

Deep Learning for Satellite Image Semantic Segmentation

Chinatip Lawansuk

Department of Electrical Engineering and Computer Science

National Taipei University of Technology

Email: t112998405@ntut.edu.tw

Abstract—Aerial image semantic segmentation is a crucial task in remote sensing and geographic information systems (GIS). This report provides an overview of methods and techniques used in semantic segmentation of aerial images, discusses the challenges involved, and highlights recent advancements in the field. Utilising the U-Net architecture and its deep variants, we demonstrate how these models address limitations of traditional methods by improving accuracy and efficiency in land use surveys. Our study includes a performance comparison of different activation functions, with ReLu achieving the highest Mean IoU at 0.896, followed by Leaky ReLu at 0.829, and Tanh at 0.763. We also compare Multiple Model Single Channel (MMS) and Single Model Multiple Channel (SMMC) architectures, finding SMMC more effective with 8.45 million parameters and 32.27 MB memory usage. Real-world application on Google Maps data showcases the model’s capabilities and areas for improvement, emphasising its potential for urban planning, environmental monitoring, and resource management.

I. INTRODUCTION

Surveying land usage is critical for sustainable development, environmental conservation, and efficient resource management. Traditional land surveying techniques, while foundational, are often limited by their scalability, accuracy, and reliance on extensive human intervention. The advent of aerial imagery has revolutionised this field, offering high-resolution and comprehensive coverage of vast areas, significantly enhancing the efficiency and accuracy of land surveys. However, interpreting these images using traditional algorithms, such as manual interpretation, supervised and unsupervised classification, edge detection, and segmentation, poses challenges including high labour costs, human error, and sensitivity to data variability.

In recent years, deep learning, particularly convolutional neural networks (CNNs), has emerged as a powerful tool for image segmentation. Among these, the U-Net architecture stands out for its ability to combine high-level contextual information with precise localisation, making it particularly effective for complex image segmentation tasks. U-Net’s symmetric encoder-decoder structure, augmented with skip connections, facilitates the retention of fine-grained details, which is crucial for accurately delineating land features in aerial imagery. Enhanced variants such as Deep U-Net further improve segmentation accuracy by incorporating deeper networks and advanced techniques like attention mechanisms and multi-scale processing.

This work explores the application of U-Net and its deep variants in the segmentation of aerial images, demonstrating how these architectures address the limitations of traditional methods. Additionally, we focus on the critical aspects of data preparation and preprocessing, which are essential for improving the performance of deep learning models. Proper data preparation, including tasks such as noise reduction, image enhancement, and augmentation, ensures that the models are trained on high-quality datasets, thereby enhancing their robustness and generalisation capabilities.

By leveraging the strengths of deep learning and thorough data preparation, this approach aims to enhance the precision, scalability, and efficiency of land use surveys, ultimately contributing to better-informed decision-making in urban planning, environmental monitoring, and resource management.

II. DATASET

In our work, we utilise the dataset titled “Semantic Segmentation of Aerial Imagery”, which comprises 72 high-resolution aerial images. Each image in this dataset is up to 1800×2000 pixels, providing a substantial amount of detail for our analysis. The dataset can be accessed via the following link: <https://www.kaggle.com/datasets/humansintheloop/semantic-segmentation-of-aerial-imagery>.

Data Preprocessing

1) *Crop*: To enhance training efficiency, we randomly select a point on the image and crop a 128×128 patch, using the selected point as the top-left corner of the cropped image. This method ensures diverse and representative samples, capturing various features and contexts from the original images. As shown in Fig. 1.

2) *Split*: For the split, the original RGB mask, with different colours for each type, is split into 6 binarised black-white images, each representing a specific class. These binary images are combined to form a 6 channels input when loading into the model, allowing it to process each class separately and effectively. As shown in Fig. 1

Additionally, we have published the processed dataset on Kaggle for sharing, allowing others to cite it in their work. The dataset can be accessed at <https://www.kaggle.com/datasets/lchinatip/aerial-image-dataset-for-semantic-segmentation/data>.

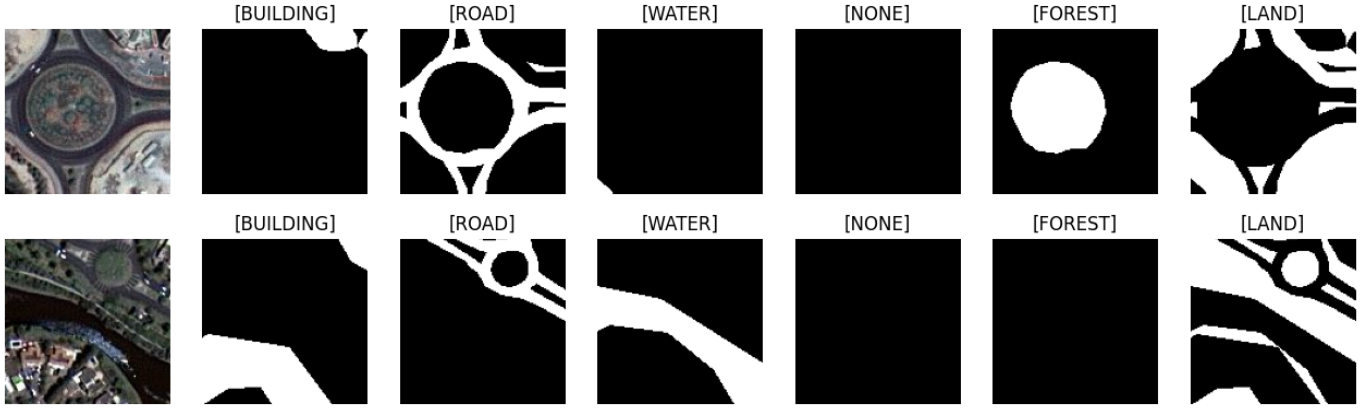


Fig. 1: Dataset shows original image and splitted masks

III. MODEL ARCHITECTURE

In this work, we utilise the U-Net model, delving into both its architectural design and activation functions. By optimising these aspects, we aim to enhance the model's performance and accuracy in image segmentation tasks. Our exploration includes experimenting with various configurations to identify the most effective combination for our specific applications.

A. Specifications and Constraints

- Input Layer Dimension: 128×128 pixels
- Input Layer Channel: 3 channels (RGB)
- Convolution Depth: 4 (Feature Maps from 64, 128, 256, 512)
- Output Layer Dimension: 128×128 pixels
- Output Layer Channel: 6 channels (Multiclass)
- Model Optimisation: Adam
- Training Epoch : 5000

B. Architectural Design

1) *Multiple Model Single Channel (MMSC)*: This approach involves using multiple submodels, each dedicated to a single class. As shown in Fig. 4.

2) *Single Model Multiple Channel (SMMC)*: In this approach, a single model is designed to have multiple channels on each layer, representing a different class. As shown in Fig. 3.

We then evaluate the model, as shown in Table I.

TABLE I: Model Comparison Between Multiple Model Single Channel (MMSC) and Single Model Multiple Channel (SMMC)

Properties	MMSC	SMMC
Parameters	12.47 M	8.45 M
Memory	47.59 MB	32.27 MB

The Multiple Model Single Channel approach takes smaller parameter count and memory usage compared to the Single Model Multiple Channel method. This is because the former splits the task into 6 distinct submodules, each handling a

single channel, resulting in smaller individual models. However, training the Multiple Model Single Channel configuration takes more time since it requires updating gradients 6 times, once for each submodule.

In contrast, the Single Model Multiple Channel approach, despite being larger, tends to yield better performance. This improvement is primarily due to fully connected among each channel, as it maintains the interconnection between different classes within a unified model. The Multiple Model Single Channel setup lose these interconnections, which can lead to suboptimal performance. By preserving the relationships between classes, the Single Model Multiple Channel approach can achieve more accurate and coherent segmentation results. Resulting in lower overlaps output among the channel.

Hence, We decided to use the Single Model Multiple Channel.

C. Activation Functions

As shown in the model architecture in Fig . Each layer is consisted of Activation layer. We work on multiple activation functions to evaluate the characteristic and performance.

1) *ReLU*: Rectified Linear Unit is a popular yet simple activation function in neural networks. ReLU introduces non-linearity into the model, enabling the learning of complex patterns. It is computational efficiency, due to the simple operations, and it is able to mitigate the vanishing gradient problem.

$$\text{ReLU}(x) = \max(0, x)$$

2) *Leaky ReLU*: Leaky ReLU is is a variant of the ReLU activation function. It allows a small, non-zero gradient when the input is negative. This addresses the dying ReLU problem by ensuring that neurons do not become inactive. Leaky ReLU introduces slight non-linearity, improving model performance and convergence.

$$\text{LeakyReLU}(x) = \begin{cases} x & \text{if } x \geq 0 \\ \alpha x & \text{if } x < 0 \end{cases}$$

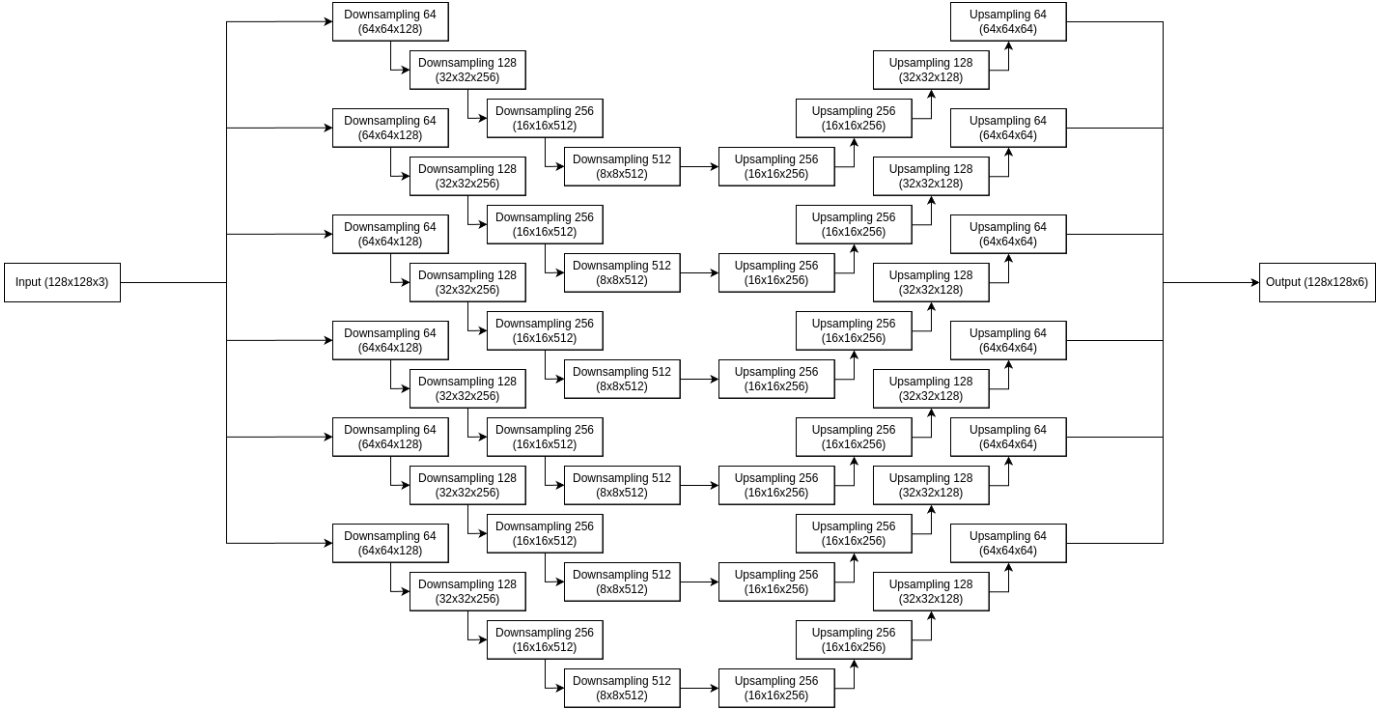


Fig. 2: Multiple Model Single Channel (MMSC)

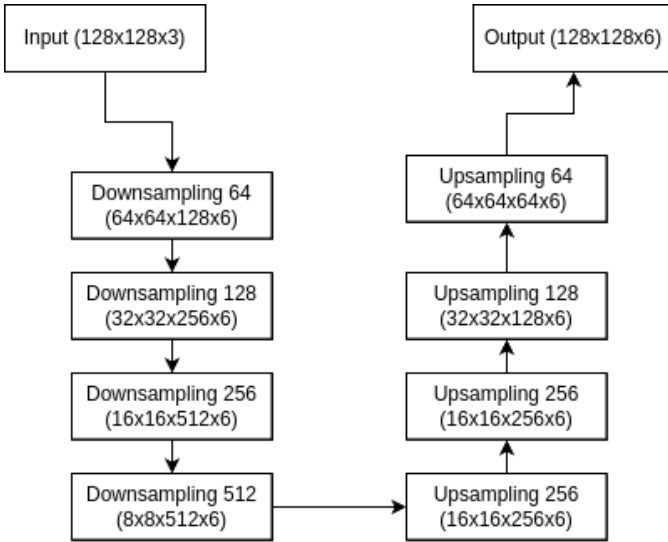


Fig. 3: Single Model Multiple Channel (SMMC)

IV. METRICS AND EVALUATION

We introduce the metrics of our model and evaluation method that had been use in our work.

A. Pixel-wise Accuracy

Pixel-wise accuracy is a metric used to evaluate the performance of image segmentation models. It measures the proportion of correctly classified pixels over the total number of pixels in an image or set of images. Mathematically, it is defined as

$$\text{Pixel-wise Accuracy} = \frac{\text{No. Correctly Classified Pixels}}{\text{Total No. of Pixels}}$$

This metric provides a straightforward way to assess how well the segmentation model distinguishes between different regions or objects within an image. High pixel-wise accuracy indicates that the model effectively assigns the correct labels to most pixels, leading to precise and reliable segmentation results.

B. Mean Intersect over Union (MeanIoU)

Mean Intersection over Union (Mean IoU) is a performance metric for image segmentation models. It measures the average overlap between predicted and ground truth segments across all classes. IoU is calculated as the intersection of predicted and true regions divided by their union. Mean IoU, the average IoU across all classes, provides a robust evaluation by accounting for both false positives and false negatives, reflecting model accuracy comprehensively.

3) *Tanh*: Tanh It maps input values to a range between -1 and 1, providing zero-centered outputs. Tanh introduces non-linearity, enabling the model to capture complex patterns. It also helps mitigate the vanishing gradient problem compared to the sigmoid function.

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

$$\text{Mean IoU} = \frac{1}{N} \sum_{i=1}^N \frac{|A_i \cap B_i|}{|A_i \cup B_i|}$$

where:

N is the number of classes

A_i is the set of pixels in the predicted mask for class i

B_i is the set of pixels in the ground truth mask for class i
for intersection,

$$A \cap B = \sum_i a_i \wedge b_i$$

and union,

$$A \cup B = \sum_i a_i \vee b_i$$

$$\text{IoU} = \frac{\sum_i a_i \cdot b_i}{\sum_i (a_i + b_i - a_i \cdot b_i)}$$

where:

a_i is binarised value of the i -th pixel in the predicted mask

b_i is binarised value of the i -th pixel in the ground truth mask

C. RMSE

Root Mean Square Error (RMSE) is a widely used metric for evaluating the accuracy of a model's predictions. It measures the square root of the average of the squared differences between the predicted and actual values. RMSE provides a way to quantify the difference between predicted and observed values, with lower values indicating better model performance. It is particularly useful because it gives a high penalty to large errors, making it sensitive to outliers.

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (a_i - \hat{b}_i)^2}$$

where:

a_i is the actual values

b_i is the predicted values

n is the number of pixel

Loss function

We customise Loss function to serve our own work. This is done by combine MeanIoU with RMSE with different ratio.

Metric

We prefer using Validation MeanIoU to be a key for tracking the performance. Since it gives overview performance of a model among all channel.

TABLE II: Model Comparison Between Multiple Model Single Channel (MMS) and Single Model Multiple Channel (SMMC)

Model	MeanIoU
ReLU	0.896
LeakyReLU	0.829
tanh	0.763

V. RESULTS

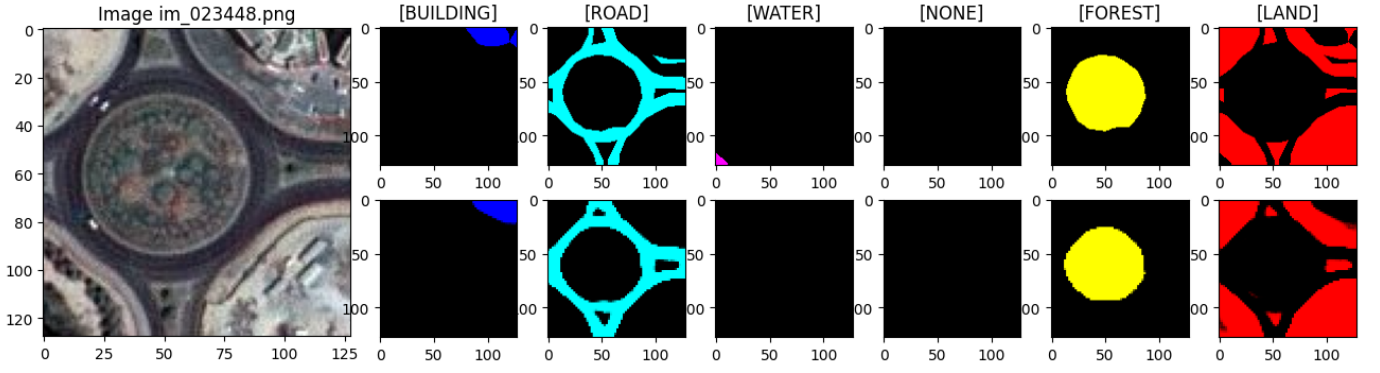
A. Performance Comparison

In this evaluation of mean Intersection over Union (Mean IoU), the Rectified Linear Unit (ReLU) activation function delivered the best performance with a Mean IoU of 0.896. This high score reflects ReLU's ability to facilitate robust and stable learning, contributing to accurate segmentation results. As shown in Fig. 4a. In contrast, the Leaky Rectified Linear Unit (Leaky ReLU) showed slightly lower performance, achieving a Mean IoU of 0.829. As shown in Fig. 4b. Although Leaky ReLU addresses the "dying ReLU" problem, it still falls short of ReLU's superior performance. The hyperbolic tangent (tanh) activation function performed the worst among the three, with a Mean IoU of 0.763. As shown in Fig. 4c. This lower score highlights tanh's inconsistency and slower convergence, making it less effective for precise segmentation tasks. As shown in Table II.

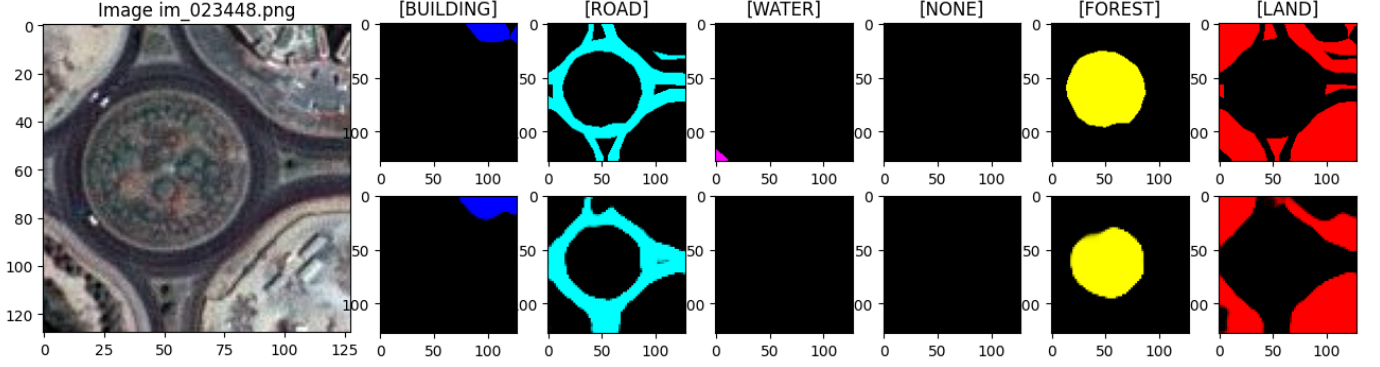
B. Performance on Training

In this work, we observed that the Rectified Linear Unit (ReLU) activation function delivered the best performance among the tested activation functions. ReLU not only facilitated faster convergence but also maintained stability throughout the training epochs. This consistency is likely due to ReLU's inherent ability to mitigate the vanishing gradient problem, ensuring that the model learns effectively even in deep networks. As shown in 5. On the other hand, the Leaky Rectified Linear Unit (Leaky ReLU) showed a slightly slower convergence compared to ReLU. Although it addressed the "dying ReLU" problem by allowing a small, non-zero gradient when the input is negative, this modification did not translate into faster learning. Despite this, Leaky ReLU performed reasonably well and maintained a relatively stable mean Intersection over Union (Mean IoU) metric, although not as consistently as ReLU. As shown in 6. Conversely, the tanh activation function gave the poorest performance in our tests. Tanh not only converged the slowest but also showed significant inconsistency throughout the training process. The Mean IoU for tanh fluctuated dramatically, alternately rising and dropping sharply across epochs. This volatility suggests that tanh struggles to maintain a stable learning trajectory, possibly due to its outputs saturating at large values, which worsen the vanishing gradient problem. As shown in 7.

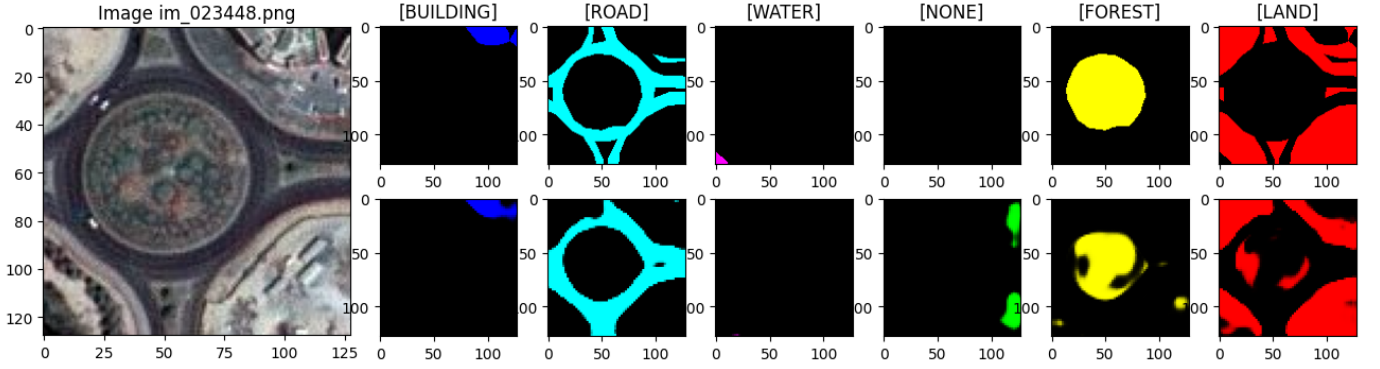
Both Leaky ReLU and tanh demonstrated inconsistency in their Mean IoU metrics during the training epochs. Unlike ReLU, whose performance was consistently robust, Leaky ReLU and tanh alternately rose and fell in their Mean IoU,



(a) Prediction Result From ReLu model



(b) Prediction Result From LeakyReLu model



(c) Prediction Result From tanh model

Fig. 4: Prediction result from various model

indicating unstable learning dynamics. This instability can hinder the overall effectiveness of the model, making these activation functions less desirable.

VI. APPLICATION

A. Application with Real Data from Google Maps

In this work, we employed the ReLu model with real images sourced from Google Maps, using the Google Maps API. We tested the model on three locations: Shezi District, Abu Dhabi, and another diverse location. Shezi District, with its urban landscape surrounded by small forest areas and a river, presented a varied environment. Abu Dhabi's arid land adjacent to a river provided a stark contrast. The results indicated that

the model effectively segmented major areas such as roads, water bodies, and vegetation. However, it struggled with high-detail classes like buildings, leading to segmentation errors. This discrepancy is likely due to the differing characteristics of buildings in Abu Dhabi and Shezi District, highlighting the model's limited tolerance to highly varied data. These findings suggest that while the model performs well overall, improvements are needed to enhance its accuracy for detailed structures across diverse landscapes.

VII. CONCLUSION

We explore the application of deep learning techniques, particularly the U-Net architecture, for the semantic segmentation

further improvements in segmenting highly detailed structures.

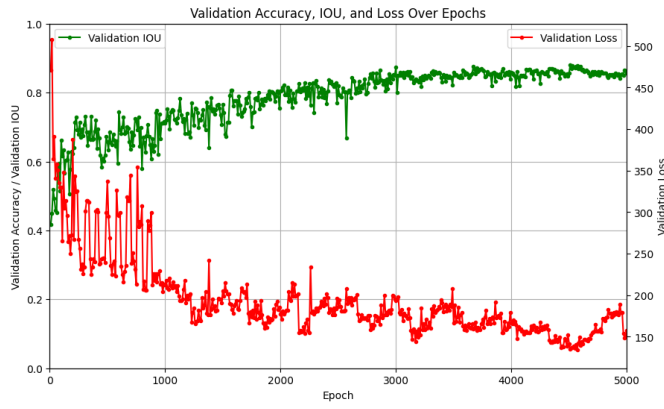


Fig. 5: MeanIoU over Training Epoch for ReLu

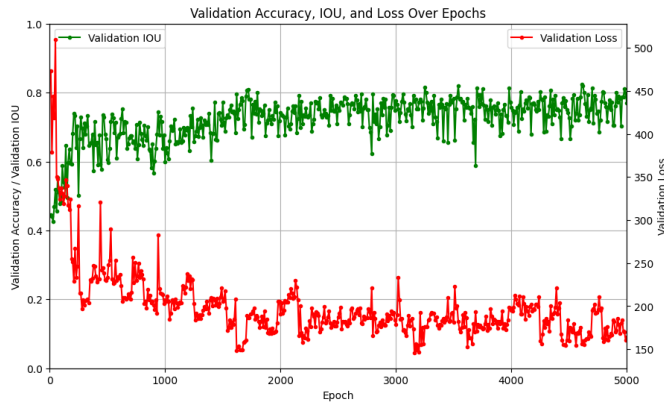


Fig. 6: MeanIoU over Training Epoch for LeakyReLu

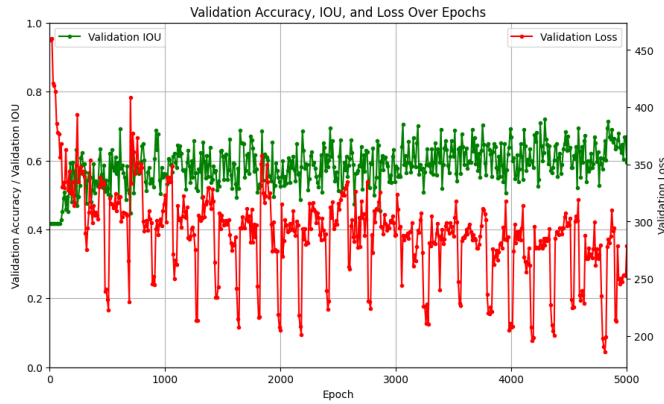
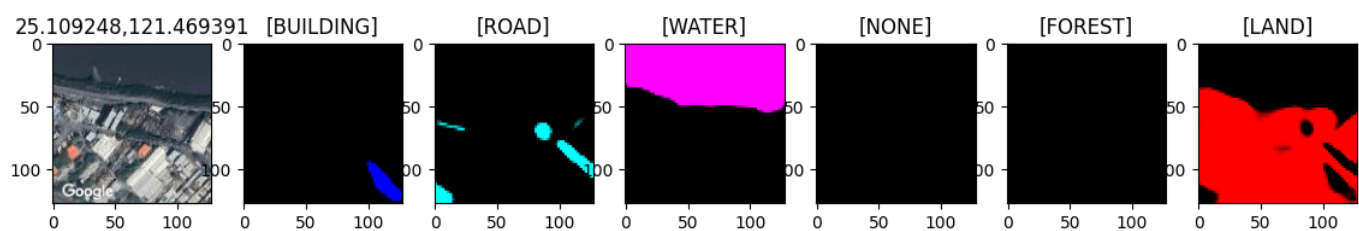
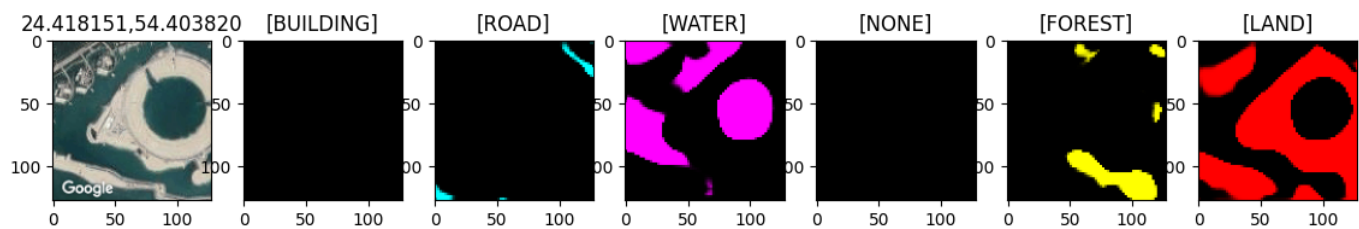


Fig. 7: MeanIoU over Training Epoch for tanh

of aerial images. The study demonstrates the advantages of deep learning over traditional methods in terms of accuracy and efficiency. We compare the performance of different activation functions, with ReLu achieving the highest Mean IoU at 0.896, followed by Leaky ReLu at 0.829, and Tanh at 0.763. Through detailed data preparation and preprocessing, we ensure robust model training. The application of the model on real-world data from Google Maps showcases its potential in diverse environments, though it highlights the need for



(a) Prediction Result From Shezi area



(b) Prediction Result for Abu Dhabi

Fig. 8: Prediction result on real data from Google Maps