Image Segmentation: A Case Study of Deep Learning vs. Traditional Computer Vision Approaches

## Tausif Khan

*Master of Science*

University of Maryland, College Park College Park, USA

tkhan001[@umd.edu](mailto:saisurya@umd.edu)

## Zeren Li

*Master of Science*

University of Maryland, College Park College Park, USA

zli06211[@umd.edu](mailto:smayank@umd.edu)

# Abstract

*This study compares traditional image processing techniques with a U-net based approach for image segmentation using the PASCAL VOC dataset. We implemented a modified Canny edge detection followed by a flood fill algorithm, involving grayscale conversion, Gaussian filtering, Sobel gradient estimation, non-maxima suppression, double thresholding, edge tracking with hysteresis, and morphological closing to create segmentation masks. Concurrently, we used a U-net model, known for its contracting and expansive paths with skip connections, to improve segmentation accuracy. Our evaluation showed that the U-net model achieved higher accuracy and detailed segmentation with an IoU of 0.48, whereas the traditional method, while faster and training-free, had an IoU of 0.34. This highlights the trade-offs between computational efficiency and segmentation precision, providing insights into the strengths and limitations of each approach for different applications.*

# Introduction

In the realm of computer vision, segmentation stands as a fundamental process pivotal for image analysis and understanding. This technique simplifies image representation by partitioning it into discernible parts, thereby aiding in boundary localization and object identification. The segmentation process involves attributing each pixel in an image to a label based on shared features, facilitating subsequent analysis. Various methodologies, including deep learning algorithms like U-net, have been developed to perform segmentation tasks efficiently. However, they come with their own limitations.

The primary objective of this project is to conduct a case study comparing traditional computer vision techniques with neural network implementations in the domain of image segmentation. Specifically, the study aims to employ edge-based detection methods as the traditional approach and implement the U-net model as the neural network approach. By undertaking this comparative analysis, the project seeks to evaluate the efficacy and performance of both methodologies in segmenting objects within images.   
  
The evaluation will be conducted based on the quality and accuracy of resulting binary masks.

The scope of this project encompasses the implementation and evaluation of two distinct approaches to image segmentation: traditional computer vision techniques utilizing Canny edge detection and neural network-based methodologies employing the U-net model. The evaluation will be centered on their performance in segmenting objects within images sourced from the PASCAL VOC dataset. However, it's important to note the inherent limitations of this study, including but not limited to computational resources, dataset constraints, and algorithmic complexities.

# Literature review

We found the original paper of the U-net model that was published in 2015. It was initially used for biomedical image segmentation tasks. The neural network consists of a contracting path, which encodes the data and extract the necessary features for the model to learn, and an expansive path, which decodes the data and locates the feature. The contracting path behaved like a usual convolutional neural network, which can classify the object belonging to which class. The expansive path is the key of difference to the traditional neural network. It operates at the pixel level, identifying which pixel belongs to which class. Rather than outputting a final layer for classification, it outputs the results with the same dimension of the inputs. So for image segmentation, it not only can classify the object but also locate where the object is.

Moreover, the model consists of several skip connection bridges, which determine the parameters of some layers within the neural network not only based on the previous layers but also on the layers before that. The main function of the skip connection bridges is to deal with the well known problem of the vanishing gradient, by directly passing the information through the network. Thus, it improves the accuracy of the segmentation mask.

The Original U-net model is effective in performing biomedical image segmentation tasks. The original dataset consists of some micro-electronic cell images. To be specific, the task was to predict whether the pixel is a cell wall or not based on the original images. Just to reference here, the IoU (intersection of union) of the predicted mask against the label is about 0.9203 for PhC-U373 cells and about 0.7756 for DIC-HeLa cells. However, to train such a neural network is tremendously time consuming. It took about 10 hours on a NVidia Titan GPU (6 GB) with only 30 ssTEM (serial section Transmission Electron Microscopy images (Ronneberger, 2015). Thus, it could be extremely beneficial if we can generate a mask not based on the traditional neural network method. And this motivates us to first locate the edges of the original image that was for segmentation. Then, generate the mask based on the edges using some methodologies. Thus, we can compare the results of our method with the traditional neural network implementation in terms of performance and efficiency.

# Methodology

The dataset we used is the Pascal Voc 2012 dataset that was for image segmentation. It consists of 21 classes of objects(for example, train, person) in realistic scenarios. You can view the sample of the datasets below.

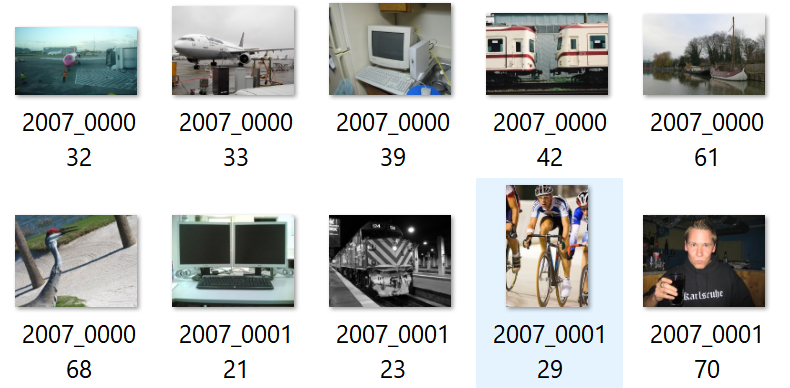


Fig2.1

The training dataset size is 1,464 and the validation dataset size is 1,449. The dataset is available on Kaggle, and can be downloaded directly from python library by using the following command:

from torchvision.datasets import VOCSegmentation

dataset\_train = VOCSegmentation(root='./data', year='2012', image\_set='train', download=True)

dataset\_val = VOCSegmentation(root='./data', year='2012', image\_set='train', download=True)

The whole dataset, which consists of both the training data and the testing data, is contained in the folder named “JPEGImages”. While the label is contained in the folder named ”segmentation class”.



Fig 2.2

Preprocessing: Outline any preprocessing steps applied to the data (e.g., normalization, augmentation).

The original dataset images are actually different in dimensionality. So, we resize the data image to the same size, which results in dimensionality of 3X256X256.



Fig2.3

For the label images, we converted the original label image with color to a binary image, because we are only interested in generating the mask. It results in a dimensionality of 1X256X256 for label images.

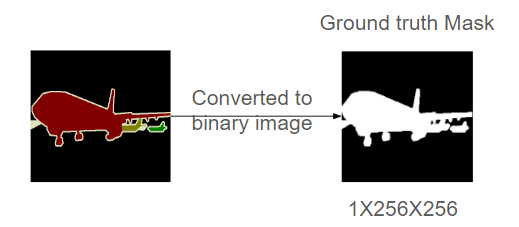


Fig 2.4

In addition, we transformed the original data image to tensors for later training.

Model Architecture:

Here is the original U-net model that was used in 2015. The model consists of a contracting path, which consists of convolutional layers followed by activation function and pooling layers, and an expansive path, which consists of up-convolutional layers(still convolutional layers but reduced in the first dimension) followed by activation function and pooling layers. The black path is the skip connection, which is used a lot in other neural networks like ResNet, and which can deal with the vanishing gradient problem by passing the information from the previous layers to the next layers (Ronneberger, 2015).

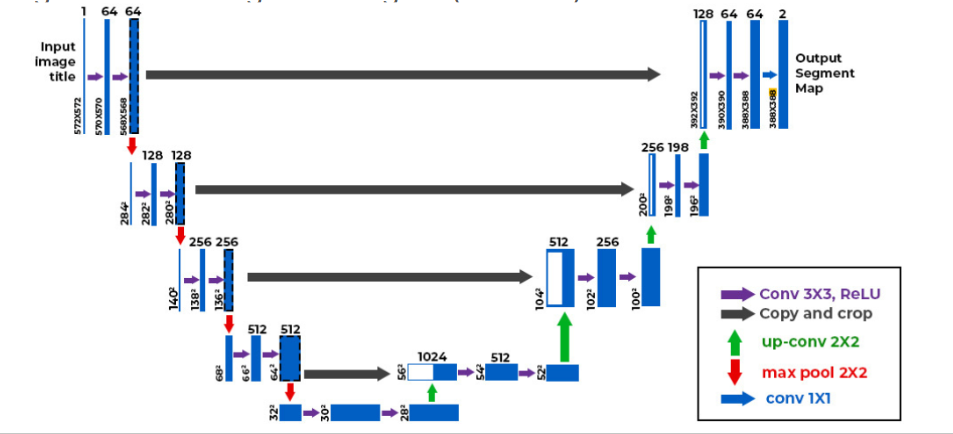


Fig 2.5

Our model is essentially the same as the U-net model that was used in 2015. The only difference lies within the dimension of the input and output images. The original data has an input image of dimensions of 1X572X572 and output of dimensions of 2X388X388. For our image, we have input dimensions of 3X256X256(first dimension is 3, since we are using a color image) and an output dimension of 1X256X256. We used padding to preserve the image dimensionality.The complete structure of the model is shown in Figure 2.6.

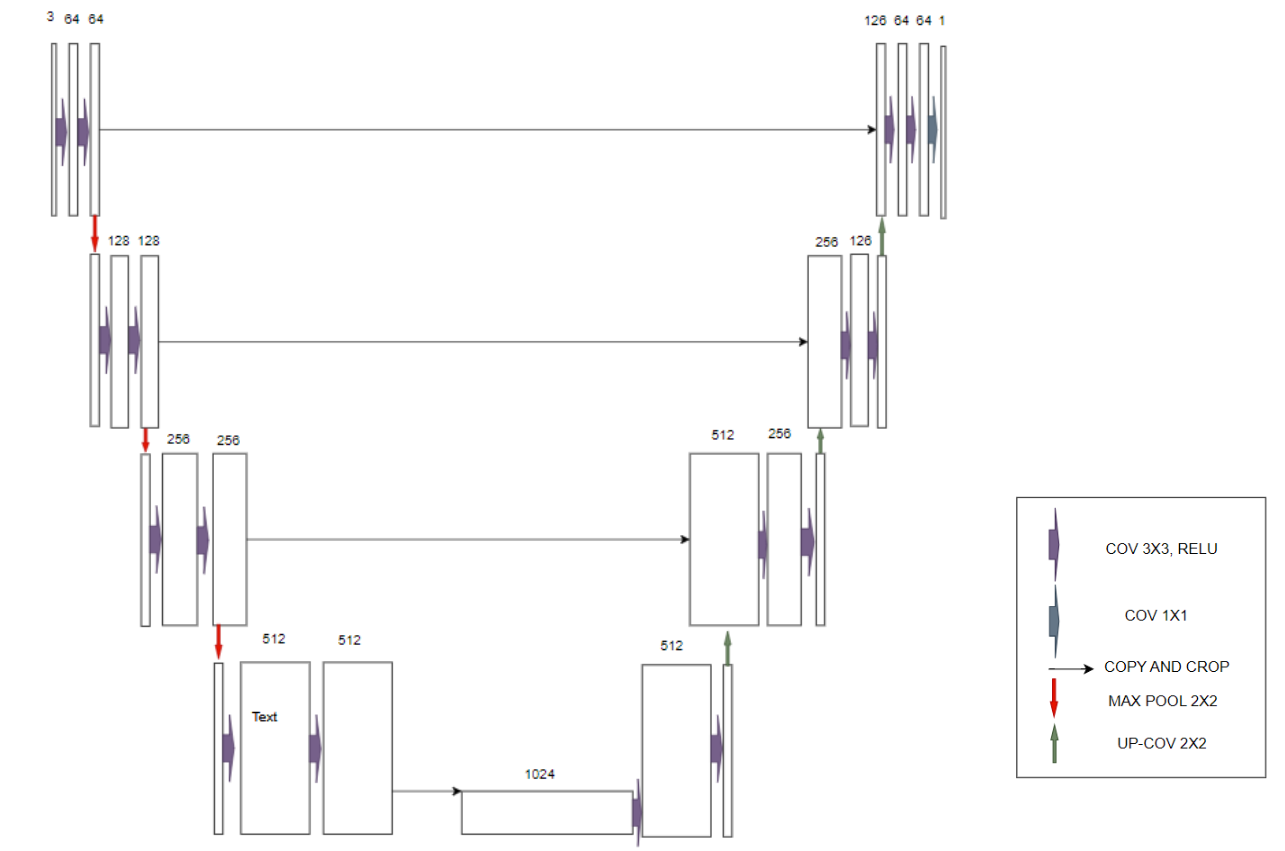


Fig 2.6

Training Procedure: Detail the training process, including hyperparameters and optimization techniques.

Then, we initialized this model with our U-net model that was described previously. We sent the input of image data to our local GPU (Nvidia GTX 1060(6G)). We used binary cross entropy loss for the loss function, and Adam for the optimizer with an initial learning rate of 0.0001. We tried using stochastic gradient descent for the optimizer, but it resulted in a very slow training process, so we used Adam finally. In our training we set the number of epochs to 8. For the training, we forward pass the parameter and computed loss and then update the parameter using backpropagation.

We set the batch size for both the training data and testing data to 8. During training, each epoch takes about 3 minutes. So, 8 epochs definitely takes about 0.5h. The initial loss after one epoch is about 0.597, and after training, it reduces to 0.4402. The whole training process is contained in Figure 2.7.



Fig 2.7

Traditional Image Segmentation Implementation:

To evaluate traditional image processing techniques against a U-net based approach for image segmentation, we implemented a structured process leveraging a modified Canny edge detection method to identify object boundaries alongside a filling algorithm to create solid object masks. We started off by preprocessing the images, converting them from color to grayscale. This enhances contrast and simplifies the image by focusing on intensity variations which emphasize an image’s structural features. Next, we applied a Gaussian filter with a 5x5 kernel to the grayscale images. This step is designed to reduce noise while preserving critical edge details. This size provides balance between smoothing and edge preservation to ensure that fine details are maintained.

Following noise reduction, we calculated the gradient magnitude and direction using several edge detection filters, including Sobel, Prewitt, and Robert. After comparative analysis, we selected the Sobel filter for its superior performance in consistently capturing detailed edge information across a diverse set of images. The Sobel filter computes the gradient by combining horizontal and vertical gradients, providing robust edge detection capabilities.

To then refine the detected edges, we implemented non-maxima suppression which retains only the local maxima pixels, representing potential edges, by comparing the intensity of each pixel with its neighbors along the gradient direction. This step effectively thins the edges, giving us a precise and accurate delineation of object boundaries.

Next, we employed a double thresholding technique to classify pixels into three categories: strong edges, weak edges, and non-edges. By applying fixed low and high thresholds, we classify the pixels based on their intensity values, and this classification is needed for edge tracking with hysteresis, where weak edges connected to strong edges are preserved, enhancing the robustness and continuity of the edge map. Although the fixed thresholds may require adjustment for images with varying complexity, this method proved effective in our study. Additionally, we explored Otsu’s method for adaptive thresholding, which yielded comparable results, further validating the reliability of double thresholding.  
  
Next we applied morphological closing, a combination of dilation followed by erosion, to close small gaps within the object boundaries. This step ensures that the edges form continuous, closed outlines, essential for accurate segmentation. The final step involved using the flood fill algorithm to fill the outlined objects, creating a solid mask that represents the segmented objects. This technique is effective for generating clear and distinct object masks, suitable for comparison with U-net generated segmentation masks.

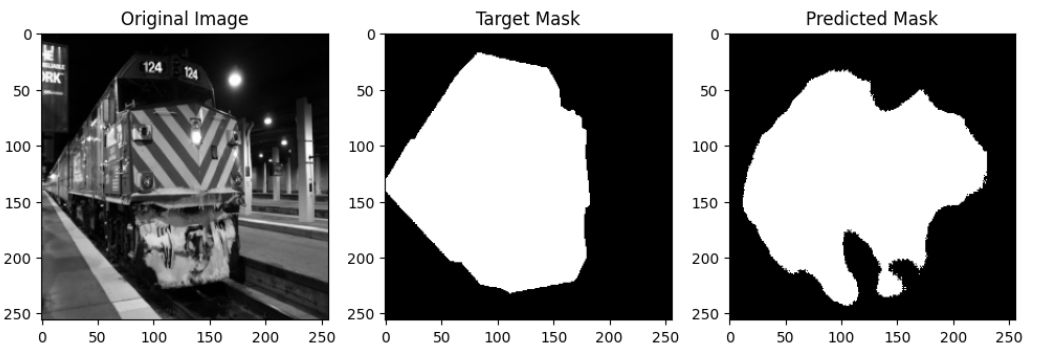
# Results

We used three different metrics to evaluate the performance of our U-net model. They are accuracy, dice score, and IoU (intersection of unions) score. Accuracy is calculated by count(intersection)/count(ground\_truth). Dice score is calculated by 2 \* count(intersection) / (count(ground\_truth) + count(detection)). And IoU is calculated by count(intersection)/count(union). The explanation for these formulas is that for accuracy, we need to count how many we predicted is true based on the label, and for dice score and IoU score, we are interested in looking at how much intersection we have based on the truth label and the predicted mask. At the end of the training, we found the results as follows:

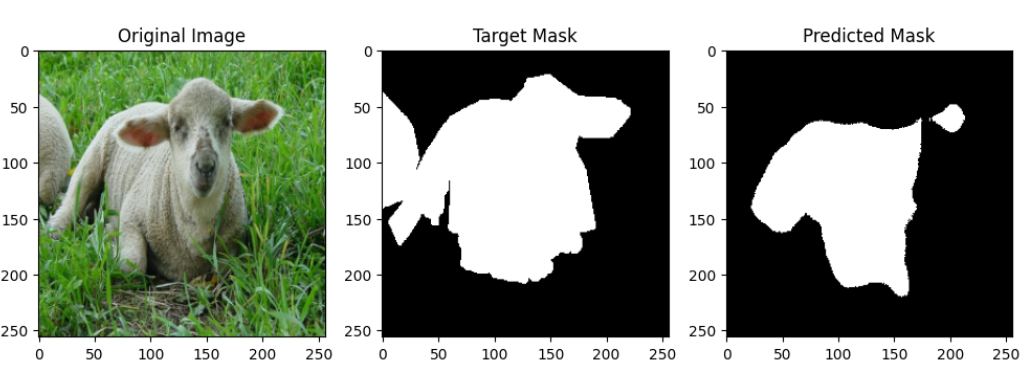
| Accuracy | 0.78 |
| --- | --- |
| Dice score | 0.64 |
| IoU | 0.48 |

table 2.1

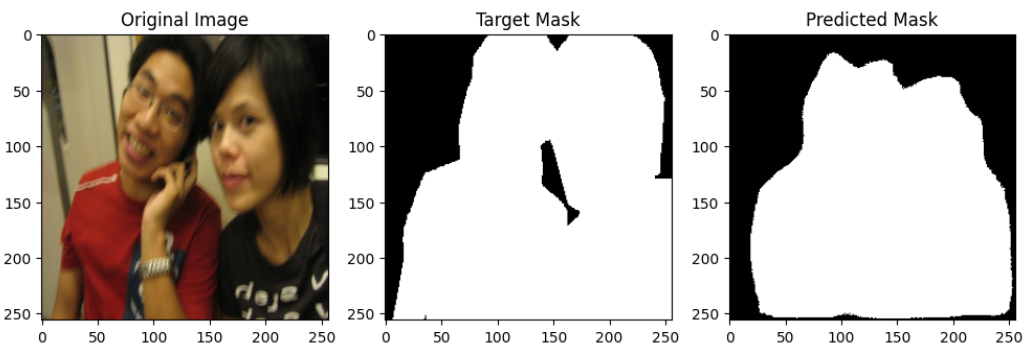
The following images shown some good prediction of our model:



The model has little issue predicting where the train is. However, it is likely to predict a little bit more than it should.



We saw that the predicted Mask covers a great area of the label, though some parts are still missing.

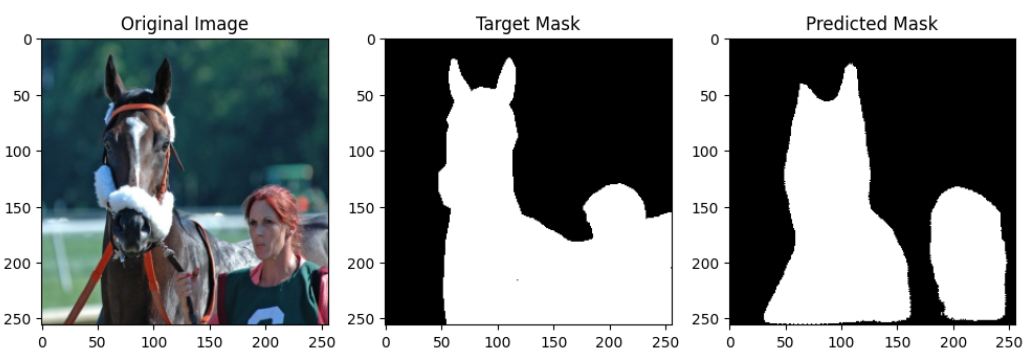


The model predicted well where the two guys are, though some details of the contours are missing.

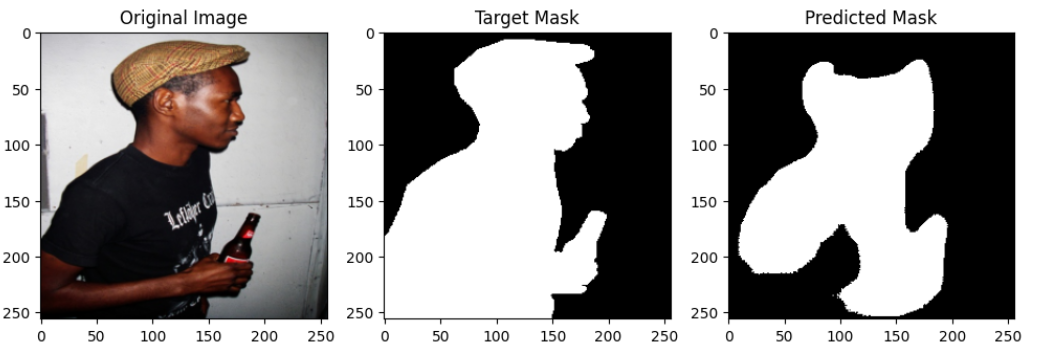
And the following images are some examples that are not “very good”:



We saw that the fence is also included in the image, however, it is not the main object.

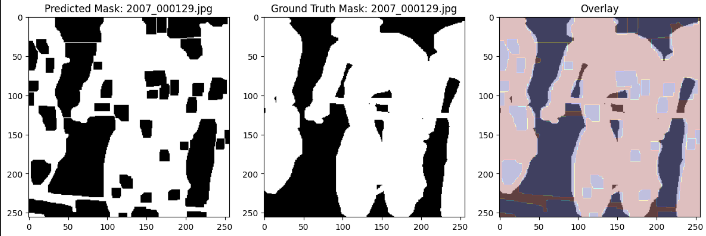
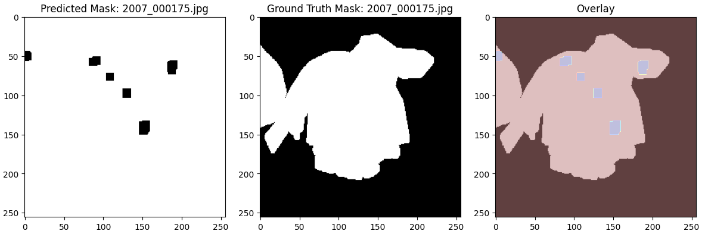
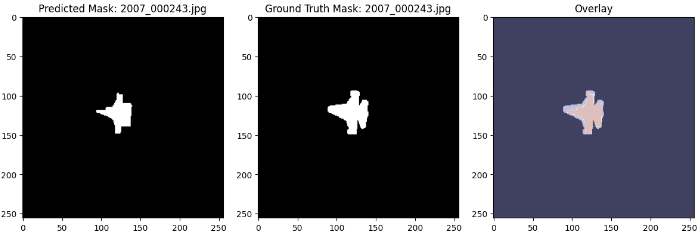


The target mask is a connected and closed mask. However, for the predicted mask, it is not connected, though it correctly interprets two objects as distinct objects.



The model has no problem predicting the “man” as the main object, however, there are some parts (arms) that were missing compared to the label.

Overall, the model generated the mask of the original image pretty well, though we saw that for some images, our predictions are little bit more or less than the ground truth mask.

Traditional Implementation Results:  
Using the same dataset, the traditional image processing techniques yielded moderate results:  
  
  
The method effectively recognized gaps such as the areas between the bikers that are considered part of the background.  
  
  
However, the method struggled when there is a gap in the image or had discontinuities often resulting in the entire background being covered.  
  


The approach performed best with a single object centered against a clear background.

| Accuracy | 0.67 |
| --- | --- |
| Dice score | 0.48 |
| IoU | 0.34 |

table 2.2

Comparison:

However, if we compare the results with the original paper. We saw that in 2015 the U-net on DIC-Hela cell has IoU about 0.77, while we have IoU of 0.48 on the Pascal Voc 2012. And there are several reasons for this disparity of performance:

1. Dataset structure difference. U-net model in 2015 is for Biomedical Image Segmentation, while our datasets incorporate a variety of objects in real senario. Due to the increase in the variety of the objects of the dataset, the IoU score tends to decrease.
2. Output size difference. The dimension of the output image used in 2015 is smaller than its input, while we used padding and preserved the output dimension. The IoU score tends to be higher on a smaller size image rather than on a larger size image.
3. Input type difference: 2015 U-net model accepts grayscale image, while we used color image as input.
4. Label difference: the original is just to predict where cell walls exist, while our label usually consists of one main object which is covered by a mask. Thus, there is an increase in the difficulty of performing segmentation.

Thus, though we saw from the image illustration that the mask generated by the model is considered “solid”, the qualitative results were not as good as we once expected. However, we still consider the model is doing a good job at predicting the mask since it has no problem predicting the general location of the object and only has some issues describing the boundary of the object.

# Conclusion

In conclusion, our study evaluates the U-net model and traditional edge detection-based methods for image segmentation using the PASCAL VOC dataset. The U-net model demonstrated superior performance in generating accurate segmentation masks, albeit with minor boundary nuances. The key strengths of the U-net include its ability to learn complex features and generate precise masks, making it suitable for applications requiring high accuracy.

The U-net model has very good accuracy in predicting the mask, even though the boundary of the generated mask has some nuances compared to the ground truth mask as we saw in some sample outputs earlier.

However, there are also several limitations with this method. Firstly, the model requires a great amount of relevant dataset to feed. That means for different dataset, the whole training process has to start again if the dataset is changed too much. And the resulting accuracy also varies a lot due to differences in the structure of the previous dataset and current dataset. Secondly, the U-net model tends to be very time consuming. In the project, we trained the model with 8 epochs and it took about 0.5 hour. To achieve higher accuracy, we have to try more epochs and it definitely will cost much more time.

Conversely, the traditional edge detection method, while less accurate, offers notable advantages in computational efficiency. It requires no training and rapidly generates segmentation predictions, making it suitable for applications with constrained computational resources or those requiring quick results.

The trade-off between computational efficiency and segmentation precision is evident:

* U-net Model: High precision, suitable for complex segmentation tasks, but computationally intensive and time-consuming.
* Traditional Methods: Faster and less resource-intensive, suitable for simpler tasks or when computational resources are limited.

This study underscores the importance of selecting segmentation techniques based on specific application requirements, balancing the need for precision with available computational resources. Future work could explore hybrid approaches, combining the strengths of both methods to enhance segmentation performance while maintaining efficiency.

Group member contribution

Zeren Li: The Pascal Voc dataset collection, implementation of U-net Model, results comparison of U-net model on Pascal Voc dataset and on original biomedical image dataset, searching for U-net related references.

Tausif Khan: Implementation of traditional computer vision techniques for image segmentation, evaluation of results, and discussion of implementation

# References

List all the references cited in the report following

a consistent citation style (e.g., APA, IEEE).

# Abidin, Z. Z., Asmai, S. A., Abas, Z. A., Zakaria, N. A., & Ibrahim, S. (2020). Development of edge detection for image segmentation. *IOP Conference Series. Materials Science and Engineering*, *864*(1), 012058. https://doi.org/10.1088/1757-899x/864/1/012058

# Ronneberger, O., Fischer, P., & Brox, T. (2015, May 18). U-NET: Convolutional Networks for Biomedical Image Segmentation. arXiv.org. https://arxiv.org/abs/1505.04597

1. Sekehravani, E. A., Babulak, E., & Masoodi, M. (2020). Implementing canny edge detection algorithm for noisy image. Bulletin of Electrical Engineering and Informatics, 9(4), 1404-1410.