Review of Unsupervised Segmentation Techniques on Long Wave Infrared Images

Mohammed Abuhussein, Aaron L. Robinson, Iyad Almadani
School of Electrical and Computer Engineering,
University of Memphis, Memphis, TN, 38111, USA

***Abstract*—This paper studies the different unsupervised segmentation algorithms that have been proposed and their efficacy on thermal images. The scope of this research is to develop a generalized approach to blindly segment urban thermal imagery to assist the system in identifying regions by shape instead of pixel values. Most methods can be classified as thresholding, edge-based, region-based, clustering, or texture analysis. We explained methods, worked before applying the methods of interest on thermal images of 8-bit and 16-bit resolution, and evaluated the performance. The evaluation section discusses where each method succeeded, where it failed, and how the performance can be enhanced. Finally, we study the time complexity of each method to assess the feasibility of implementing a fast, and generalized method of pixel labeling.**

***Keywords*—Unsupervised segmentation; thermal images; texture analysis; pixel labeling; Gabor; GMM; image analysis; K-Means; MRF; Otsu's; DNN; region-based clustering**

I. Introduction

Image segmentation [1] is an area of focus primarily due to its potential usefulness in numerous fields of application. Given that images allow for the transfer of information, understanding them and the associated methods of extracting information is essential. Image segmentation often serves as the first step in the process of image interpretation. It aims to change, simplify, or partition the representation of an image into a more meaningful collection of segments for enhanced analysis [2], [3]. The importance and applicability of image processing cannot be overemphasized. In practice, many image processing algorithms do not focus on the entire image but only require information from the image regions that share certain features. For example, consider an application such as medical imaging where surgery decisions need accurate information about the images to either initiate or speed up patient recovery [4], [5]. Image segmentation supplies the critical image processing function of aiding object location and boundaries in patient imagery in these situations. It effectively assigns labels to every image pixel and enables necessary identifications such as foreground and background regions and other objects of interest in the scene [6].

As implied by the name, the outcome of an image segmentation procedure is a set segment that, when combined, covers the whole image. Each one of these individual segments is called a mask [7]. Masks are pixels in a particular region that share certain texture, color, and intensity characteristics. Image segmentation converts images into sets of masks which can then be interpreted as labeled images. Consequently, the labeled regions produced by the segmentation allow one the capability of only processing the important parts of an image rather than processing the whole image [5].

So far, there has been a plethora of effective segmentation techniques developed for multiple applications and platforms. These techniques include threshold segmentation [8], region [9], and edge-based segmentation [10], clustering, texture-based segmentation [11], and Partial Differential Equation (PDE) based segmentation [12]. There are a plethora of segmentation approaches. However, the underlying question becomes how does one identify the technique that offers the best image analysis results and performance?

This paper will apply the aforementioned methods to long-wave infrared images (LWIR) and analyze the results. We will discuss each of these methods in general and provide variation details concerning implementation, effect on accuracy, the difference in performance on eight vs. 16-bit data, number of tunable parameters, response to texture and uniform surfaces, and lastly, their time complexities.

In this paper, we present a comprehensive review of unsupervised segmentation techniques applied to long-wave infrared (LWIR) images. The paper has main six sections, starting with the motivation section and ending with the conclusion. The motivation for this study arises from the growing need for effective image analysis in LWIR applications. The literature review section provides an overview of the existing research, highlighting unsupervised segmentation techniques specifically designed for LWIR images. The evaluation section presents the comparative analysis results, showcasing the effectiveness of each technique. The discussion section offers insights into the findings, identifying trends and potential areas for improvement. The conclusion summarizes the key takeaways from the review, emphasizing the most promising techniques. This review serves as a valuable resource for researchers and practitioners in LWIR image segmentation, facilitating the development of accurate and efficient segmentation methods.

II. Motivation

LWIR is one of the three commonly defined wavelength bands in which infrared imaging operates. The other two are Medium Wavelength Infrared (MWIR) and Very Long Wavelength Infrared (VLIR). LWIR infrared is commonly defined as covering the wavelengths that range from 8,000nm to 14,000nm (8μm to 14μ m) [13]. Generally, LWIR cameras detect the thermal emissions of animals, vehicles, and people as they stand out when the environment's temperature differs by an amount greater than the camera's sensitivity. LWIR imaging is commonly utilized as a solution for night vision,

thermal imaging, and in degraded visual environments because the longer wavelengths make it less susceptible to scattering from obscurants, such as fog, rain, smoke, dust, and sand. LWIR imaging is instrumental in distinguishing targets at night since traditional imaging employs visible light and cannot reveal sufficient information in these scenarios due to a lack of signal.

The fields of computer vision and image processing are responsible for developing many methods designed to resolve the problems arising in the image segmentation process. However, for infrared/thermal images, the traditional techniques face some additional restrictions, which result in the segmentation being a more challenging problem. For example, when attempting to apply pixel-based segmentation methods to infrared images, the lack of disparity in pixel intensities poses a challenge in grouping/defining the objects' pixels with respect to their background. That can mainly be due to insufficient temperature differences between the object and the background. Another example occurs when utilizing image gradients as edge indicators. In these cases, the LWIR segmentation may fail to accurately identify appropriate object boundaries within the scene due to the non-uniform nature of the pixel intensities and the resulting the poor edge identification.

In this study, we focus on evaluating traditional segmentation algorithms and their feasibility in segmenting thermal images. The challenges mentioned above will be the main scope of this work to create a user-friendly tool to provide labeled data with minimal human input. For some algorithms, the human input will be selecting the number of thresholds, clusters, or objects. Meanwhile, other methods, such as region-growing, will take starting seeds as inputs. Texture segmentation takes sample texture patches as inputs. Ideally, the tool will include a standalone method that will only require the semantic labels from the user.

III. Literature Review

In this treatment, we have reviewed publications from the last 20 years addressing image segmentation. This period can be divided into the pre-popularization and post-popularization of the deep learning era. The authors note that the deep-learning methods are very efficient with RGB representations of visible light images and are widely used due to this fact. More importantly, the authors note that very few deep-learning algorithms are applied to segment infrared images. This is most likely due to the difficulties mentioned above associated with infrared image segmentation.

The next sections present common methods used in unsupervised segmentation. We discuss thresholding as the first and most common pre-processing step, then discuss other prevalent and promising segmentation techniques. Lastly, we evaluate these techniques by visually analyzing the results and providing quantitative performance evaluation, and discussing scenarios where each method fails in the results sections.

This section explains the methods examined in this study, including the different variations of the same general approach. We begin with thresholding since it is an essential step in most segmentation approaches. Section III-B discusses the various edge detection approaches. Sections III-C, III-D, and III-F delve into region growing, clustering, and texture analysis, respectively.

A. Thresholding

Thresholding image segmentation techniques have gained significant attention due to their simplicity and effectiveness. They are especially useful when dealing with images that have distinct foreground and background intensities. The basic idea behind thresholding [14] is to select a threshold value that separates the desired objects or regions from the rest of the image. Thresholding is the simplest and probably the most common image segmentation technique. The underlying principle relies upon setting a number of pixel intensity thresholds to divide the image pixels into multiple categories. Each category or mask is intended to represent a region of the input image with common features. Common features include color/grayscale characteristics or other common transformation characteristics. If the technique is based on a single threshold value, the effective result is to change a grayscale image into a binary one. If more than one threshold is desired, the thresholding is referred to as multi-level. Binary segmentation and other multi-level thresholding techniques all share the same core issue of effectively selecting optimal thresholds based on certain criteria [1]. Thresholding techniques can be categorized based on global, local, or image histograms. Global Thresholding is the simplest form of thresholding, where a single threshold value is applied to the entire image. Pixels with intensities above the threshold are classified as foreground, while those below the threshold are classified as background. Local thresholding, also known as adaptive thresholding, is a technique used for image segmentation where different threshold values are determined for different regions or pixels of an image. Unlike global thresholding, which applies a single threshold value to the entire image, local thresholding takes into account the local characteristics of the image to handle variations in illumination, contrast, and noise. In local thresholding, the threshold value for each pixel is computed based on the neighborhood around that pixel. The neighborhood can be defined as a fixed window size or a variable size depending on the algorithm or application. The threshold is calculated using statistical measures such as the mean, median, or standard deviation of the pixel intensities within the neighborhood. The main advantage of local thresholding is its ability to adapt to local variations in image properties. This makes it particularly useful in situations where the lighting conditions or intensity characteristics change across different regions of the image. By adjusting the threshold values locally, local thresholding can effectively segment objects or regions with varying illumination or contrast levels. Image histogram thresholding techniques analyze the histogram of the image to determine the threshold values. These techniques can be either global or local. They involve examining the distribution of pixel intensities in the histogram and selecting appropriate threshold values based on certain criteria or statistical measures. Examples of image histogram thresholding methods include Otsu's method, which finds an optimal threshold by maximizing the between-class variance, and the Maximum Entropy method, which selects the threshold that maximizes the entropy of the image.

The most popular variable thresholding method is Otsu's maximum variance approach. Formulated by Nobuyuki Otsu,

the Otsu method is also known as the variance threshold and is a popular algorithm in image segmentation. The optimal threshold is obtained by maximizing class variance functions [5]. It partitions the input image grayscale levels into foreground and background regions. The maximum inter-class variance difference between the two is obtained when the threshold is set to the "optimal" value. It is the preferred method for real-world images based on shape measures and uniformity. However, if the variances among classes differ significantly, the Otsu method cannot offer suitable thresholds for separating the classes [1]. Despite these shortcomings, the Otsu method has a simple algorithm that makes it feasible, convenient, and widely implemented. We will briefly summarize the implementation steps. The first step is to determine the highest grayscale intensity value in the image and denote that level as $L-1$. The threshold K is then calculated by considering each gray level from 0 to L-1. Then the threshold probability is calculated and summed by the weight.

The average gray level of the pixel μ_i is then calculated as the following:

$$\begin{aligned} \omega_2 &= \sum_{i=k+1}^{l-1} p_i \\ \mu_1 &= \frac{1}{\omega_1} \sum_{i=k}^{L-1} ip_i \\ \mu_2 &= \frac{1}{\omega_2} \sum_{i=k}^{L-1} ip_i \end{aligned} \tag{1}$$

The overall gray value of the image μ is given by $\mu = \sum_{j=0}^{i-1} ip_i$. Follow-on stages calculate the variance σ_B and finally the maximum threshold T.

$$\sigma^2 = \omega_1(\mu_1 - \mu)^2 + \omega_2(\mu_2 - \mu)^2 \tag{2}$$

The optimal threshold is obtained by maximizing σ^2.

In multiple/bi-modal thresholding, multiple threshold values such as T0, T1, T2, and T3 exist. Calculation of these levels permits the subsequent multiple category image representation. For example, if a segmented image containing three levels is desired, the output image $B(x, y)$ can be obtained from the pixels of an input image $A(x, y)$ using the following formula:

$$B(x,y) = \begin{cases} m & \text{if } A(x,y) > T_1 \\ n & \text{if } T_0 < A(x,y) \leqslant T_1 \\ 0 & \text{if } A(x,y) \leq T_0 \end{cases} \tag{3}$$

Threshold values can be calculated from the peak values of the image histogram when obvious differences exist in the gray levels of the background and foreground. Both the object and the background contribute to peaks in the histogram. The boundary between them produces a valley. Image segmentation yields perfect results when the segment threshold is at the valley. The threshold method is advantageous because of its simplicity and faster-operating speed. When both the target and the background have high contrast, one can easily obtain the segmentation effect [5]. However, the technique is not without limitations. First, this technique does not provide accurate results for image segmentation when grayscale differences are insignificant. The underlying reason is that it only considers the pixel intensity information and ignores the spatial information contained in the image. Its sensitivity to grayscale unevenness and noise explains why it is fused with other methods to process images [1]. Additionally, in cases requiring more than two segments, the multiple threshold method is not applicable for images with low cluster variances.

Although both the maximum variance and bi-modal method take a short time, the former offers a more robust algorithm because it can segment the foreground from the background faster and more accurately when dealing with images where image contrast is not obvious.

As mentioned above, another limitation of threshold-based methods is that they tend to focus on intensity alone and ignore the relationship among pixels. This is especially problematic in cases where it is not immediately obvious that the identified pixels are contiguous. There is also the possibility of including extraneous pixels which are not part of the target region. Similarly, one can easily miss isolated pixels in the target region. The effects worsen as noise increases because the intensity of the pixel does not necessarily depict normal intensity [15]. Thus, thresholding can lead to too much information loss or the inclusion of an excess number of extraneous pixels. Over and above, in global thresholding, changes in the illumination may make some parts darker and others brighter in ways unrelated to the objects within the image [16]. This challenge is addressed by the inclusion of a variable threshold applied across the image.

B. Edge-based Segmentation

Edge-based segmentation is an image processing method based on identifying object boundaries or edges in an input image. In almost all cases, this technique works by detecting discontinuities in brightness [17]. The method effectively detects and links edge pixels to form contours.

A major feature of an image is its edges. Edges are a crucial aspect of many computer vision and pattern recognition algorithms. As such, the detection of edges is an essential step in image processing [15]. The process may be enumerated as follows:

(1) The primary stage involves identifying edges present in the thermal image. To achieve this, different algorithms designed for edge detection, like the Canny edge detector, Sobel operator, or Laplacian of Gaussian (LoG), can be employed. However, when it comes to thermal images, only a limited number of these algorithms produce satisfactory outcomes. One such effective combination is the utilization of Gabor with Histogram of Oriented Gradients (HOG) technique [18]. This algorithm analyzes the gradients and extracts features of the image to identify regions of rapid intensity changes, which are indicative of edges.

(2) Edge Linking: Once the edges are detected, the next step is to link or connect the individual edge segments to form continuous boundaries. This can be done using techniques like edge linking by Hough transform, region growing, or contour

tracing algorithms. The goal is to create closed curves or contours that represent the boundaries of the objects or regions of interest.

(3) Edge Refinement: In some cases, the detected edges may contain noise or artifacts. Therefore, edge refinement techniques can be applied to enhance the quality and accuracy of the edges. These techniques may involve smoothing or filtering the edges, filling gaps, or removing small or spurious edge segments.

(4) Region Segmentation: Once the edges are obtained and refined, they can be used to segment the image into different regions or objects. This can be achieved by performing operations such as region growing, active contours (snakes), or graph cuts, which utilize the information provided by the detected edges to partition the image into meaningful segments.

Given that images have many redundant data, Kaganami and Beiji pointed in [19] out that the essential information is on the edges of an image. They correspond to texture, object boundaries, as well as changes in surface orientation [15]. In essence, an edge usually corresponds to points in the image wherein the grayscale values differ considerably from pixel to pixel. For this reason, detecting edges helps to extract valuable image feature information in regions in which there are sudden and rapid alterations [20].

Finally, edge detection is an integral step toward understanding the characteristics of an image. Edges have important features and contain information that is meaningful for determining the spatial relationship of neighboring pixels. They can be used to decrease significantly the amount of memory required to store the image, filter out less pertinent information, and preserve the vital structural properties of the image. We will explore some edge detection methods in the following sections.

1) Gradient Edge Detection Method: Various methods in the literature use convolutional kernels to extract edge features from images. However, most of them belong to two groups: gradient-based–methods and Laplacian-based methods. Gradient-based methods, as Jahne mentioned in [15], detect the edges of an image by searching for both the minimum and the maximum values in the image's first derivative. For instance, the popular Sobel, Prewitt, and Roberts operators detect horizontal and vertical edges of an image based on the value of this derivative. Appropriate thresholding can be used in separating sharp edges [19]. As an edge-detection method, the Sobel edge operator shown in equation 4 carries out a two-dimensional spatial gradient measurement on a particular image and hence emphasizes regions of high spatial frequency, which correspond to the image edges. This operator finds the estimated absolute gradient magnitude at every point in an input grayscale image [21]. Theoretically, the Sobel operators are two 3×3 convolution kernels. One kernel is essentially the other kernel rotated by ninety degrees. The Sobel operator is illustrated in the following kernels:

$$G_x = \begin{bmatrix} +1 & 0 & -1 \\ +2 & 0 & -2 \\ +1 & 0 & -1 \end{bmatrix} \; and \; G_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \tag{4}$$

The Prewitt operator computes the maximum response of a set of convolution kernels to find the local edge orientations for every pixel. It is suitable for estimating both the orientation and magnitude of the edge of an image [20]. For this operator, one kernel is sensitive to image edges in the horizontal direction and the other to the vertical direction. The directional kernels are illustrated below:

$$G_x = \begin{bmatrix} +1 & 0 & -1 \\ +1 & 0 & -1 \\ +1 & 0 & -1 \end{bmatrix} \; and \; G_y = \begin{bmatrix} +1 & +1 & +1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix} \tag{5}$$

The Kirsch edge detector uses four filters to detect edges. These filters are essentially a rotation of a basic compass convolution filter [20]. Kirsch convolution kernels are shown below:

$$\begin{gathered} N = \begin{bmatrix} +5 & +5 & +5 \\ -3 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix}, \; W = \begin{bmatrix} +5 & -3 & -3 \\ +5 & 0 & -3 \\ +5 & -3 & -3 \end{bmatrix} \\ S = \begin{bmatrix} -3 & -3 & -3 \\ -3 & 0 & -3 \\ +5 & +5 & +5 \end{bmatrix} \; and, \; E = \begin{bmatrix} -3 & -3 & +5 \\ -3 & 0 & +5 \\ -3 & -3 & +5 \end{bmatrix} \end{gathered} \tag{6}$$

The direction of the edge operator is defined by the mask that produces the maximum edge results.

2) Laplacian Edge Detection Method: The Laplacian method detects the edges by looking for zero crossings in the second derivative of the image's pixel intensity values. Common approaches include the Laplacian-of-Gaussian (LoG) and Marr-Hildreth [22].

To find the edges of an image, the Marr-Hildreth method of edge detection will first filter the image with the LoG filter matrix, which is calculated using the input value of the standard deviation [19]. The standard deviation value determines the filter matrix's width. It also controls the amount of smoothing that the Gaussian component produces. The LoG filtering then smooths the image and enhances all of its edges. The Laplacian of Gaussians response can be estimated by convolving the image with the kernel 7.

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix} \; and \; \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix} \tag{7}$$

As soon as filtering is completed, edge localization is processed by finding zero crossings at every pixel for every direction [21]. Overall, Marr-Hildreth edge detection is used in finding edges through second-order differentiation. In most edge-detection approaches, the main idea is to compute local image change indicators, which include both first-order and second-order derivatives. In image processing, the gradient is the first-order derivative of choice, and it could be utilized in detecting the presence of an edge in an image [21]. Conversely, second-order derivatives are usually calculated with the use of the Laplacian. Notably, the second derivative's sign determines if the pixel of an image is on the light or dark side of an edge [22].

3) Canny Edge Detection: Canny edge detection, which was introduced by J Canny in [23] is a multi-stage approach to detect edges in images using the gradient calculated by the Sobel operator in the X and Y direction followed by non-max suppression, double thresholds, and edge tracking by hysteresis. As the case for any gradient operation, Gaussian smoothing is a critical preprocessing step since all gradients are sensitive to noise. Then the intensity of the edges is calculated by finding the gradient in the image by convolving the Sobel kernel in (4) in the x and y directions. The magnitude matrix G and the gradient slope θ is calculated as the following:

$$\|G\| = \sqrt{I_x^2 + I_y^2} \tag{8}$$

$$\theta = arctan(\frac{I_y}{I_x}) \tag{9}$$

In the next step, non-maximum suppression uses the pair of magnitude and direction of the gradient to find the most intense pixel in the direction of the gradient θ, and the rest of the less-intense pixels are removed or set to zero. This will result in thinner edges with varying edge intensities. The double-threshold stage suppresses false-edge pixels, and eliminates variations in edge intensities. In the final step, the edges pixels are connected by applying hysteresis. Low intensity pixels that fall between string edges are considered strong while the ones with no neighboring edge-pixels are set to zero. This will result in final edge array. Canny edges operate on grayscale images. In the results section, we demonstrate how Canny edges are highly sensitive to noise and shadowing effects in thermal images.

C. Region-based Segmentation

An image is partitioned into regions based upon the similarity of the pixels. In essence, this technique groups sub-regions or pixels into more prominent regions based on pre-set criteria. The procedure usually begins with a set of seed points. New regions are grown from these points by attaching to every seed those adjacent pixels that have properties comparable to the seed, for instance, particular ranges of gray level or intensity. In other words, the region growing image segmentation approach entails growing regions by recursively including nearby pixels which are similar and linked to the seed pixel [24]. Notably, connectivity is required to ensure that pixels do not connect in different parts of the image.

In region growing, homogeneity of regions is the main criterion for segmentation. The homogeneity criteria are as follows: shape, texture, color, gray level, and model. Pixel aggregation is the simplest of all the region growing approaches. After one region has been fully grown by appending adjacent pixels, another seed pixel that does not yet belong to any region will be chosen and then begins the process once more. The entire process continues until every pixel belongs to some particular region [21]. It is a bottom-up approach. Region growing approach requires human interference in choosing the starting seeds.

1) Split-and-Merge Segmentation: The split-and-merge approach is the opposite of the region growing technique. This approach entails separating the image into regions based on a particular similarity measure. The regions are then merged based upon a different or the same similarity measure [25]. Another name of this technique is quadtree division. Initially, some criteria for what is a uniform area are set. Then, the whole image is split into four sub-images. Every sub-image is checked, and if they are not uniform, they are divided into four new sub-images. After every iteration, the adjacent regions are compared. They are then merged if they are uniform as per the similarity measure. The split-and-merge approach entails splitting an image recursively into smaller and smaller parts until every individual region is coherent and then merging them recursively to produce more significant coherent regions [26].

When merging the regions, the approach can begin with small regions, such as 4 x 4 or 2 x 2 regions, and regions which have similar characteristics, for instance, variance or gray level is then merged [24]. Splitting and merging are usually utilized iteratively.

2) Watershed Segmentation: The term watershed is broadly understood as a ridge that divides areas drained by a variety of river systems. The geographical area that drains into a reservoir or river is known as a catchment basin. Catchment basins and watersheds have a connection to image processing [27]. A watershed transform is a crucial tool that can be used to solve image segmentation problems. The watershed transform method grows regions of pixels around an image's local minima. It ensures that the boundaries of nearby areas lie by the side of the crest lines of the gradient image. This method of image segmentation combines features of both the region-based and edge-based segmentation methods. An image in watershed segmentation is considered as a topographic landscape that has valleys and ridges. The landscape's elevation values are defined by their gradient magnitude or gray levels of the respective pixels. The watershed transform decomposes a given image into catchment basins. A catchment basin, for every local minimum, consists of all the points whose path of steepest descent ends at this minimum [28] similar to the previous example. Basins are separated from each other by watersheds. The watershed transform decomposes an image; hence it allocates every pixel to a watershed or a region. Numerous small regions come up with noisy medical image data, and this is typically referred to as the over-segmentation problem [27]. It is the main drawback of the watershed segmentation approach.

The advantage of region-based image segmentation is that region-based methods are usually better in noisy images, where detecting borders is complex. Moreover, region-based image segmentation approaches tend to be more robust than edge-based approaches because regions typically cover more pixels than edges. Hence the scientist has more information available to characterize his/her image. Furthermore, when detecting a particular region, the scientist can utilize texture which is difficult whenever one deals with edges [26]. In addition, region growing techniques usually give good image segmentation, which matches well with the observed edges. However, the disadvantage is that the output of region-growing methods is either too few regions (under-segmented), or too many regions (over-segmented) [25]. Objects such as quantum semiconductor dots, DNA micro-array elements, blood cells, toner spots on a printed page, or any other type of object that may span several disconnected regions cannot be found.

Also, region-based segmentation algorithms are generally more complex than edge-based approaches and multiple other image segmentation methods [26]. The other shortcoming is that the regions obtained in region-based segmentation strongly depend on the initial pixel chosen and the order in which the border pixels are examined. Furthermore, the results are susceptible to the threshold value.

Visualizing the watershed: the image on the left can be topographically represented as the image on the right.

D. Clustering-based Segmentation

Clustering is another powerful image segmentation technique. It is an unsupervised learning task that involves identifying a finite set of clusters to classify the pixels in a digital image. Cluster analysis entails partitioning an image data set into several disjoint clusters or groupings [29]. During the partitioning, two criteria must be maintained, namely low coupling property and high cohesive property. When processing an image, its features are first extracted and then put together into properly-separated clusters based on each class of an image [30]. Notably, the clustering algorithm aims at developing the partitioning decisions based upon the first set of clusters updated following every iteration [31]. The number of clusters in these clustering-based approaches is referred to as priors, and image pixels are classified into suitable clusters based upon the principle of inter-cluster similarity minimization or intra-cluster similarity maximization. There are two main categories of clustering-based segmentation algorithms, namely soft or fuzzy clustering and hard clustering.

E. Fuzzy C-Means

The Fuzzy C-Means (FCM) clustering algorithm was conceptualized in the year 1981 by Jim Bezdek. It is undoubtedly the most common soft clustering approach. It is a clustering method that allows one piece of data to belong to at least two clusters. It is an unsupervised clustering algorithm. Through FCM, an image is segmented by grouping pixels with identical or almost identical values into one cluster, in which every group of pixel's values belonging to one cluster are similar to each other and differ from pixel's values belonging to other clusters [32]. The clusters represent the segments of the image that has been segmented to indicate group membership. Notably, the FCM algorithm is an iterative method of clustering which yields an optimal c partition by reducing the weighted within-group sum of squared error objective function [33]. The algorithm is based upon minimization of the objective function shown below:

$$J = \sum_{i=1}^{N} \sum_{j=1}^{C} u_{ij}^{m} \parallel x_i - c_j \parallel^2 \text{ for} 1 \leq m < \infty \qquad (10)$$

In equation 10, m is a real number greater than 1, u_{ij} is the member of the pixel value x_i in the cluster j. While c_j is the center of the cluster and x_i is the pixel intensity measured data. We use $\parallel * \parallel_p$ to denote the $p-th$ norm used to express the similarity between the pixel intensity and the center of the clusters [34]. The pixel intensity memberships u_{ij} are calculated as follows:

$$u_{ij} = \frac{1}{\sum_{k=1}^{C} (\frac{\|x_i - c_j\|}{\|x_i - c_k\|})^{\frac{2}{m-1}}} \qquad (11)$$

While the centers of cluster values are calculated as the following:

$$c_j = \frac{\sum_{i=1}^{N} u_{ij}^{m} \cdot x_i}{\sum_{i=1}^{N} u_{ij}^{m}} \qquad (12)$$

In equation 11, k is the steps in the iteration. The procedure will converge when the stopping criteria σ is reached [33].

$$\sigma < \parallel U^{(k+1)} - U^{(k)} \parallel \qquad (13)$$

To summarize, the FCM algorithm starts by initializing membership matrix $U^{(0)}$, then calculates the cluster centers vectors c_j. It then updates the values of the membership based on the new cluster centers using equation 11. The algorithm then makes the decision stop if the stopping criteria are met, otherwise calculate the new cluster centers, and begin the process again.

The main advantage of the FCM algorithm is that it is capable of preserving a lot more information than other clustering algorithms. Consequently, it also provides better results than other algorithms such as K-Means.

algorithm and k-nearest neighbors (KNN) algorithm [32]. In addition, the algorithm is renowned for giving the best result for overlapped data sets. Unlike the KM algorithm in which a data point has to belong only to a single cluster center, a data point in FCM clustering is allocated membership to every cluster center, and hence data point can belong to multiple cluster centers [33]. Finally, we mention that another major advantage of the FCM algorithm is computational efficiency. It is widely utilized in the medical field for soft segmentation, such as brain tissue models.

We end this section by mentioning a few shortcomings of the FCM method. First, the algorithm can be sensitive to image noise. It does not consider the pixels' spatial information and therefore can produce excessive output result variance in the presence of noise. The result is somewhat inaccurate image segmentation [34]. Another shortcoming is that the FCM algorithm is time-consuming due in large part to its iterative nature. Moreover, Euclidean distance measures with the Fuzzy c-means algorithm could unequally weigh underlying factors [35]. Besides, although better results can be obtained with lower values of σ, these are obtained to the detriment of more iterations [33]. A priori specification of the number of clusters is also listed as a limitation of the method, so we repeat it here to inform the reader.

F. Texture Based

A texture is broadly understood as the regular repetition of a particular pattern or element on a surface. It represents aspects of the surface pattern, including regularity, directionality, color, brightness, and coarseness[36]. It is utilized in identifying dissimilar non-textured and textured areas in an image, segmenting/classifying distinct texture areas in an image, and extracting boundaries between major texture

regions [37]. An image is partitioned into several regions with dissimilar textures containing a comparable group of pixels during texture segmentation. In essence, a textured image is segmented into various regions that have similar patterns. Segmentation of textures necessitates the choice of good texture-specific features with excellent discriminating power. In general, techniques for extracting texture features could be categorized into three main classifications: spectral, structural, and statistical. In spectral techniques, the textured image, as Madasu and Yarlagadda pointed out in [38], is changed into the frequency domain. After that, extract the texture features can be carried out by assessing the power spectrum. In structural–based feature extraction techniques, the fundamental facet of texture, known as texture primitive, is utilized in forming more intricate patterns of texture through the application of grammar rules that stipulate how texture patterns are generated. Lastly, in statistical techniques, texture statistics, for instance, the moments of the gray-level histogram, are founded upon gray-level co-occurrence matrix and are calculated for discriminating different textures [38]. Over the years, many different methods have been developed for texture-based segmentation. The main ones include Gabor filters, Markov random fields, and wavelets.

1) Gabor Filter: A Gabor filter essentially refers to a combination of a sinusoidal term and a Gaussian filter. Dennis Gabor conceptualized this method, and it is a linear filter. It is notable that frequency and orientation representations of Gabor filters are comparable to those of the human visual system and are suitable for texture discrimination and representation [39]. A two-dimensional (2D) Gabor filter in the spatial domain is a Gaussian kernel function modulated by a sinusoidal plane wave. In 2D, a Gabor filter is as illustrated in equation 14.

$$g_{\lambda,\theta,\psi,\sigma,\gamma}(x,y) = exp(-\frac{x^2+\gamma^2y^2}{2\sigma^2})cos(2\pi\frac{x}{\lambda}+\psi) \quad (14)$$

In this equation, λ represents the wavelength of the cosine factor, θ represents the orientation of the normal to the parallel stripes of a Gabor function in degrees, ψ is the phase offset in degrees, and γ is the spatial aspect ratio indicating the elliptical nature of the Gabor function support, and σ is the standard deviation of the Gaussian that determines the (linear) size of the receptive field.

While the sinusoidal component of the Gabor filter provides the directionality, the Gaussian provides the weights. The impulse response of the Gabor filter, as Haralick pointed out in [36], is defined by a harmonic function multiplied by a Gaussian function. A Gabor filter applies to a wide range of image-processing applications. Aside from texture segmentation, it can also be applied to image representation, retina identification, edge detection, and document analysis [36]. One of the advantages of Gabor filters is that they satisfy the minimum space-bandwidth product according to the uncertainty principle. As such, these filters provide simultaneous optimum resolution in both the spatial-frequency and space domains. They are utilized in solving problems that involve intricate images comprising textured regions [36]. Texture segmentation with the use of Gabor filters involves three steps. In the first step, a filter bank is used to decompose the input image, using the equation 14.

The second step is feature extraction. The following nonlinear sigmoidal function that saturates the output of the filters is used in this step:

$$tanh(\alpha t) = (\frac{1-e^{-2\alpha t}}{1+e^{-2\alpha t}}) \quad (15)$$

Where σ is the standard deviation that determines the receptive window size.

Lastly, the pixels in the Gabor responses are grouped together using a clustering algorithm such as K-Means.

2) Markov Random Fields (MRF): Markov Random Fields (MRF) is a highly sophisticated texture-based segmentation method. It is a probabilistic model. Regions in natural images are usually homogeneous. Pixel homogeneity means that adjacent pixels often have similar properties. For instance, these properties include common characteristics such as texture, color, and intensity. MRF captures such contextual constraints. MRF-based segmentation approaches have been extensively utilized for classification and segmentation in remote sensing applications [40]. MRF is extensively studied and also has a solid theoretical background. According to [40], the MRF segmentation can only be applied to a Markovian image. A Markovian image is an image where the probability distribution of gray levels depends on the neighboring pixels' gray levels, and it is represented by Gibbs fields. The conditional probability for the pixel Z_i with a grey value of g_i belonging to a cluster of pixel values depends on the neighboring pixels Z^i with pixel values of g^i. It is denoted as the following:

$$P(Z_i = g_i|Z^i = g^i) = \frac{1}{S}e^{-H(g_i,G^i)} \quad (16)$$

$$and$$

$$S = \sum_{g=0}^{G} e^{-H(g_i,G^i)} \quad (17)$$

The partition sum S is calculated by summing the energy function of the Markov random fields for the partition. This characterization of the energy function is defined by the parameter vector $\theta = [b_0, b_1.....]^T$. The parameter vector θ is used for the segmentation and characterization of texture.

G. Deep Unsupervised Segmentation Models

In recent years, image segmentation has attracted interest in computer vision research. Object detection, texture recognition, and image compression are some applications of image segmentation. A set consisting of pairs of images and pixel-level semantic labels, such as street or car, is used to train supervised image segmentation. In contrast, unsupervised image segmentation is used to predict more general labels. However, there are no training images or ground truth labels for pixels in unsupervised image segmentation. Therefore, once a target image is input, the pixel labels and feature representations are jointly optimized, and the gradient descent updates their parameters. In [41], the proposed approach, label prediction

and network parameter learning are alternately iterated to meet the following criteria:

1) Pixels of similar features should be assigned the same label.
2) Spatially continuous pixels should be assigned the same label.
3) The number of unique cluster labels should be large.

In order to satisfy these criteria, Wonjik et al. present a CNN-based approach that optimizes both feature extraction and clustering functions at the same time [41]. They proposed a novel end-to-end differentiable network of unsupervised image segmentation, and in order to enable end-to-end learning of a CNN, an iterative approach to predict cluster labels using differentiable functions has been proposed. This study extends the previous research published (ICASSP) [42]. In the previous work, superpixel extraction using simple linear iterative clustering was employed for criterion (2) from the criteria mentioned above. However, the previous algorithm had a limitation that the boundaries of the segments were fixed in the superpixel extraction process. In this study, a spatial continuity loss is proposed as an alternative to mitigate the limitation mentioned above. Moreover, they presented an extension of the proposed method for segmentation with scribbles as user input, which showed better accuracy than existing methods while maintaining efficiency. In addition, they introduced another extension of the proposed method: unseen image segmentation by using networks pre-trained with a few reference images without re-training the networks.

1) Differentiable Feature Clustering: The following is a description of the picture segmentation problem that has been solved. For the sake of simplicity, let $(\{\})$ denote $(\{\}_n^N = 1)$ Unless otherwise stated, where N is the number of pixels in input color image $I = V_n \in R^3$. Consider $(f : R3 \rightarrow R_p)$ be a function for extracting features. And $(X_n \in R_p)$ group of p-dimensional feature vectors of image pixels. By using $C_n = G(X_n)$, cluster labels $C_n \in Z$ has been assigned to all of the pixels, where $g : Rp \rightarrow Z$ is a mapping function. G can be an assignment function that returns the label of the cluster centroid that is closest to X_n in this case. The equation mentioned above is used to derive C_n in the scenario when f and g are fixed. In contrast, if f and g are trainable but C_n is fixed, the equation, as mentioned earlier, can be considered a conventional supervised classification issue. If f and g are differentiable, the parameters for f and g can be optimized using gradient descent. Unknown C_n are predicted in this work while training the parameters of f and g in an entirely unsupervised way. The following two sub-problems were addressed to put this into practice: prediction of the optimal C_n with fixed f and g, and training of the parameters of f and g with fixed C_n. In particular, the three criteria presented in Section I are mutually exclusive and can never be ultimately achieved. Applying K-means clustering to X_n for criterion (a), performing graph cut algorithm using distances to centroids for (b), and finding k in K-means clustering using a non-parametric technique for (c) is one feasible solution for tackling this problem utilizing a traditional method (c). However, because these traditional approaches are only applicable to fixed X_n, the solution may be suboptimal. As a result, a CNN-based algorithm is presented as a solution. All of the requirements above are satisfied by jointly optimizing the feature extraction functions for X_n and C_n. An iterative strategy to forecast C_n using differentiable functions is suggested to enable end-to-end learning of a CNN. The input image I was fed into the CNN to extract deep features X_n using a feature-extraction module. The response vectors Rn of the features in q-dimensional cluster space were then calculated using a one-dimensional $1D$ convolutional layer, where $q = 3$ in this example. The three axes of the cluster space were represented by $z1$, $z2$, and $z3$. The response vectors were then standardized across the cluster space's axes using a batch normalization method. Furthermore, cluster labels C_n have been established by utilizing an $argmax$ function to give cluster IDs to response vectors. The feature similarity loss was then computed using the cluster labels as pseudo targets. Finally, the spatial continuity loss and the feature similarity loss have been computed and backpropagated.

2) Superpixel Learning: Ilyas et al. propose a novel approach for unsupervised segmentation in using superpixels within a CNN framework in [43]. Superpixels are the outcome of perceptual pixel grouping, or, to put it another way, the effect of image over-segmentation. Superpixels contain more information than pixels and match with image borders better than rectangular image patches. The local contrast and distance between pixels in the image's RGB color space are used by superpixel extraction methods. In [43] the authors we extract P superpixels that are more detailed and unique in the input image. After that, each pixel in each superpixel is given the same semantic name. The fewer iterations the CNN must do to produce the final segmented image, the finer the pixels generated by the technique. Too many generated categories (superpixels) will cause the CNN to produce more iterations. To avoid similar situations, input images are pre-processed image by applying contrast enhancement and blurring. Many structures use the simple linear iterative clustering (SLIC) methodology to produce superpixels. However, Ilyas et al. chose the Felzenswalb algorithm because it utilizes a graph-based image segmentation method. In comparison to the other algorithms, this one does an excellent job with image details. Moreover, its time complexity is linear, and it is quicker than the other available methods.

In their approach, Ilyas et al. computed the n-dimensional feature vector from this RGB image through their network's N convolutional blocks. SE-ResNet (detailed later) is the first block, followed by batch normalization and ReLu activation. The dimensions with the highest value were then taken from the feature vector output of the last convolutional block. As a result, we were able to extract the labels from the resulting feature vector. To achieve feature recalibration, we used the bespoke squeeze and excitation networks (SE-Net) initially developed by Jie Hu et al. In order to obtain a SE-ResNet block. We chose to combine SE-Net with ResNet because of its increased representational power. Moreover, we name it SE-Block for simplicity of notation. CNN's extract hierarchical information from images using convolutional filters. Deeper layers detect more abstract features and geometry of the objects present in the images, whereas shallow layers find trivial features from contexts such as edges or high frequencies. Each phase extracts more and more critical information to complete the work at hand at each phase efficiently. In SE-Net, each output channel is weighted adaptively, which is the significant difference between SE-Net and Normal convolutional networks. We add a single parameter to each channel and shift it

linearly based on how relevant each channel is. This is done by obtaining a global understanding of each channel by squeezing the feature maps to a single numeric value using global average pooling (GAP). The results go through the neural network's two fully connected (FC) layers, which produce a vector of the same size as the input. Each original output channel may now be scaled based on its relevance using this n-dimensional vector. As the last step, we utilized K-means to eliminate noise from the final segmented image. In order to apply K-means, we have to find the number of K, which represents the number of clusters. Because of the unsupervised scenario, we do not know how many segmented areas will be in the final segmented image. So, in order to solve this issue, we count the number of disjointed segmented regions in the final segmented image and assign that value to K.

IV. Pre-evaluation

A. Dataset

For this study, we will be using the ADAS dataset provided by FLIR. This dataset contains 8-bit and 14-bit LWIR images and non-annotated RGB images of the same scenes for reference. The dataset was collected by mounting an infrared camera next to a true color camera with center lines approximately 2 inches apart [44]. The two cameras were mounted on a vehicle driving around, collecting synced segments of video and images in Santa Barbra, CA streets and highways. The image capture rate is two frames per second, and the rate of the video is at 30 fps. The infrared frames have a resolution of 640×512 with a 45-degree horizontal field of view and a 37 vertical field of view. The RGB images have 1280×1024 with a field of view set to match the infrared camera. The dataset contains 10,228 synced frames and includes a variety of categories/labels, such as, persons, cars, bicycles, dogs. The demonstrated test cases in the results table are selected by the dynamic pixel value range. for example, the road image has very low dynamic range i.e. all pixel values are limited to a very small number of bins in the histogram while other images have wider ranges. The FLIR dataset contains labels of bounding boxes of several object used for training object detection models. However, we do not employ any of the bounding box labels or details. the dataset is merely chosen since it provides pairs of RGB and thermal images with relatively high resolution. to that point there is also the KIAST dataset and several other face datasets.

B. Preprocessing

First, the image was sharpened to give each item in the image a clear border in order to make a higher-quality image. The image then applied to bilateral filter which reduced unnecessary noise while maintaining the sharpness of the object edges. The filter can be applied in a variety of sizes $n \times n$. We avoid using a values higher than n = 5, since this would result in extreme smoothing and leads to lose a lot of useful information.

C. Evaluation Methods

In the field of image processing, evaluating the performance of a segmentation algorithm is a crucial step. The primary key in evaluating segmentation algorithms is how each method performs in a system or a specific application. For example, in some object detection and tracking applications, the evaluation of how well the segmentation algorithm performs is determined by how well the approach can distinguish the target object from the rest image being considered the background. After extracting the object from the image, the image is furtherly processed . In this case, the target is measured and compared with the ground truth, and the result is evaluated. In their paper, Zhang et al. classify and discuss assessment methods of image segmentation [45]. Additionally, the difference between supervised and unsupervised evaluation methods is examined in detail. In [46], a thorough study about the evaluation approaches in different applications is provided. In this paper, we will provide a qualitative evaluation of the segmentation results for each algorithm and visually compare the results. Furthermore, we will provide a quantitative and analytical evaluation of each algorithm using a semi-supervised approach.

The results reported in this study are calculated using the Dice index, Specificity, Sensitivity, and the Jaccard index as demonstrated in equations Equations (18) to (21).

$$Dice = 2 \times TP/(2 \times TP + FP + FN) \tag{18}$$

$$Specificity = TN/(TN + FP) \tag{19}$$

$$Sensitivity = TP/(TP + FN) \tag{20}$$

$$Jacc = TP/(TP + FN + FP) \tag{21}$$

The Dice index is the intersection between the generated segmentation and the ground truth given in 18. The specificity 19 is the correctly assigned pixels in the image. The sensitivity is the number of uniformly distributed pixels object pixels can be calculated as shown in equation 20. Equation 21 is the Jaccard index which is the relation between the two segmentations, the predicted and the ground truth.

V. Evaluation

A. Threshold

Even though we already know that threshold or image "binarization" does not make sense for this application, we have implemented it as an essential step compared to the other segmentation techniques investigated in this study. The optimal number of thresholds for each image is determined by counting the number of peaks in the histogram. Here, we assumed that our objects have uniform temperatures and that this results in constant pixel values across a single object. This assumption means that the significant peaks will determine the optimal number of thresholds in the image. In order to standardize the peak finding process, a median filter was applied to the frames to provide a uniform range of pixel values. Since all the different techniques of locating the thresholds in the histogram returned close results, we will discuss and calculate the accuracy Otsu's approach since it is the commonly used approach and the most robust.

While determining the optimal number of thresholds, we assumed that objects with uniform temperatures create homogeneous regions or segments. Realistically, objects normally do not have consistent surface temperatures. This temperature discrepancy and environmental and sensor noise lead to the common characteristic of thermal images not containing well-defined regions. Therefore, it causes the thresholding process to often fail when dealing with a histogram with a small variance or a histogram with its peaks concentrated in a small portion. sample result is demonstrated in Fig. 1.

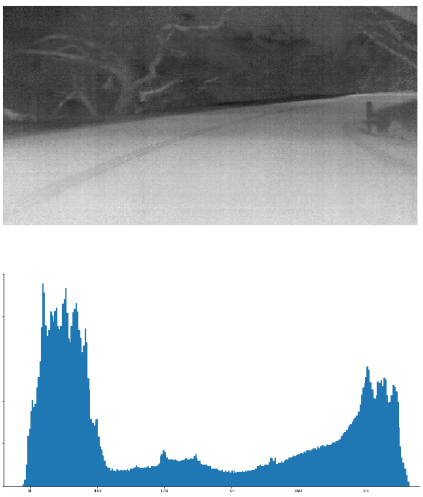


Fig. 1. This histogram has two major peaks and several local peaks which will cause the thresholding process to fail.

In Table I, we demonstrate more examples of the binarized images using Otsu's thresholds.

B. Region and Edge-based Segmentation

When applying edge detectors to thermal images, we notice the overlapping objects, although not at the same depth, with the same pixel values are grouped and have no separating edges between them, as demonstrated in the Fig. 2.

Fig. 2. Over-lapping objects of different depths.

In the case of applying the Canny edge detector, in the active regions of the image, the detector returns many false positives due to the variation in pixel intensities. In Fig. 3, the brick road forms multiple closed regions where it could be mistaken for multiple local regions when in reality, they belong to the same object.

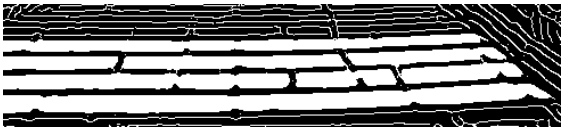


Fig. 3. Over-segmentation of the brick road due to visible gradient in the temperatures.

Watershed segmentation relies on finding the topographic elevation in the image intensities. We notice that watershed segmentation provides the best results when there is a significant disparity within a region that contains two objects of similar pixel intensities. But it also causes over-segmentation in other cases where the same object contains prominent edges. As shown in Table I, watershed generates qualitatively best results in terms of assigning a uniform labels to objects with respect to their edges and their local maxima.

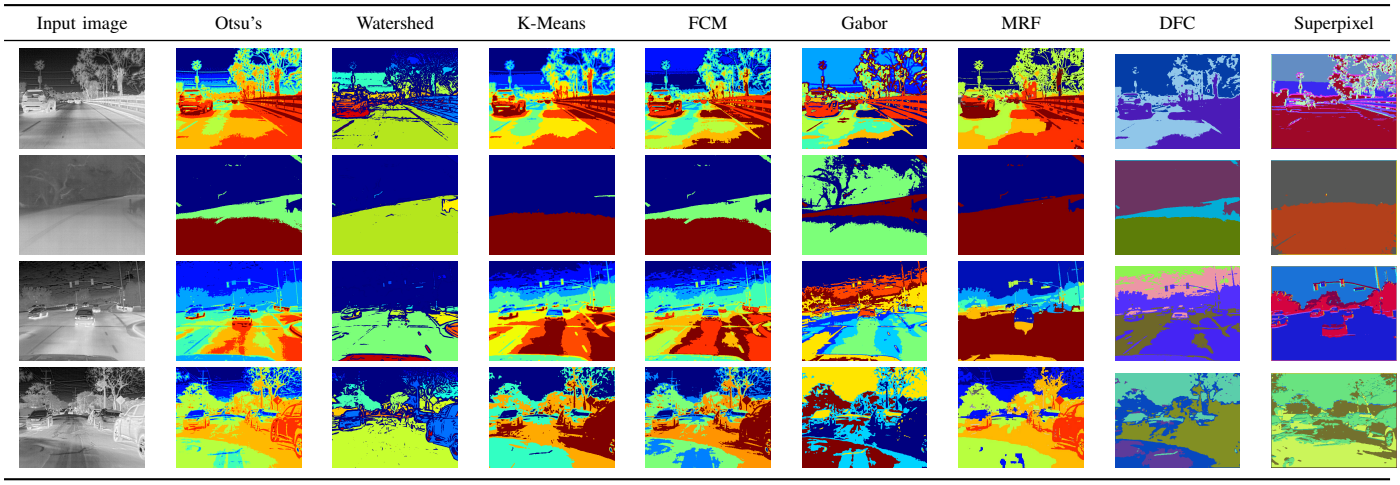
C. Clustering

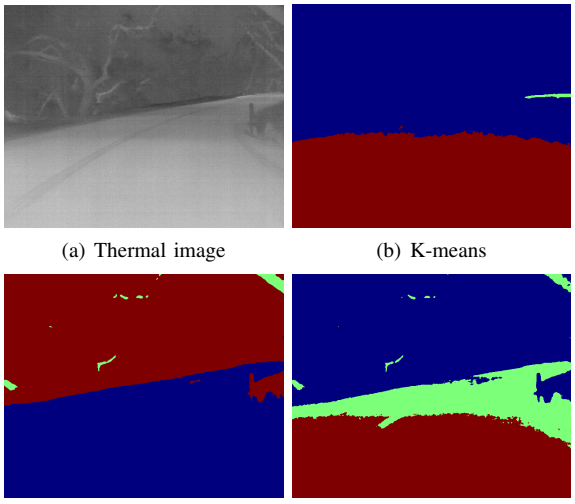
Both K-means and Gaussian mixtures play an essential role in unsupervised machine learning. They offer simple and intuitive approaches to clustering and are straightforward to implement. Typically, they are included in any significant machine learning software package. When K-means was applied to the set of test images, it returned results similar to those achieved by multi-modal thresholds. When K-means fails, GMM comes in. Since K-means can do good enough on most images, we use GMM only for those cases where K-means cannot detect good boundaries. We use all the cluster centers calculated by K-means to initiate the GMM model for the same number of mixtures. Then for each given image, we calculate the probability. We then threshold and normalize them to create a black and white image similar to what we get from K-means. FCM has the disadvantages of sensitivity to initial cluster values, sensitivity to noise, and the solution provided does not consider any relevant spatial information from neighboring pixels. Applying fuzzy clustering on pixel values without any additional features will result in better segmentation when compared to the results from K-Means and multi-modal thresholds, as demonstrated in Fig. 4.

D. Texture Analysis

The results shown provide some insight into how these texture-based feature extraction techniques are performed. The Gabor method performed decently in the given segmentation tasks, although more processing was required to achieve accuracy. Additionally, the Gabor method takes several parameters as initial input to the program, and these parameters require a lot of experimentation and errors. However, the Gabor parameters that have always made the most significant contribution to the method's output were the window sizes. Whether it was the size of the moment mask, the size of the Gabor filter, or the smoothing window after the activation function had been applied, these window sizes caused drastic changes in the results of the segmentation results.

TABLE I. SEGMENTATION RESULTS

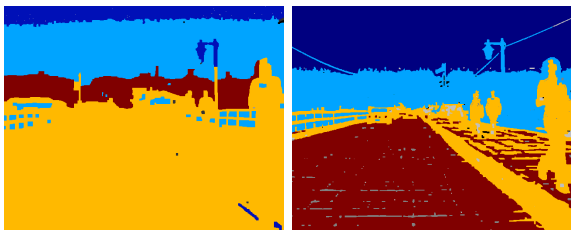




(a) Thermal image (b) K-means

(c) GMM (d) Fuzzy C-Means

Fig. 4. Comparison between clustering methods K-means, GMM, and FCM.



(a) Blind MRF (b) Labeled MRF

Fig. 5. The difference in the performance after providing hard labels for MRF segmentaion. The left image is the blind segmentation result without providing labels while the right image is the result when providing sample segments for each label.

The MRF model did not segment the image properly. Possibly, because exploiting only pixel values does not give enough segmentation power to the model. However, incorporating complex labels of each class's mean and variance provided more accurate segmentation for the labeled classes. Therefore, the aggregate four features: pixel intensity, mean, variance, and the sum of the log of the intensities of neighboring pixels, are used on the MRF model satisfying segmentation. Fig. 5 demonstrates the difference in the performance between the unsupervised segmentation and the hard-labeled segmentation.

E. Unsupervised Deep Learning Models

As shown in the qualitative and quantitative results, unsupervised deep learning models provide similar results to the classical clustering algorithms. This poor performance can be due to the lack of feature representation in the images. If the feature representation is not well-suited to the task or too limited in scope, the model may struggle to accurately segment the image. Another reason would be due to having these models need to be fine-tuned to the dataset used in this study. Also, the choice of hyperparameters would affect the overall performance of these models. There are several hyperparameters involved in unsupervised segmentation models, such as the learning rate, regularization, and optimization method. If the hyperparameters are not chosen correctly, the model's performance can suffer (Table II).

VI. DISCUSSION

This paper reviewed the most common approaches for providing labels for training purposes using unsupervised segmentation algorithms. The first three sections covered the theory behind each approach. In the results section, we quantitatively and visually analyzed each approach and discussed cases where the method failed and the reasoning for the failure. The results indicated that we could not rely solely on pixel values for segmentation, even for such low-rank images as thermal images. Segmentation methods such as thresholds or clustering performed poorly in more complicated scenes with several objects of the same temperature in the scene. Therefore, extra information must be incorporated in the segmentation approach to producing a more accurate result. Approaches that rely on edges to separate different objects fail due to the

TABLE II. QUALITATIVE AND QUANTITATIVE RESULTS

	8-bit Images				16-bit Images				RGB Images			
	Dice	Spe	Sens	Jacc	Dice	Spec	Sens	Jacc	Dice	Spe	Sens	Jacc
Otsu's	0.33	0.80	0.79	0.20	0.68	0.43	0.24	0.63	0.61	0.52	0.55	0.41
Watershed	0.60	0.34	0.92	0.49	0.77	0.47	0.17	0.11	0.71	0.84	0.92	0.67
KM	0.37	0.32	0.14	0.23	0.37	0.32	0.14	0.23	0.40	0.61	9.14	0.33
GMM	0.37	0.32	0.14	0.23	0.71	0.12	0.31	0.78	0.42	0.67	9.14	0.31
FCM	0.34	0.93	0.98	0.21	0.34	0.93	0.98	0.21	0.49	0.54	0.98	0.30
Gabor	0.36	0.31	0.55	0.2	0.39	0.30	0.21	0.19	0.52	0.39	0.55	0.35
MRF	0.35	0.96	0.88	0.21	0.32	0.63	0.52	0.20	0.46	0.60	0.88	0.23
DFC	0.36	0.31	0.55	0.15	0.31	0.55	0.20	0.71	0.44	0.22	0.71	0.20
Superpixel	0.35	0.96	0.88	0.21	0.26	0.88	0.21	0.86	0.58	0.19	0.86	0.47

TABLE III. PERFORMANCE EVALUATION OF STUDIED METHODS FOR ALL THREE TYPES OF INPUT IMAGES

Approach	Time complexity
Otsus	$O(N + L^2)$
Watershed	$O(K \times N)$
K-Means	$O(K \times N \times T)$
FCM	$O(K \times N \times T)$
GMM	$O(N \times K \times D^3)$
Gabor	$O(M^2 \times N^2)$
MRF	$O(N \times M \times K \times T)$

lack of depth information. This issue comes in when there are several overlapping objects with the same temperature in the scene. Finally, we see that texture analysis often delivers the best performance since they consider the spatial relations between neighboring pixels. In the case of Gabor segmentation, this approach requires empirical determination of several parameters to return better results. It is worth mentioning that the enhanced results produced by these texture-based methods are not without significant increases in computational requirements, algorithmic complexity, and significant barriers to real-time implementation.

A. Time Complexity

Table III lists the time complexities for each of the studies algorithms. Where N is number of pixels in the image, L is histogram length, K number of clusters, T is the time to calculate the distance between two objects, D is the problem dimension, and M is the window size. We notice that texture analysis is more complex and require more analysis than thresholding or clustering. It is evident that in order to build a labeling GUI using any of those algorithms, it would need high computing capabilities to make the GUI easy to use and provide results quickly.

VII. CONCLUSION AND FUTURE WORK

In conclusion, this paper has provided a comprehensive review of unsupervised segmentation techniques for long wave infrared (LWIR) images. Through the evaluation and analysis of various methods, several key findings have emerged. Firstly, it is evident that unsupervised segmentation techniques play a crucial role in extracting meaningful information from LWIR images, despite the challenges posed by noise, low contrast, and temperature variations. The reviewed techniques have shown varying degrees of effectiveness in segmenting LWIR images, with some demonstrating superior performance in specific scenarios.

Moving forward, there are several avenues for future research in this domain. Firstly, further investigation is needed to explore the combination of multiple unsupervised segmentation techniques to enhance the overall segmentation accuracy in LWIR images. Fusion methods that leverage the strengths of different algorithms could potentially yield superior results. Additionally, incorporating domain-specific knowledge and priors, such as thermal physics, object characteristics, and context information, may further improve segmentation accuracy and robustness. Furthermore, the evaluation of unsupervised segmentation techniques on LWIR video sequences warrants attention. Temporal consistency and motion information can be leveraged to improve the accuracy of segmentation results over time. Investigating the use of unsupervised segmentation techniques for real-time applications, such as tracking and object recognition, is another area of interest.

In conclusion, this review has shed light on the current landscape of unsupervised segmentation techniques for LWIR images. While notable progress has been made, there is ample room for further exploration and improvement. By addressing the identified research gaps and leveraging emerging technologies, we can advance the state-of-the-art in LWIR image segmentation, ultimately facilitating more effective and reliable analysis in LWIR applications such as surveillance, target detection, and autonomous systems.

ACKNOWLEDGMENT

The authors express their gratitude to themselves for their dedicated efforts in conducting and presenting this remarkable research.

REFERENCES

[1] D. Kaur and Y. Kaur, "Various image segmentation techniques: a review," *International Journal of Computer Science and Mobile Computing*, vol. 3, no. 5, pp. 809–814, 2014.

[2] S. S. Al-amri, N. V. Kalyankar, and K. S. D., "Image segmentation by using threshold techniques," *CoRR*, vol. abs/1005.4020, 2010. [Online]. Available: http://arxiv.org/abs/1005.4020

[3] P. Wang, P. Chen, Y. Yuan, D. Liu, Z. Huang, X. Hou, and G. Cottrell, "Understanding convolution for semantic segmentation," in *2018 IEEE winter conference on applications of computer vision (WACV)*. Ieee, 2018, pp. 1451–1460.

[4] D. D. Patil and S. G. Deore, "Medical image segmentation: a review," *International Journal of Computer Science and Mobile Computing*, vol. 2, no. 1, pp. 22–27, 2013.

[5] D. Li and Y. Wang, "Application of an improved threshold segmentation method in SEM material analysis," *IOP Conference Series: Materials Science and Engineering*, vol. 322, p. 022057, mar 2018. [Online]. Available: https://doi.org/10.1088%2F1757-899x%2F322%2F2%2F022057

[6] J. Ruiz-Santaquiteria, G. Bueno, O. Deniz, N. Vallez, and G. Cristobal, "Semantic versus instance segmentation in microscopic algae detection," *Engineering Applications of Artificial Intelligence*, vol. 87, p. 103271, 2020.

[7] J. Rogowska, "Overview and fundamentals of medical image segmentation," in *Chapter 5*, 2009.

[8] W. ya Guo, X. fei Wang, and X. zhi Xia, "Two-dimensional otsu's thresholding segmentation method based on grid box filter," *Optik*, vol. 125, no. 18, pp. 5234–5240, 2014.

[9] R. Kashyap and P. Gautam, "Modified region based segmentation of medical images," in *2015 International Conference on Communication Networks (ICCN)*. IEEE, 2015, pp. 209–216.

[10] R. Priyadharsini and T. S. Sharmila, "Object detection in underwater acoustic images using edge based segmentation method," *Procedia Computer Science*, vol. 165, pp. 759–765, 2019.

[11] J. Wang, Z. Xu, and Y. Liu, "Texture-based segmentation for extracting image shape features," in *2013 19th International Conference on Automation and Computing*. IEEE, 2013, pp. 1–6.

[12] C. Tian and Y. Chen, "Image segmentation and denoising algorithm based on partial differential equations," *IEEE Sensors Journal*, vol. 20, no. 20, pp. 11 935–11 942, 2019.

[13] F. A. Smith, E. L. Jacobs, S. Chari, and J. Brooks, "LWIR thermal imaging through dust obscuration," in *Infrared Imaging Systems: Design, Analysis, Modeling, and Testing XXII*, G. C. Holst and K. A. Krapels, Eds., vol. 8014, International Society for Optics and Photonics. SPIE, 2011, pp. 148 – 159. [Online]. Available: https://doi.org/10.1117/12.884351

[14] D.-h. Xie, M. Lu, Y.-f. Xie, D. Liu, and X. Li, "A fast threshold segmentation method for froth image base on the pixel distribution characteristic," *PloS one*, vol. 14, no. 1, 2019.

[15] P.-Y. Pai, C.-C. Chang, Y.-K. Chan, M.-H. Tsai, and S.-W. Guo, "An image segmentation-based thresholding method," *Journal of Imaging Science and Technology*, vol. 56, no. 3, pp. 30 503–1, 2012.

[16] A. K. Chaubey, "Comparison of the local and global thresholding methods in image segmentation," *World Journal of Research and Review*, vol. 2, no. 1, 2016.

[17] A. Fabijańska and D. Sankowski, "Segmentation methods in the selected industrial computer vision application," in *Computer Vision in Robotics and Industrial Applications*. World Scientific, 2014, pp. 23–48.

[18] I. Almadani, M. Abuhussein, and A. L. Robinson, "Sow localization in thermal images using gabor filters," in *Advances in Information and Communication: Proceedings of the 2022 Future of Information and Communication Conference (FICC), Volume 1*. Springer, 2022, pp. 617–627.

[19] H. G. Kaganami and Z. Beiji, "Region-based segmentation versus edge detection," in *2009 Fifth International Conference on Intelligent Information Hiding and Multimedia Signal Processing*. IEEE, 2009, pp. 1217–1221.

[20] Z. Hussain and D. Agarwal, "A comparative analysis of edge detection techniques used in flame image processing," *International Journal of Advance Research In Science And Engineering IJARSE*, no. 4, 2015.

[21] R. C. Gonzalez and R. E. Woods, *Digital Image Processing (3rd Edition)*. USA: Prentice-Hall, Inc., 2006.

[22] Z. Lu-Bin and Z. Wei, "Analysis and application of image segmentation and edge detection operator," in *2015 10th International Conference on Computer Science & Education (ICCSE)*. IEEE, 2015, pp. 720–724.

[23] J. Canny, "A computational approach to edge detection," *IEEE Transactions on pattern analysis and machine intelligence*, no. 6, pp. 679–698, 1986.

[24] D. Palaz, M. Magimai-Doss, and R. Collobert, "Joint phoneme segmentation inference and classification using crfs," in *2014 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, 2014, pp. 587–591.

[25] B. L. Price, B. S. Morse, and S. Cohen, "Livecut: Learning-based interactive video segmentation by evaluation of multiple propagated cues," in *2009 IEEE 12th International Conference on Computer Vision*. IEEE, 2009, pp. 779–786.

[26] M. T. Wanjari, V. K. Yeotikar, K. D. Kalaskar, and M. P. Dhore, "Document image segmentation using k-means clustering technique," *International Journal of Electronics, Communication and Soft Computing Science & Engineering (IJECSCSE)*, p. 95, 2015.

[27] J. B. Roerdink and A. Meijster, "The watershed transform: Definitions, algorithms and parallelization strategies," *Fundamenta informaticae*, vol. 41, no. 1, 2, pp. 187–228, 2000.

[28] "The watershed transform: Strategies for image segmentation." [Online]. Available: https://www.mathworks.com/company/newsletters/articles/the-watershed-transform-strategies-for-image-segmentation.html

[29] N. Dhanachandra, K. Manglem, and Y. J. Chanu, "Image segmentation using k-means clustering algorithm and subtractive clustering algorithm," *Procedia Computer Science*, vol. 54, pp. 764–771, 2015.

[30] D.-Q. Zhang and S.-C. Chen, "Kernel-based fuzzy and possibilistic c-means clustering," in *Proceedings of the International Conference Artificial Neural Network*, vol. 122, 2003, pp. 122–125.

[31] K.-S. Chuang, H.-L. Tzeng, S. Chen, J. Wu, and T.-J. Chen, "Fuzzy c-means clustering with spatial information for image segmentation," *computerized medical imaging and graphics*, vol. 30, no. 1, pp. 9–15, 2006.

[32] J. Chen, C. Yang, G. Xu, and L. Ning, "Image segmentation method using fuzzy c mean clustering based on multi-objective optimization," in *JournalofPhysics: Conference Series*, vol. 1004, no. 1, 2018, pp. 012–035.

[33] M. J. Christ and R. Parvathi, "Fuzzy c-means algorithm for medical image segmentation," in *2011 3rd International Conference on Electronics Computer Technology*, vol. 4. IEEE, 2011, pp. 33–36.

[34] H. R. Mohammed, H. H. Alnoamani, and A. A. Jalil, "Improved fuzzy c-mean algorithm for image segmentation," *Int J Adv Res Artif Intel*, vol. 5, pp. 7–10, 2016.

[35] Z. Ji, Y. Xia, Q. Chen, Q. Sun, D. Xia, and D. D. Feng, "Fuzzy c-means clustering with weighted image patch for image segmentation," *Applied soft computing*, vol. 12, no. 6, pp. 1659–1667, 2012.

[36] R. M. Haralick, K. Shanmugam, and I. Dinstein, "Textural features for image classification," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-3, no. 6, pp. 610–621, 1973.

[37] V. V. Bhosle and V. P. Pawar, "Texture segmentation: different methods," *International Journal of Soft Computing and Engineering (IJSCE)*, vol. 3, no. 5, pp. 69–74, 2013.

[38] V. K. Madasu and P. Yarlagadda, "An in depth comparison of four texture segmentation methods," in *9th Biennial Conference of the Australian Pattern Recognition Society on Digital Image Computing Techniques and Applications (DICTA 2007)*, 2007, pp. 366–372.

[39] R. Zwiggelaar and E. R. E. Denton, "Texture based segmentation," in *Digital Mammography*, S. M. Astley, M. Brady, C. Rose, and R. Zwiggelaar, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 433–440.

[40] R. R. A, "Markov random fields (mrf)-based texture segmentation for road detection."

[41] W. Kim, A. Kanezaki, and M. Tanaka, "Unsupervised learning of image segmentation based on differentiable feature clustering," *IEEE Transactions on Image Processing*, vol. 29, p. 8055–8068, 2020. [Online]. Available: http://dx.doi.org/10.1109/TIP.2020.3011269

[42] A. Kanezaki, "Unsupervised image segmentation by backpropagation," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 1543–1547.

[43] T. Ilyas, A. Khan, M. Umraiz, and H. Kim, "Seek: A framework of superpixel learning with cnn features for unsupervised segmentation," *Electronics*, vol. 9, no. 3, 2020. [Online]. Available: https://www.mdpi.com/2079-9292/9/3/383

[44] "FLIR ADAS dataset description," https://www.flir.com/oem/adas/adas-dataset-form/, accessed: 2018-07-26.

[45] H. Zhang, J. E. Fritts, and S. A. Goldman, "Image segmentation evaluation: A survey of unsupervised methods," *computer vision and image understanding*, vol. 110, no. 2, pp. 260–280, 2008.

[46] Z. Wang, E. Wang, and Y. Zhu, "Image segmentation evaluation: a survey of methods," *Artificial Intelligence Review*, vol. 53, no. 8, pp. 5637–5674, Dec. 2020. [Online]. Available: https://doi.org/10.1007/s10462-020-09830-9