Semantic Segmentation of Aerial Images using Deep Networks

Rucha Pendharkar
Dept of Electrical and Computer Engineering
Northeastern University
Boston, MA
pendharkar.r@northeastern.edu

***Abstract*—In this project, the application of U-Net, a convolutional neural network architecture, for aerial image segmentation tasks is explored. U-Net has shown remarkable success in segmenting medical images due to its unique architecture, which incorporates both contracting and expansive pathways. This study focuses on applying U-Net to segment aerial images. The model is trained on an annotated dataset containing aerial images of Dubai and evaluate its performance using standard metrics such as accuracy. Additionally, we investigate techniques to enhance U-Net's performance, including attention and transfer learning. Through experiments and analysis, we aim to demonstrate the efficacy of U-Net for semantic segmentation for aerial images.**

***Index Terms*—semantic segmentation, convolutional neural networks, aerial images**

I. Introduction

Semantic Segmentation is a task in the field of Computer Vision which involves categorizing each pixel in an image into a class or object. Aerial images are a popular field for applying semantic segmentation for different analyses. Manually classifying each pixel into a class or category can be quite time-consuming and tedious. Segmentation and classification of images has become very efficient since Convolution Neural Networks (CNNs) came into the picture. The objective of this project is to perform semantic segmentation using deep networks.

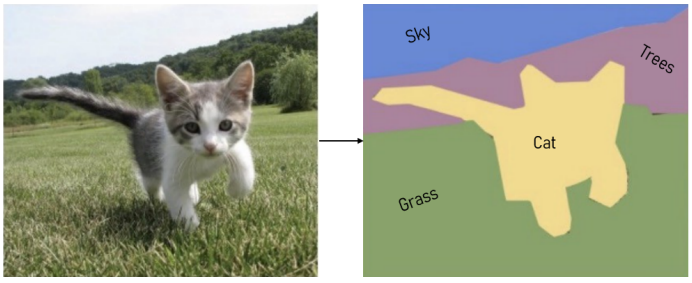

Fig. 1. Example semantic segmentation.

In image segmentation tasks, the network tries to learn a mapping between pixels of the input image and the pixels of the segmentation mask (hand-labelled segmented images). These masks are the ground truths. This trained network can then be applied to segment new untrained images. The network architecture is inspired from the **U-Net** model which is widely used for image segmentation tasks. This model is generally used in medical science fields to detect anomalies in medical images like tumors, but is known to perform well for other segmentation tasks as well.

The dataset used, consists of aerial imagery of Dubai obtained by MBRSC (Mohammed Bin Rashid Space Center in Dubai, the UAE) satellites and annotated with pixel-wise semantic segmentation in 6 classes. It consists of 72 images of various sizes, grouped into 8 tiles. Results from the U-Net model are compared with a pretrained model.

II. Related Work

A. A Survey on Deep Learning Methods for Semantic Image Segmentation in Real-Time

This survey paper effectively compares and contrasts multiple datasets, deep learning approaches for semantic segmentation such as Fully Convolutional Networks, Conditional Random Fields with Neural Networks, Encoder-Decoder, Feature Fusion, RNNs etc. It also talks about some modern concepts such as adding attention to the model for performance enhancement. It also summarizes basic metrics used to evaluate different semantic segmentation approaches. These metrics can be dividied into two - accuracy based (how close is the prediction to the ground truth) or efficiency based (how much memory was allocated and training time). This paper was useful in model selection, and determining the metric used for evaluating the performance on the dataset.

B. U-Net: Convolutional Networks for Biomedical Image Segmentation

U-Net is a type of machine learning architecture proposed in 2015 [1]. The U-Net has a unique structure which makes it useful in tasks which have a high resolution input and output. It consists of an encoder and decoder. The encoder extract features from the input images, and the decoder is responsible for up-sampling the intermediate features and producing the final image outputs. The encoder and decoder are symmeterical in shape, making it extremely effective. This architecture outperformed the sliding window method and proved that U-Nets can be used on relatively smaller datasets and still produce pretty good results, in lesser training time. Since the dataset is relatively smaller, this model was chosen for the project.

C. Semantic Segmentation using Unet-VGG16: A case study in Yunlin, Taiwan

In this paper, a modified U-Net segmentation network is proposed to segment rice regions in the aerial images of Yunlin, Taiwan. Transfer learning using the well-known **VGG16** model is then employed to enhance U-Net's performance. This research shows that the training process using UNet-VGG16 is faster, more convergent, and more stable with smoother movements. A feature in the preprocessing stage is modification of the aerial images by converting them from *R-G-B* to *RG-NIR*. The blue color channel was replaced with the Near-InfraRed (NIR) channel. 91% accuracy was obtained for the experiments obtained. Although images were processed in RGB color channels in the project, this paper was useful in understanding the different data preprocessing techniques to be performed before the images can be fed into the model. It also served as the inspiration for adding and experimenting with vanilla U-Net.

III. Methodology

A. Dataset

The open access satellite imagery dataset for semantic segmentation published by Humans in the Loop was used.

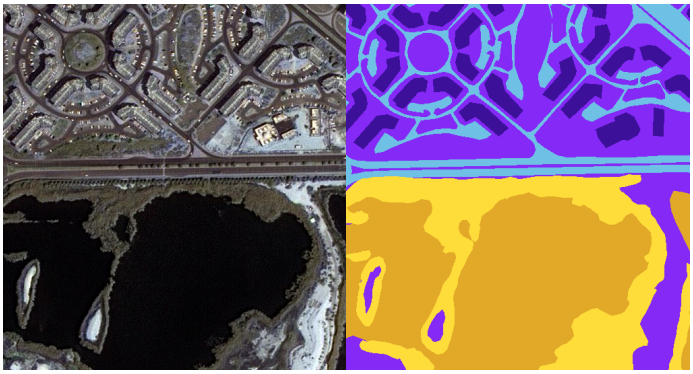

Fig. 2. Image and corresponding mask from the dataset.

It consists of 72 images of various sizes, grouped over 8 tiles. The images are in JPG format and the masks are in the PNG image format. The data is divided into 6 classes given in Table I.

TABLE I
Hex Code and Corresponding Classes

Building	#3C1098
Land (unpaved area)	#8429F6
Road	#6EC1E4
Vegetation	#FEDD3A
Water	#E2A929
Unlabeled	#9B9B9B

B. Image Pre Processing

The images and their corresponding masks were broken into smaller patches of size 256×256. All of the images were cropped and normalized. This increased the size of the dataset to 1305 images

C. One Hot Encoding of Labels

In the data preprocessing pipeline, the RGB-to-label conversion process plays a pivotal role. This process involves mapping RGB color values from labeled images to corresponding class labels. Each class is associated with a unique Hex color code, which serves as its identifier. A hexadecimal color code consists of three pairs of characters representing the intensity of the red, green, and blue components, respectively. Each pair corresponds to a value ranging from 00 to FF, where 00 represents the absence of the color and FF represents the maximum intensity. To begin with, these Hex values are converted to RGB values. The RGB values of each pixel in the labeled image are compared with predefined color mappings for classes such as buildings, land, roads, vegetation, water, and unlabeled areas. When a match is found, the pixel is assigned the corresponding class label.

Furthermore, to facilitate model training and evaluation, the class labels are transformed into a one-hot encoding format using TensorFlow. This step converts the integer labels into binary class matrices, where each class is represented by a binary array. This also helps in avoiding to create biases, and gets rid of any that may arise due to natural ordinality in the data. The resulting one-hot encoded labels enable efficient representation of categorical data and are essential for training convolutional neural networks (CNNs) for semantic segmentation tasks.
The dataset was then split into training and testing data.

D. Defining the U-Net Model Architecture

U-Net architecture was chosen as it is effective on smaller datasets and requires lesser training time. Each layer contributes to the U-Net architecture, which consists of an encoder path for feature extraction and a decoder path for upsampling and generating segmentation masks. The skip connections between encoder and decoder paths help in preserving spatial information and capturing detailed features necessary for semantic segmentation tasks.

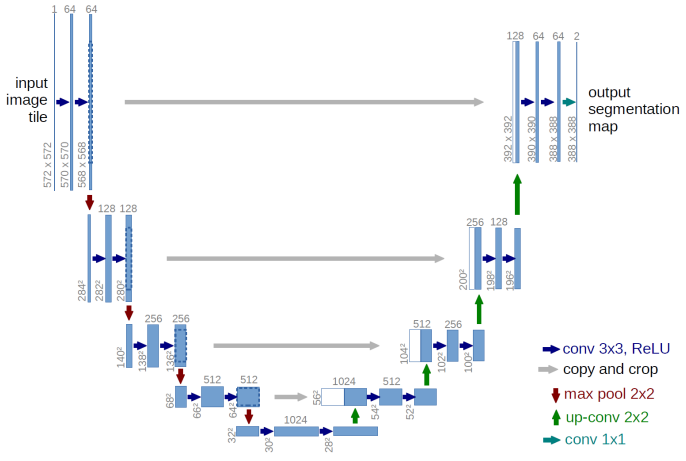

Fig. 3. Architecture of original U-Net.

1) **Encoder Blocks**: There are 5 encoder blocks. Each encoder block consists of two convolutional layers followed by a max-pooling layer and a dropout layer, with ReLU activations. These blocks progressively downsample the input image and extract features.

2) **Decoder Blocks**: There are 4 decoder blocks. Each decoder block consists of a convolutional transpose layer for upsampling followed by two convolutional layers and a dropout layer, with ReLU activations. These blocks progressively upsample the feature maps and recover spatial information.

3) **Skip Connections**: Additionally, each decoder block is connected to the corresponding encoder block through concatenation, forming skip connections to retain spatial information and capture fine-grained details.

4) Here the number of the decoders are smaller than the encoder. Encoders need to be deep in order to capture the features in a similar fashion to their classification counterparts. Decoders, however have to upsample the compressed feature space in order to provide pixel-level classification. The latter can be achieved with a much less deep architecure providing significant computational savings [3]

E. Selecting the Loss Function and Evaluation Metric

Loss functions are an essential part of training the model because they show the accuracy of its performance, helping it learn with each epoch

Cross-entropy loss refers to the difference between two random variables. It measures the variables to extract the difference in the information they contain, showcasing the results. There are two types of cross-entropy losses - binary and categorical. Binary loss is used where classification is binary. We utilize categorical cross-entropy loss in multi-class classification tasks with more than two mutually exclusive classes.

Since the task at hand is a multi-class segmentation (6 classes) problem, the **categorical cross-entropy loss** was chosen

$$\mathrm{L} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C} y_{i,c}\cdot\log(\hat{y}_{i,c})$$

Where:

- N is the total number of samples.
- C is the number of classes.
- $y_{i,c}$ is the ground truth probability that sample i belongs to class c.
- $\hat{y}_{i,c}$ is the predicted probability that sample i belongs to class c.

This formula computes the average negative log likelihood of the predicted class probabilities compared to the true class labels across all samples.

Accuracy was chosen as the evaluation metric.

F. Experimenting with Attention

An attempt was made to tweak the architecture of the U-Net model to see what difference would be obtained in the results. Attention mechanisms enhance deep learning models by selectively focusing on important input elements, while ignoring the others, thus improving prediction accuracy and increasing computational efficiency. In an attempt to increase performance, an attention mechanism was added to one of the encoder blocks

This attention block consists of two convolutional layers, max pooling and a transposed convolutional layer, followed by concatenation. After concatenation, the combined features undergo further convolutional layers with ReLU activation, followed by a sigmoid activation.

G. Comparison with Pretrained Model

To compare and contrast the results, a pretrained U-Net model was used from the segmentation_models library. The Backbone architecture used was the *resNet34* model with pre-trained weights from the ImageNet dataset, which can help in transfer learning.

IV. Experiments and Results

All of the models were trained for **70 epochs** using the adam optimizer, using a batch size of 16.

A. U-Net

The following figures 4 and 5 denote the accuracies and losses on the training and validation data. Table II summarizes the metrics at the end of training. Figure 6 shows the results of the model on the test data and the corresponding mask. The model is able to correctly classify all of the regions. It struggles to capture finer or smaller regions, or incorrectly classifies regions. The boundaries between classes are not very smooth.

TABLE II
U-Net Model Metrics

Training Accuracy	0.904
Validation Accuracy	0.823
Training Loss	0.08
Validation Loss	0.234

B. Modified U-Net

TABLE III
Modified U-Net Model Metrics

Training Accuracy	0.915
Validation Accuracy	0.843
Training Loss	0.056
Validation Loss	0.203

As seen in Figure 9, the segmentation seems to be only slightly better. Figures 7 and 8 show no drastic change in the accuracy and loss values. They are only slightly better. The boundaries between classes seem to have improved a little bit. It can be seen that it is faster to learn and improve, than vanilla U-Net.

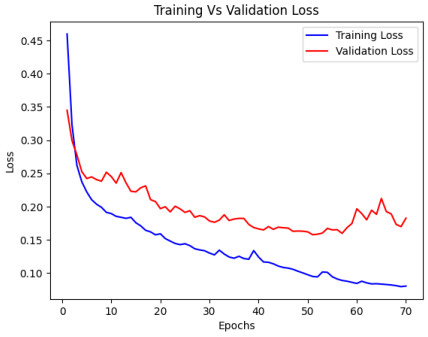

Fig. 4. Loss Plot for U-Net

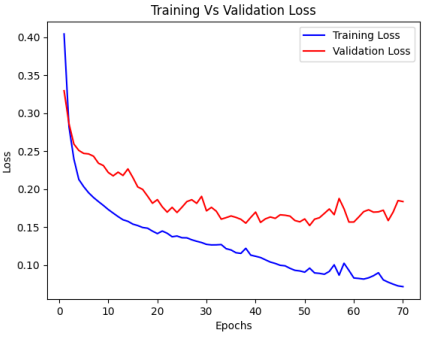

Fig. 7. Loss Plot for modified U-Net

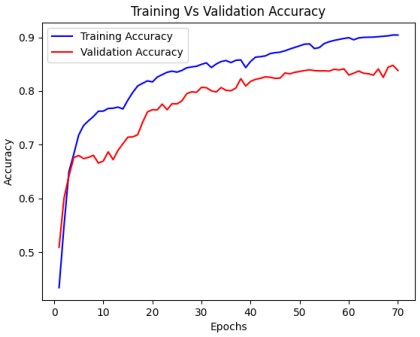

Fig. 5. Accuracy Plot for U-Net

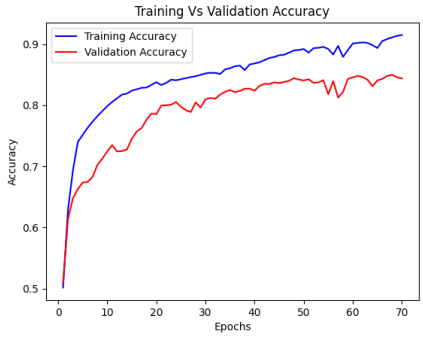

Fig. 8. Accuracy Plot for modified U-Net

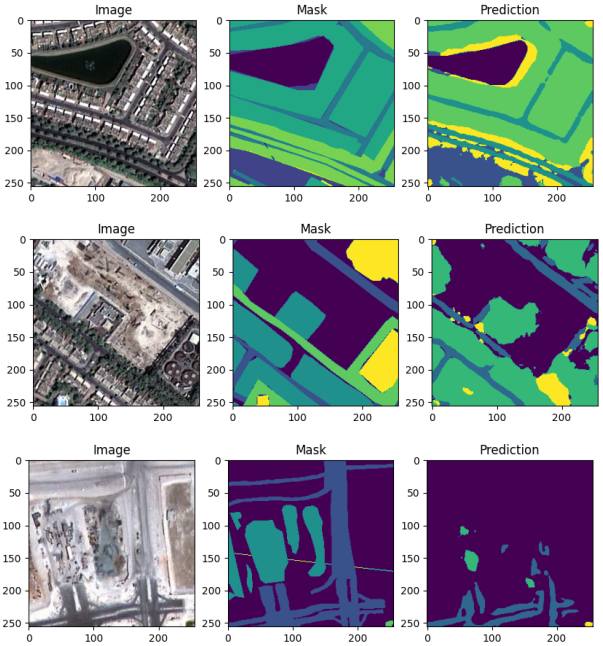

Fig. 6. Results of vanilla U-Net on test images

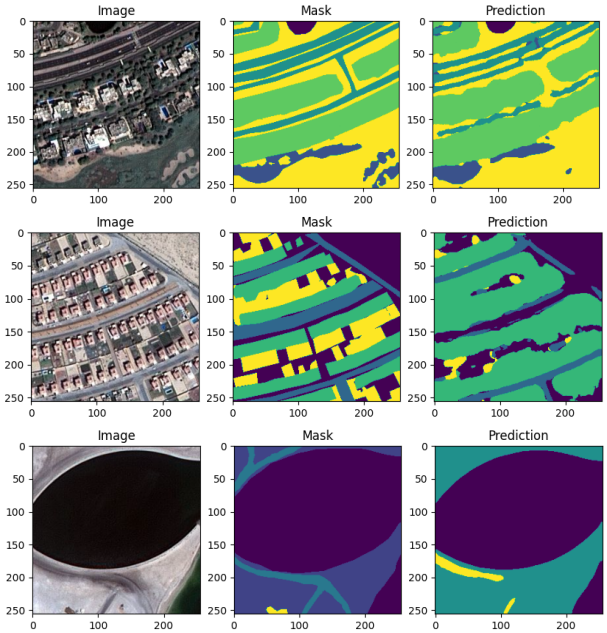

Fig. 9. Results of modified U-Net on test images

C. Pre-trained U-Net

TABLE IV
PRETRAINED U-NET MODEL METRICS

Training Accuracy	0.9572
Validation Accuracy	0.8648
Training Loss	0.0362
Validation Loss	0.1512

The pretrained model had much better segmentation results. The boundaries were sharper, and denser and finer details were effectively captured. Table IV summarizes the training and validation losses and accuracies.

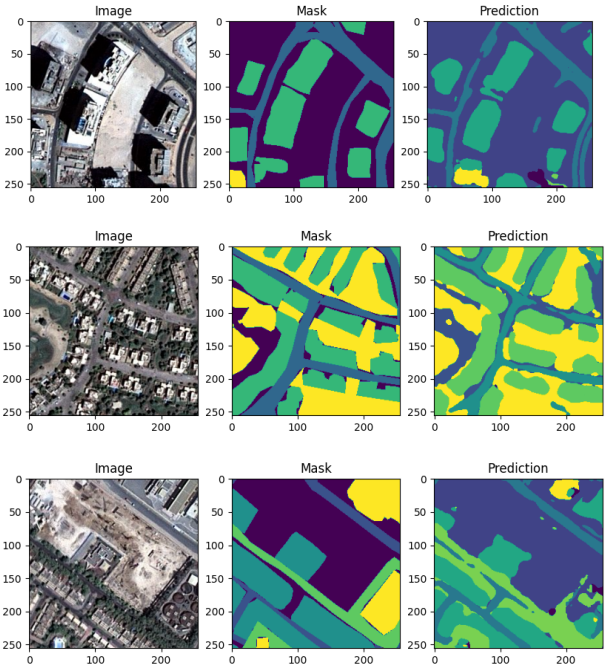

Fig. 10. Results of pretrained model

V. CONCLUSION

All of the three trained models were tested on the same input image to better understand their performance.Transfer learning yeilded the best results. While U-Net in itself seems to perform well enough, the modification adds in one more layer and gives better results. However, the pre trained model gave the best results. There were fewer instances of incorrect classifications and the boundaries between the classes were much smoother. One way would be to use erosion and dilation filters during preprocessing of data to get a better boundaries between classes.

Fine-tuning the hyper parameters such as number of epochs, batch size, size of training and testing data may make the results much more robust.

Adding in more attention layers or experimenting with different loss functions may also help in determining the optimal set of hyperparameters. More metrics such as IoU (Intersection over Union), Jaccard's co-efficient can also be used for evaluating the model performance.

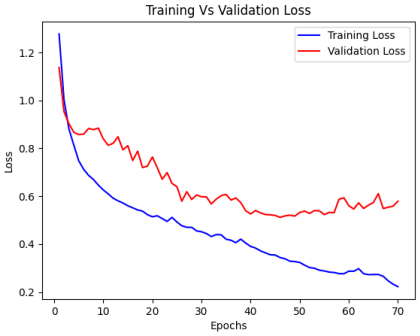

Fig. 11. Loss Plot for pretrained U-Net

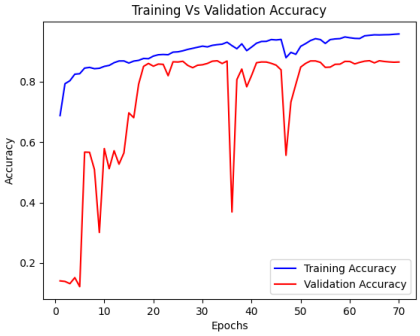

Fig. 12. Accuracy Plot for pretrained U-Net

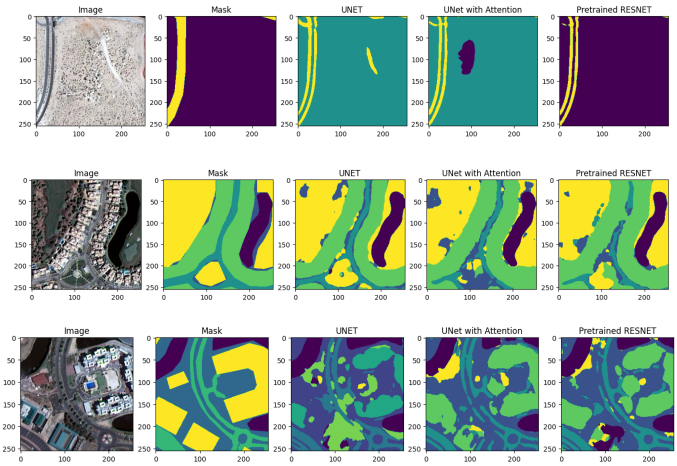

Fig. 13. Comparing results from different models

ACKNOWLEDGMENT

I would like to thank Prof Bruce Maxwell for his invaluable guidance and approach to solve this problem. I would also like to thanks CS5330 TAs for their help in debugging small mistakes in my code.

REFERENCES

[1] O. Ronneberger, P. Fischer, T. Brox, *U-Net: Convolutional Networks for Biomedical Image Segmentation*, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, 2015, arXiv:1505.04597 [cs.CV].

[2] I. Wahyuni, W.-J. Wang, D. Liang, C.-C. Chang, *Rice Semantic Segmentation Using Unet-VGG16: A Case Study in Yunlin, Taiwan*, *2021 International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS)*, 2021, pp. 1-2, doi: 10.1109/ISPACS51563.2021.9651038.

[3] G. Takos, *A Survey on Deep Learning Methods for Semantic Image Segmentation in Real-Time*, *CoRR*, vol. abs/2009.12942, 2020, https://arxiv.org/abs/2009.12942.

[4] "Humans in the Loop Semantic Segmentation Dataset," Humans in the Loop, https://humansintheloop.org/resources/datasets/semantic-segmentation-dataset-2/.

[5] https://paperswithcode.com/task/semantic-segmentation.