of superpixels, the algorithm searches the indices with maximum similar value. Then these indices are compared to the threshold and are arranged in an array. The similarity index must meet the criterion of being greater than or equal to the threshold. In this regard, the algorithm organizes the indices in a descending order starting from the maximum similar value. For each particular noise level, the number of consistent superpixels is averaged over the whole set of images.

3.5 Algorithm 1

As a summary, Algorithm 1 gives a complete description of the proposed method.

Input: Image **I**, the expected number of superpixels S*p*, blur filter, impulse noise or combined (blur+impulse noise) and Threshold $\boldsymbol{\tau}$

Output: Consistent superpixels

1. Generate superpixels S*p* on input image **I,** save the boundaries mask and labels
2. Apply selected noise to **I** up to 4 levels (increase noise value at each level)
   a. Repeat step 1 for the noisy image ***Io***
3. Set the threshold $\boldsymbol{\tau}$ ={0.3, 0.5, 0.8}
4. Compute similarity indices of **I** and ***Io*** (eq.1)
   a. First, take the superpixel segmentation results of the original and noisy images and calculate the similarity indices based on JSC equation 1.
   b. Second, Compare the similarity indices values with the coefficient threshold and arrange the indices in descending order.
   c. Compute and average the number of Consistent Superpixels over 500 images
5. Repeat the process for each noise level

4. EXPERIMENTS AND RESULTS

The algorithm presented in this study is implemented using Matlab 2017b. We create a new framework in which we used different superpixels algorithms with our new algorithm to compute and extract the desired results. These superpixel algorithms are also employed in Matlab with the extension of C++. We performed three different experiments. For experiment one, we applied 2D Gaussian blur, for the experiment two impulse noise and the experiment three both blur and impulse noise is combined and applied to the images to compute more robust results. Also, these three experiments are repeated based on the values of the threshold parameter, which is kept {0.3, 0.5, and 0.8}. First, the value of the threshold is kept {0.3}. It is the lowest similarity value. The comparison at this value is very loose. Second, the threshold value is {0.5}, which is the middle value. Third, the value is kept {0.8}, which is the highest value for comparison. The consistent superpixels computed for {0.8} is said to be high-level consistent superpixels, as these superpixels remained constant over all four noise levels under a stringent condition. The superpixels remained consistent on threshold level {0.5} are medium-level, and threshold level {0.3} are the low-level consistent superpixels. The number of superpixels for each experiment is kept the same, i.e., {500}. The experiments are performed over Berkeley BSDS500 benchmark [35]. The database contains a set of 500 images, and the performance and results are averaged over the whole database.

To achieve more quantitative assessments and observe the effect of blur filter and impulse noise over the performance of superpixel segmentation algorithm, we also computed the Achievable segmentation accuracy (ASA), Under-Segmentation error (USE), and Boundary Recall (BR) for all three experiments. These performance parameters are computed to show the performance degradation of all algorithms by adding noise to the images. The ASA [36] measures the accuracy attained by an algorithm by comparing the superpixels with the labels of ground truth segmentation. In other words, it figures out the maximum achievable accuracy by labeling each superpixel with the label of ground truth segmentation that has a significant overlap area. The USE [24] for an algorithm is calculated concerning flow out of a segment output produced by an algorithm when placed over ground truth segments. Thus, the measure penalizes any superpixels that do not firmly fit the ground truth segment. The BR [26] measures the fraction of the ground truth edges overlapped within a small disk-shaped neighborhood of the superpixels boundaries, and for all the three experiments, the disk radius is set as 2 pixels.

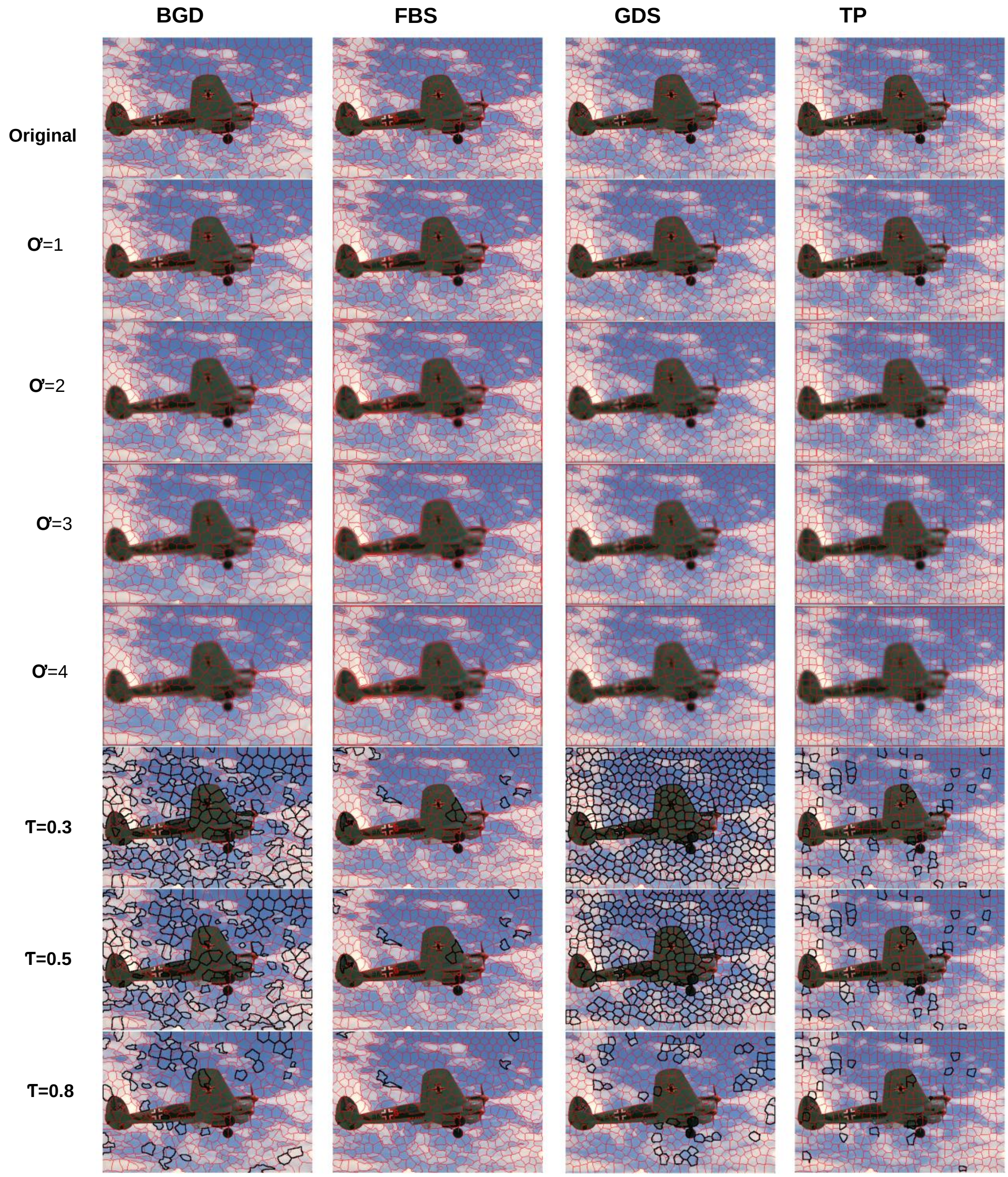


FIGURE 4: The visual result of the effect of blur filter for the values {1, 2, 3, and 4} and the extraction of consistent superpixels for all threshold values.

4.1 Experiment One (2D Gaussian Blur)

Figure 4 shows the visual results of our algorithm for 2D Gaussian blur and validates the influence of consistent superpixels over the threshold parameter {0.3, 0.5, 0.8}. The different threshold values affect the selection of final consistent superpixels. In figure 5, we presented the individual result of each algorithm, which shows the consistency comparison of each blur level and the decrease in the number of consistent superpixels with the increase of blur level. In figure 6, we presented the combined comparison of all four algorithms concerning the threshold values. The results shown in figure 6 reveal that GDS performed well among all algorithms. The structure sensitive nature of GDS algorithm using geodesic distance utilizes the density of an image structure to produce superpixels. This property helps the algorithm to produce and maintain more consistent superpixel superpixels for all threshold values. At the second BGD managed more consistent superpixels. The bilateral geodesic distance defined by BGD sets a new dynamic seed-sensitive image gradient operator that computes the distance information of two-pixel positions which evaluates the levels of color difference and allows the algorithm to generate and maintain consistent superpixels. The FBS and TP produced the lowest number of consistent superpixels.

To observe the effect of blur filter over images causing algorithm performance degradation more quantitatively, we presented the quantitative assessment in figure 7. For the ASA and USE, at the initial value of blur filer value (1), all algorithms slightly improve their performances. It is because the low levels of blur filter smooth out the image plan and remove some of the weak boundaries causing the algorithms to produce better superpixels. But as the blur increases, it also destroys the firm boundaries causing the algorithms to reduce its performance. With the increasing values of blur filter shows that all the algorithms decreased the ASA and BR, whereas increased the USE.

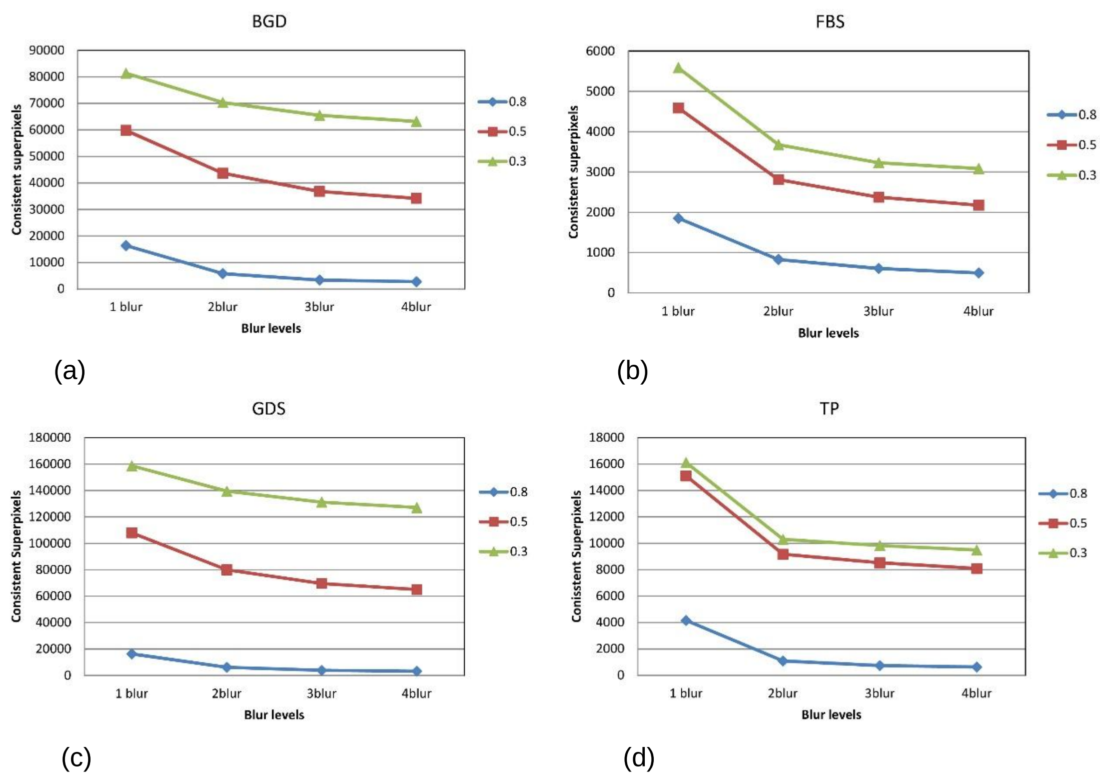


FIGURE 5: The number and the degradation of consistent superpixels for 4 blur levels of each individual algorithm for the threshold {0.3, 0.5, and 0.8} values. (a) BGD, (b) FBS, (c) GDS, and (d) TP.

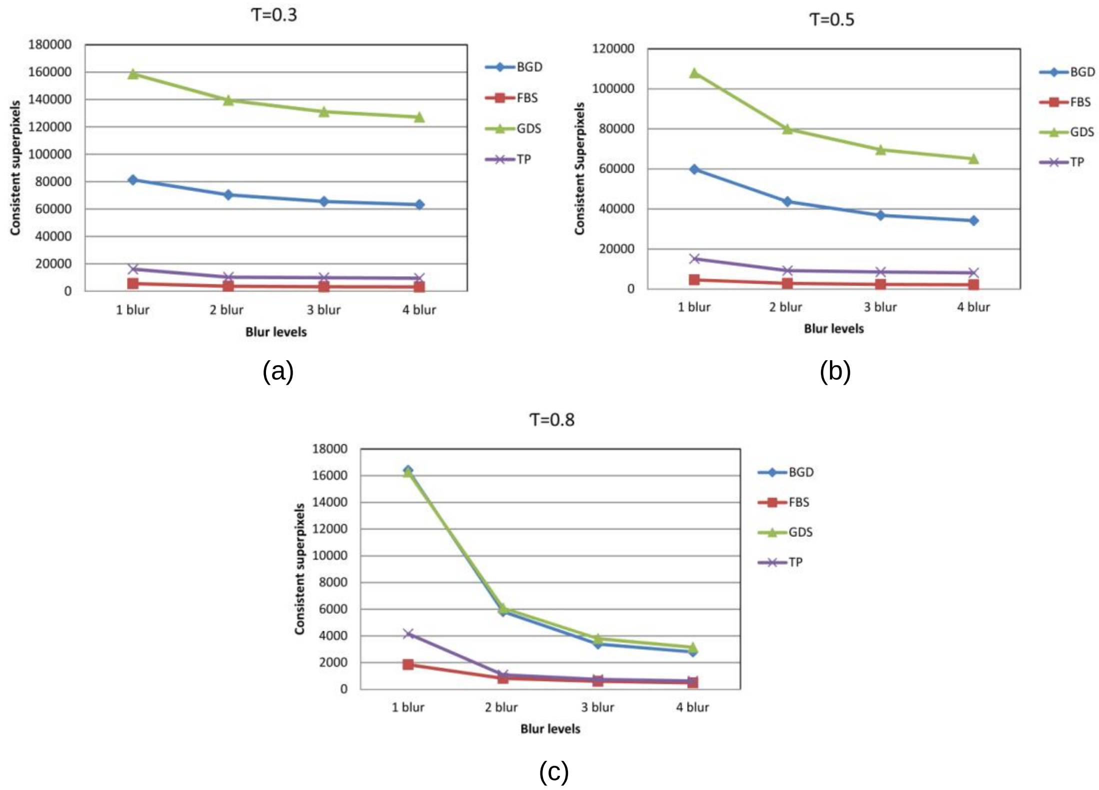


FIGURE 6: The superpixel consistency degradation comparison between the four superpixels algorithms for each value of the threshold for four blur levels. (a) {0.3}, (b) {0.5}, and (c) {0.8}.

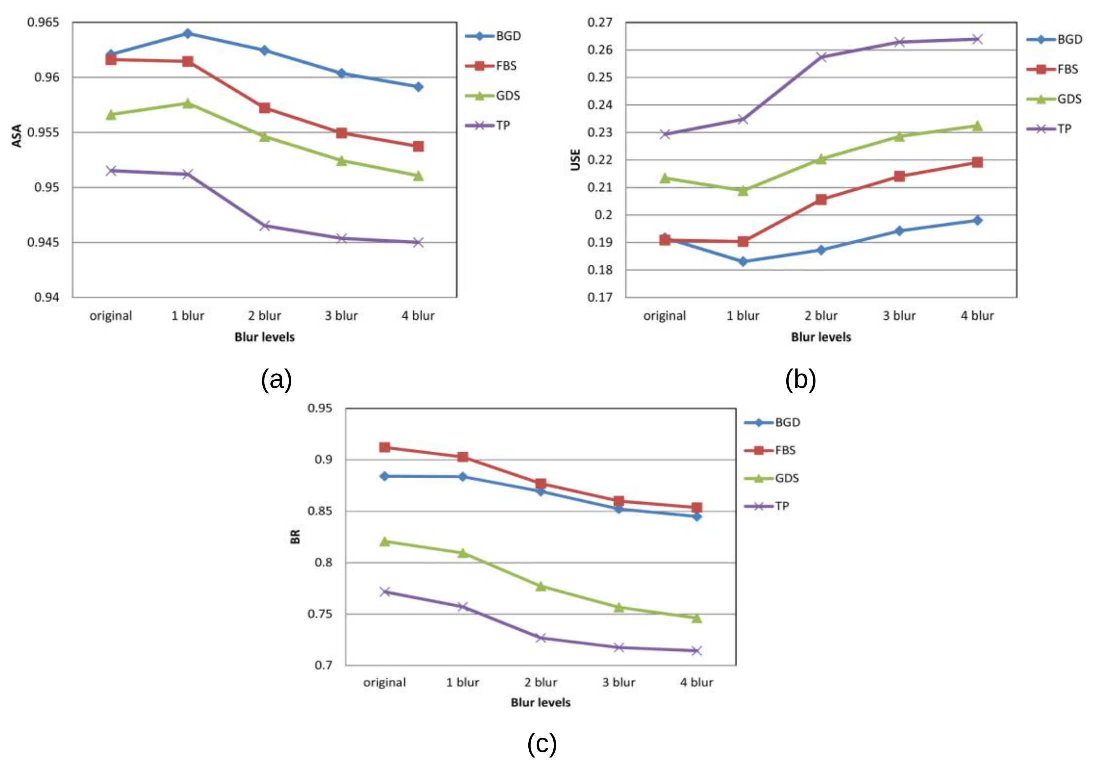


FIGURE 7: Quantitative evaluation results of BGD, FBS, GDS and TP for 2D Gaussian blur {1, 2, 3, 4} for 500 superpixels (a) ASA, (b) USE, and (c) BR.

4.2 Experiment Two (Impulse Noise)

The impulse noise is randomly distributed over the image plane changing the pixel value and appeared as black and white spots. It is independent and uncorrelated to the image pixels, and the noisy pixels take either the salt value (grey level -225) or the pepper value (grey level -0). We applied the noise up to four levels to the set of images. The noise value is kept {0.01, 0.05, 0.1, and 0.5}. The visual results for evaluating and extracting consistent superpixels for impulse noise are given in figure 8. The impulse noise profoundly affected the performance of algorithms. All algorithms maintained a low number of consistent superpixels for all noise levels.

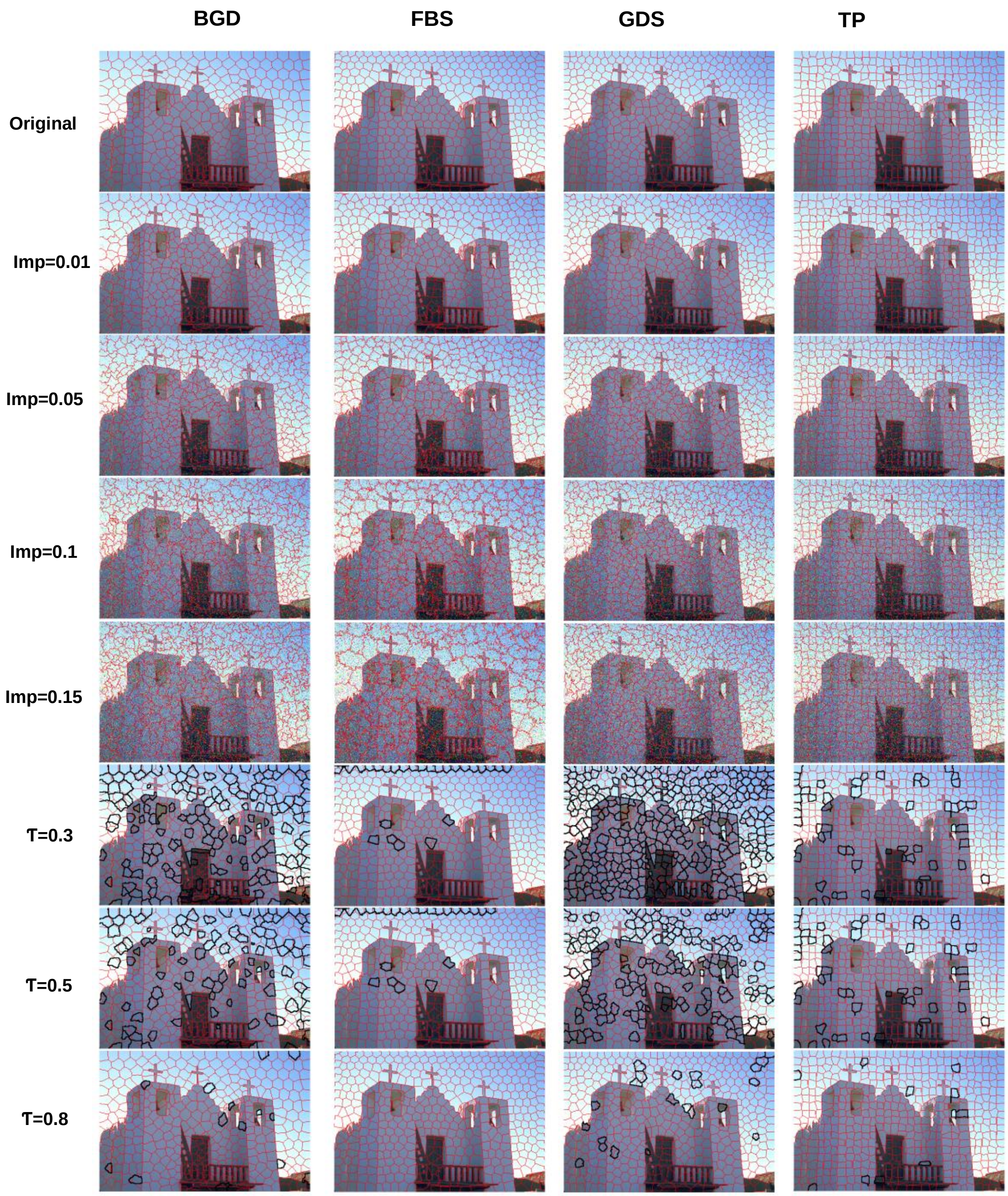


FIGURE 8: The visual result of the effect of impulse noise for the values {0.01, 0.05, 0.1, and 0.15} and the extraction of consistent superpixels for all threshold values.

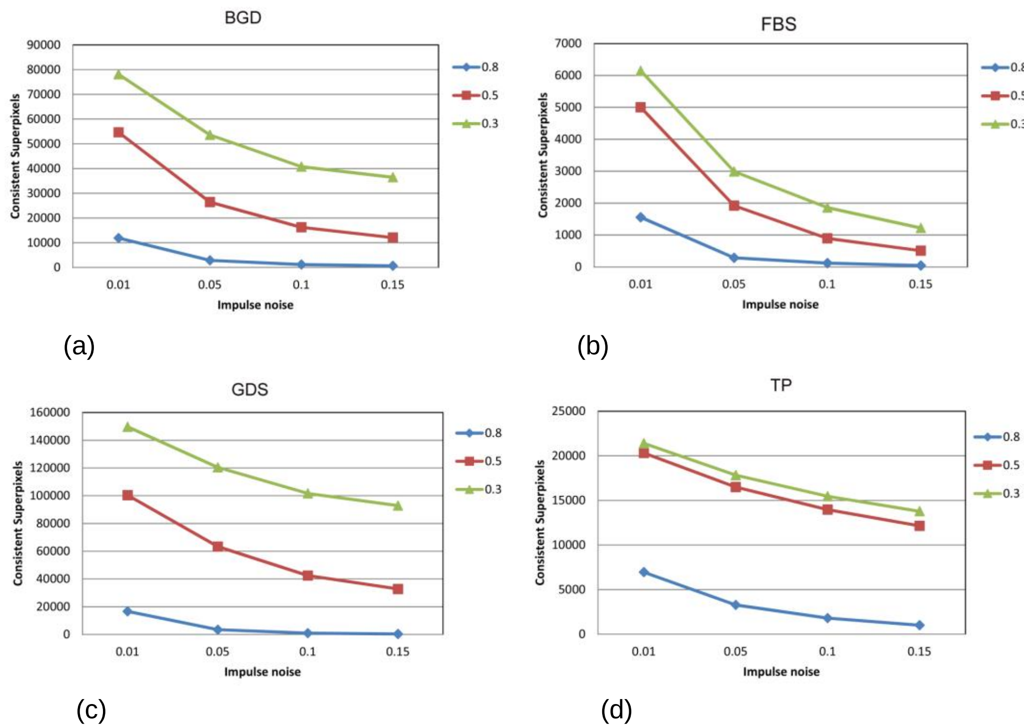


FIGURE 9: The number and the degradation of consistent superpixels for four levels of impulse noise of each algorithm for the threshold {0.3, 0.5, and 0.8} values. (a) BGD, (b) FBS, (c) GDS, and (d) TP.

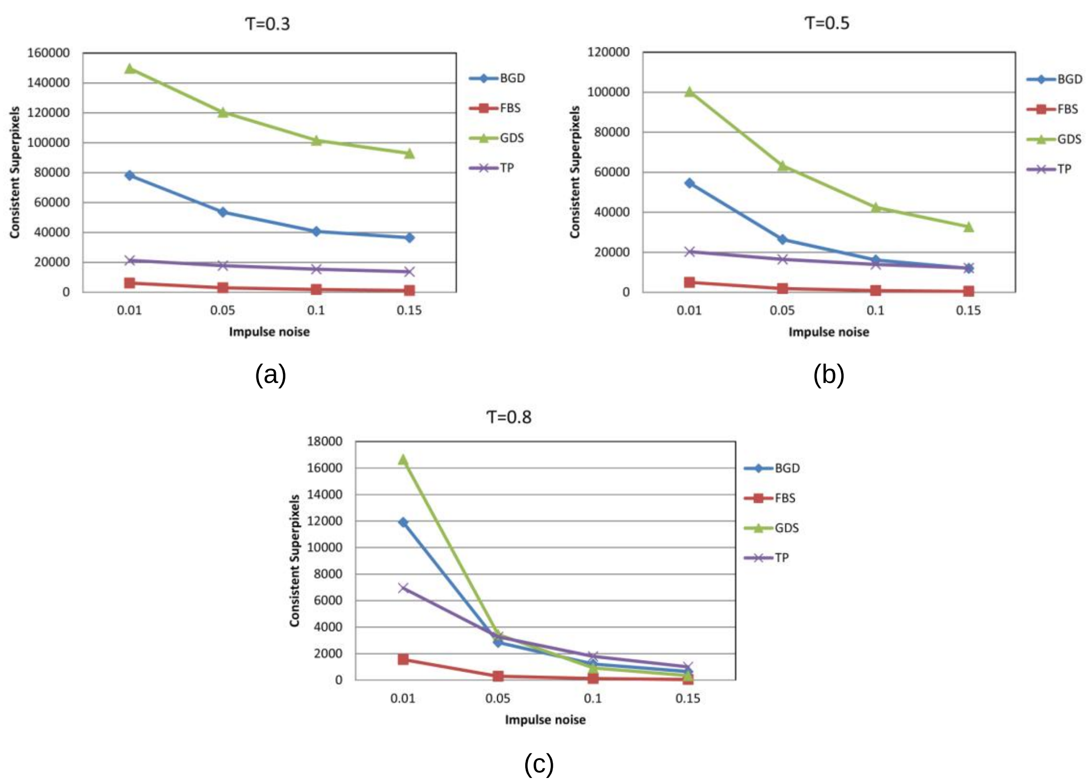


FIGURE 10: The superpixel consistency degradation comparison between the four superpixels algorithms for each value of the threshold for four levels of Impulse Noise. (a) {0.3}, (b) {0.5}, and (c) {0.8}.

Figure 9 shows the result of each algorithm and demonstrates the average number of consistent superpixels for the three different threshold values. For the value of the threshold {0.8}, the number of consistent superpixels decreased. Figure 10 shows the comparison of all four algorithms for three different values of the threshold. It demonstrates that among all algorithms GDS maintained more and produced more consistent superpixels then other algorithms for threshold value {0.3} and {0.5}. For threshold value, {0.8} and impulse noise level {0.05, 0.1, 0.15} GDS, TP, and BGD maintained and produced approximately the same number of consistent superpixels. FBS produces the lowest number of consistent superpixels among all algorithms for all three values of the threshold.

Moreover, to analyze the effect of Impulse noise on the performance of the superpixel segmentation algorithm, we quantitatively analyze the performances of all algorithms. Figure 11 shows the quantitative assessment of all algorithms by computing their ASA, USE, and BR. For ASA and BR the decreased values, where the increase in USE shows that all the algorithms decrease their performances for all levels of Impulse noise.

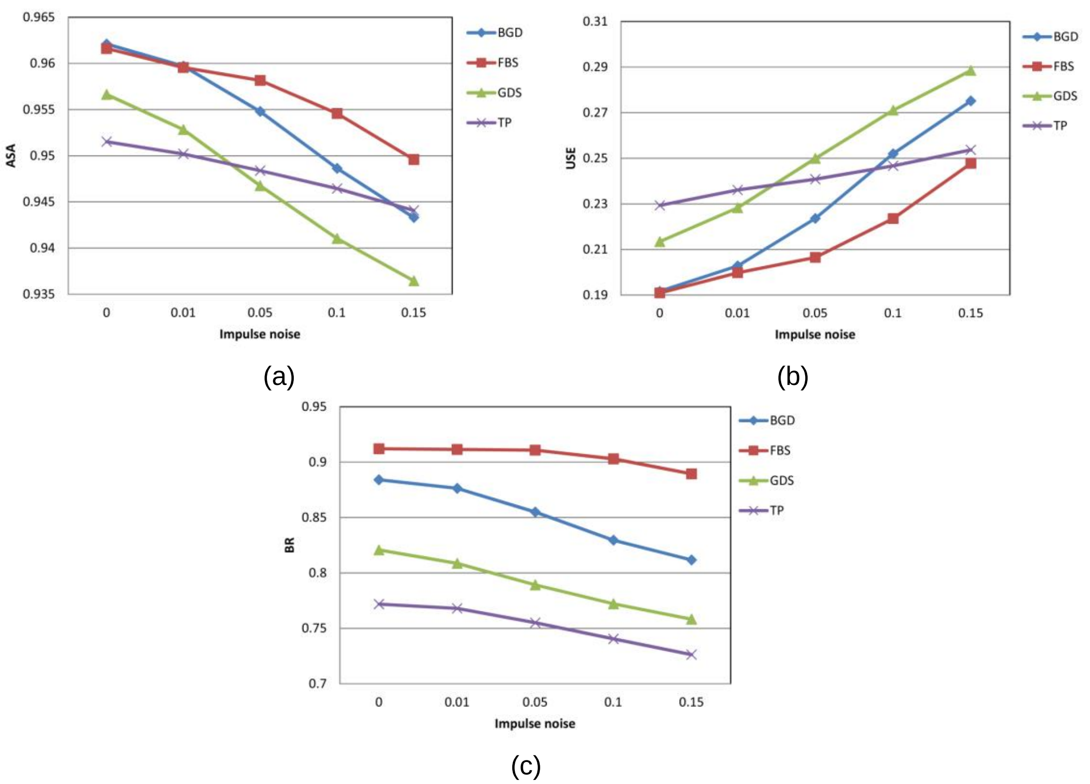


FIGURE 11: Quantitative evaluation results of BGD, FBS, GDS and TP for Impulse Noise {0.01, 0.05, 0.1, and 0.5} for 500 superpixels (a) ASA, (b) USE, and(c) BR.

4.3 Experiment Three (Impulse Noise + 2D Gaussian Blur)

To further increase the effect of noise and make the assessment process more robust, we combine the 2D Gaussian blur and impulse noise. We applied a combination of both the images and computed the results. We kept the values {(1+0.01), (2+0.05), (3+0.1), (4+0.15)}. The visual result of combined blur filter and impulse noise and computation of consistent superpixels is given in figure 12. The results obtained for consistent superpixels demonstrate that in the presence of both noises the image quality is highly reduced. The blur filter destroyed some of the essential information from the image plan whereas the impulse noise changes the pixels values randomly. The effect of both profoundly troubled the uniform intensity of pixels and deviated the superpixels to adhere to the object boundary well.

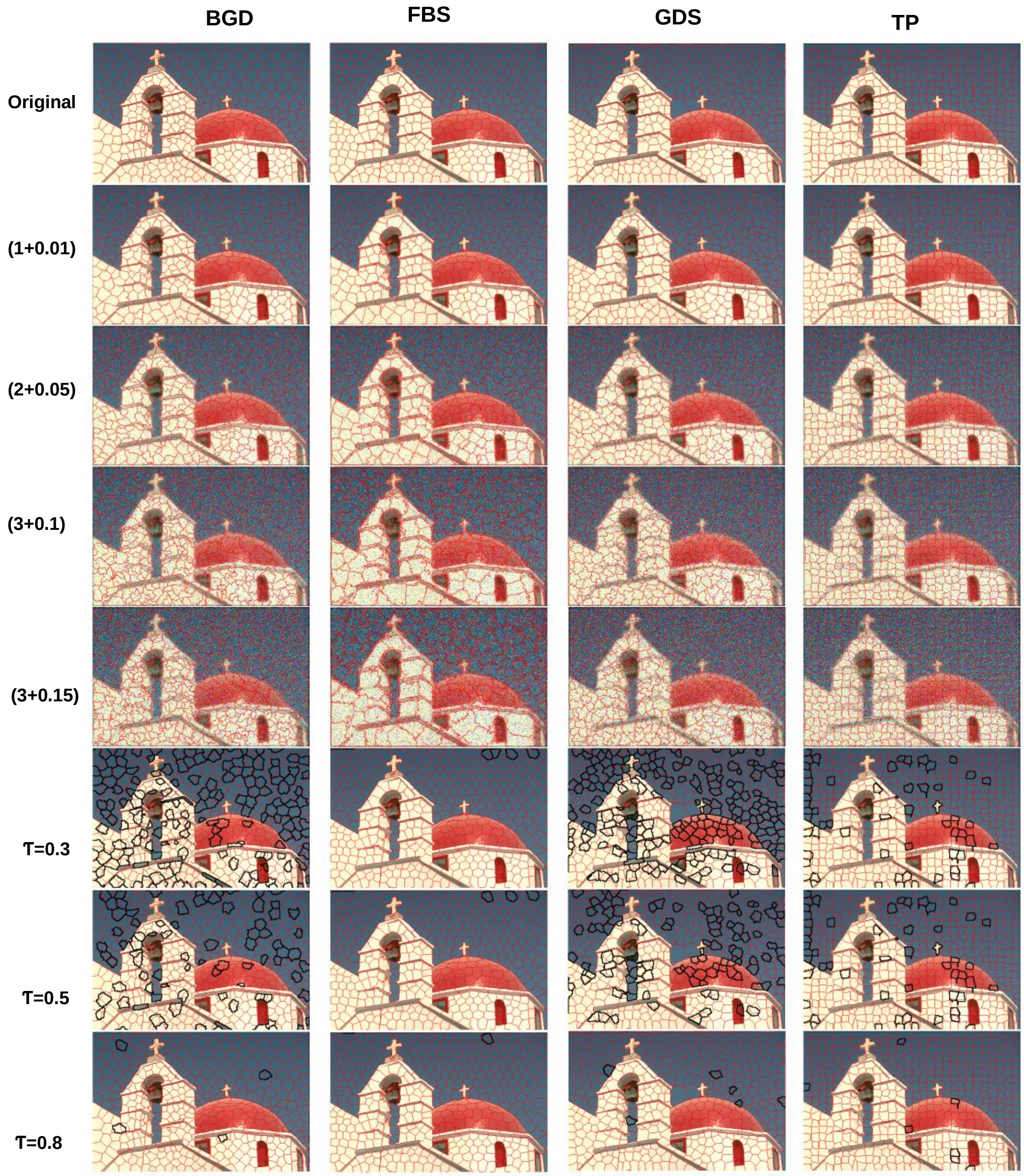
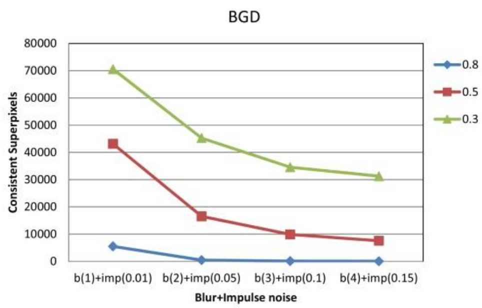
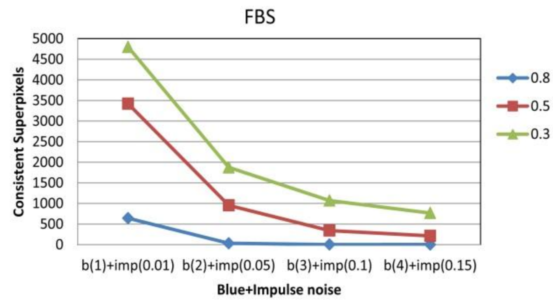


FIGURE 12: The visual result of the effect of blur filter + impulse noise for the values {(1+0.01), (2+0.05), (3+0.1), (4+0.15)} and the extraction of consistent superpixels for all threshold values.

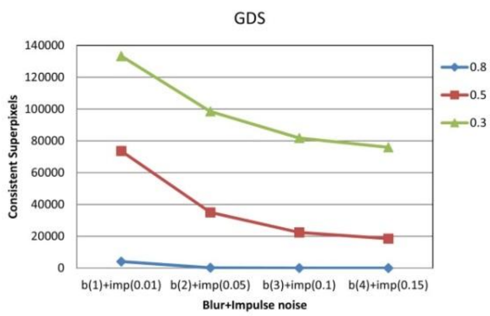
Figure 13 shows that each algorithm fails to produce consistent superpixels at threshold value {0.8}. We can see in figure 14 the GDS produces and maintained more consistent superpixels among other algorithms for threshold values {0.3, 0.5}, whereas at threshold value {0.8} all algorithm produced a deficient number of consistent superpixels. For all threshold values, FBS maintained the lowest number of consistent superpixels among all algorithms. The quantitative assessment of both blur filter and impulse noise is presented in figure 15. It shows that the noise affected each algorithm's performance and they somehow fail to show robustness to noise. Each algorithm reduces the ASA and BR while showing a high increase in USE.



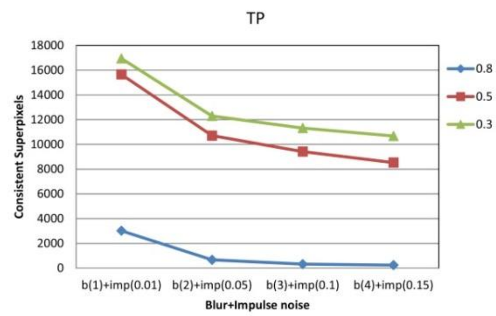
(a)



(b)

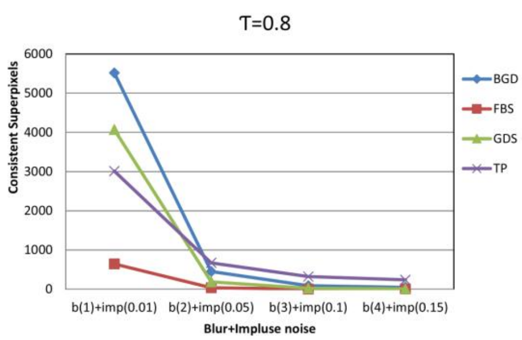


(b)

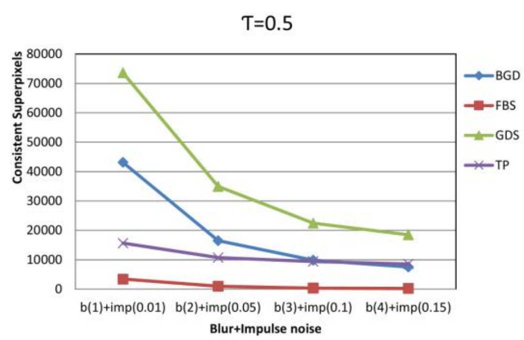


(d)

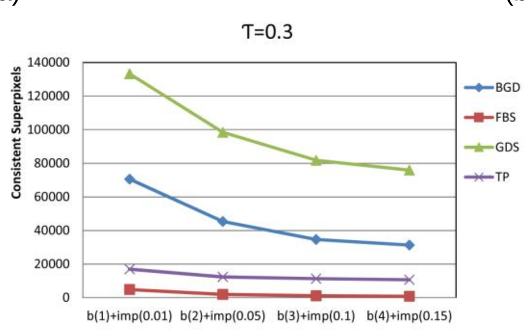
FIGURE 13: The number and the degradation of consistent superpixels for four levels of blur filter + impulse noise of each algorithm for the threshold {0.3, 0.5, and 0.8} values. (a) BGD, (b) FBS, (c) GDS, and (d) TP.



(a)



(b)



(c)

FIGURE 14: The superpixel consistency degradation comparison between the four superpixels algorithms for each value of the threshold for four levels of blur filter + impulse noise. (a) {0.3}, (b) {0.5}, and (c) {0.8}.

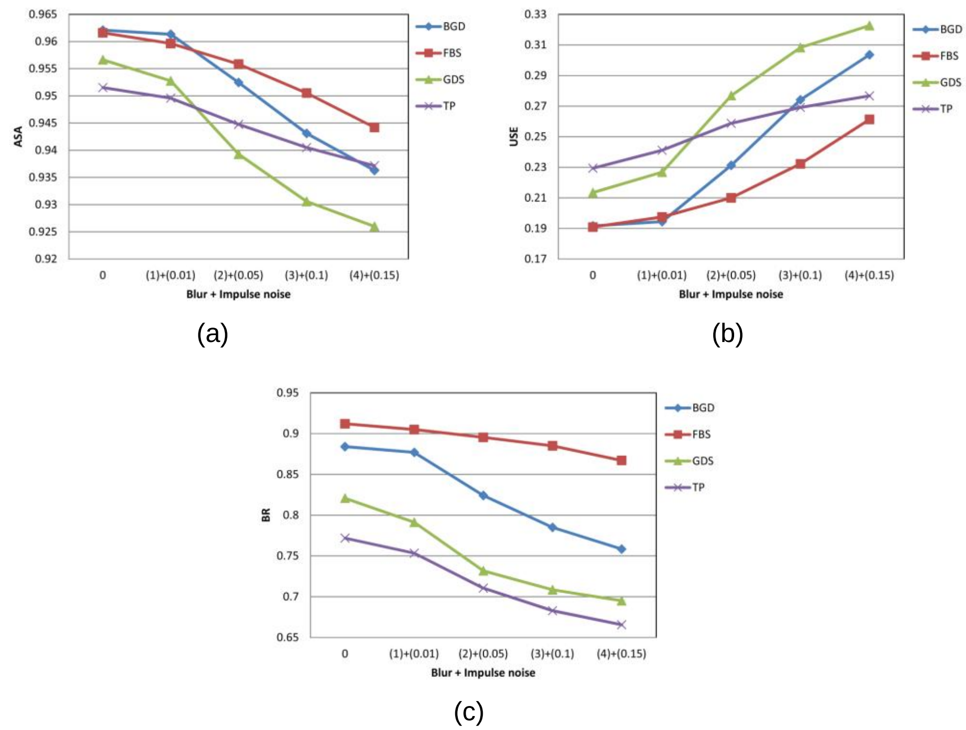


FIGURE 15: Quantitative evaluation results of BGD, FBS, GDS and TP for 2D Gaussian blur filter + Impulse noise {(1+0.01), (2+0.05), (3+0.1), (4+0.15)} for 500 superpixels (a) ASA, (b) USE, and (c) BR.

4.5 Discussion

Based on the results analyzed and presented earlier it is concluded that the proposed algorithm successfully evaluated the superpixel consistency over different levels of noise added to the set of images and extracted consistent superpixels. The overall results of all experiments indicate that all algorithms are sensitive to noise. The blur filter and impulse noise affected the performance of all algorithms. The algorithm fails to produce and maintain more consistent superpixels in the presence of noise. The number of consistent superpixels profoundly decreased with higher noise and the strict condition of the coefficient threshold. The quantitative results show the performance degradation of each algorithm in the presence of noise. To improve the robustness of algorithms to the noise, they can be enhanced to capture image texture well and extract the neighboring pixel information in a better way to produce good superpixels. With high levels of blur filter, the object boundaries become weak; algorithms can be improved to capture weak boundaries well based on the similarity of color and distance with neighboring pixels. The primary purpose of the proposed algorithm is to analyze the consistency of performance and robustness of superpixel segmentation algorithms under different conditions of noise. Other methods presented in the literature worked on the evaluation of superpixels using different properties of superpixels [38-42]. Most of these studies have taken out the evaluation process over original images which are noise free.

The study in ref [38] presented an extensive evaluation of superpixel segmentation algorithms on original images which are noise free. They ranked the evaluation of these algorithms by comparing their visual quality, algorithm runtime, and performance keeping the superpixel connectivity parameter relevant. The study in ref [39], also presented a comparative evaluation study of superpixel segmentation algorithms on noise-free images. In Ref [40] a metric is proposed to measure the compactness of superpixel which evaluates the superpixel segmentation algorithms to investigate the effects of compactness on specific applications. The study in ref [41] presented an evaluation of superpixels segmentation algorithms and proposed a metric for the regularity of superpixel in natural images. The study in ref [42] proposed an evaluation framework based on the color homogeneity, respect of image objects, and shape regularity. The major contribution of the present paper to the literature is that it proposes a way to evaluate superpixel segmentation algorithms under different levels of noise by assessing the consistency of superpixels.

In the future, we have the plan to work with more challenging strategies to analyze the superpixel algorithms for other image artifacts such as colorimetric changes and geometrical distortions, etc. Furthermore, we aspire to continue working on the categorization of consistent superpixels to separate the foreground and background superpixels of an image over common types of noise.

5. CONCLUSION

In this article, we presented a new idea of consistent superpixels. A superpixel is said to be consistent if it occupies the same object and does not deviate from its object boundaries over a certain level of noise applied to the same image. To accomplish this task, we introduced a new method that combines our new algorithm and existing superpixel segmentation algorithms. The new algorithm is intended to evaluate the consistency of superpixels for the selected superpixel segmentation algorithms in the presence of noise using the Jaccard Similarity coefficient (JSC). The superpixel segmentation algorithms include bilateral geodesic distance, superpixels via geodesic distance, Flooding based superpixels generation, and Turbopixel. The 2D Gaussian blur, impulse noise and a combination of both are chosen to corrupt the images. A coefficient threshold in the range of {0 to 1} is defined which controls the degree of similarity. We chose three different thresholds values {0.3, 0.5, and 0.8} to compute various results. The increment of threshold values restricts the criteria of similarity to work out the consistent superpixels. The consistency evaluation results obtained by the proposed algorithm demonstrate that the noise affects the performance of each algorithm and they somehow failed to produce and maintain the consistent superpixels for each noise level. The number of consistent superpixels decreased gradually with the increase of noise. To further explore the effect of noise over the results of superpixel segmentation algorithms quantitatively, we used the performance parameters including Achievable Segmentation Accuracy (ASA), Under Segmentation Error (USE) and Boundary Recall (BR). The results of these parameters showed that the effect of blur and impulse noise profoundly affected and reduced the performance of all algorithms and they failed to maintain the consistency of superpixels with the increase of noise. Furthermore, we aspire to continue working on the categorization of consistent superpixels to separate the foreground and background superpixels of an image over common types of noise.

Conflicts of Interest: The author(s) declare(s) that there is no conflict of interest regarding the publication of this paper.

Funding: This work was supported in part by the NSFC Joint Fund with Zhejiang Integration of Informatization and Industrialization under Key Project, No.U1609218, NSFC Fund (No.61772312), the key research and development project of Shandong province (2017GGX10110).

6. REFERENCES

[1] REN, X. (2003). Learning a classification model for segmentation. *Proc. 9th ICCV, 2003*, *1*, 10-17.

[2] Shen, J., Du, Y., Wang, W., & Li, X. (2014). Lazy random walks for superpixel segmentation. IEEE Transactions on Image Processing, 23(4), 1451-1462.

[3] Kaufhold, J., Collins, R., Hoogs, A., & Rondot, P. (2006, August). Recognition and segmentation of scene content using region-based classification. In 18th International Conference on Pattern Recognition (ICPR'06) (Vol. 1, pp. 755-760). IEEE.

[4] Dong, X., Shen, J., Shao, L., & Van Gool, L. (2016). Sub-Markov random walk for image segmentation. IEEE Transactions on Image Processing, 25(2), 516-527.

[5] Peng, B., Zhang, L., & Zhang, D. (2011). Automatic image segmentation by dynamic region merging. IEEE Transactions on image processing, 20(12), 3592-3605.

[6] Zhang, L., Gao, Y., Xia, Y., Lu, K., Shen, J., & Ji, R. (2014). Representative discovery of structure cues for weakly-supervised image segmentation. IEEE Transactions on Multimedia, 16(2), 470-479.

[7] Wang, W., Shen, J., Li, X., & Porikli, F. (2015). Robust video object cosegmentation. IEEE Transactions on Image Processing, 24(10), 3137-3148.

[8] Xiang, D., Tang, T., Zhao, L., & Su, Y. (2013). Superpixel generating algorithm based on pixel intensity and location similarity for SAR image classification. IEEE Geoscience and Remote Sensing Letters, 10(6), 1414-1418.

[9] Cheng, J., Liu, J., Xu, Y., Yin, F., Wong, D. W. K., Tan, N. M., ... & Wong, T. Y. (2013). Superpixel classification based optic disc and optic cup segmentation for glaucoma screening. IEEE transactions on Medical Imaging, 32(6), 1019-1032.

[10] Mahmood, A., Mian, A., & Owens, R. (2014). Semi-supervised spectral clustering for image set classification. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 121-128).

[11] Chai, Y., Lempitsky, V., & Zisserman, A. (2011, November). Bicos: A bi-level co-segmentation method for image classification. In 2011 International Conference on Computer Vision (pp. 2579-2586). IEEE.

[12] Wang, Y., & Zhao, Q. (2015, June). Superpixel tracking via graph-based semi-supervised SVM and supervised saliency detection. In 2015 IEEE International Conference on Multimedia and Expo (ICME) (pp. 1-6). IEEE.

[13] Dong, X., Shen, J., & Shao, L. (2017). Hierarchical superpixel-to-pixel dense matching. IEEE Transactions on Circuits and Systems for Video Technology, 27(12), 2518-2526.

[14] Pan, X., Zhou, Y., Li, F., & Zhang, C. (2017). Superpixels of RGB-D images for indoor scenes based on weighted geodesic driven metric. IEEE transactions on visualization and computer graphics, 23(10), 2342-2356.

[15] Wang, W., Shen, J., Yang, R., & Porikli, F. (2018). Saliency-aware video object segmentation. IEEE transactions on pattern analysis and machine intelligence, 40(1), 20-33.

[16] Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., & Süsstrunk, S. (2012). SLIC superpixels compared to state-of-the-art superpixel methods. IEEE transactions on pattern analysis and machine intelligence, 34(11), 2274-2282.

[17] Wang, J., & Wang, X. (2012). VCells: Simple and efficient superpixels using edge-weighted centroidal Voronoi tessellations. IEEE Transactions on pattern analysis and machine intelligence, 34(6), 1241-1247.

[18] Liu, Y. J., Yu, C. C., Yu, M. J., & He, Y. (2016). Manifold SLIC: A fast method to compute content-sensitive superpixels. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 651-659).

[19] Shen, J., Hao, X., Liang, Z., Liu, Y., Wang, W., & Shao, L. (2016). Real-time superpixel segmentation by DBSCAN clustering algorithm. IEEE Transactions on Image Processing, 25(12), 5933-5942.

[20] Van den Bergh, M., Boix, X., Roig, G., & Van Gool, L. (2015). Seeds: Superpixels extracted via energy-driven sampling. International Journal of Computer Vision, 111(3), 298-314.

[21] Mori, G. (2005, October). Guiding model search using segmentation. In Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1 (Vol. 2, pp. 1417-1423). IEEE.

[22] Malik, J., Belongie, S., Leung, T., & Shi, J. (2001). Contour and texture analysis for image segmentation. International journal of computer vision, 43(1), 7-27.

[23] Brekhna, B., Mahmood, A., Zhou, Y., & Zhang, C. (2017). Robustness analysis of superpixel algorithms to image blur, additive Gaussian noise, and impulse noise. Journal of Electronic Imaging, 26(6), 061604.

[24] Levinshtein, A., Stere, A., Kutulakos, K. N., Fleet, D. J., Dickinson, S. J., & Siddiqi, K. (2009). Turbopixels: Fast superpixels using geometric flows. IEEE transactions on pattern analysis and machine intelligence, 31(12), 2290-2297.

[25] Zhou, Y., Pan, X., Wang, W., Yin, Y., & Zhang, C. (2017). Superpixels by bilateral geodesic distance. IEEE Transactions on Circuits and Systems for Video Technology, 27(11), 2281-2293.

[26] Pan, X., Zhou, Y., Zhang, C., & Liu, Q. (2014, October). Flooding based superpixels generation with color, compactness and smoothness constraints. In 2014 IEEE International Conference on Image Processing (ICIP) (pp. 4432-4436). IEEE.

[27] Rajaraman, A., & Ullman, J. D. (2010). Finding similar items. Mining of Massive Datasets, 77, 73-80.

[28] Niwattanakul, S., Singthongchai, J., Naenudorn, E., & Wanapu, S. (2013, March). Using of Jaccard coefficient for keywords similarity. In Proceedings of the international multiconference of engineers and computer scientists (Vol. 1, No. 6, pp. 380-384).

[29] Aydin, B., & Tamir, M. (2015). U.S. Patent Application No. 14/084,903.

[30] Bank, J., & Cole, B. (2008). Calculating the jaccard similarity coefficients with map reduce for entity pairs in wikipedia. Wikipedia Similarity Team, 1-18.

[31] Chen, D. J., Chen, H. T., & Chang, L. W. (2016, September). Fast defocus map estimation. In 2016 IEEE International Conference on Image Processing (ICIP) (pp. 3962-3966). IEEE.

[32] Indu, S., & Ramesh, C. (2007, March). A noise fading technique for images highly corrupted with impulse noise. In 2007 International Conference on Computing: Theory and Applications (ICCTA'07) (pp. 627-632). IEEE.

[33] Ibrahim, H., Neo, K. C., Teoh, S. H., Ng, T. F., Chieh, D. C. J., & Hassan, N. N. (2012). Impulse noise model and its variations. International Journal of Computer and Electrical Engineering, 4(5), 647.

[34] Wang, P., Zeng, G., Gan, R., Wang, J., & Zha, H. (2013). Structure-sensitive superpixels via geodesic distance. International journal of computer vision, 103(1), 1-21.

[35] Martin, D., Fowlkes, C., Tal, D., & Malik, J. (2001, July). A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. Vancouver:: Iccv.

[36] Zhang, Y., Li, X., Gao, X., & Zhang, C. (2017). A simple algorithm of superpixel segmentation with boundary constraint. IEEE Transactions on Circuits and Systems for Video Technology, 27(7), 1502-1514.

[37] Stutz, D., Hermans, A., & Leibe, B. (2018). Superpixels: An evaluation of the state-of-the-art. Computer Vision and Image Understanding, 166, 1-27.

[38] Neubert, P., & Protzel, P. (2012, January). Superpixel benchmark and comparison. In Proc. Forum Bildverarbeitung(Vol. 6).

[39] Schick, A., Fischer, M., & Stiefelhagen, R. (2012, November). Measuring and evaluating the compactness of superpixels. In Proceedings of the 21st international conference on pattern recognition (ICPR2012) (pp. 930-934). IEEE.

[40] Wang, M., Liu, X., Gao, Y., Ma, X., & Soomro, N. Q. (2017). Superpixel segmentation: a benchmark. Signal Processing: Image Communication, 56, 28-39.

[41] Giraud, R., Ta, V. T., & Papadakis, N. (2017). Evaluation framework of superpixel methods with a global regularity measure. Journal of Electronic Imaging, 26(6), 061603.