The Development of Image Semantic Segmentation

Zinan Li
12011517

***Abstract*—Image semantic segmentation is an important task in computer vision that aims to divide an image into different segments, each corresponding to a specific object or class, by assigning a class label to each pixel. This process is crucial for many computer vision applications as it enables a deeper understanding and interpretation of the image content. Throughout the years, various techniques have been developed for semantic segmentation, starting from traditional methods and evolving to contemporary deep-learning-based approaches. In this essay, an overview of the progress in semantic segmentation will be presented, highlighting the advantages and disadvantages of different methods, and illustrating some representative algorithms. It will be discussed how these techniques have been improved over time to achieve better accuracy and how deep-learning approaches have revolutionized the field. Furthermore, the versatility of these various methods in different computer vision tasks will also be explored.**

***Index Terms*—Artificial intelligence, Semantic segmentation, Computer vision**

I. Introduction

Image semantic segmentation is a task in computer vision that involves dividing an image into different segments, each corresponding to a specific object or class, by assigning a class label to every pixel. This is distinct from traditional image classification, which pertains to the assignment of a single semantic label to the entire image without the responsibility to precisely identify the edges of each object in one image. And that is almost the hardest part, edge detection. In addition, semantic segmentation, as a pixel-specific labeling process, serves as a vital intermediate step in attaining a comprehensive and nuanced understanding of the scene shown in the image. It is a highly demanding job and a crucial intermediate process in many computer vision applications, such as object recognition and tracking (i.e. automatic driving [1], video surveillance [2], video editing [3] [4]), medical imaging [5], robotics, reality augmentation [6], and environmental monitoring.

The history of image segmentation dates back to the early days of computer vision, where thresholding, edge detection, region-growing, fuzzy mathematical methods, and graphic models were commonly used. These early approaches, however, faced several limitations, such as a lack of robustness and the inability to capture fine details. Semantic segmentation has advanced alongside computer vision as a field. Deep learning techniques have completely changed the sector in the last ten years, inspiring the creation of potent architectures including fully convolutional networks (FCN) [21], U-Net [22], and Mask R-CNN [27]. These architectures have been widely adopted in a variety of computer vision applications and have achieved cutting-edge performance on a variety of benchmark datasets.

The evolution of semantic segmentation will be briefly discussed in this essay, starting with early methods and going on to more modern deep learning techniques. We will emphasize the most significant achievements and difficulties in the area of semantic segmentation through this investigation, as well as any potential long-term effects.

II. Early approaches to semantic segmentation

An early survey [7] divided the segmentation procedures into three categories based on the reasoning behind the various segmentation methods: (1) distinctive feature thresholding or clustering; (2) edge detection; and (3) region extraction. In a subsequent assessment, Pal, N. R. and Pal, S. K. generally divided the approaches into two categories: the classical approach and the fuzzy mathematical approach [8]. Minor methods like the Markovian random field and neural network do not fall under these two categories. Under this classification, traditional image processing methods, such as threshold and edge detection, are used in traditional methods for image segmentation to distinguish between an image's foreground and background. These techniques rely on made-up features and presumptions regarding the image data. On the other hand, fuzzy mathematical approaches make use of ideas from fuzzy set theory to model ambiguity and uncertainty in the image data. These techniques sometimes include grouping pixels into several segments based on how similar they are to various cluster centers using fuzzy clustering algorithms, such as fuzzy c-means. Fuzzy mathematical approaches may handle image data with non-uniform distributions and do not make firm assumptions about the content of the images, in contrast to classical methods.

In this essay, we will follow the second classification and provide an overview of the following methods: thresholding (and clustering), edge detection, region growing, fuzzy mathematical methods, and PDE-based segmentation.

A. Thresholding and Clustering

Thresholding, sometimes referred to as binarization, is a fundamental method of image processing that thresholds the intensity values of the pixels in a grayscale image to create a binary image. The foundation of this strategy is the notion of thresholding a picture into background and foreground areas. Indicators of the picture, such as the intensity distribution and desired degree of detail, are taken into account when choosing the threshold value.

Global thresholding, which uses the same threshold value over the whole image, is one of the simplest thresholding techniques. This method is useful for images that have a

clear distinction between the background and the foreground. However, it is sensitive to variations in lighting and contrast, and it has difficulty handling images with multiple classes or regions with varying intensity levels. Opposite to global threshold, local threshold, also called adaptive method is a technique to adaptively threshold an image, by dividing the image into small neighborhoods or sub-regions and computing a threshold value for each neighborhood. It is useful for images with varying illumination or non-uniform backgrounds, where the global thresholding can lead to suboptimal results.

However, for successful operation, both global and local threshold approaches depend on an ideal threshold value. Finding the right threshold value may be challenging and time-consuming, especially when working with photographs that have uneven backgrounds or different lighting. Many automatic threshold approaches, like Otsu's method [9], have been developed to obviate the manual selection of the threshold value. The technique computes the intensity histogram of the picture and uses it to determine the threshold value that optimizes the variation between the pixels in the foreground and background. It does this by iterating over all possible threshold values and calculating a measure of the variance for each one. The threshold value that maximizes the variance is selected as the optimal threshold value.

Although automatic thresholding methods conquer the problems of selection of the threshold value and are more robustness to image variability and less sensitivity to noise. It also has its own limitations such as poor performance in images without obvious peaks and ignoring spatial information [10].

Liu D. and Yu J. have demonstrated that the goal function of Otsu's approach in multilevel thresholding is mathematically comparable to that of the K-means method [11]. Clustering algorithms groups similar pixels within an image based on a specific similarity metric. These metrics can include color, texture, and intensity. Common clustering methods employed in image segmentation include k-means, mean shift, and normalized cut. These algorithms typically involve a two-step process: first, features are extracted from the image, then pixels are grouped based on those features. Both methods fundamentally follow the philosophy of Fisher's discriminative analysis – maximizing between-group variance and minimizing within-group variance. Since clustering algorithms are adaptable and may be used with a wide range of image formats, they can successfully handle complicated photos that contain several objects and backgrounds. However, they are sensitive to initialization, significantly rely on the feature set, and don't have a defined standard for measuring the success. It is significant to remember that there is no one-size-fits-all clustering technique because it depends on the characteristics of the photos and the particular task at hand.

B. Edge Detection

Compared with thresholding methods, edge detection methods focus more on detecting the rapid changes of color or intensity in a local area instead of figuring out an optimal threshold of color or intensity distribution. It segments an image by identifying the boundaries of objects within an image. Edge detection techniques were divided by Davis into two groups: sequential technique and parallel technique [12]. The image is processed sequentially, one pixel at a time, using the sequential technique, also referred to as the "serial" technique. This means that each pixel is analyzed and its value is used to determine the edge strength at that location. This method is relatively simple to implement and is well-suited for small images or images with simple edges. The parallel technique, also known as the "parallel" technique, processes the image in parallel, meaning that multiple pixels are analyzed simultaneously. This is typically done by dividing the image into small sub-regions, called "tiles," and then processing each tile in parallel. This method is more complex to implement, but it is more efficient and can handle larger images or images with more complex edges.

C. Region Growing

Region growing is the third main approach in image segmentation which involves starting with a seed point and iteratively merging adjacent regions that have similar characteristics. Typically, the user's input or the desired class are used to determine the seed point. The concept of region expansion is the joining of regions that share the same characteristics, such as color, intensity, or texture. This approach is able to handle images with multiple classes, but it is sensitive to noise, boundaries with low contrast, and it can be computationally expensive [13]. An example of an algorithm is hierarchical region growth, which starts with the full image as a single region and recursively divides the image into smaller regions depending on some criteria until the desired number of regions or degree of homogeneity is reached.

D. Fuzzy Mathematical Methods

Fuzzy mathematical methods in image segmentation are techniques that use concepts from fuzzy set theory to model uncertainty and ambiguity in the image data. These techniques divide a picture into various sections or segments based on how closely the pixels resemble the various cluster centers.

The fuzzy c-means (FCM) algorithm, a popular technique that can handle picture data with non-uniform distributions, is a version of the k-means algorithm [18]. The FCM algorithm assigns each pixel a membership value between 0 and 1 for each cluster center, indicating the degree to which the pixel belongs to each cluster. Another fuzzy mathematical method is the Fuzzy-Connectedness(FC) algorithm [19], which is based on the idea of fuzzy connectedness among pixels. This algorithm uses a fuzzy relation to define the similarity between pixels and then uses a graph-based approach to segment the image. Fuzzy mathematical methods are particularly useful in image segmentation because they can handle image data with non-uniform distributions and do not make strong assumptions about the image content, which makes them suitable for images with varying lighting, noise, and texture.

E. PDE Based Segmentation

PDE-based methods are often done by formulating an optimization problem that seeks to minimize an energy function defined by a PDE, which can take different forms depending on the specific application. PDEs can be used to model various image features such as edges, textures, and shapes, and can be combined with other methods such as graph cuts or level sets to improve the segmentation accuracy.

The active contours, sometimes referred to as snakes, are one of the most widely used PDE-based image segmentation techniques [14]. The progression of an initial contour approaching the borders of the object of interest is modeled using a PDE by the active contours algorithm. The energy function is described as the product of an external energy component that pushes the contour toward the object limits and an internal energy term that maintains the contour's smoothness and regularity. The Geodesic Active Contours algorithm was also proposed and was based on snakes. [15]. It combines topologically adaptable active contours with geodesic distances. The active contour's progression toward the item of interest's borders is modeled using a PDE in this approach, but the geodesic distance is employed as the distance metric, which allows for the contour to change topology during the evolution.

Another popular PDE-based image segmentation method is the level sets [16]. The level set algorithm uses a PDE to evolve the level set function of an object's boundary over time. The energy function is defined as the sum of an internal energy term, which keeps the level set function smooth and regular, and an external energy term, which drives the level set function towards the object boundaries.

The PDE-based algorithms can also be formulated in different ways, for example, the Chan-Vese Model is a PDE-based model that defines an energy function that is minimized by the desired contour [17]. The energy functional is defined as a combination of the internal energy term which promotes smoothness and the external energy term which promotes fitting the contour to the object boundaries.

The Mean Curvature Flow algorithm is another illustration. The PDE that underpins this technique defines how a surface changes over time in relation to its mean curvature. The algorithm can be used to smooth an initial contour by evolving it according to the mean curvature of the surface. This algorithm is useful when the initial contour is noisy or has small inaccuracies and the goal is to smooth it out.

Finally, it is important to note that PDE-based algorithms can be combined with other methods to improve the accuracy and robustness of the segmentation. For example, the level set method can be combined with graph cuts to improve the accuracy of segmentation by incorporating prior knowledge about the image. Similarly, the active contour method can be combined with level sets to allow for topological changes during the evolution of the contour.

III. Deep Learning Era of semantic segmentation

The advent of deep learning marked a significant turning point in the field of semantic segmentation. After the introduction of convolutional neural networks (CNNs), researchers overcame the limitations of pre-deep learning methods and achieved great accuracy on several standard datasets. Deep learning methods in semantic segmentation can be summarized into 11 categories:

A. Fully Convolutional Networks

In 2015, the concept of Fully Convolutional Networks (FCNs) have been introduced for image segmentation [21]. A convolutional neural network (CNN), for example, may be transformed into a completely convolutional network by swapping out its fully connected layers for convolutional layers. This is the basic idea of FCNs. This makes the network more versatile and adaptive to many sorts of pictures and applications by allowing the network to receive an image of any size as input and create a segmented image of the same size.

In traditional CNNs, fully connected layers are utilized in order to classify input image into a fixed number of classes. However, these layers can only process images of fixed size, and the output is also a fixed-size vector that corresponds to the class scores. To use CNNs for image segmentation, one must either resize or crop the image so that it can be processed by the fully connected layers, or use a sliding window to scan the image and process it in small patches.

FCNs, in contrast, utilize an architecture that allows the network to accept an picture of any size as input and then output a segmented image of the same size. This is accomplished by substituting convolutional layers, which can handle pictures of any size, for the fully linked layers. The use of convolutional layers enables the network to learn spatial hierarchies of features from the input image, and produce a dense prediction for each pixel in the output image.

B. Convolutional Models with Graphical Models

Convolutional models, such as convolutional neural networks (CNNs), are designed to extract features from images using convolutional layers. These models are typically trained to classify images into a fixed number of classes. However, they can also be used for image segmentation by modifying the architecture and the training process.

Graphical models that specify a probability distribution across the picture pixels include Conditional Random Fields (CRFs) and Markov Random Fields (MRFs). These models are typically used to model the dependencies between the pixels, and they can be used to perform image segmentation by finding the most probable configuration of the image pixels.

Combining graphical models and convolutional neural networks (CNNs) can increase the reliability and accuracy of picture segmentation. One method involves using a CNN as a feature extractor, where the CNN is trained to extract features from an image and then these features are fed to a graphical model such a Markov Random Field (MRF) or Conditional Random Field (CRF) to optimize the segmentation. Another approach is to use a CNN to predict the unary and pairwise potentials of a CRF model, which can be used to refine the

segmentation produced by the CNN. Additionally, CNNs can be combined with MRF model, Graph cut algorithm to refine the segmentation produced by the CNN. The graphical models can be used to capture the dependencies between the pixels, while the CNNs can be used to capture the features of the image. By combining these two approaches, it is possible to achieve more accurate and robust image segmentation results.

C. Encoder-decoder Models

Encoder-decoder models are intended to accept a picture as input and produce a segmented version of the same image, with each pixel labeled to indicate the class of the item to which it belongs.

The main idea behind encoder-decoder based models is to use a convolutional neural network (CNN) as an encoder to extract features from the input image, and a transposed convolutional neural network (also known as a deconvolutional network) as a decoder to upsample the features and produce a segmented image. While the decoder is used to boost spatial resolution and line up the output with the input picture, the encoder is used to make the image have less spatial resolution.

Some of the most representative algorithms based on encoder-decoder models are: U-Net: A popular model use encoder and decoder that with skip connections between them to keep the semantic information and improve the accuracy of the segmentation and decrease the number of parameters [22]; SegNet: Another popular encoder-decoder architecture that uses an added max-pooling indices layer to preserve the spatial information during the upsampling process [23];

Encoder-decoder based models have several advantages such as handling images of varying sizes and resolutions, allowing for end-to-end training, and the ability to learn spatial hierarchies of features from the input image. However, they also have some disadvantages such as computational complexity and difficulty in capturing fine details in the output.

D. Multiscale and Pyramid Network Models

Multiscale and pyramid network models leverage the hierarchical structure of images by utilizing multiple scales in the processing of the images and incorporating information from these different scales to enhance the precision of the segmentation.

The architecture of these models typically comprises of multiple branches, each of which processes the input image at a distinct scale. These branches are often implemented using a combination of convolutional layers and pooling layers, which are utilized to reduce the spatial resolution of the image and extract features at various scales. The features extracted from each branch are then combined through concatenation or sum operations to produce a final feature vector, which is subsequently passed through multiple of convolutional layers to produce the segmentation.

An example of a representative network that employs the encoder-decoder architecture and utilizes a pyramid of features to improve the accuracy of the segmentation by combining information from multiple scales is Feature Pyramid Networks (FPN) [25].

E. R-CNN models

R-CNN models are designed to take advantage of the temporal and spatial dependencies in image sequences by processing the images in a sequential manner. The main idea behind RNN based models is to use recurrent connections to propagate information across time and space, which allows the network to learn long-term dependencies in the image sequences. The temporal dependencies between image frames in video sequences or spatial dependencies between pixels in an image can be modeled using RNNs. In this way, the network can segment objects that are moving or changing over time with higher accuracy.

Some of the most representative algorithms based on RNN models are: Recurrent Scale Space CNN (RSS-CNN) : it uses a scale space CNN to segment images in a recurrent manner; Recurrent Residual Convolutional Neural Network (R-ResNet) [26]: it uses residual connections to improve the accuracy of model in segmentation task; Mask R-CNN [27]: it is edited from the two-stage architecture of the R-CNN. By adding a second branch that creates a mask for each object in the image, Mask R-CNN expands the basic R-CNN architecture. This method, which has demonstrated success on numerous datasets, enables more accurate object localization.

F. Dilated Convolutional Models and DeepLab Family

Dilated convolutional models use these dilated layers to increase the network's receptive field while keeping the feature maps' spatial resolution. [28]. The main idea behind dilated convolutional models is to use dilated convolutional layers in the network architecture. Dilated convolution is a variation of convolutional layer where the filters have holes or gaps in them, without increasing the number of parameters and increases the network's receptive field. This enhances the segmentation accuracy of the network and enables it to extract more contextual information from the input image.

The DeepLab family of models uses dilated convolutional layers in conjunction with an encoder and decoder to increase segmentation accuracy as well decrease the number of parameter. DeepLab V3+: An encoder-decoder architecture that uses atrous convolution to increase the receptive area of the network and improve the accuracy of the segmentation task [24].

G. Recurrent neural network models,

Recurrent neural network (RNN) models are a type of deep learning model architecture that have been applied in image segmentation tasks. These models are designed to take advantage of the temporal and spatial dependencies in image sequences by processing the images in a sequential manner. The main idea behind RNN based models is to use recurrent connections to propagate information across time and space, which allows the network to learn long-term dependencies in the image sequences.

Modeling the temporal connections between image frames in video sequences is one of the key benefits of utilizing RNNs for image segmentation, which improves the accuracy of the model to segment objects that are moving or changing over

time. Additionally, RNNs can be used to model the spatial dependencies between pixels in an image, which improves the ability of the model to segment objects that are not necessarily connected in the spatial domain.

H. Generative models and Adversarial Training

Deep neural network architectures such as generative models and adversarial training have been applied to picture segmentation challenges. These models are designed to improve the accuracy of the segmentation by generating realistic images and using them to train the network.

The fundamental principle of generative models and adversarial training is to create realistic images that are similar to the input images using a generative model, such as a Variational Autoencoder (VAE) or a Generative Adversarial Network (GAN). Following that, the network is trained using these produced images and other training data that is representative of real images.

Adversarial training is a technique that is applied to improve the robustness of the model by training it to be resistant to adversarial examples, which are images that have been modified in a way that is intended to fool the network. A discriminator network will be used to determine if the input photos are real or fraudulent, and then to use the gradients from the discriminator to do back propagation and adjust the weights of the generator and segmentation network.

Some of the most representative algorithms based on generative models and adversarial training are:

GAN-based image segmentation: GANs are used to generate realistic images that are similar to the input images, and then used to train the segmentation network [29].

Adversarial training for robust image segmentation: Adversarial training is used to improve the robustness of the network by training it to be resistant to adversarial examples [29].

VAE-based image segmentation: Variational autoencoder (VAE) is used to learn a probabilistic generative model of the data, and then utilize this model to raise the segmentation's accuracy.

I. Convolutional models with active contour models

Deep neural network architectures that employ convolutional models with active contour models have been applied to image segmentation applications. These models were created to combine the advantages of active contour models and convolutional neural networks (CNNs).

Main idea behind convolutional models with active contour models is to use CNNs to provide a high-quality initialization value for the active contour model, so that the active contour model can refine more accurate segmentation results. The active contour model then uses this data to evolve a contour that precisely follows the limits of the object of interest after the CNN gives a rough segmentation of the image.

Active contour models, commonly referred to as snakes or level sets, are a well-known and popular image segmentation method in computer vision. These models are designed to evolve a contour to fit the boundaries of the object of interest based on energy minimization.

Some of the most representative algorithms based on convolutional models with active contour models are:

Hybrid CNN-active contour models: A hybrid strategy that boosts segmentation accuracy by combining active contour models with CNNs.

CNN-based active contour models: The segmentation output is refined using the active contour model after a high-quality initialization is provided for it by CNNs.

CNN-integrated active contour models: CNNs are integrated into the active contour model to raise the accuracy of the segmentation tasks.

The performance of these architectures is typically evaluated using standard metrics such as Intersection over Union (IoU), accuracy, and F1-score [30]. These metrics measure the overlap between the predicted segmentation and the ground truth segmentation, and they are commonly used to evaluate the performance of semantic segmentation models [30].

In summary, the deep learning era of semantic segmentation has been marked by the introduction of convolutional neural networks (CNNs) and architectures such as FCN, U-Net, and Mask R-CNN [31]. These architectures have been able to overcome the limitations of early methods and achieve state-of-the-art performance on several benchmark datasets [31]. These architectures were able to generate dense predictions, utilize context to increase the object localization's accuracy, and the performance of these architectures is typically evaluated using standard metrics such as Intersection over Union (IoU), accuracy, and F1-score [30].

IV. Recent Advances in Semantic Segmentation

Semantic segmentation has advanced in numerous ways during the past few years. The performance and effectiveness of semantic segmentation models have been improved as a result of these developments in new architectures and methodologies.

The employment of attention processes is one of the most significant recent developments. The model's attention mechanisms enable it to focus on relatively important areas of the image. and have been shown to improve performance on various datasets. There are different types of attention mechanisms, such as channel-wise attention and spatial attention, that can be applied at different layers or channels of the network.

Another recent development is the use of lightweight architectures, such as MobileNetV2, EfficientNet and ShuffleNet. These architectures are made to use memory and compute more effectively, making them suitable for deployment on mobile devices or resource-constrained environments. It has been demonstrated that these architectures are substantially more efficient than larger architectures while still achieving good performance.

Domain adaptation is another recent development in semantic segmentation. The process of converting a model trained on one source domain to a different target domain is known as domain adaptation. This is important in semantic segmentation, as it allows the model to generalize better to new

scenarios. Adversarial adaptation, which learns to generate synthetic images that are comparable to the target domain, has been demonstrated to be successful for domain adaptation in semantic segmentation.

There are also other recent developments such as the use of self-supervised, semi-supervised and weakly-supervised learning to improve the model's performance and make them more efficient in terms of labeled data.

In conclusion, recent advancements in semantic segmentation include attention mechanisms, lightweight architectures, domain adaptation, and self-supervised, semi-supervised and weakly-supervised learning, which have improved the performance and efficiency of semantic segmentation models. These developments have the potential to significantly impact the field and open up new avenues for investigation.

V. Conclusion

In this essay, we have provided an overview of the development of semantic segmentation, starting with early approaches and moving on to recent deep learning methods. We have highlighted the key breakthroughs and challenges in the field of semantic segmentation and its potential future impact.

We have seen that early approaches, such as thresholding and region growing, faced limitations in robustness and their ability to capture fine details. Pre-deep learning methods, such as active contours, graph cuts, and CRF, overcame these limitations by incorporating more sophisticated techniques for image analysis and modeling. However, they also faced their own limitations, such as sensitivity to initialization, noise, and computational resources.

A critical turning point in the history of semantic segmentation was the introduction of deep learning. With the introduction of CNNs and architectures like U-Net, FCN, and Mask R-CNN, researchers were able to overcome the limitations of pre-deep learning methods and achieve the best performance on several baseline datasets.

There have been significant developments in the subject in recent years, such as attention mechanisms, lightweight architectures, and domain adaptation, which have improved the performance and efficiency of semantic segmentation models. These advancements open up new possibilities for research and have the potential to make a big difference in the field of semantic segmentation.

Overall, semantic segmentation as a field has made significant progress over the years, and it is expected to continue to evolve as researchers develop new methods and techniques to improve performance and efficiency. The field's impact on a wide range of applications related with computer vision and its potential to make a significant contribution to the development of AI.

References

[1] Ess, A., Müller, T., Grabner, H., & Van Gool, L. (2009, September). Segmentation-Based Urban Traffic Scene Understanding. In BMVC (Vol. 1, p. 2).

[2] Patro, B. N. (2014, September). Design and implementation of novel image segmentation and BLOB detection algorithm for real-time video surveillance using DaVinci processor. In 2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI) (pp. 1909-1915). IEEE.

[3] Yoon, Y., Jeon, H. G., Yoo, D., Lee, J. Y., & So Kweon, I. (2015). Learning a deep convolutional network for light-field image super-resolution. In Proceedings of the IEEE international conference on computer vision workshops (pp. 24-32).

[4] Hampapur, A., Weymouth, T., & Jain, R. (1994, October). Digital video segmentation. In Proceedings of the second ACM international conference on Multimedia (pp. 357-364).

[5] Clarke, L. P., Velthuizen, R. P., Camacho, M. A., Heine, J. J., Vaidyanathan, M., Hall, L. O., ... & Silbiger, M. L. (1995). MRI segmentation: methods and applications. Magnetic resonance imaging, 13(3), 343-368.

[6] Georgel, P., Schroeder, P., Benhimane, S., Hinterstoisser, S., Appel, M., & Navab, N. (2007, November). An industrial augmented reality solution for discrepancy check. In 2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality (pp. 111-115). IEEE.

[7] Fu, K. S., & Mui, J. K. (1981). A survey on image segmentation. Pattern recognition, 13(1), 3-16.

[8] Pal, N. R., & Pal, S. K. (1993). A review on image segmentation techniques. Pattern recognition, 26(9), 1277-1294.

[9] Otsu, N. (1979). A threshold selection method from gray-level histograms. IEEE transactions on systems, man, and cybernetics, 9(1), 62-66.

[10] Patil, D. D., & Deore, S. G. (2013). Medical image segmentation: a review. International Journal of Computer Science and Mobile Computing, 2(1), 22-27.

[11] Liu, D., & Yu, J. (2009, August). Otsu method and K-means. In 2009 Ninth International Conference on Hybrid Intelligent Systems (Vol. 1, pp. 344-349). IEEE.

[12] Davis, L. S. (1975). A survey of edge detection techniques. Computer graphics and image processing, 4(3), 248-270.

[13] Sharma, N., Mishra, M., & Shrivastava, M. (2012). Colour image segmentation techniques and issues: an approach. International Journal of Scientific & Technology Research, 1(4), 9-12.

[14] Kass, M., Witkin, A., & Terzopoulos, D. (1988). Snakes: Active contour models. International journal of computer vision, 1(4), 321-331.

[15] Caselles, V., Kimmel, R., & Sapiro, G. (1997). Geodesic active contours. International journal of computer vision, 22(1), 61-79.

[16] Sethian, J. A. (1999). Level set methods and fast marching methods: evolving interfaces in computational geometry, fluid mechanics, computer vision, and materials science (Vol. 3). Cambridge university press.

[17] Chan, T. F., & Vese, L. A. (2001). Active contours without edges. IEEE Transactions on image processing, 10(2), 266-277.

[18] Bezdek, J. C., Ehrlich, R., & Full, W. (1984). FCM: The fuzzy c-means clustering algorithm. Computers & geosciences, 10(2-3), 191-203.

[19] Udupa, J. K., & Samarasekera, S. (1996). Fuzzy connectedness and object definition: Theory, algorithms, and applications in image segmentation. Graphical models and image processing, 58(3), 246-261.

[20] Minaee, S., Boykov, Y. Y., Porikli, F., Plaza, A. J., Kehtarnavaz, N., & Terzopoulos, D. (2021). Image segmentation using deep learning: A survey. IEEE transactions on pattern analysis and machine intelligence.

[21] Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 3431-3440).

[22] Ronneberger, O., Fischer, P., & Brox, T. (2015, October). U-net: Convolutional networks for biomedical image segmentation. In International Conference on Medical image computing and computer-assisted intervention (pp. 234-241). Springer, Cham.

[23] Badrinarayanan, V., Kendall, A., & Cipolla, R. (2017). Segnet: A deep convolutional encoder-decoder architecture for image segmentation. IEEE transactions on pattern analysis and machine intelligence, 39(12), 2481-2495.

[24] Chen, L. C., Papandreou, G., Schroff, F., & Adam, H. (2017). Rethinking atrous convolution for semantic image segmentation. arXiv preprint arXiv:1706.05587.

[25] Lin, T. Y., Dollár, P., Girshick, R., He, K., Hariharan, B., & Belongie, S. (2017). Feature pyramid networks for object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2117-2125).

[26] Chen, L. C., Papandreou, G., Kokkinos, I., Murphy, K., & Yuille, A. L. (2014). Semantic image segmentation with deep convolutional nets and fully connected crfs. arXiv preprint arXiv:1412.7062.

[27] He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask r-cnn. In Proceedings of the IEEE international conference on computer vision (pp. 2961-2969).

[28] T. Chen, S. -H. Wang, Q. Wang, Z. Zhang, G. -S. Xie and Z. Tang, "Enhanced Feature Alignment for Unsupervised Domain Adaptation of Semantic Segmentation," in IEEE Transactions on Multimedia, vol. 24, pp. 1042-1054, 2022, doi: 10.1109/TMM.2021.3106095.

[29] Shen, D., Liu, T., Peters, T. M., Staib, L. H., Essert, C., Zhou, S., ... & Khan, A. (Eds.). (2019). Medical Image Computing and Computer Assisted Intervention–MICCAI 2019: 22nd International Conference, Shenzhen, China, October 13–17, 2019, Proceedings, Part II (Vol. 11765). Springer Nature.

[30] Virnodkar, S. S., Pachghare, V. K., Patil, V. C., & Jha, S. K. (2021). DenseResUNet: An Architecture to Assess Water-Stressed Sugarcane Crops from Sentinel-2 Satellite Imagery. Traitement du Signal, 38(4).

[31] Leibe, B., Matas, J., Sebe, N., & Welling, M. (Eds.). (2016). Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV (Vol. 9908). Springer.