Spatial Detection of Slums in Dhaka Using U-Net
(Acme AI Fellowship Program 2025- Stage 01)

1. Introduction

Urban slums pose significant challenges for city planning and development. Identifying slum areas using satellite imagery can aid in better decision-making for resource allocation and infrastructure development. This project focuses on detecting slums in Dhaka using machine learning techniques on satellite images.

2. Data Preprocessing

2.1 Dataset Description: The dataset contains 33 high-quality satellite images of urban and peri-urban areas in Mirpur, Dhaka. Each image is paired with a corresponding mask image, except for three missing mask images, which are considered unmasked images for future inference (Testing the model with unseen images during training). The goal is to classify slum and non slum areas in a picture.

2.2 Image Enhancement: To enhance the features like building density and infrastructure, the images are resized to a uniform dimension of 256x256 and 512x512 pixels. This resizing ensures consistency in input size, which is crucial for training deep learning models [1]. Additionally, pixel values are normalized to the range [0, 1] to improve model performance by reducing computational complexity and ensuring that the neural network processes the data more efficiently.

2.3 Data Augmentation: In this specific preprocessing stage, no augmentation techniques such as rotation or flipping are applied. However, the dataset is split into training (80%) and validation (20%) sets to evaluate the model's generalization capabilities.

2.4 Annotation & Labeling: Each image has an associated mask that annotates the slum areas. These masks are grayscale images where the slum areas are marked with intensity values (0 for non-slum and 1 for slum). The three unmasked images will be used as test data for inference. For the remaining images, mask paths are matched to the images for training the model.

3. Model Development

3.1 Model Architecture: A U-Net architecture was selected for this task due to its effectiveness in biomedical and remote sensing segmentation [2]. U-Net's symmetric encoder-decoder structure allows precise localization of slum areas by capturing both global and fine-grained spatial details [3], [4].

3.2 Training Methodology: The model was trained using binary cross-entropy loss to optimize for pixel-wise segmentation of slum vs. non-slum areas. The Adam optimizer was used for efficient convergence. The training was performed with 300 epochs and a batch size of 5 (256*256), 300 epochs and a batch size of 6 (512*512), 150 epochs and a batch size of 6 (512*512) balancing computational efficiency with learning stability. The models are trained with default Google Colab Environment.

3.3 Validation Strategy: The dataset was split into **80% training and 20% validation** to assess model performance. The validation accuracy and loss were tracked across epochs to monitor overfitting.

4. Results , Analysis and Challenges

4.1 Performance Metrics:

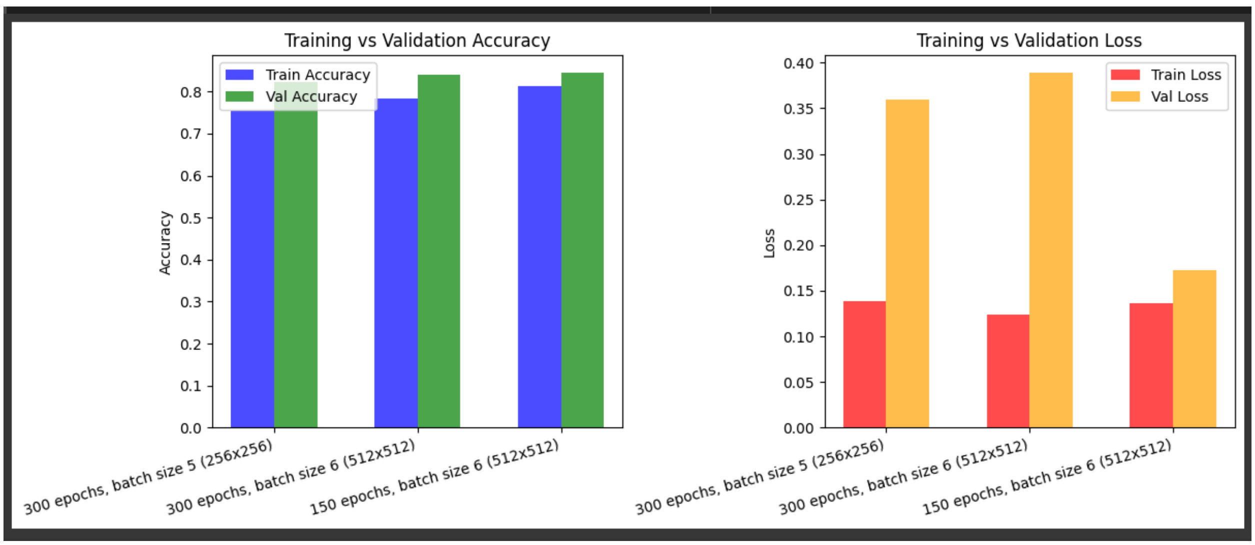

Fig 1: Training vs Validation (Accuracy and Loss) based on the last epoch result

4.2 Visualization Outputs:

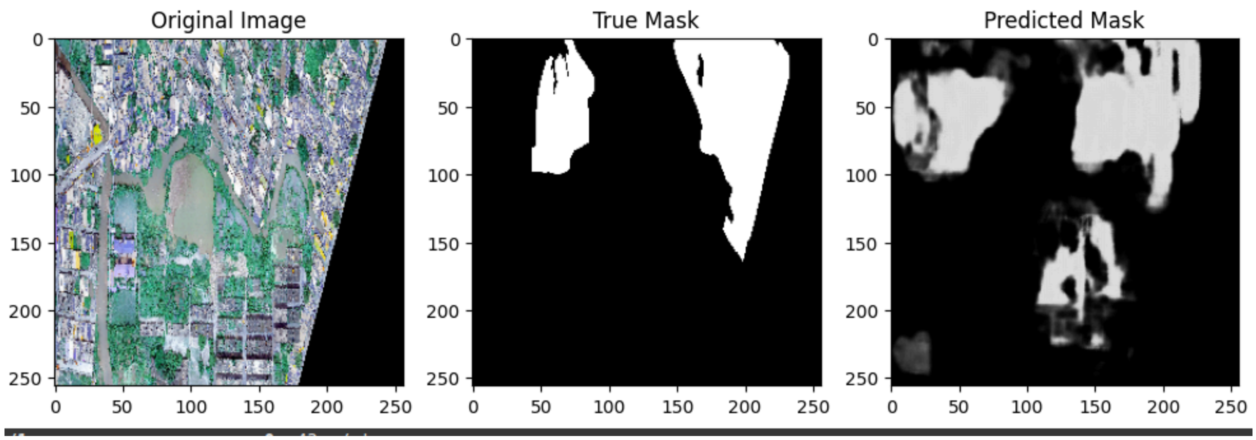

Fig 2: Validation output based (300 epochs. Batch size: 5, 256 x 256)

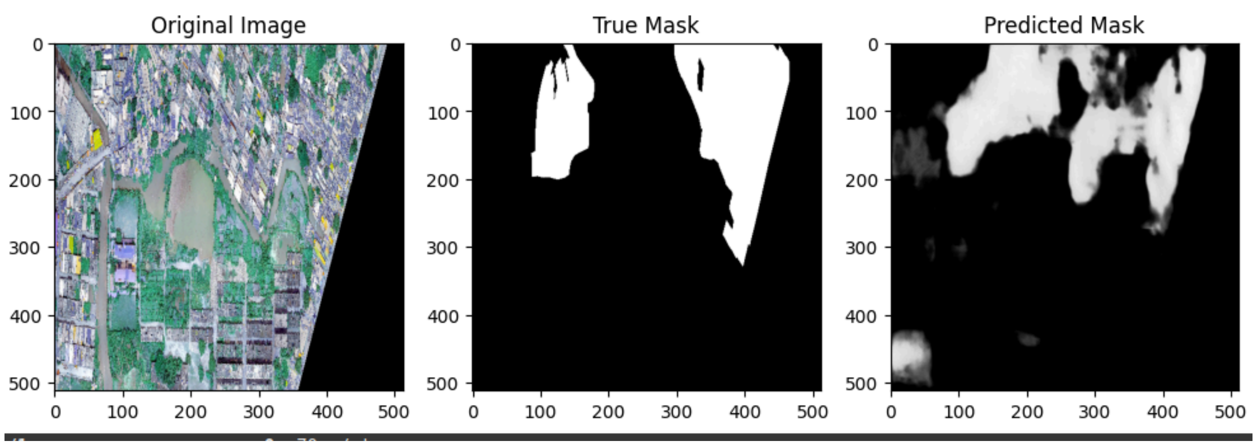

Fig 3: Validation output based (300 epochs. Batch size: 6, 512 X 512)

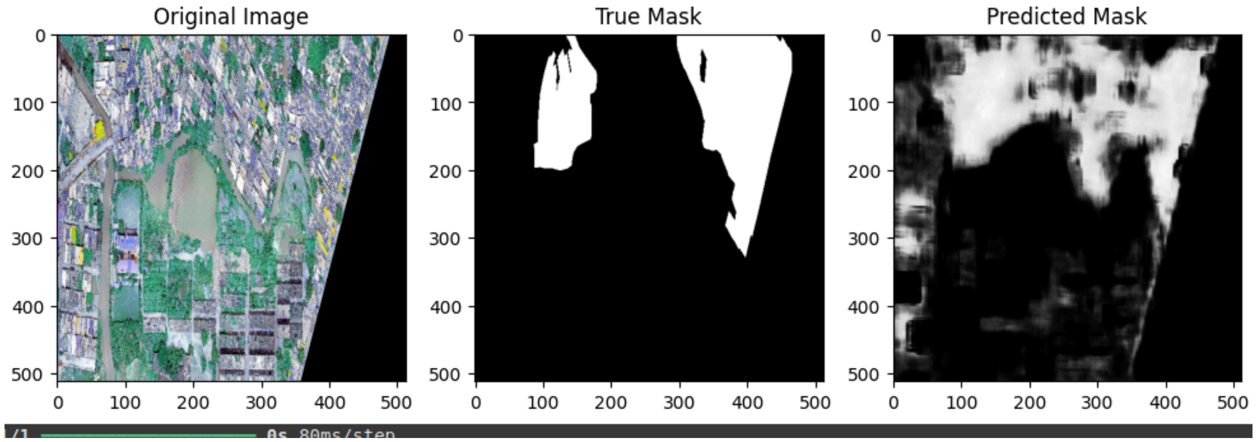

Fig 4: Validation output based (150 epochs. Batch size: 6 , 512 X 512

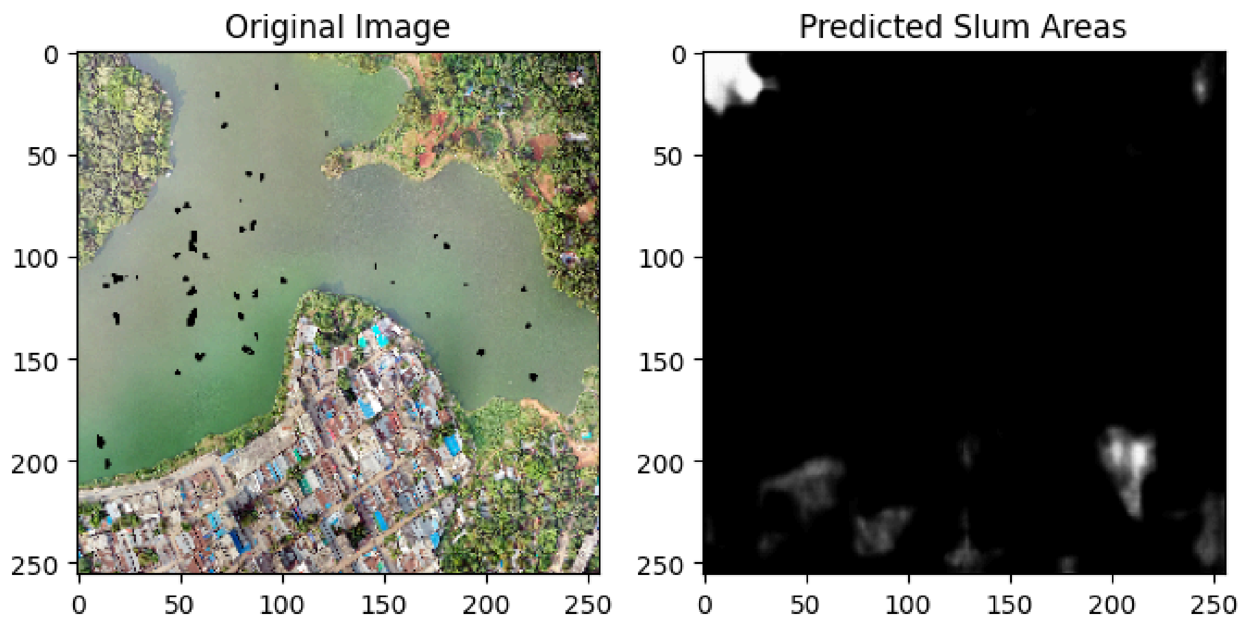

Fig 5: Validation output based (300 epochs. Batch size: 5, 256 x 256)

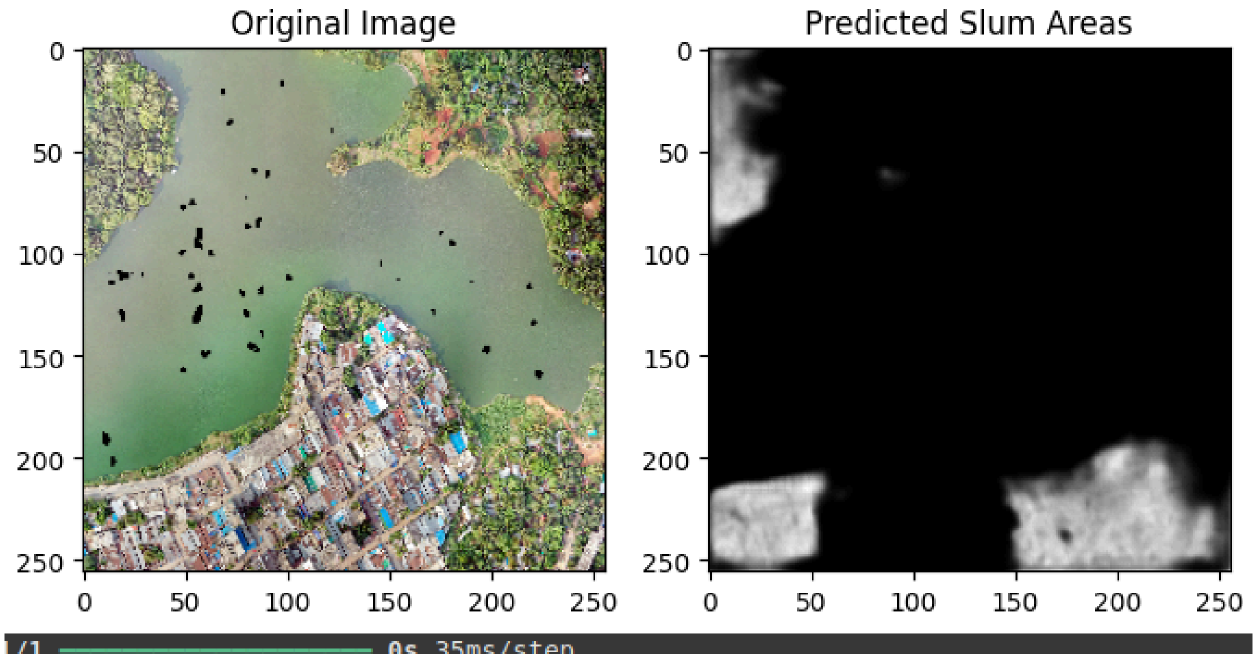

Fig 6: Validation output based (300 epochs. Batch size: 6, 512 X 512)

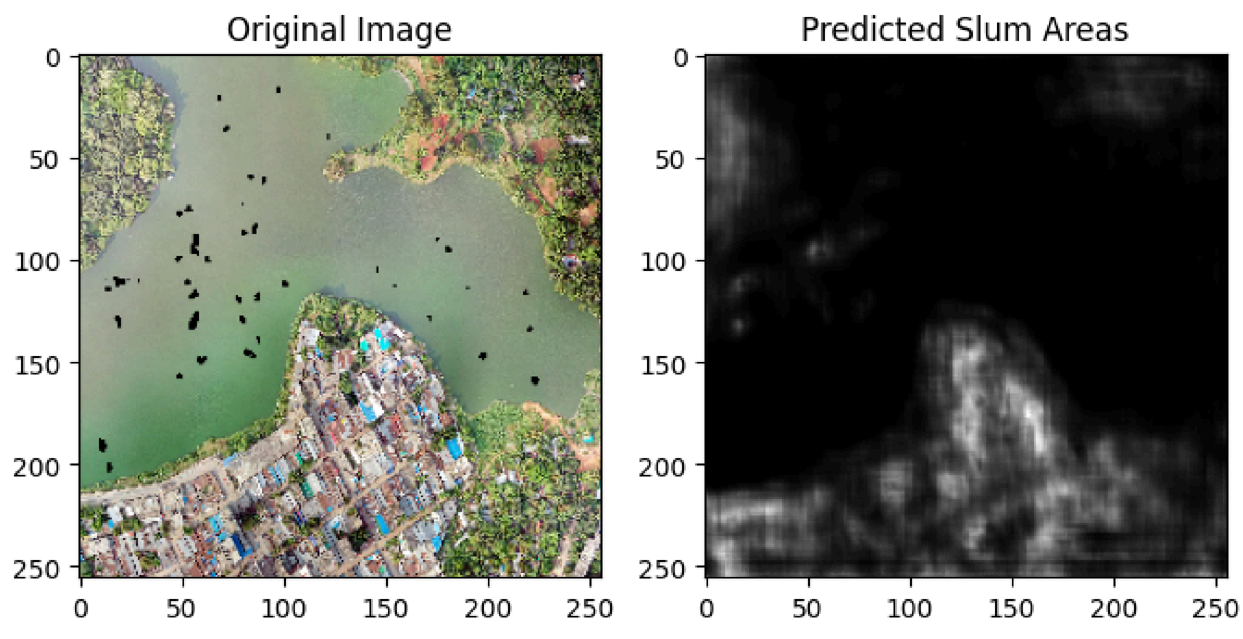

Fig 7: Validation output based (150 epochs. Batch size: 6 , 512 X 512)

4.3 Analysis and Challenges:

An interesting observation from Figure 1 is that the validation accuracy is higher than the training accuracy. This is likely due to the small dataset (24 train, 6 val), making the validation set easier to predict. Overfitting is possible but typically results in higher training accuracy. To address this, increasing the dataset size or adding data augmentation could help.

Based on the visual results, the models performed well, especially the one trained with a higher resolution (512×512), 150 epochs, and a batch size of 6, compared to other configurations.

More visual results are attached in the Google Colab and its PDF file [5].

Balancing image resolution, dataset size, epoch count, and batch size was challenging due to the small training set. Increasing epochs with a small batch size often led to stagnation in validation accuracy. For future work, a larger dataset and fine-tuned training parameters could enhance prediction accuracy. Also other deep learning models like Mask -CNN can also be used. [6]. Furthermore, using pre-trained weights from larger segmentation datasets can also be useful.

6. Conclusion

This study successfully implemented a U-Net model for slum detection in Dhaka using satellite imagery. Despite challenges related to dataset size and training stability, the model demonstrated promising results, particularly with higher-resolution images and optimized training configurations. The findings highlight the importance of balancing resolution, batch size, and epochs to achieve better segmentation accuracy. Future improvements could focus on expanding the dataset, incorporating data augmentation, and fine-tuning hyperparameters for enhanced performance.

7. References

1. Chollet, F. (2017). *Deep Learning with Python*. Manning Publications.
2. Ronneberger, O., Fischer, P., & Brox, T. (2015). *U-Net: Convolutional networks for biomedical image segmentation*. International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI).
3. Rakhlin, A., Davydow, D., & Nikolenko, S. (2018). *Land cover mapping using U-Net and Sentinel-2 imagery*. arXiv preprint arXiv:1809.04647
4. Bhatia, V. (n.d.). U-Net Implementation. GitHub repository. Retrieved from https://github.com/VidushiBhatia/U-Net-Implementation.
5. https://github.com/iamsmsr/Spatial-Detection-of-Slums-in-Dhaka-Using-U-Net-.git

6. Maiya, S. R., & Babu, S. C. (2018). Slum segmentation and change detection: A deep learning approach. arXiv preprint arXiv:1811.07896. https://arxiv.org/abs/1811.07896