drones

MDPI

Article

Graph-Based Image Segmentation for Road Extraction from Post-Disaster Aerial Footage

Nicholas Paul Sebasco [1] and Hakki Erhan Sevil [2,*]

[1] Mathematics and Statistics, University of West Florida, Pensacola, FL 32514, USA
[2] Intelligent Systems & Robotics, University of West Florida, Pensacola, FL 32514, USA
* Correspondence: hsevil@uwf.edu

Abstract: This research effort proposes a novel method for identifying and extracting roads from aerial images taken after a disaster using graph-based image segmentation. The dataset that is used consists of images taken by an Unmanned Aerial Vehicle (UAV) at the University of West Florida (UWF) after hurricane Sally. Ground truth masks were created for these images, which divide the image pixels into three categories: road, non-road, and uncertain. A specific pre-processing step was implemented, which used Catmull–Rom cubic interpolation to resize the image. Moreover, the Gaussian filter used in Efficient Graph-Based Image Segmentation is replaced with a median filter, and the color space is converted from RGB to HSV. The Efficient Graph-Based Image Segmentation is further modified by (i) changing the Moore pixel neighborhood to the Von Neumann pixel neighborhood, (ii) introducing a new adaptive isoperimetric quotient threshold function, (iii) changing the distance function used to create the graph edges, and (iv) changing the sorting algorithm so that the algorithm can run more effectively. Finally, a simple function to automatically compute the k (scale) parameter is added. A new post-processing heuristic is proposed for road extraction, and the Intersection over Union evaluation metric is used to quantify the road extraction performance. The proposed method maintains high performance on all of the images in the dataset and achieves an Intersection over Union (IoU) score, which is significantly higher than the score of a similar road extraction technique using K-means clustering.

Keywords: efficient graph-based image segmentation; road extraction; computer vision; post-disaster aerial footage

Citation: Sebasco, N.P.; Sevil, H.E. Graph-Based Image Segmentation for Road Extraction from Post-Disaster Aerial Footage. *Drones* **2022**, *6*, 315. https://doi.org/10.3390/drones6110315

Academic Editors: Chi Chen, San Jiang, Xijiang Chen, Mao Tian, Jianping Li and Jian Zhou

Received: 19 August 2022
Accepted: 24 October 2022
Published: 26 October 2022

1. Introduction

The International Disaster Database, Université Catholique de Louvain—Brussels—Belgium, shows that roughly 300–400 natural disasters have occurred annually since 2000 [1]. Natural disasters such as hurricanes, earthquakes, wildfires, and volcanic activity can cause extreme destruction of property and infrastructure, personal injury, and death. A particular threat to Floridians and other coastal states is hurricanes. It has been projected that the combined forces of coastal development and climate change will increase the amount of hurricane damage in the future [2–6]. One of the undeniable aftermaths of hurricanes is the damage and obstructions to roadways, which can inhibit emergency vehicles from rescuing people. These obstructions may include: flooding, uprooted vegetation, sinkholes, structural damage to the road, and other debris. Injuries that are sustained from a hurricane impact may be life-threatening and require immediate emergency medical services (EMS) [7]. An EMS driver does not have time to deal with blocked or defective roadways. It is imperative that an EMS driver takes the fastest traversable route in order to maximize the probability of saving a person's life.

The use of satellite imagery and unmanned autonomous vehicles equipped with cameras permits the mapping and exploration of territories that may be inaccessible and unsafe. For instance, a swarm of unmanned aerial vehicles (UAVs) can be deployed after a disaster to assess damage to critical infrastructure. Similarly, a UAV can stream video to a high-performance server with dedicated algorithms used to assess infrastructure damage and the traversability of roads after a major hurricane. Mainly, the first step in assessing the traversability of roads is to identify and extract the roads from images or video streams. The difficulty of extracting continuous road segments can indicate that the road contains debris and is potentially not traversable. After extraction, road segments can be fed into classification algorithms to further assess traversability.

In this research, high-resolution UAV imagery was taken from the aftermath of hurricane Sally at the University of West Florida (UWF). The images have been stored in a custom dataset. Ground truth masks were created for images in this dataset so that the quality of subsequent road extraction on these images could be assessed. A new approach was developed for road extraction using a particular set of pre-processing techniques, a modified version of Efficient Graph-Based image segmentation, and a novel post-processing heuristic.

The original contributions of this study can be summarized as: (i) the use of custom aerial footage data taken from the UWF campus after hurricane Sally, (ii) the development of ground truth masks to measure the quality of road extraction, (iii) the development of a new post-processing heuristic which can identify and connect segments of the road after image segmentation and help address one of the shortcomings of Efficient Graph-Based image segmentation, and (iv) performing a hue and saturation analysis for the segments neighboring road segments so that the segments that are similar enough can be merged. The intention of this paper is to present our modified Efficient Graph-Based image segmentation framework design with analysis and to show proof-of-concept results of the entire pipeline on custom aerial footage data recorded after hurricane Sally in 2020. Further analysis, such as robustness to noise and comparison with other methods, are not included in the scope of the originally intended contribution in this article, and they are left for a future work. Additionally, the planned future work also includes another important aspect of comprehensive computational time analysis.

The remainder of the paper is organized as follows. Related work on image segmentation is discussed in Section 2, followed by the details of Efficient Graph-Based Image Segmentation in Section 3. Road Extraction is presented in Section 4, followed by the results and discussion in Section 5. Conclusions are presented in Section 6.

2. Related Work

Image segmentation is a computer vision technique that divides an image into several specific components with unique attributes. Modern image segmentation algorithms are usually categorized as semantic (pixel-wise association with class label), instance (accurate delineation of each object in an image), and panoptic (assigning class labels to objects in images). Early image segmentation methods include threshold-based, centroid-based, density-based, graph-based, fuzzy theory-based, hierarchical, and distribution-based methods. The goal of this section is to outline some of the most popular techniques that are well-suited to solving the problems of road extraction and traversability assessment. A comprehensive survey of clustering algorithms can be found in [8]; furthermore, a survey of clustering algorithms used in Image segmentation can be found in [9].

Centroid-based image segmentation and clustering involve finding an arbitrary number of centroids in a dataset and then grouping together data points with the smallest distance to a particular centroid. One of the most algorithms, K- means, is a centroid-based clustering algorithm [10]. However, K-means has significant drawbacks as it lacks flexibility in the shape of the clusters, and there are no probabilities associated with cluster assignments [11]. In other studies, the K-means algorithm is used to segment images; of particular relevance, in [12], the road is extracted from images using the K- means algorithm and morphological operations. The K-means algorithm is used to generate segments, and then a simple geometric post-processing heuristic is applied, which attempts to identify the segments that are road.

Hierarchical clustering has also been used to segment images, both as a standalone segmentation method and in combination with other segmentation methods. In [13], images were first pre-segmented with the well-known graph-based segmentation method of Normalized Cuts. Afterward, hierarchical segmentation was applied. In [14], the study proposes an agglomerative hierarchical clustering-based high-resolution remote sensing image segmentation algorithm. The algorithm showed favorable results over standard K-means image segmentation. In [15], an agglomerative clustering technique within a feature matrix is presented. It is shown to compare favorably with the image segmentation algorithm and has the advantage of not needing to know the number of clusters in advance.

The fuzzy sets-based method is another approach presented in the literature for image segmentation [16]. In [17], the authors developed a fuzzy system to identify roads in aerial images with five fuzzy membership functions (Good values, Up-Probable values, Down-Probable values, Up-Bad values, and Down-Bad values), which help classify pixels as either being road or non-road with a certain probability. Some limitations of this method are that it is designed for 8-bit images, cannot handle regions of road less than 5 pixels in width, relies on hard-coded values, and cannot handle shadows. In [18], the authors develop an approach based on Fuzzy C-Means (FCM) to extract roads from foggy aerial images. The authors point out that aerial image quality is susceptible to weather conditions, variations in lighting, and properties of the ground. Fog can obscure the gray scale difference between road and non-road regions in an image; thus, a defogging procedure should be applied if fog is present. Other segmentation methods include density-based algorithms [19,20], Mean Shift-based methods [21–26], and Gaussian mixture models (GMM)-based approaches [27–32].

In graph-based image segmentation methods, images must first be converted into a graph where each pixel is a node in the graph. Next, a decision must be made as to which edges to add to the graph. One option is to add edges between a pixel and all other pixels that are in its neighborhood. Two obvious neighborhoods that can be used are the Von Neumann 4-pixel neighborhood and the Moore 8-pixel neighborhood [33,34]. In [35], Normalized Cuts is used to segment high-resolution satellite images. Although the experiments showed “good operability,” the slow running time of the algorithm was discouraging. In [36], the Normalized Cuts method was used for the detection of roads in aerial images. The aerial image is first segmented into 20 components. Next, color and shape are used to determine which components are roads.

In terms of road extraction, there are several studies with various methods presented in the literature, some of them worth mentioning here. The U-net neural network is one of the methods presented for road extraction [37]. In [38], the authors propose an object-based classification approach for automatic road detection from orthophoto images. Additionally, a deep convolutional neural network (CNN)-based framework for road detection and segmentation from aerial images is presented in [39]. Similarly, another advanced algorithm, namely Fully Convolutional Network (FCN) and conditional Generative Adversarial Networks (GAN), are used for road extraction using RGB images captured by a UAV [40]. In [41], the author used the Mask-R-CNN neural network to detect flood water on roads, and in [42], a Mask-R-CNN approach was used to segment images in an effort to monitor the road surface condition. There are numerous other high-performing neural network architectures used for image segmentation presented in the literature. The lack of availability of large labeled datasets and the unsuitability for sparse scenes limit the effectiveness of neural network-based methods [43]. A comparison of a large number of image segmentation methods, including neural network and non-neural network-based methods, is conducted in [43].

There are also other notable techniques that have been used to segment images. In the Affinity Propagation method, real number messages are exchanged between data points until a high-quality set of clusters gradually emerges [44,45]. In [46], the authors present MRF (Markov Random Fields) models that can accurately capture road network topologies in synthetic aperture radar (SAR) images. In [47], the author proposes a genetic algorithm to find the initial contour points so that an Active Contour Model can be used to find the road in images. In [48], ant colony optimization is used for parameter selection of fuzzy object-based image analysis to extract the roads from remotely sensed images.

Although significant research has been conducted on image segmentation, road extraction based on image segmentation after a major disaster using aerial footage is still an open area of research. Little attention has been imparted to the development of custom pre- and post-processing steps. In this paper, an Efficient Graph-Based Image Segmentation approach with improvements performed via developed algorithms is introduced. This approach shows advantages over K-mean clustering with a higher Intersection over Union (IoU) score.

3. Efficient Graph-Based Image Segmentation

The image segmentation approach that was used in this study to isolate segments of traversable road is based on the Efficient Graph-Based image segmentation algorithm. In this section, an overview of Efficient Graph-Based image segmentation (EGS) and the modifications that are made is presented. The word *efficient* in the name of the algorithm refers to the algorithm's fast running time.

Consider an undirected graph $G = (V, E)$, with V being a set of vertices of and E being the set of edges consisting of two vertices (v_i, v_j) and a weight w_{ij}. In EGS, edge weights represent how dissimilar two pixels in an image are. Similarity can be measured in terms of pixel attributes, including hue and intensity, and a distance function such as the L^2 norm can be used for that purpose. EGS constructs a segmentation S by partitioning V into k connected components $C \in S$, $|S| = k$ (|•| represents the cardinality, numbers of elements in that set). Next, the issue of deciding if there is evidence for a boundary between components is considered. EGS introduced a novel pairwise region comparison predicate that measures the similarity between boundary elements of two components relative to the similarity inside of each component. The difference inside of each component is quantified by $Int(C)$, which is the largest edge weight in the minimum spanning tree (MST) of the component [34].

$$Int(C) = max(w(e)),\ e \in \text{MST}(C) \tag{1}$$

In Equation (1), if e is a graph's edge, $w(e)$ returns the weight of that edge. The difference between components is captured by the term $Dif(C_1, C_2)$, which represents the smallest edge weight connecting the components [34].

$$Dif(C_1, C_2) = min(w(e)) \mid v_i, v_j \in e,\ \ v_i \in C_1,\ v_j \in C_2 \tag{2}$$

In Equation (2), $min(w(e))$ refers to this minimum edge weight, that is, the smallest (minimum) edge weight that exists connecting two distinct components C_1 and C_2. The minimum edge weight must be used because using some quantile, such as the median, produces an NP-hard computational problem. The region comparison predicate of EGS is defined to be [34]:

$$D(C_1, C_2) = \begin{cases} True, & Dif(C_1, C_2) > MInt(C_1, C_2) \\ False, & Otherwise \end{cases} \tag{3}$$

where the minimum internal difference, $MInt(C_1, C_2)$, is defined as [34]:

$$MInt(C_1, C_2) = min(Int(C_1) + \tau(C_1),\ Int(C_2) + \tau(C_2)) \tag{4}$$

where $\tau(C)$ is a threshold function used to control how different two components must be from each other in order for a boundary to exist between them. In Equation (4), $Int(C)$ is the largest edge weight in the MST. The default threshold function given in the EGS algorithm is defined as [34]:

$$\tau(C) = \frac{k}{|C|} \tag{5}$$

where k is the lone parameter of EGS and can be thought of as the affinity for larger components or, more succinctly, as a parameter to control scale. Any function that is not negative can be used as $\tau(C)$.

The EGS algorithm works by first constructing a graph, G, from the image with n vertices and m edges. The output of the algorithm is a segmentation of the vertices into k components. The following steps define the algorithm [34]:

1. Sort the edges by non-decreasing edge weight.
2. Start with segmentation S^0 where each vertex is its own component.
3. Construct the q-th segmentation S^q from S^{q-1} by doing the following. Let $v_i,\ v_j$ be the vertices connected by the q-th edge. If $v_i,\ v_j$ are not in the same components $C_i,\ C_j\ \in S^{q-1}\ \big|\ C_1\ \neq C_2$ and the weight is small compared to the internal difference of $C_i,\ C_j$ then join the two components; otherwise, take no action.
4. Repeat (3) for each $q = 1,\ \ldots,\ m$.
5. Return the final segmentation as $S = S_m$.

There are several highly appealing features of the EGS algorithm. Reference [34] emphasizes that in order for a segmentation algorithm to be of practical real-time use, it should run in time approximately linear in the number of image pixels. The default implementation given by the author runs in $\mathrm{O}(n * log(n))$ time where n is the total number of pixels in the image; however, this can be improved to $\mathrm{O}(n + k)$ by using integer edge weights and a constant time sorting algorithm. For images, the use of integer edge weights is a great option because the color of each pixel in an image is usually represented by one or more integers. Further details about the implementation of EGS on aerial images can be found in [49].

The results section (Section 5) will show the application of EGS in road extraction that the higher number of unique edge weights generated from the Euclidean distance is not worth the sacrifice of computational speed. A lookup table can be pre-computed to increase computational time efficiency if needed. Alternatively, using a sorting optimization can decrease the computational time. However, even without using any sorting optimization, the EGS method is incredibly fast and possibly the fastest method that can be used to segment an image while considering all of the pixels in the image.

In this study, original contributions include the changes that were made to the default algorithm as: (*i*) a simple way of automatically computing the parameter k and (*ii*) a new adaptive isoperimetric quotient threshold function designed for road extraction. The basis for this formula is that k should increase with image size to prevent tiny segments from forming and attenuate the burden of post-processing.

$$k(z, n, m) = \frac{\alpha}{z}\sqrt{nm} \tag{6}$$

The variable z captures the elevation of the UAV. The constant α is some real number that controls the degree to which the elevation of the UAV affects the size of the road segments. Presumably, the higher the elevation, the smaller the road segments will be. In this study, $\frac{\alpha}{z}$ was set equal to 2.5. Additionally, a new threshold function is introduced in this study, which has been given the name *adaptive isoperimetric quotient threshold function*

$$\begin{gathered} p = 2\pi r \ \rightarrow r = \tfrac{p}{2\pi} \\ A_{iso} = \pi\left(\tfrac{p}{2\pi}\right)^2 = \tfrac{p^2}{4\pi} \\ \tau(\mathrm{C}) = \tfrac{A_{iso}k}{|\mathrm{C}|^2} \end{gathered} \tag{7}$$

where p is the perimeter of a segment, r is the radius of a circle that encloses the segment, and A_{iso} is the isoperimetric area of the circle that encloses the segment. The key feature of this threshold function is that more compact circular components are less favorable than long and skinny geometries. This provides a great improvement in the algorithm's ability to extract paths and roads. Further modifications to the EGS algorithm can be found in [49].

4. Road Extraction

4.1. Data

The dataset consists of high-resolution aerial images captured of the UWF campus after hurricane Sally [49–51]. The original dataset consisted of both images and videos taken by the UAV. Out of these images, the 10 images that achieved the best job of capturing the road were selected. Additionally, 21 screenshots from the UAV videos were taken. These images were combined into a validation dataset, and ground truth masks were generated for each image. The ground truth masks were created by assigning each pixel in the image to one of three categories: road, non-road, or uncertain. Each category was associated with a color: road with black, non-road with green, and uncertain with red. Each image is approximately 5472 pixels by 3648 pixels. The "jpeg" file format of the source images was converted to the non-lossy "png" file format. An image from the dataset and its corresponding mask can be seen in Figure 1. In the right image (Figure 1), the black region is road, the green region is non-road, and the red region is a region of uncertainty.

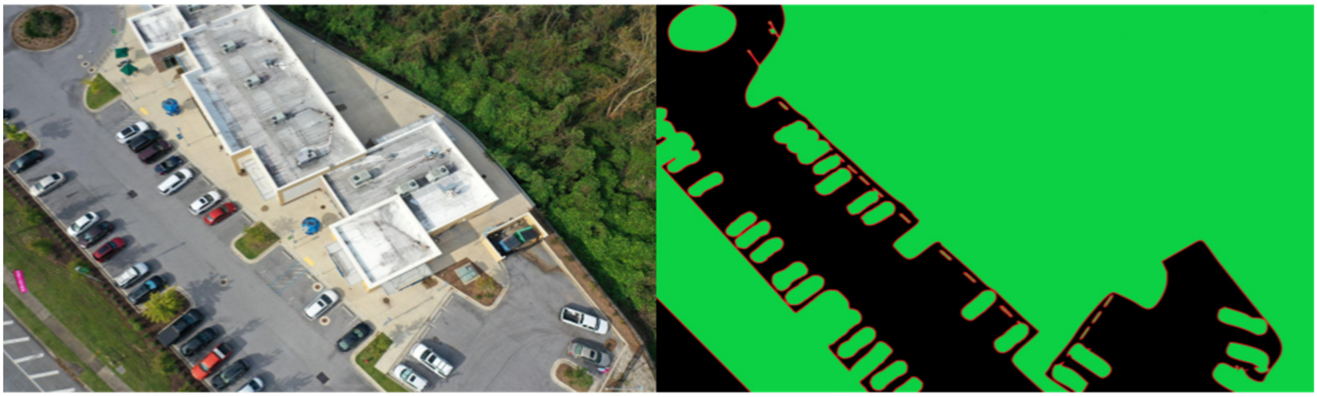

Figure 1. A sample frame from aerial footage and its corresponding mask.

There are several interesting properties of the road in these images. One interesting feature of the roads is that they are often located next to trees or large structures such as buildings or poles. These objects can cast large shadows on the road, which creates a variation in the color of the road. The road is generally located several inches below the surrounding land, which is separated by a curb. Therefore, during a storm, the road serves as a repository for dirt, sand, mud, leaves, and a myriad of other substances from the surrounding land. Additionally, car tires can end up transferring dirt and other matter across the road leading to discolorations and tire tracks. The UWF campus is a heavily wooded environment. In particular, the longleaf pine grows all over the campus, and the needles turn brown annually and shed all over the surrounding road. Similar to other urban roads, the roads on the UWF campus have many painted markings and symbols. All of these properties have a profound impact on road extraction and traversability assessment; they tend to obfuscate these tasks. Figures 2 and 3 highlight various properties of the roads in the dataset. In Figure 2, it can be seen from the top image that different parts of road in an aerial image can vary drastically depending on various effects, such as dirt, mud, and faded parking line marks. The bottom three images in Figure 2 show that trees can partially block large chunks of road. Determining whether or not these trees lie on the road and lead to non-traversability is an important question.

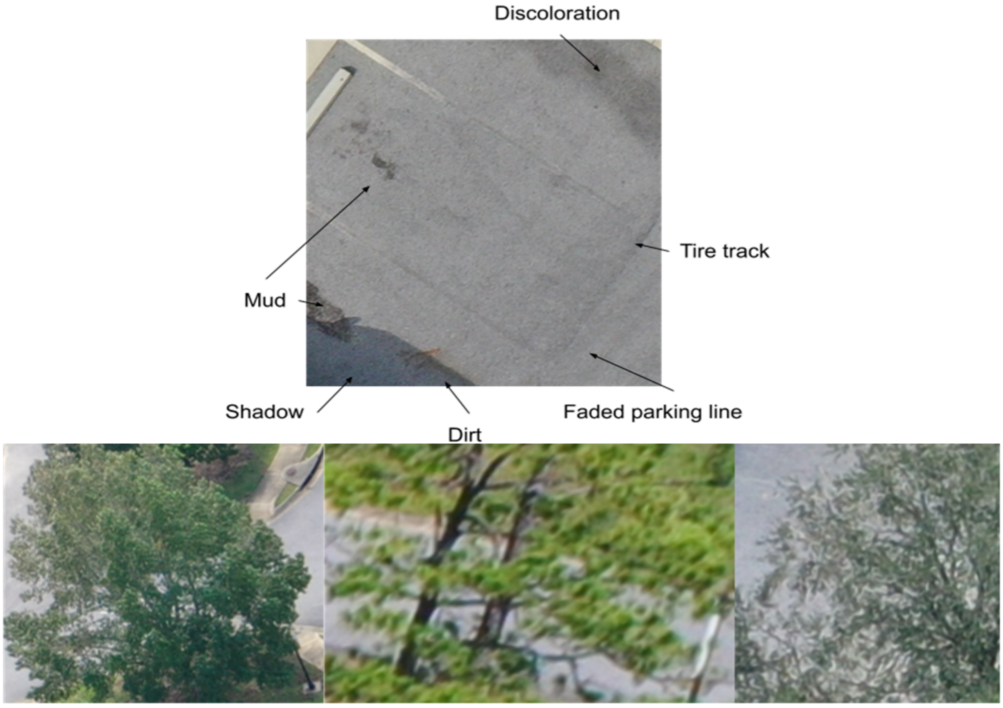

Figure 2. Top row: example of various effects on the road surface. Bottom row: examples of trees blocking view of the road in aerial footage.

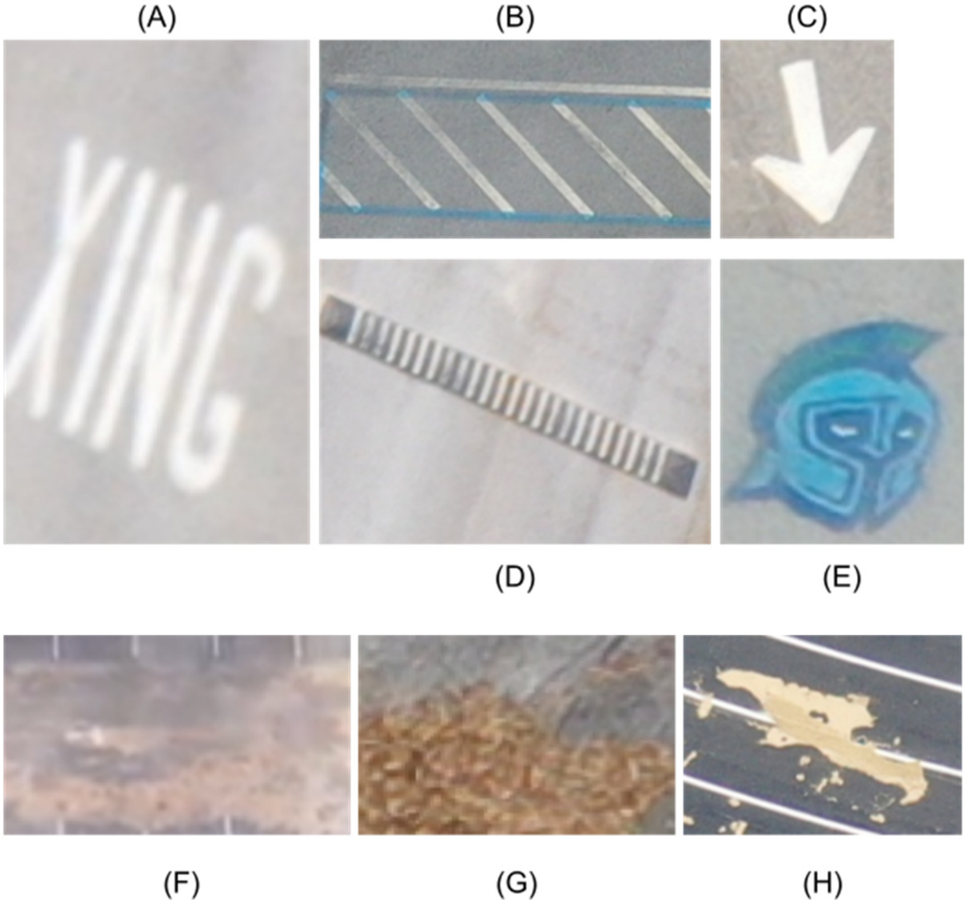

Figure 3. (**A**–**H**) Various road markings, pine needles, sand, tree leaves, dirt, and other debris can be found on the road in the UAV images.

4.2. Pre-Processing

This section introduces a high-performing pre-processing stack that aids in quality road extraction. First, the image is resized using bicubic Catmull–Rom interpolation. Next, a median filter with a kernel size proportional to the ratio of the original image size to the resized image is applied. Finally, the color space is transformed from RGB to HSV. This pre-processing stack was chosen in an effort to maximize the performance of road extraction.

4.2.1. Catmull–Rom Bicubic Interpolation

The first step in the pre-processing stack is Catmull–Rom bicubic interpolation. All subsequent operations (pre-processing, EGS, and post-processing) run more effectively after this step. Although the nearest neighbor's interpolation runs slightly better, the performance-to-quality tradeoff observed from [52] is reason enough to stick with Catmull–Rom.

Catmull–Rom resizing works by generating an interpolating spline curve to find the pixel values in the down-sampled image. A Catmull–Rom spline with centripetal parameterization can be computed as follows: let $p = [x\ y]^T$ denote some point. For a curve segment C_i defined by control points $p_{i-1},\ p_i,\ p_{i+1},\ p_{i+2}$ and knot sequence $\tau_{i-1},\ \tau_i,\ \tau_{i+1},\ \tau_{i+2}$. The Catmull–Rom spline can be plotted with the following function [53]:

$$\begin{aligned} C_i &= \tfrac{\tau_{i+1}-\tau}{\tau_{i+1}-\tau_i} L_{012} + \tfrac{\tau\ -\tau_i}{\tau_{i+1}-\tau_i} L_{123} \\ L_{012} &= \tfrac{\tau_{i+1}-\tau}{\tau_{i+1}-\tau_{i-1}} L_{01} + \tfrac{\tau\ -\tau_{i-1}}{\tau_{i+1}-\tau_{i-1}} L_{12} \\ L_{123} &= \tfrac{\tau_{i+2}-\tau}{\tau_{i+2}-\tau_i} L_{12} + \tfrac{\tau\ -\tau_i}{\tau_{i+2}-\tau_i} L_{23} \\ L_{01} &= \tfrac{\tau_i-\tau}{\tau_i-\tau_{i-1}} p_{i-1} + \tfrac{\tau\ -\tau_{i-1}}{\tau_i-\tau_{i-1}} p_i \\ L_{12} &= \tfrac{\tau_{i+1}-\tau}{\tau_{i+1}-\tau_i} p_i + \tfrac{\tau\ -\tau_i}{\tau_{i+1}-\tau_i} p_{i+1} \\ L_{23} &= \tfrac{\tau_{i+2}-\tau}{\tau_{i+2}-\tau_{i+1}} p_{i+1} + \tfrac{\tau\ -\tau_{i+1}}{\tau_{i+2}-\tau_{i+1}} p_{i+2} \\ \tau_{i+1} &= |p_{i+1} - p_i|_\alpha + \tau_i,\ \ \alpha\ \in [0,1] \end{aligned} \tag{8}$$

4.2.2. Median Filter

The default EGS algorithm utilizes Gaussian blurring into the algorithm as a pre-processing step to remove noise. For the process of road and path extraction, preserving edges and linear structures is critical. The median filter has been shown to exhibit better edge-preserving properties relative to linear image filters [54]. Since the EGS paper was released, tremendous work has been conducted on improving the computational speed of the median filter [55]. Thus, the Gaussian filter was removed from the algorithm and replaced by a median filter. The kernel size of the median filter is chosen based on the ratio of the original image size to the image size after resizing. The kernel is chosen this way so that the compound effect of resizing and blurring does not result in potentially hazardous objects on the road being blurred out of existence.

4.2.3. HSV Color Space

The color space is transformed to HSV in the developed pre-processing algorithm. The numerical value of each channel remains an integer from 0–255 in order to use the sorting optimizations that are presented in [49]. Given an 8-bit RGB image, the transformation to HSV is defined as follows:

(1) The 8-bit B, G, R integers are scaled to fit in the range 0 to 1.
(2) $V = max(B, G, R)$
(3) $S = V - min(B, G, R)\ if\ V\ \neq 0$; otherwise *0*

(4) $H = \begin{cases} (60G - 60B)(V - min(B,G,R))^{-1} \ if \ V = R \\ 120 + (60B - 60R)(V - min(B,G,R))^{-1} \ if \ V = G \\ 240 + (60R - 60G)(V - min(B,G,R))^{-1} \ if \ V = B \end{cases}$

(5) $H = H + 360 \, if \; H < 0$

(6) $0 \leq V \leq 1, \; 0 \leq S \leq 1, \; 0 \leq H \leq 360$

(7) The H, S, V values are then scaled to fit back into the 0–255 range

$$V = 255(V), \;\; S = 255(S), \;\; H = \frac{H}{2} \tag{9}$$

4.2.4. Compound Effect of Blur and Resize

In this study, an object in the road is analyzed before and after resizing and blurring to assess their effects of them. Blurring after resizing has the obvious computation benefit of needing to process fewer pixels than blurring prior to resizing. Additionally, the range of usable kernel sizes will be smaller and more restricted, which will lead to more effective blurring. In [49], before and after resizing and blurring images shows an interesting difference between the Catmull–Rom interpolation plus median filter (CRM) combo versus the nearest neighbor plus Gaussian filter (NG) combo. Specifically, when blurring is completed after interpolation, for larger kernels, the CRM combo does a great job at smoothing the road while preserving the edges along the side. Unnatural lighting effects, discolorations, noise, and even road markings not relevant to the application at hand are smoothed over. However, a glaring issue is that the speed bumps are virtually blurred out of existence if the kernel gets too large and the image size is reduced too much. If those had been potentially hazardous non-traversable objects instead of speed bumps, then it is crucial that they are identified at some point. On the other hand, the NG combo does not smooth over-the-road discolorations and lighting variations, as well as the CRM combo. The NG combo image just becomes progressively blurrier for larger kernels. By strategically choosing the percentage by which the image size is reduced and the median filter kernel size, it is shown that the CRM combo performs better than the NG combo for pre-processing [49].

4.3. Post Processing

EGS has a post-processing operation baked into the algorithm. Specifically, for each edge connecting two nodes *a* and *b* and for some user-defined integer, minSize:

1. If *a* is not *b*, go to (2); otherwise, keep iterating.
2. If the size of *a*'s segment is less than minSize or the size of *b*'s segment is less than minSize, then merge segments *a* and *b*.

This post-processing operation allows the user to adjust how large the segments should be. Although having an additional hyperparameter can be convenient in some situations, for the purpose of automated road extraction, it is better to have a way to automatically compute a reasonable value. This heuristic relies on the fact that neighboring segments have a higher probability of belonging to a larger unified segment. However, the main drawback of this approach is that the characteristics of each segment are not compared in any way. This means that merging segments with no relation to each other is a likely outcome. Since this method does not consider the similarity between neighboring segments that are merged, large values should be used cautiously. A simple method to auto-compute a value for this parameter is:

$$minSize = \frac{k}{2 * (2.5)^2} = \frac{\sqrt{width * height}}{5} \tag{10}$$

After this, a new method is introduced, which is called Median Color Quantization (MCQ). Median color quantization simply assigns a color to each segment, and that assigned color is the median color of all pixels in that segment. Median color quantization can be observed in Figure 4.

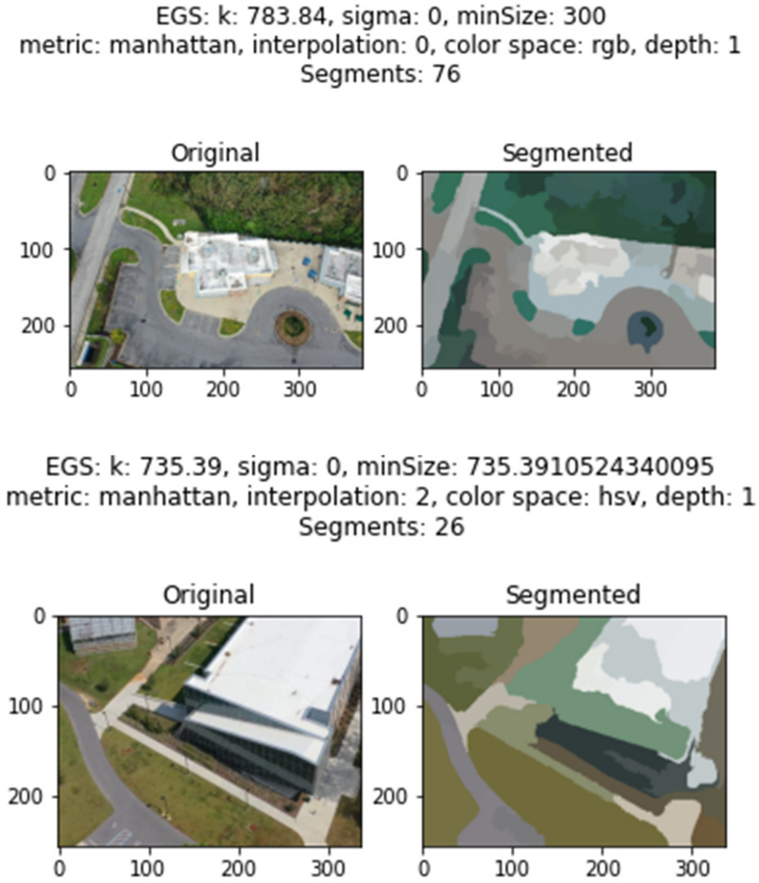
EGS: k: 783.84, sigma: 0, minSize: 300
metric: manhattan, interpolation: 0, color space: rgb, depth: 1
Segments: 76
Original
Segmented
0
100
200
0
100
200
300
EGS: k: 735.39, sigma: 0, minSize: 735.3910524340095
metric: manhattan, interpolation: 2, color space: hsv, depth: 1
Segments: 26
Original
Segmented
0
100
200
0
100
200
300

Figure 4. Segmentation followed by EGS post-processing and MCQ.

The next step is to determine which segments are actually roads. The shape of the road segments varies not only naturally but as a function of the location and angle of the UAV camera. Furthermore, deposits of pine needles or other debris at the road-curb interface can obfuscate the geometry of the road segments. The adaptive isoperimetric quotient threshold function introduced earlier already searched for road-like geometries. Additionally, in some cases, the road is not all in one segment. The road might be broken up into multiple large sub-segments. These subsegments need to be merged. Before the road segments are merged, if more than one exists, it is important to identify at least one segment that is likely to be a road segment. This segment is called the Nucleation Site, from which any remaining road segments will be merged. The nucleation site is identified via a process called the Road Segment Identification (RSI) heuristic, defined below.

1. For each segment, compute the Road Segment Similarity Metric (δ_{RSS}).
2. Use the developed algorithm with k=1 to choose the best road segment, S_b.
3. Asses probability that S_b is, in fact, road and not non-road. If S_b is determined to be non-road, conclude that there are no road segments in the image and halt. Otherwise, go to step 4.
4. For each neighbor of S_b, compute the difference in hue and saturation between S_b and the neighboring segment S_{nbr}. If the change in hue and saturation is small, then merge the two segments.
5. For all neighbors that were merged, repeat steps 4–5 using each neighbor in place of S_b.

The Road Segment Similarity Metric (δ_{RSS}) is defined as:

$$\delta_{RSS}(c,\ rS) = \frac{dist(c,\ rS)}{max(dist(x, rS))} \in [0, 1] \tag{11}$$

where *dist* is just some Minikowski distance, and $max(dist(x, rS))$ refers to the largest possible distance from rS. The parameter c is the color value of the segment, and rS (road sample) is a predefined value for the ideal color of the road. Two ways to compute rS are as follows:

1. Crop a small piece of road out of an image or some small number of images. Then use these samples to compute a value for rS. These road sample images can be taken from the UAV dataset and treated as calibration images, or they can be obtained from some pre-existing database.
2. Simply choose some shade of gray.

Some problems with this approach include: road markings can decrease the saturation of a segment, and lighting effects and discoloration can cause the road to have a lighter gray color. Using K-nearest neighbors and setting a k value assumes that there will be at least one road segment in the image. δ_{RSS} can be treated as a probability, and values below a certain threshold can be deemed non-road. Therefore, if δ_{RSS} is small enough for S_b, it can be concluded that no road segments exist in the image. Another issue arises with this approach if the road segments are not connected and cannot be chained together; a sample case is depicted in Figure 5. In the figure, if region B is taken to be the nucleation site, then region A would never be labeled as a road since the two segments are not connected. To address this issue, after RSI completes for S_b, it can be run again, excluding all segments that have already been labeled as road. This process can continue until a δ_{RSS} for S_b is computed.

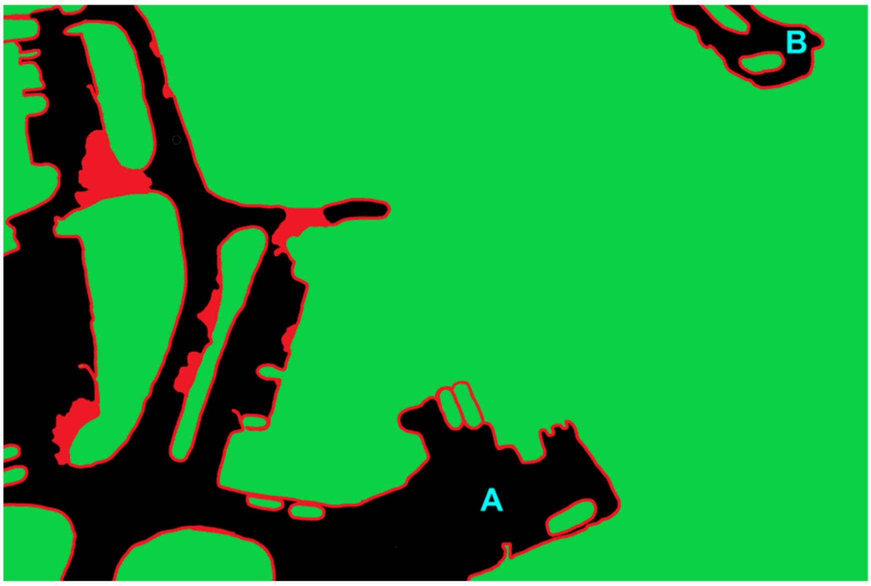

Figure 5. A sample case of image mask depicting multiple road segments that are not connected.

Reference [28] uses the minimum weight edge connecting disparate segments to look for evidence of a boundary between them; the median or any other quantile edge weight cannot be used because it results in an NP-hard problem. Since MCQ was applied, each segment is represented by the median color of the underlying pixels in that segment. Now, it is possible to make a more robust comparison of the median colors of neighboring segments and decide whether or not to merge segments. The value color channel is not considered in an effort to merge segments that were separated because of shadows. Shadows will cause a spike in the value channel in an HSV image. The difference in HSV color between a segment of road with a shadow and a segment of road without a shadow is depicted in Figure 6.

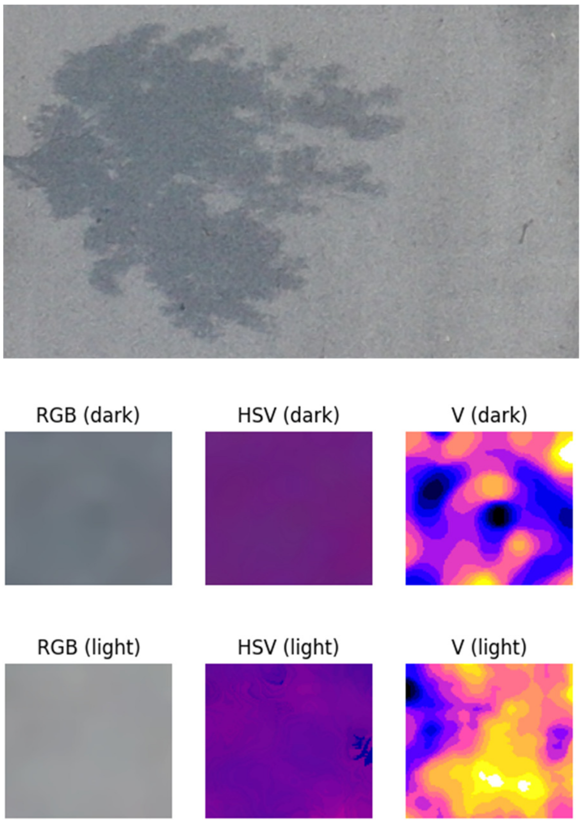

Figure 6. HSV difference between region of road with shadow and region of road without shadow.

5. Results and Discussions

In order to assess the quality of the road extraction, the Intersection over the Union (IoU) technique was used. The IoU quantifies the extent to which the predicted segmentation overlaps with the ground truth segmentation. The parameters used in the presented results are percent reduction ($n\%$), minimum size (minSize), pixel neighborhood (k), threshold function ($\tau(C)$), and recursion depth (d).

5.1. Resizing and Median Filter Kernel

For the median blur operation that occurred during pre-processing, three different kernel sizes were used. Their value is computed as a function of the image size. The size of these three kernels corresponds to three levels of blurring: subtle, moderate, and high. It can be seen from Table 1 that the moderate median kernel size of 15 results in the highest IoU scores. These results show that not enough or too much median blurring can result in poorer road extraction. For resizing, three different levels of size reduction were considered. The image dimensions were reduced by $n\%$. The n values used were: 50, 75, and 90. From the data in Table 1, it can be seen that the highest IoU scores come from the images that were reduced the least. Unfortunately, these higher IoU scores come at the cost of longer algorithm running times. Figures 7–10 show results overlaying the ground truth mask on the predicted mask for different images.

Table 1. Catmull–Rom Interpolation and Median Filter. Comparison of IoU for different image and median filter kernel sizes.

% Resized	Subtle	Moderate	High
50%	0.835	0.892	0.882
75%	0.788	0.825	0.823
90%	0.683	0.785	0.692

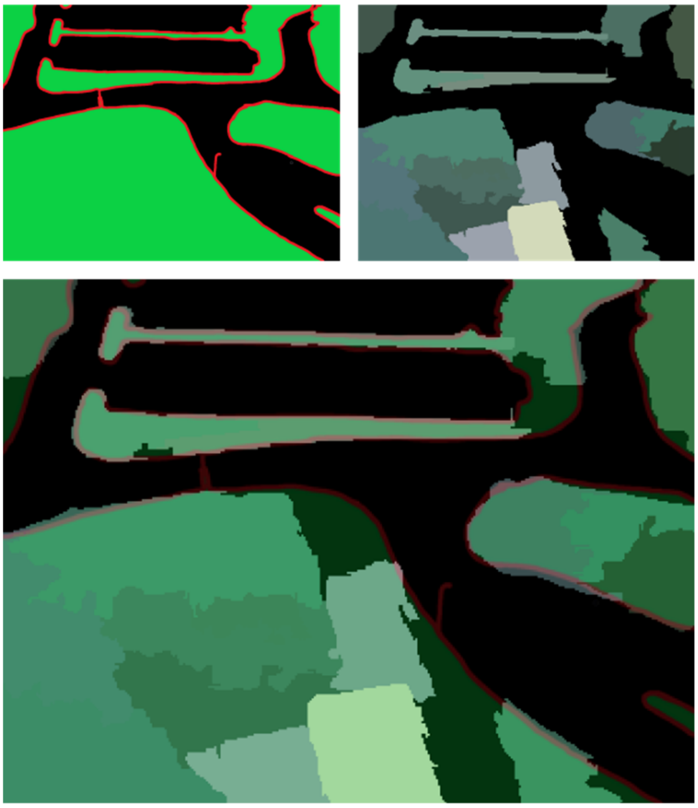

Figure 7. Case-1. **Top** row: Comparison of predicted mask to the actual mask for different median filter values. **Bottom** row: overlaying the ground truth mask on the predicted mask.

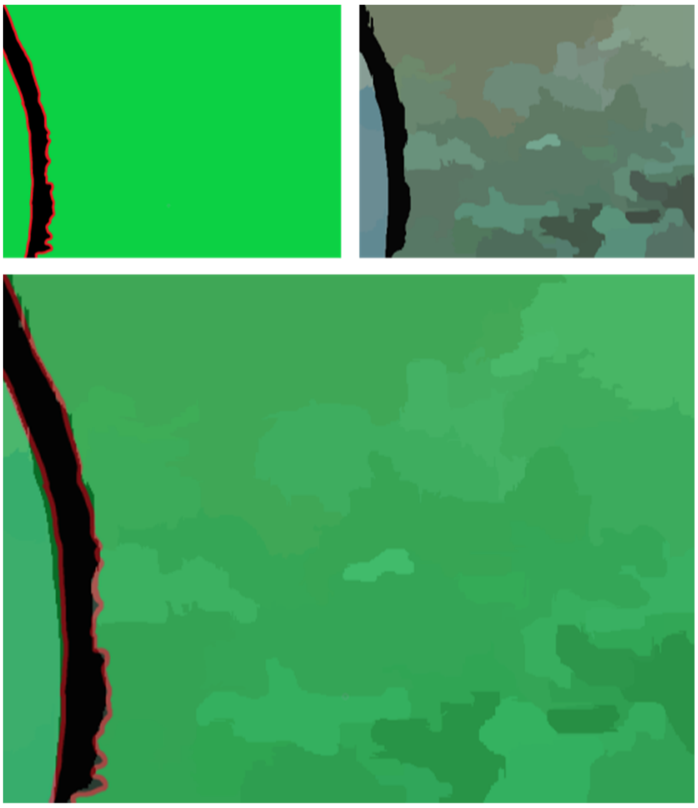

Figure 8. Case-2. **Top** row: Comparison of predicted mask to the actual mask for different median filter values. **Bottom** row: overlaying the ground truth mask on the predicted mask.

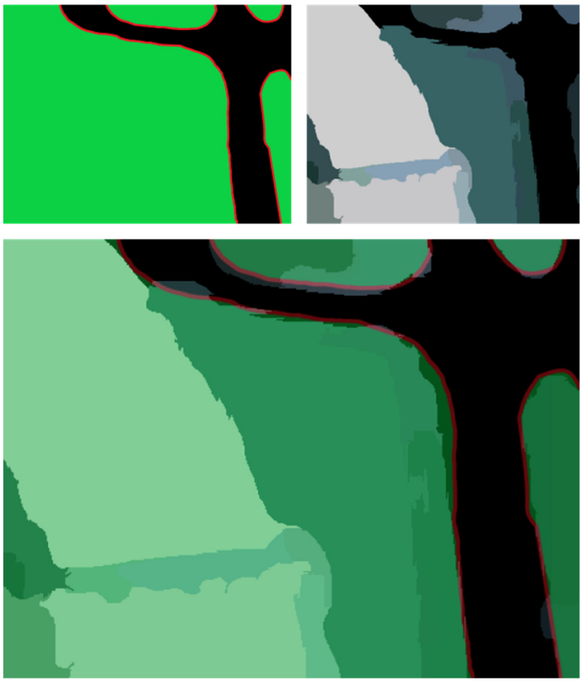

Figure 9. Case-3. **Top** row: Comparison of predicted mask to the actual mask for different median filter values. **Bottom** row: overlaying the ground truth mask on the predicted mask.

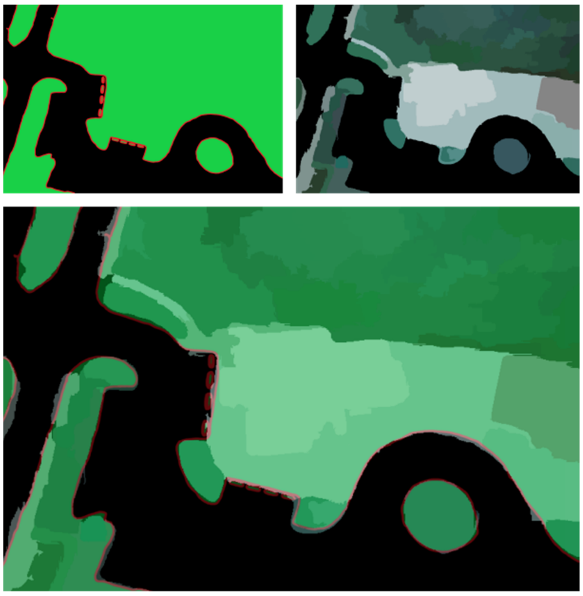

Figure 10. Case-4. **Top** row: Comparison of predicted mask to the actual mask for different median filter values. **Bottom** row: overlaying the ground truth mask on the predicted mask.

5.2. Pixel Neighborhood and the MinSize Parameter

Three different pixel neighborhoods—extended, Von Neumann, and Moore, were tested along with three values for the minSize parameter: k, $k/2$, and $k/5$. The extended neighborhood is the Moore neighborhood plus the extra ring of pixels that border the Moore neighborhood. This results in a maximum of 24 possible neighbors per pixel. Table 2 shows that segmentation using the Von Neumann neighborhood achieves the highest overall IoU score. This is interesting because building a graph while using the Von Neumann pixel neighborhood results in the sparsest graph of the three neighborhoods used in this study. The extended neighborhood performs the worst, even though it yields roughly six times as many graph edges as the Von Neumann neighborhood. The Moore neighborhood shows the least amount of variance for different values of the minSize parameter. Figures 11–13 depict the results for each neighborhood.

Table 2. Pixel neighborhood and minSize. Comparison of IoU for different pixel neighborhoods and values of minSize.

	minSize		
Neighborhood	k	$\frac{k}{2}$	$\frac{k}{5}$
Von Neumann	0.825	0.878	0.755
Moore	0.846	0.816	0.814
Extended	0.707	0.758	0.786

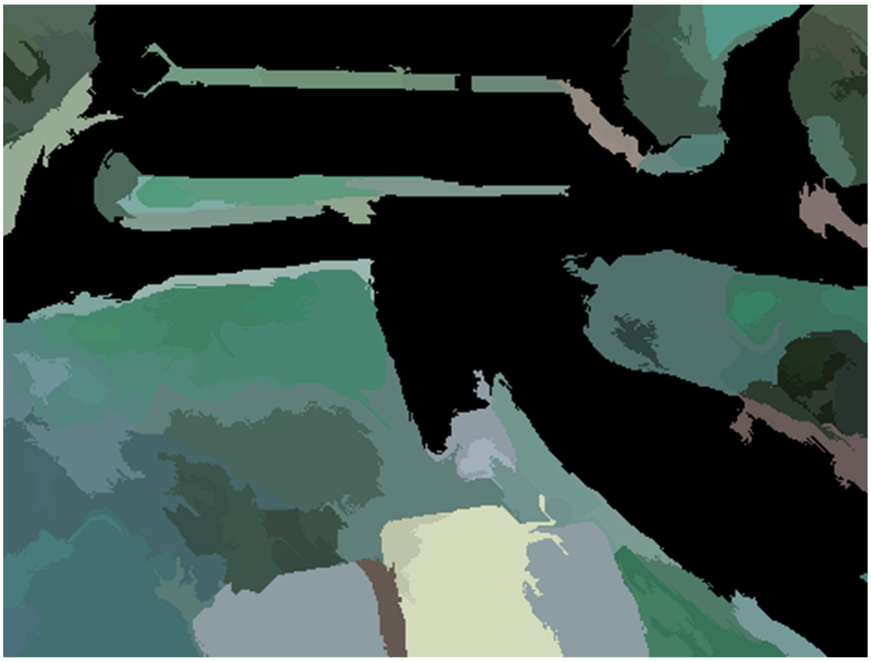

Figure 11. Extended neighborhood with minSize set at $k/5$.

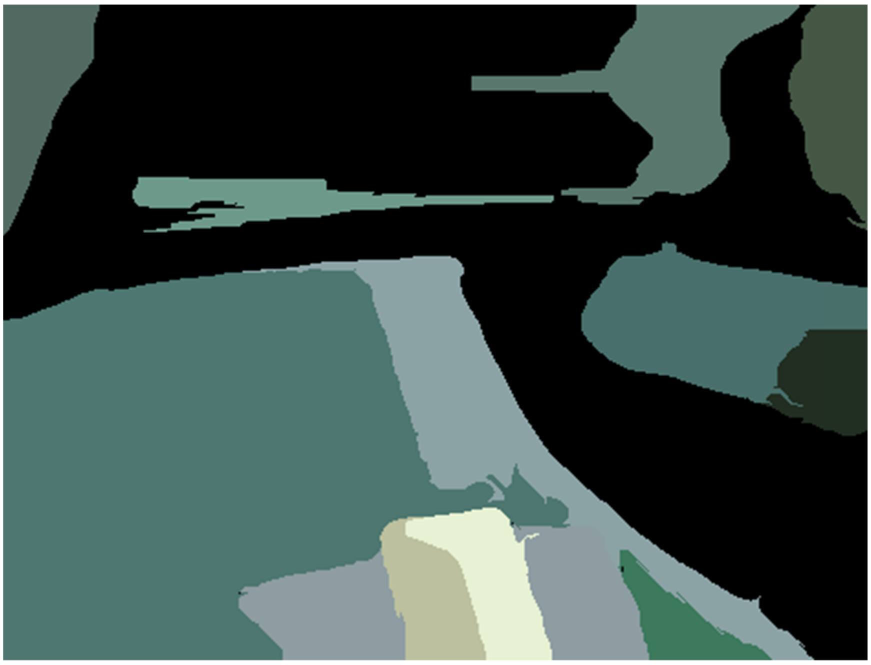

Figure 12. Moore neighborhood with minSize set at k.

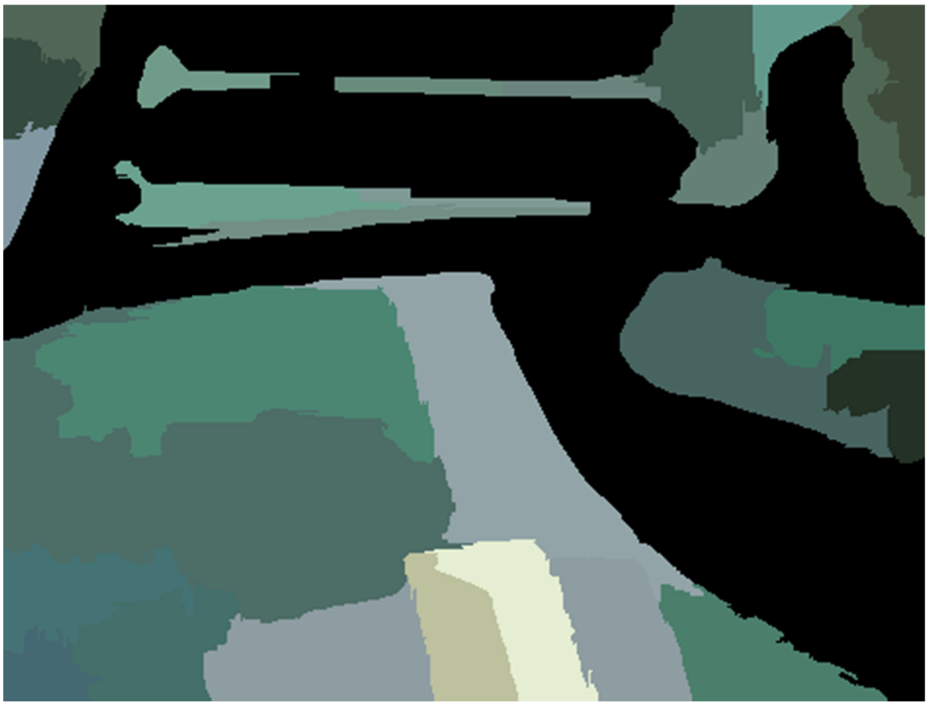

Figure 13. Von Neumann neighborhood with minSize set at $k/2$.

5.3. Threshold Function and Distance Function

In this study, different distance functions and threshold functions were also analyzed. The integer Euclidean distance is computed from a pre-computed lookup table. The lookup table is created by calculating the square of each unique Euclidean distance and assigning it a unique integer. The Euclidean distance performed very poorly for these images. Another drawback of using the Euclidean distance is that it results in slower algorithm running times. The newly proposed adaptive isoperimetric quotient threshold function yields the highest overall IoU score when coupled with the Manhattan distance. The standard threshold function seems to work better with Euclidean distances. The IoU scores are given in Table 3. Figures 14–16 depict the results from modifying the threshold and distance functions.

Table 3. Threshold Function and Distance Function. Comparison of IoU for different threshold and distance functions.

	Distance		
$\tau(C)$	**Manhattan**	**Euclidean**	**Integer Euclidean**
Standard	0.756	0.488	0.615
Adaptive Isoperimetric Quotient	0.877	0.246	0.370

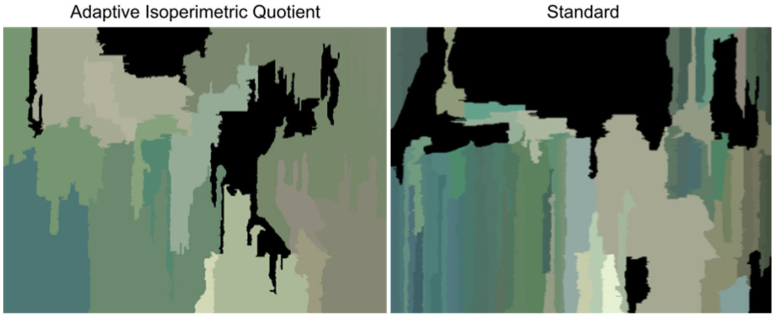

Figure 14. The adaptive isoperimetric quotient threshold function is compared to the standard threshold function. The Euclidean distance was used to create the graph edges.

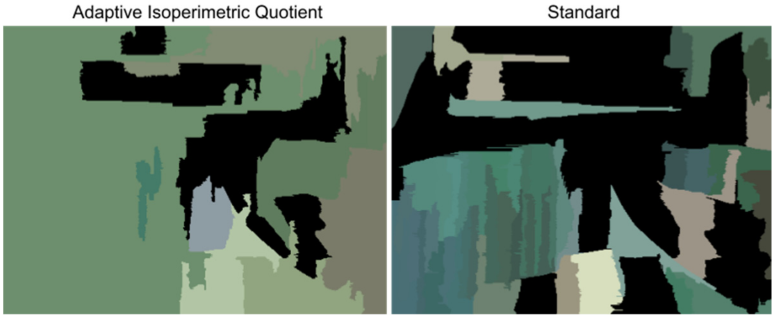

Figure 15. The adaptive isoperimetric quotient threshold function is compared to the standard threshold function. The integer Euclidean distance was used to create the graph edges.

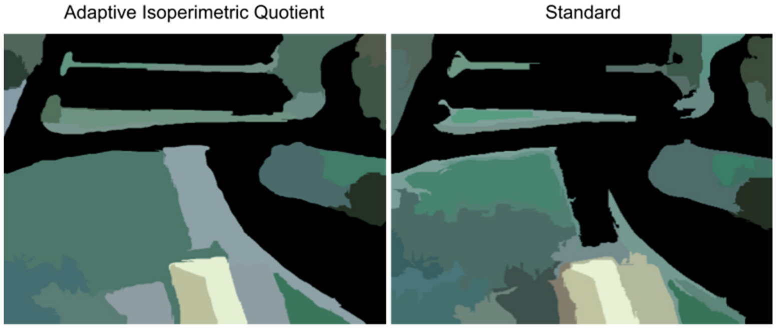

Figure 16. The adaptive isoperimetric quotient threshold function is compared to the standard threshold function. The integer Manhattan distance was used to create the graph edges.

5.4. Recursive Graph-Based Segmentation

Further, an experiment was conducted where the proposed algorithm was recursively called on its output. Table 4 shows results for different values of k, $minSize$, and the recursion depth. Two groups were created, group A and group B. In both groups, $minSize$ and k grow as a function of the recursion depth. The rate of growth is higher for group B, but the initial values of $minSize$ and k are also much lower. Although the overall IoU scores were lower than simply letting k and $minSize$ automatically compute according to the newly proposed Equation (6), it can be observed that the IoU scores increased with recursion depth. Furthermore, group B, with a recursion depth of three, showed an impressive IoU score of 0.872. Figures 17–19 depict the results for different recursion depths.

Table 4. Recursive Graph-Based Segmentation. Comparison of IoU for different growth functions and recursion depths.

	Recursion Depth (d)		
Growth Function	**1**	**2**	**3**
Group A $k_1 = k/2,\ minSize_1 = k/4$ $minSize_d = minSize_{d-1} + 30(d-1)$ $k_d = k_{d-1} + 50(d-1)$	0.804	0.807	0.815
Group B $k_1 = k/4,\ minSize_1 = k/8$ $minSize_d = minSize_{d-1} + 50(d-1)$ $k_d = k_{d-1} + 100(d-1)$	0.795	0.798	0.872

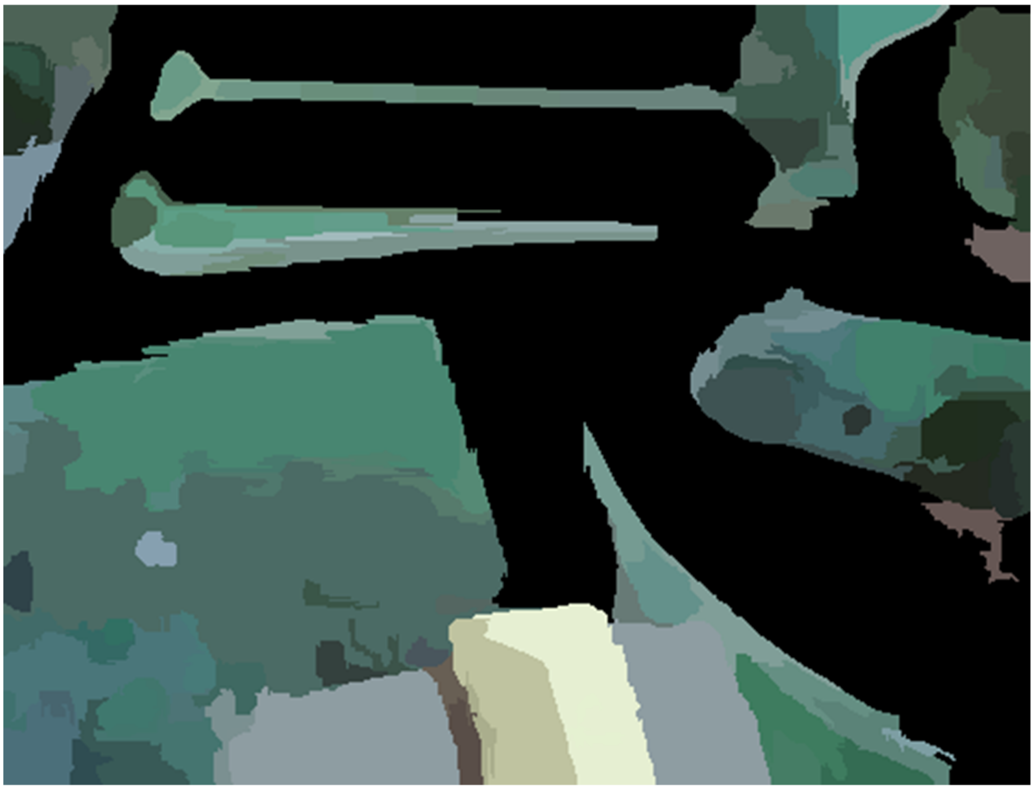

Figure 17. The recursion depth is 1 for this result. k was set to $k/2$, and minSize was set to $k/4$.

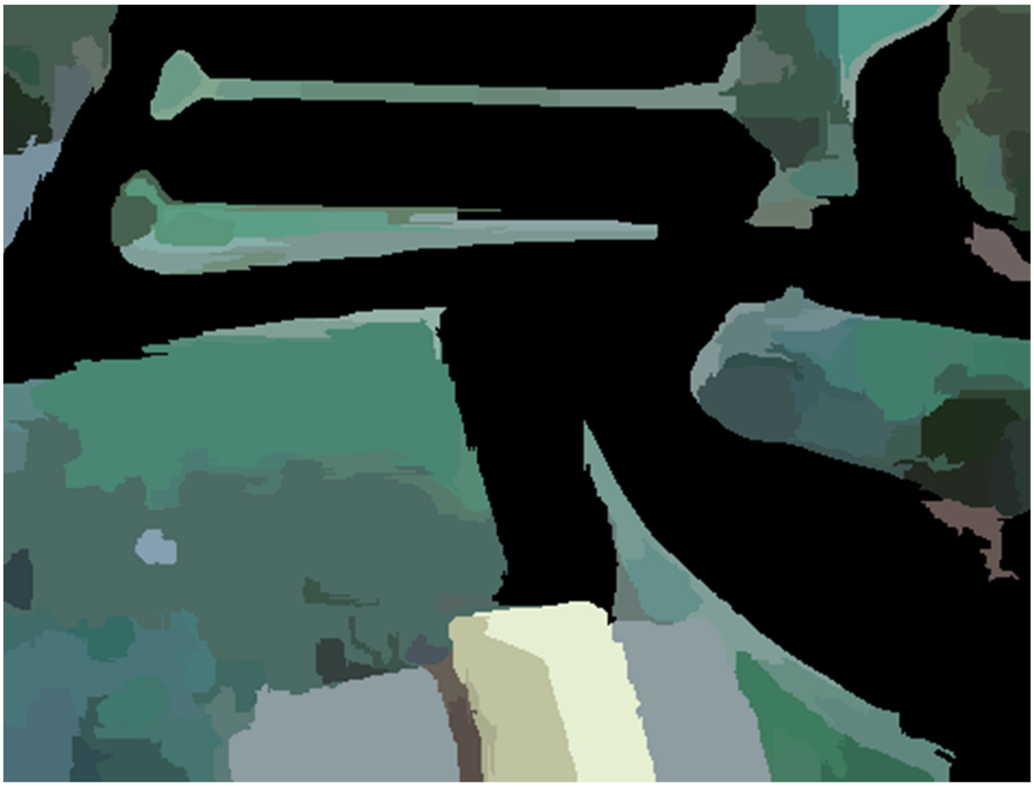

Figure 18. The recursion depth is 2 for this result. k was set to $k/2$, and minSize was set to $k/4$.

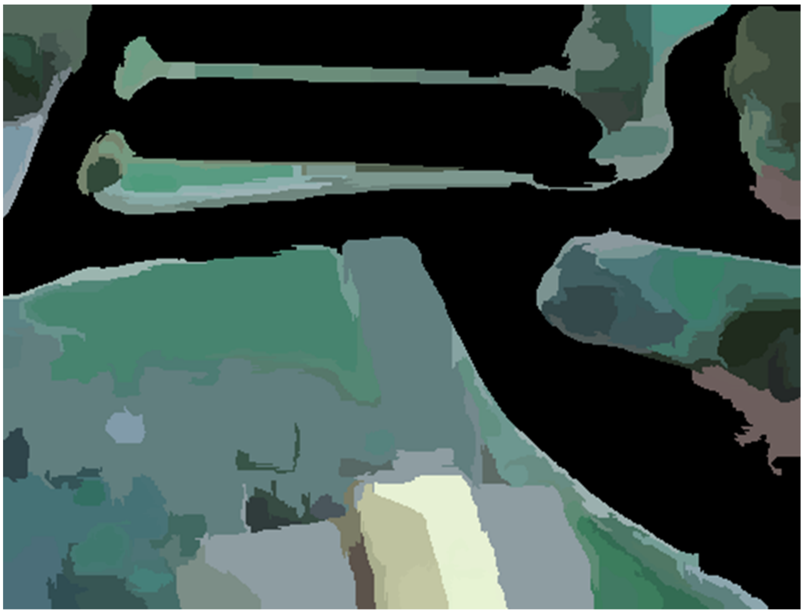

Figure 19. The recursion depth is 3 for this result. k was set to $k/4$, and minSize was set to $k/8$.

5.5. Comparison with K-Means

As a final analysis, the K-means algorithm was used to try and extract the road from images, and the results were compared to the approach presented in this study. This comparison is depicted in Figure 20. Three different values for k were chosen. When using K-means for road extraction, the post-processing presented in this paper is no longer valid. In K-means, the segments do not necessarily represent objects in the image. Completely unrelated objects on opposite sides of an image that happen to have similar colors can belong to the same segment. When the proposed post-processing is applied after K-means segmentation, the nucleation site that is discovered could just be some stray pixel. Therefore, to extract the roads, a simple thresholding operation was applied. The image was converted into a binary image, and then morphological operations were applied. It can be observed from Tables 5 and 6 that the proposed method in this paper performs significantly better than the K-means-based road extraction. The low IoU scores primarily stem from labeling non-road as road.

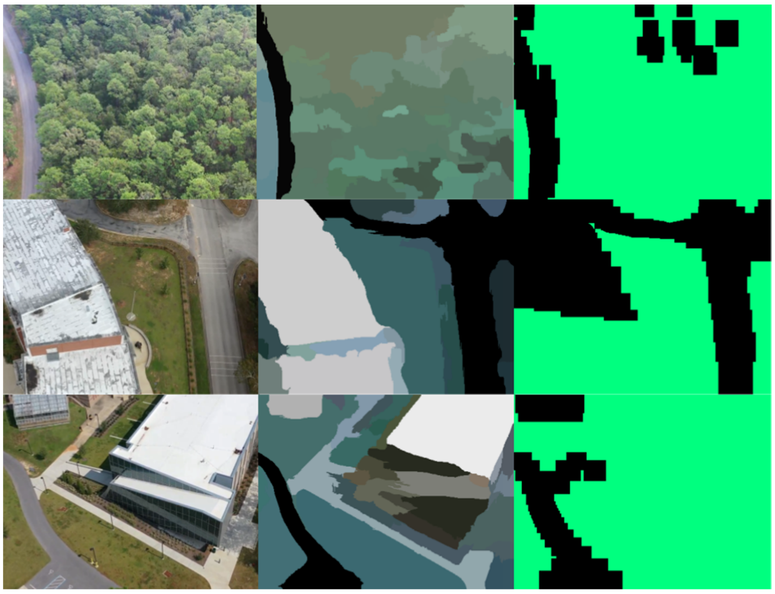

Figure 20. The source images are on the left, the segmentations produced from the novel method proposed in this paper are in the center, and segmentations from K-means are on the right.

Table 5. K-means Based Road Extraction. Comparison of IoU for different values of k.

# Centroids	**25**	**50**	**100**
IoU	0.361	0.352	0.337

Table 6. EGS versus K-means. The IoU scores of EGS and K-means are compared for the same images in the database. For K-means, the k value was chosen that resulted in the highest IoU.

Image No.	**EGS**	**K-Means**
0	0.745	0.234
2	0.858	0.442
4	0.846	0.403
Mean IoU (All Images)	0.832	0.422

6. Conclusions

In this study, an improved Efficient Graph-Based image segmentation algorithm applied to road extraction from post-disaster aerial footage is presented. The modifications proposed to EGS in this study led to significant improvements in the segmentation quality and Intersection over Union (IoU) scores. Although less size reduction seems to yield higher-quality segmentations, it is important to factor in the running time required to process more pixels. From the experiments conducted, a 75% reduction in image size yields high-quality segmentation results and also allows the algorithm to run efficiently for large images. The Von Neumann pixel neighborhood coupled with integer Manhattan distance edge weights led to the highest quality segmentation results. The Moore neighborhood showed strong performance, and it is worth noting that the change in IoU as minSize changed was less extreme. The introduction of the new adaptive isoperimetric quotient threshold function showed great promise for the application of road extraction and showed a significant improvement in the IoU score when compared to the standard threshold function. Recursively calling the algorithm and incrementing the parameters *k* and minSize resulted in progressively better IoU scores. One issue that can be seen with the IoU measure is that it does impose a strict enough penalty on the dissolution of non-road segments. For the application of traversability, a non-road segment being labeled road should be far costlier than a road segment labeled non-road. The comparison between K-means and the developed algorithm in this paper shows that the algorithm has significantly better IoU scores than the K-means method. Additionally, the major issue in the K-means method is the large number of non-road segments being labeled as roads. It is conceivable to think that a more rigorous post-processing method could yield even better results with the developed algorithm. As a future work, the novel pre-processing, image segmentation, and MCQ combination presented in this research will be used to generate superpixels which could then be used as input to a state-of-the-art neural network architecture such as a U-net. The comparison of our developed framework with other methods in the literature, robustness analysis, and computational time analysis is not included in the scope of the originally intended contribution in this article, and they are left for a future work.

Author Contributions: Conceptualization, methodology, investigation, writing—original draft preparation, N.P.S.; supervision, project administration, writing—review and editing, H.E.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. The Centre for Research on the Epidemiology of Disasters (CRED). Emergency Events Database (EM-DAT). Available online: www.emdat.be (accessed on 16 January 2022).
2. Peduzzi, P.; Chatenoux, B.; Dao, H.; De Bono, A.; Herold, C.; Kossin, J.; Mouton, F.; Nordbeck, O. Global trends in tropical cyclone risk. *Nat. Clim. Chang.* **2012**, *2*, 289–294. [CrossRef]
3. Dapena, K. The Rising Costs of Hurricanes. Wall Street Journal 2018. Available online: www.wsj.com/articles/the-rising-costs-of-hurricanes-1538222400 (accessed on 16 January 2022).
4. Koks, E.E.; Rozenberg, J.; Zorn, C.; Tariverdi, M.; Vousdoukas, M.; Fraser, S.A.; Hal, J.W.; Hallegatte, S. A global multi-hazard risk analysis of road and railway infrastructure assets. *Nat. Commun.* **2019**, *10*, 2677. [CrossRef]
5. Bjarnadottir, S.; Li, Y.; Stewart, M.G. Social vulnerability index for coastal communities at risk to hurricane hazard and a changing climate. *Nat. Hazards* **2011**, *59*, 1055–1075. [CrossRef]
6. Horner, M.W.; Widener, M.J. The effects of transportation network failure on people's accessibility to hurricane disaster relief goods: A modeling approach and application to a Florida case study. *Nat. Hazards* **2011**, *59*, 1619–1634. [CrossRef]
7. Hsia, R.Y.; Huang, D.; Mann, N.C.; Colwell, C.; Mercer, M.P.; Dai, M.; Niedzwiecki, M.J. A US national study of the association between income and ambulance response time in cardiac arrest. *JAMA Netw. Open* **2018**, *1*, e185202. [CrossRef]
8. Xu, D.; Tian, Y. A comprehensive survey of clustering algorithms. *Ann. Data Sci.* **2015**, *2*, 165–193. [CrossRef]
9. Cheng, G.; Liu, L. Survey of image segmentation methods based on clustering. In Proceedings of the 2020 IEEE International Conference on Information Technology, Big Data and Artificial Intelligence (ICIBA), Chongqing, China, 6–8 November 2020; IEEE: Piscataway Township, NJ, USA, 2020; Volume 1, pp. 1111–1115.
10. Manning, C.D.; Raghavan, P.; Schütze, H. *Introduction to Information Retrieval*; Cambridge University Press: Cambridge, UK, 2008; p. 335.
11. VanderPlas, J. *Python Data Science Handbook: Essential Tools for Working with Data*; O'Reilly Media, Inc.: Newton, MA, USA, 2016.
12. Maurya, R.; Gupta, P.R.; Shukla, A.S. Road extraction using k-means clustering and morphological operations. In Proceedings of the 2011 International Conference on Image Information Processing, Shimla, India, 3–5 November 2011; IEEE: Piscataway Township, NJ, USA, 2011; pp. 1–6.
13. Chen, Q. Hierarchical segmentation for color images. In Proceedings of the 2015 8th International Congress on Image and Signal Processing (CISP), Shenyang, China, 14–16 October 2015; IEEE: Piscataway Township, NJ, USA, 2015; pp. 934–938.
14. Rongjie, L.; Jie, Z.; Pingjian, S.; Fengjing, S.; Guanfeng, L. An agglomerative hierarchical clustering based high-resolution remote sensing image segmentation algorithm. In Proceedings of the 2008 International Conference on Computer Science and Software Engineering, Wuhan, China, 12–14 December 2008; IEEE: Piscataway Township, NJ, USA, 2008; Volume 4, pp. 403–406.
15. Mukherjee, D.P.; Mohanta, P.P.; Acton, S.T. Agglomerative clustering of feature data for image segmentation. In Proceedings of the International Conference on Image Processing, Bordeaux, France, 16–19 September 2002; IEEE: Piscataway Township, NJ, USA, 2002; Volume 3, p. III.
16. Bezdek, J.C. *Pattern Recognition with Fuzzy Objective Function Algorithms*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2013.
17. Mohammadzadeh, A.; Zoej, M.J.V.; Tavakoli, A. Automatic main road extraction from high resolution satellite imageries by means of particle swarm optimization applied to a fuzzy-based mean calculation approach. *J. Indian Soc. Remote Sens.* **2009**, *37*, 173–184. [CrossRef]
18. Fengping, W.; Weixing, W. Road extraction using modified dark channel prior and neighborhood FCM in foggy aerial images. *Multimed. Tools Appl.* **2019**, *78*, 947–964. [CrossRef]
19. Kriegel, H.; Kröger, P.; Sander, J.; Zimek, A. Density-based clustering. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* **2011**, *1*, 231–240. [CrossRef]
20. Ankerst, M.; Breunig, M.M.; Kriegel, H.P.; Sander, J. OPTICS: Ordering points to identify the clustering structure. *ACM Sigmod Rec.* **1999**, *28*, 49–60. [CrossRef]
21. Chacón, J.E.; Monfort, P. A comparison of bandwidth selectors for mean shift clustering. *arXiv* **2013**, arXiv:1310.7855.
22. Craciun, S.; Kirchgessner, R.; George, A.D.; Lam, H.; Principe, J.C. A real-time, power-efficient architecture for mean-shift image segmentation. *J. Real-Time Image Process.* **2018**, *14*, 379–394. [CrossRef]
23. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.
24. Pooransingh, A.; Radix, C.A.; Kokaram, A. The path assigned mean shift algorithm: A new fast mean shift implementation for colour image segmentation. In Proceedings of the 2008 15th IEEE International Conference on Image Processing, San Diego, CA, USA, 12–15 October 2008; IEEE: Piscataway Township, NJ, USA, 2008; pp. 597–600.
25. Siqiang, L.; Wei, L. Image segmentation based on the Mean-Shift in the HSV space. In Proceedings of the 2007 Chinese Control Conference, Zhangjiajie, China, 26–31 July 2007; IEEE: Piscataway Township, NJ, USA, 2007; pp. 476–479.
26. Revathi, M.; Sharmila, M. Automatic road extraction using high resolution satellite images based on level set and mean shift methods. In Proceedings of the 2013 Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT), Tiruchengode, India, 4–6 July 2013; IEEE: Piscataway Township, NJ, USA, 2013; pp. 1–7.
27. Bishop, C.M.; Nasrabadi, N.M. *Pattern Recognition and Machine Learning*; Springer: New York, NY, USA, 2006; Volume 4, p. 738.
28. Szeliski, R. *Computer Vision: Algorithms and Applications*; Springer Nature: Berlin/Heidelberg, Germany, 2022.
29. Govindaraju, V.; Raghavan, V.; Rao, C.R. *Big Data Analytics*; Elsevier: Amsterdam, The Netherlands, 2015.

30. Nguyen, T.M. Gaussian Mixture Model Based Spatial Information Concept for Image Segmentation. Ph.D. Dissertation, University of Windsor, Windsor, ON, Canada, 2011.
31. Bhagavathy, S.; Newsam, S.; Manjunath, B.S. Modeling object classes in aerial images using texture motifs. In Proceedings of the 2002 International Conference on Pattern Recognition, Quebec, Canada, 11–15 August 2002; IEEE: Piscataway Township, NJ, USA, 2002; Volume 2, pp. 981–984.
32. Zhou, H.; Kong, H.; Alvarez, J.M.; Creighton, D.; Nahavandi, S. Fast road detection and tracking in aerial videos. In Proceedings of the 2014 IEEE Intelligent Vehicles Symposium Proceedings, Dearborn, MI, USA, 8–11 June 2014; IEEE: Piscataway Township, NJ, USA, 2014; pp. 712–718.
33. Shi, J.; Malik, J. Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2000**, *22*, 888–905.
34. Felzenszwalb, P.F.; Huttenlocher, D.P. Efficient graph-based image segmentation. *Int. J. Comput. Vis.* **2004**, *59*, 167–181. [CrossRef]
35. Akinina, A.V.; Nikiforov, M.B.; Savin, A.V. Multiscale image segmentation using normalized cuts in image recognition on satellite images. In Proceedings of the 2018 7th Mediterranean Conference on Embedded Computing (MECO), Budva, Montenegro, 10–14 June 2018; IEEE: Piscataway Township, NJ, USA, 2018; pp. 1–3.
36. Grote, A.; Butenuth, M.; Gerke, M.; Heipke, C. Segmentation based on normalized cuts for the detection of suburban roads in aerial imagery. In Proceedings of the 2007 Urban Remote Sensing Joint Event, Paris, France, 11–13 April 2007; IEEE: Piscataway Township, NJ, USA, 2007; pp. 1–5.
37. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention, Singapore, 27 September–1 October 2015; Springer: Berlin/Heidelberg, Germany, 2015; pp. 234–241.
38. Yigit, A.Y.; Uysal, M. Automatic road detection from orthophoto images. *Mersin Photogramm. J.* **2020**, *2*, 10–17.
39. Ichim, L.; Popescu, D. Road detection and segmentation from aerial images using a CNN based system. In Proceedings of the 2018 41st International Conference on Telecommunications and Signal Processing (TSP), Athens, Greece, 4–6 July 2018; IEEE: Piscataway Township, NJ, USA, 2018; pp. 1–5.
40. Varia, N.; Dokania, A.; Senthilnath, J. DeepExt: A convolution neural network for road extraction using RGB images captured by UAV. In Proceedings of the 2018 IEEE Symposium Series on Computational Intelligence (SSCI), Bangalore, India, 18–21 November 2018; IEEE: Piscataway Township, NJ, USA, 2018; pp. 1890–1895.
41. Sarp, S.; Kuzlu, M.; Cetin, M.; Sazara, C.; Guler, O. Detecting floodwater on roadways from image data using Mask-R-CNN. In Proceedings of the 2020 International Conference on INnovations in Intelligent SysTems and Applications (INISTA), Novi Sad, Serbia, 24–26 August 2020; IEEE: Piscataway Township, NJ, USA, 2020; pp. 1–6.
42. You, J. Weather data integrated mask R-CNN for automatic road surface condition monitoring. In Proceedings of the 2019 IEEE Visual Communications and Image Processing (VCIP), Sydney, Australia, 1–4 December 2019; IEEE: Piscataway Township, NJ, USA, 2019; pp. 1–4.
43. Soni, P.K.; Rajpal, N.; Mehta, R. A comparison of road network extraction from High Resolution Images. In Proceedings of the 2018 First International Conference on Secure Cyber Computing and Communication (ICSCCC), Jalandhar, India, 15–17 December 2018; IEEE: Piscataway Township, NJ, USA, 2018; pp. 525–531.
44. Frey, B.J.; Dueck, D. Clustering by passing messages between data points. *Science* **2007**, *315*, 972–976. [CrossRef]
45. Bedawi, S.M.; Kamel, M.S. A comparative study of clustering methods for urban areas segmentation from high resolution remote sensing image. In Proceedings of the 2009 Ninth International Conference on Intelligent Systems Design and Applications, Pisa, Italy, 30 November–2 December 2009; IEEE: Piscataway Township, NJ, USA, 2009; pp. 169–174.
46. Xu, G.; Sun, H.; Yang, W.; Shuai, Y. An improved road extraction method based on MRFs in rural areas for SAR images. In Proceedings of the 2007 1st Asian and Pacific Conference on Synthetic Aperture Radar, Huangshan, China, 5–9 November 2007; IEEE: Piscataway Township, NJ, USA, 2007; pp. 489–492.
47. Huber, R.; Lang, K. Road extraction from high-resolution airborne SAR using operator fusion. In *IGARSS 2001. Scanning the Present and Resolving the Future, Proceedings of the IEEE 2001 International Geoscience and Remote Sensing Symposium (Cat. No. 01CH37217), Sydney, Ausralia, 9–13 July 2001*; IEEE: Piscataway Township, NJ, USA, 2001; Volume 6, pp. 2813–2815.
48. Maboudi, M.; Amini, J.; Malihi, S.; Hahn, M. Integrating fuzzy object based image analysis and ant colony optimization for road extraction from remotely sensed images. *ISPRS J. Photogramm. Remote Sens.* **2018**, *138*, 151–163. [CrossRef]
49. Sebasco, N.P. Traversable Region Identification from Post-Disaster Aerial Footage Using Graph Based Image Segmentation 2022. Master's Thesis, University of West Florida, Pensacola, FL, USA, 2021.
50. Clevenger, A.; Lowande, R.; Sevil, H.E.; Mahyari, A. *Towards UAV-Based Post-Disaster Damage Detection and Localization: Hurricane Sally Case Study*; AIAA SciTech 2022: San Diego, CA, USA, 2022; AIAA-2022-0788.
51. Lowande, R.; Clevenger, A.; Mahyari, A.; Sevil, H.E. Analysis of Post-Disaster Damage Detection using Aerial Footage from UWF Campus after Hurricane Sally. In Proceedings of the International Conference on Image Processing, Computer Vision & Pattern Recognition (IPCV'21), Las Vegas, NV, USA, 26–29 July 2021.
52. Parsania, P.; Virparia, P.V., Dr. A Comparative Analysis of Image Interpolation Algorithms. *Int. J. Adv. Res. Comput. Commun. Eng.* **2016**, *5*, 29–34. [CrossRef]
53. Catmull, E.; Rom, R. A class of local interpolating splines. In *Computer Aided Geometric Design*; Academic Press: Cambridge, MA, USA, 1974; pp. 317–326.

54. Arias-Castro, E.; Donoho, D.L. Does median filtering truly preserve edges better than linear filtering? *Ann. Stat.* **2009**, *37*, 1172–1206. [CrossRef]
55. Perreault, S.; Hebert, P. Median filtering in constant time. *IEEE Trans. Image Process.* **2007**, *16*, 2389–2394. [CrossRef] [PubMed]