Comparison of Image Segmentaion Models: Using Satellite Land Cover Images

Daniel Witoslawski,
Huiseong Cho, Jaeyoung Jung, HyoungJoon Park, Yutak Seo, Taehyun Kim

2024.06.25 ~ 2024.07.16

UNLV Research Internship Program

INDEX

INTRODUCTION & LITERATURE REVIEW

Introduction

TOPIC

Semantic Image Segmentation

PURPOSE

Study Computer Vision Models

SIGNIFICANCE

Apply to Various Area

- Automobile
- Medical Image
- Agriculture

PROJECT GOALS:

COMPARE AI MODELS, ANALYSIS OF RESULTS

Literature Review

PRIOR RESEARCH

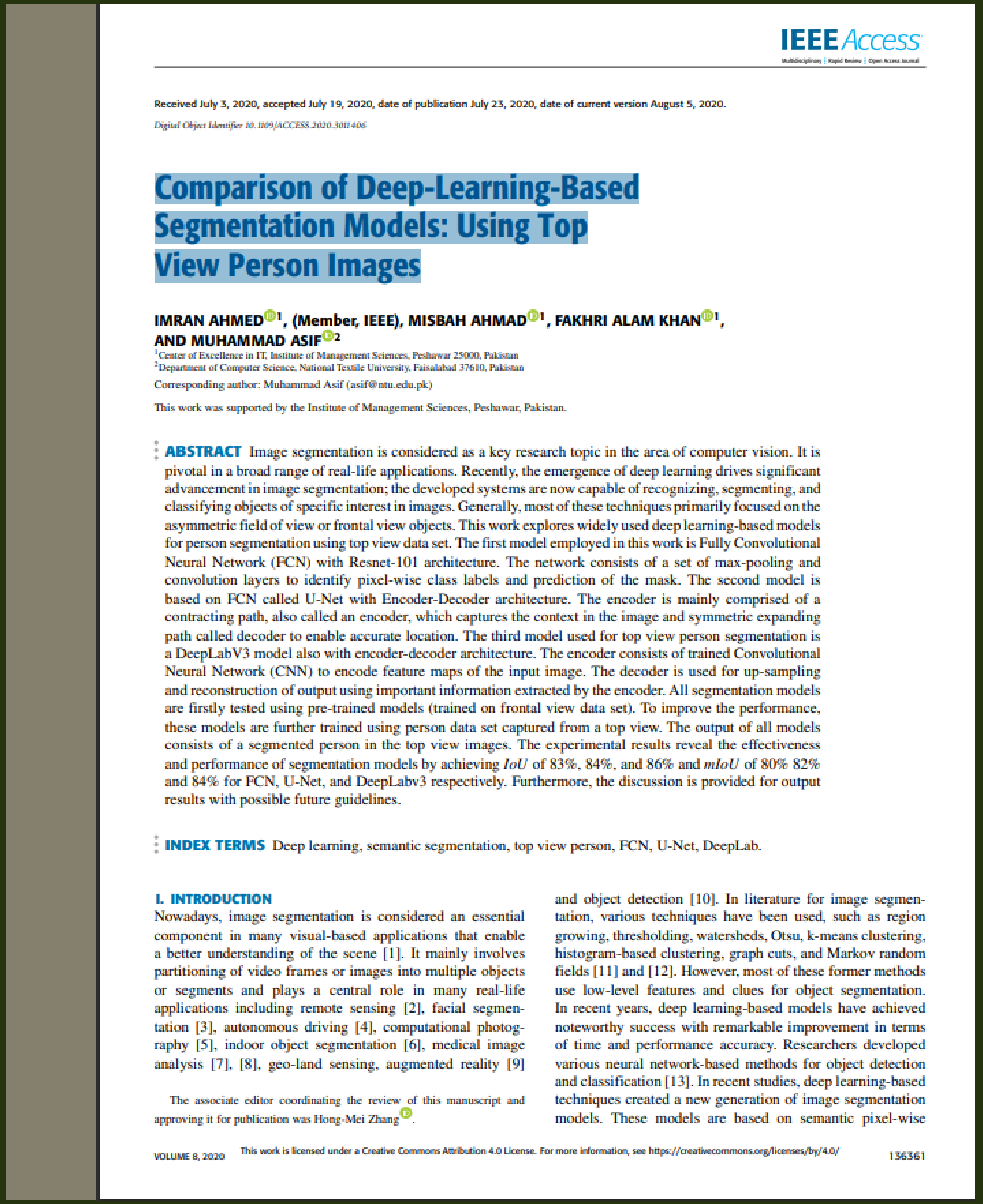
Comparison of Deep-Learning-Based Segmentation Models: Using Top View Person Images

COMPARED : FCN, U-NET, DeeplabV3

Accuracy, Loss, Time Inference , IOU

WE COMPARE : U-NET++, DeeplabV3+

More Recent Models



IEEE Access

Received July 3, 2020, accepted July 19, 2020, date of publication July 23, 2020, date of current version August 5, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3011406

Comparison of Deep-Learning-Based Segmentation Models: Using Top View Person Images

IMRAN AHMED[1], (Member, IEEE), MISBAH AHMAD[1], FAKHRI ALAM KHAN[1], AND MUHAMMAD ASIF[2]

[1]Center of Excellence in IT, Institute of Management Sciences, Peshawar 25000, Pakistan

[2]Department of Computer Science, National Textile University, Faisalabad 37610, Pakistan

Corresponding author: Muhammad Asif (asif@ntu.edu.pk)

This work was supported by the Institute of Management Sciences, Peshawar, Pakistan.

ABSTRACT Image segmentation is considered as a key research topic in the area of computer vision. It is pivotal in a broad range of real-life applications. Recently, the emergence of deep learning drives significant advancement in image segmentation; the developed systems are now capable of recognizing, segmenting, and classifying objects of specific interest in images. Generally, most of these techniques primarily focused on the asymmetric field of view or frontal view objects. This work explores widely used deep learning-based models for person segmentation using top view data set. The first model employed in this work is Fully Convolutional Neural Network (FCN) with Resnet-101 architecture. The network consists of a set of max-pooling and convolution layers to identify pixel-wise class labels and prediction of the mask. The second model is based on FCN called U-Net with Encoder-Decoder architecture. The encoder is mainly comprised of a contracting path, also called an encoder, which captures the context in the image and symmetric expanding path called decoder to enable accurate location. The third model used for top view person segmentation is a DeepLabV3 model also with encoder-decoder architecture. The encoder consists of trained Convolutional Neural Network (CNN) to encode feature maps of the input image. The decoder is used for up-sampling and reconstruction of output using important information extracted by the encoder. All segmentation models are firstly tested using pre-trained models (trained on frontal view data set). To improve the performance, these models are further trained using person data set captured from a top view. The output of all models consists of a segmented person in the top view images. The experimental results reveal the effectiveness and performance of segmentation models by achieving *IoU* of 83%, 84%, and 86% and *mIoU* of 80% 82% and 84% for FCN, U-Net, and DeepLabv3 respectively. Furthermore, the discussion is provided for output results with possible future guidelines.

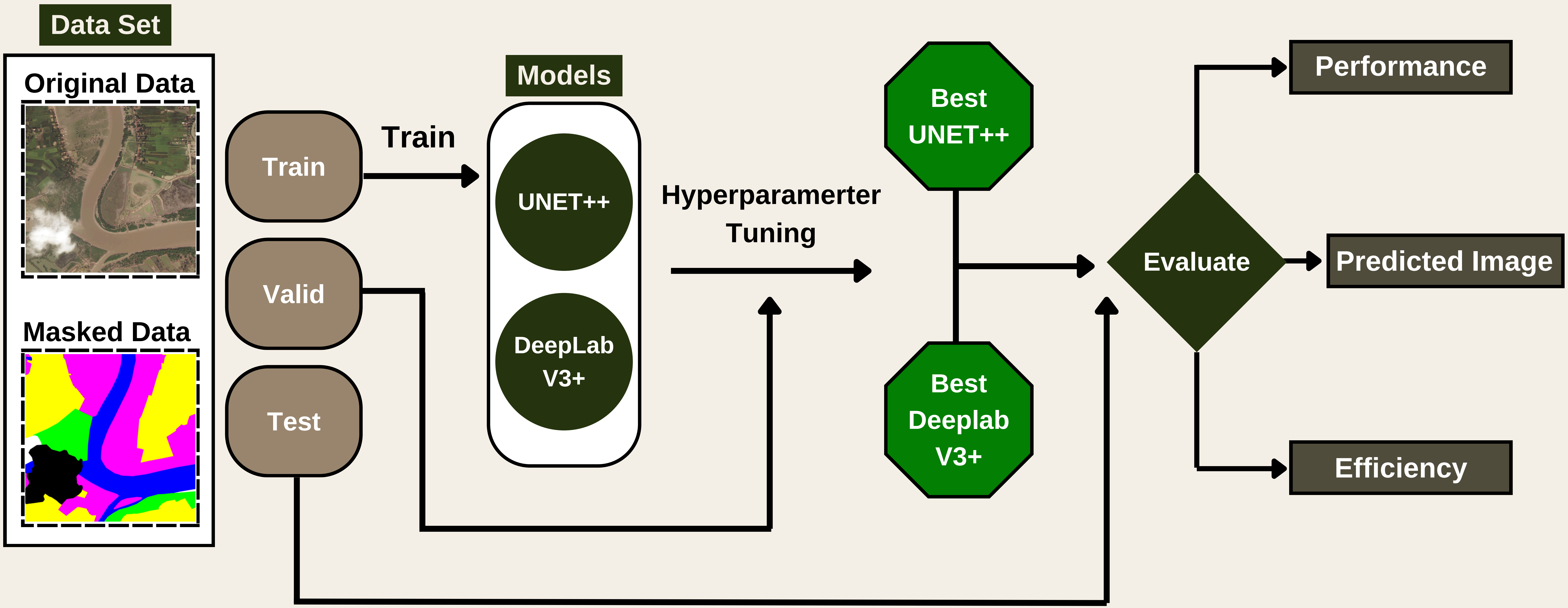
INDEX TERMS Deep learning, semantic segmentation, top view person, FCN, U-Net, DeepLab.

I. INTRODUCTION

Nowadays, image segmentation is considered an essential component in many visual-based applications that enable a better understanding of the scene [1]. It mainly involves partitioning of video frames or images into multiple objects or segments and plays a central role in many real-life applications including remote sensing [2], facial segmentation [3], autonomous driving [4], computational photography [5], indoor object segmentation [6], medical image analysis [7], [8], geo-land sensing, augmented reality [9] and object detection [10]. In literature for image segmentation, various techniques have been used, such as region growing, thresholding, watersheds, Otsu, k-means clustering, histogram-based clustering, graph cuts, and Markov random fields [11] and [12]. However, most of these former methods use low-level features and clues for object segmentation. In recent years, deep learning-based models have achieved noteworthy success with remarkable improvement in terms of time and performance accuracy. Researchers developed various neural network-based methods for object detection and classification [13]. In recent studies, deep learning-based techniques created a new generation of image segmentation models. These models are based on semantic pixel-wise

The associate editor coordinating the review of this manuscript and approving it for publication was Hong-Mei Zhang.

PROJECT FRAMEWORK



DATASET PREPROCESSING

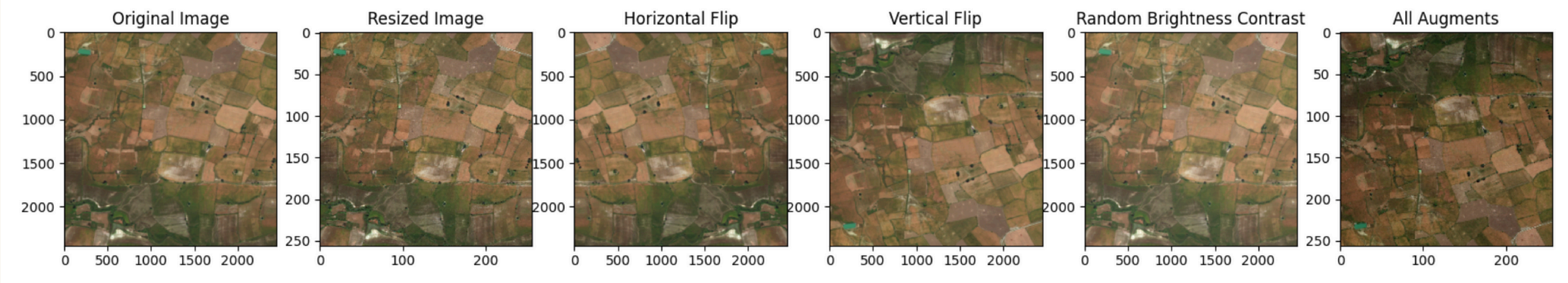
About Dataset

- 803 images
- 7 Classes : Urban, Agriculture, Rangeland, Forest, Water, Barren, Unknown

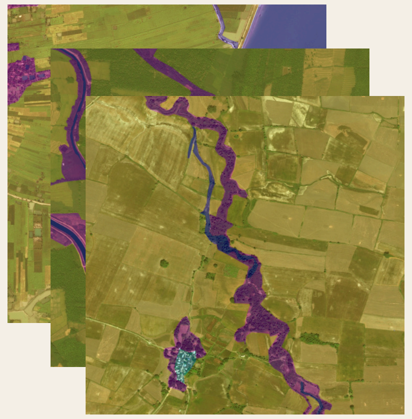
Dataset Augmentation

- Used **Albumentation** Library for Augmentation

	Flip	Resize	Random Brightness
Train	O	O	O
Test	X	O	O



Dataset Splitting



8

2

Test

8

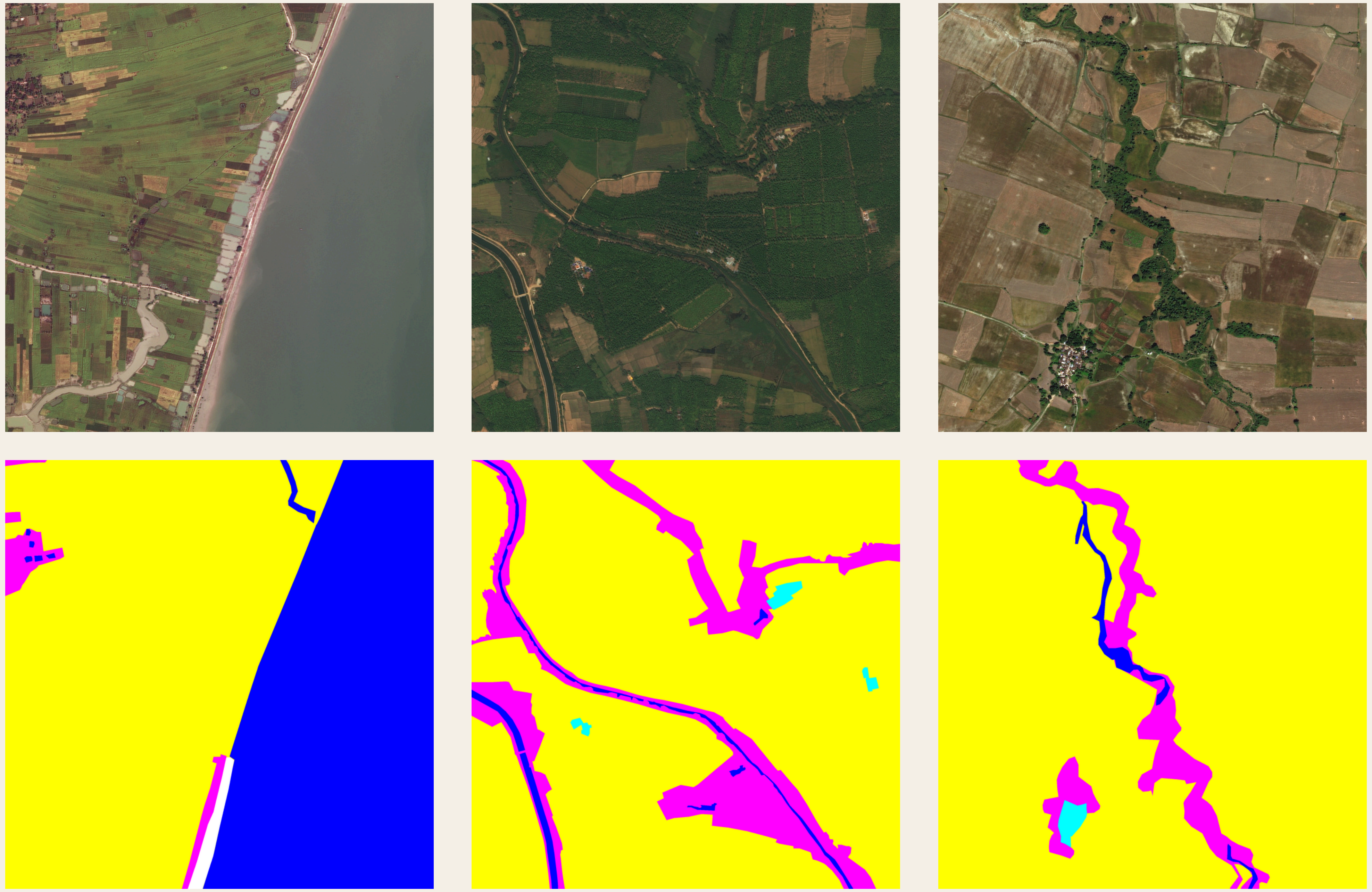
Train

2

Validation

Dataset Splitting

- Problem : Multiple Classes, Imbalanced.
- Color label Balancing for Train/Valid/Test is Essential.



Dataset Splitting

- We made **class vectors** to balance Train/Valid/Test datasets.
- Class vectors represent **existence of each own classes**.

	Urban	Agriculture	Range Land	Forest	Water	Barren	Unkown
image1	1	1	0	1	0	1	0
image2	1	1	1	0	1	0	0
image3	1	1	0	1	0	0	1
image4	0	1	1	1	1	0	0
⋮							
Total	800	722	521	191	471	358	15

Dataset Splitting

Using class vectors, we divided train/test datasets.

	Total	800	722	521	191	471	358	15
8	Train	512	459	333	123	302	268	10
	Valid	128	118	84	30	75	67	2
2	Test	160	145	104	38	94	84	3

Dataset Splitting

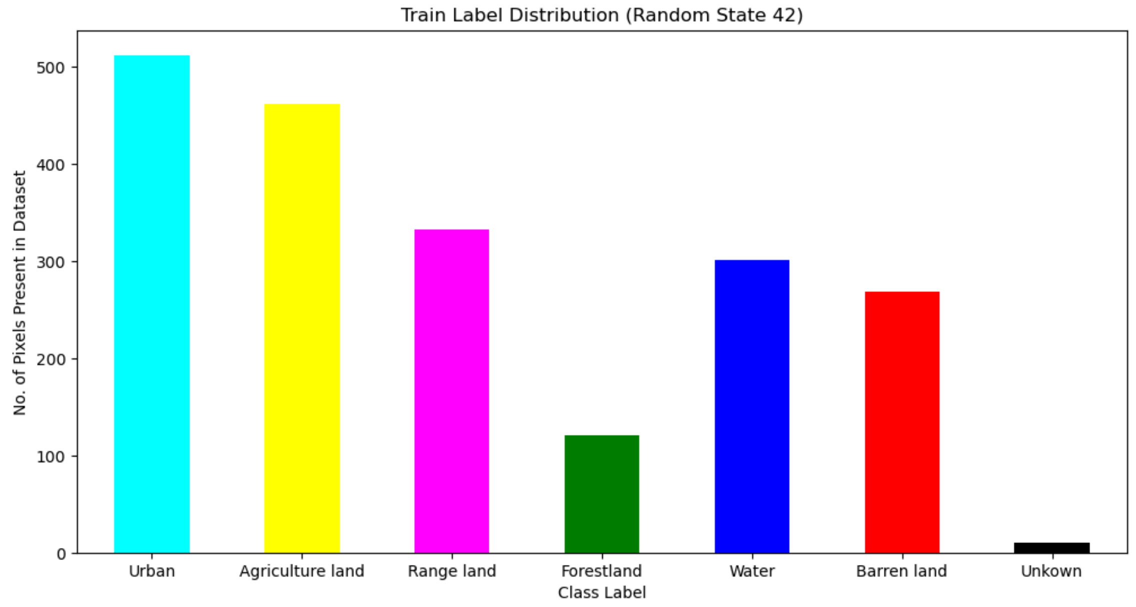
In train datasets, we also divided train/validation datasets.

	Total	800	722	521	191	471	358	15
8 ⟵	Train	512	459	333	123	302	268	10
2 ⟵	Valid	128	118	84	30	75	67	2
	Test	160	145	104	38	94	84	3

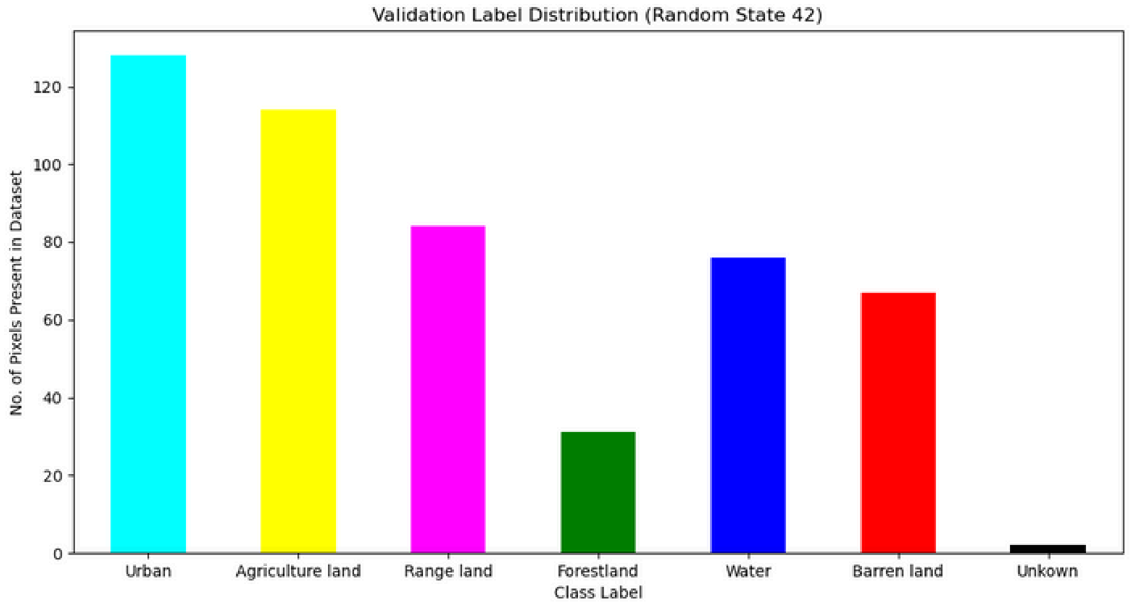
Dataset Splitting

Through data preprocessing, we have gained **balanced datasets**.

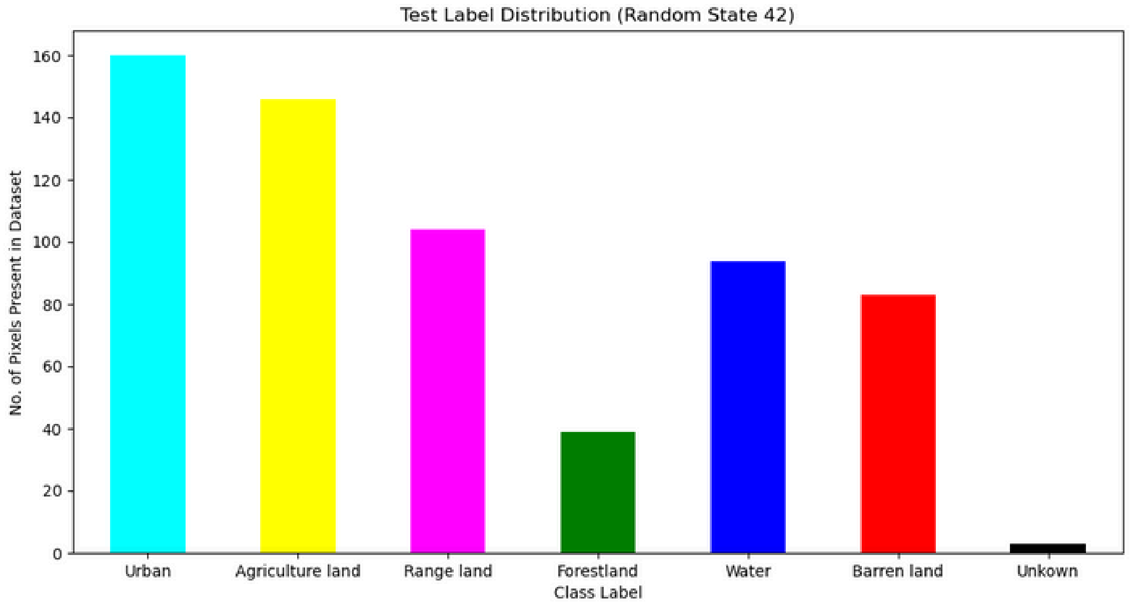
train



valid



test



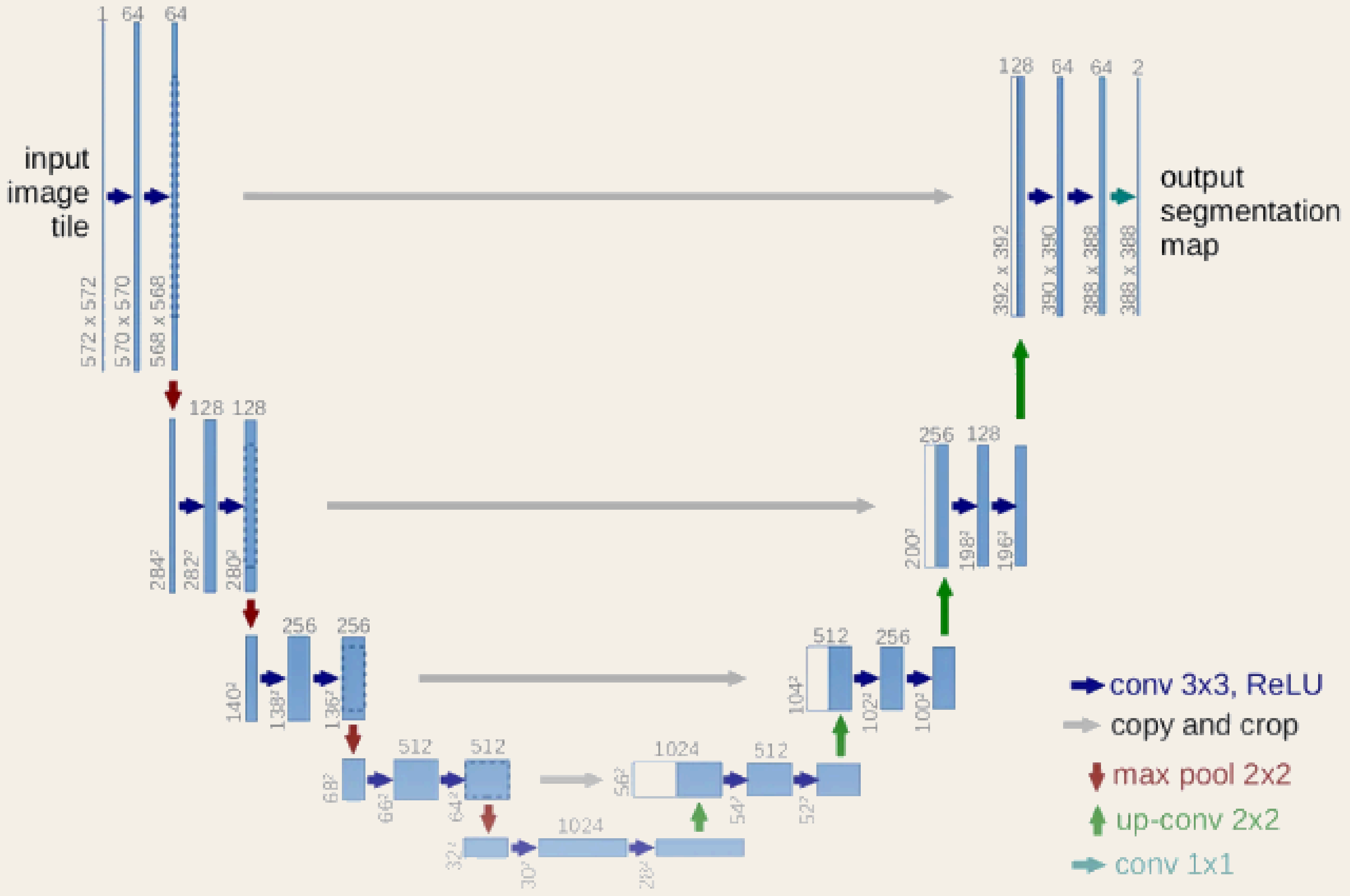
Class distribution for the training/valid/test datasets after dataset preprocessing

Random Sampling Cross-Validation

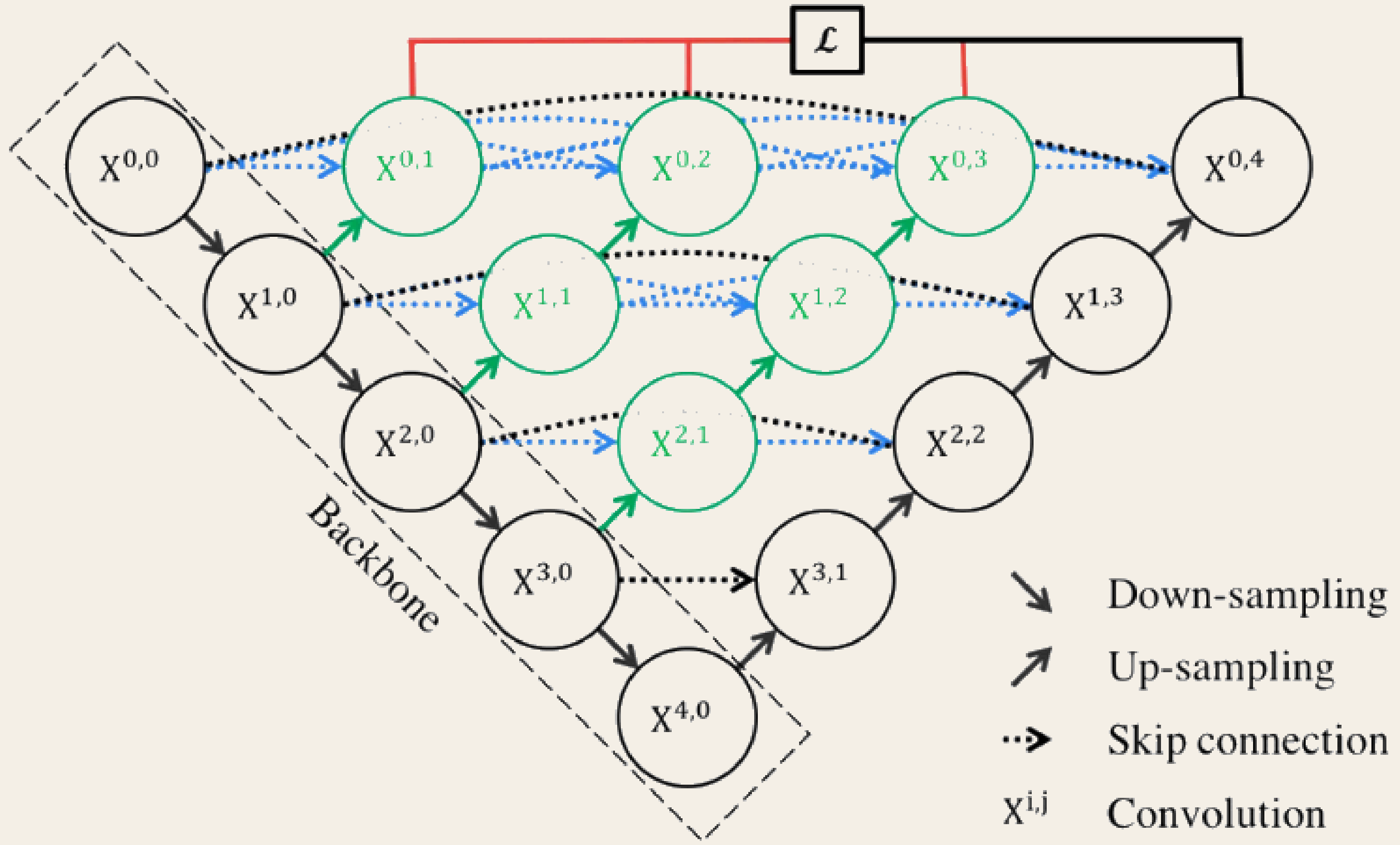
5 Experiment, Avg of Results

HYPERPARAMETER TUNING

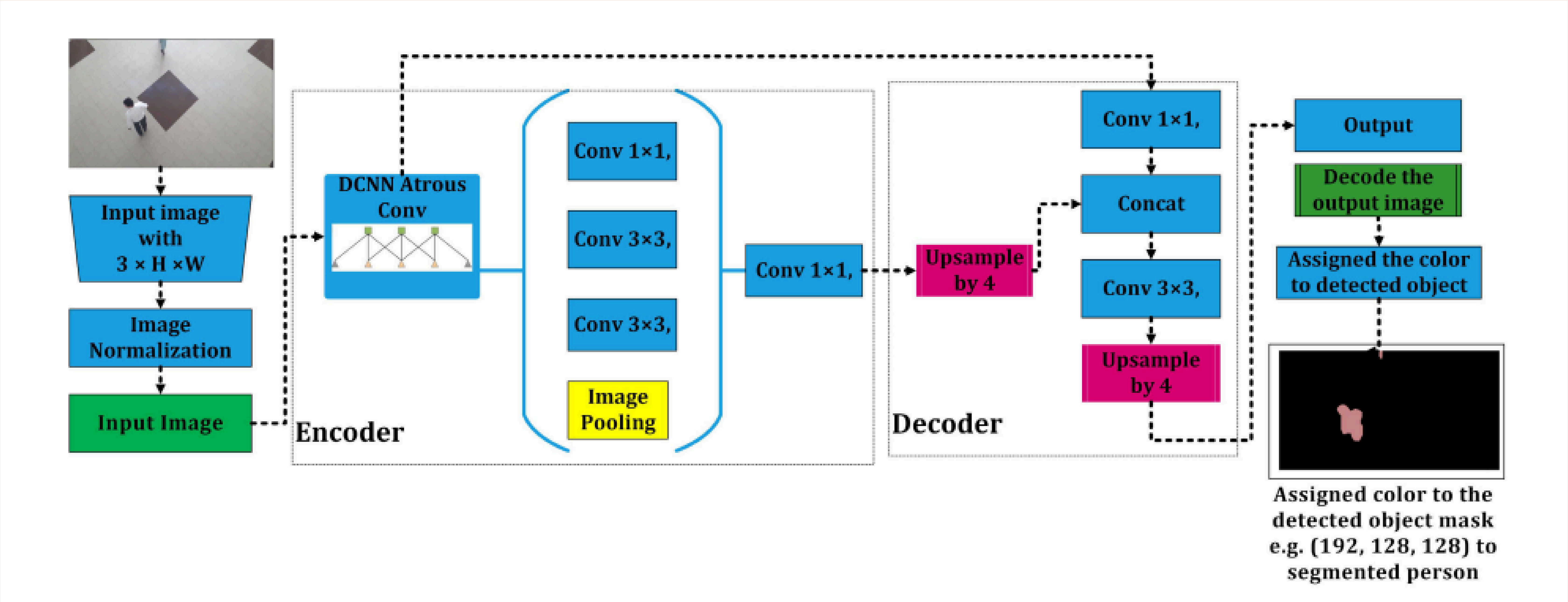
Framework of Model : UNET++



UNet architecture



Framework of Model : DeeplabV3



HYPERPARAMETER TUNING

1 LEARNING RATE

- Represents the **Speed** at which an AI Model "**Learns**"

2 BATCH SIZE

- **Num of Samples** that you feed into Model at **each learning process**

GOAL:

FINDING THE **BEST HYPERPARAMER** COMBINATION (UNET, DEEPLABV3)

Variables

Model	Encoder_Name	Encoder_Weight	Batch size	Learning Rate
DeepLabV3+	ResNet-50	ImageNet	16, 8	0.001, 0.0001
UNet++	RegNetY-120	ImageNet	16, 8	0.001, 0.0001

Segmenation Models

```
model_DeepLabV3Plus = DeepLabV3Plus(
            encoder_name='resnet50',
            encoder_weights='imagenet',
            in_channels=3,
            classes=len(CLASSES),
            activation="softmax"
        )
```

```
model_UnetPlusPlus = UnetPlusPlus(
            encoder_name='timm-regnety_120',
            encoder_weights='imagenet',
            in_channels=3,
            classes=len(CLASSES),
            activation="softmax"
        )
```

Finding Best Hyperparameter

batch_size	lr	model	test/Loss	test/IoU	test/Accuracy	test/Precision	test/Recall	test/F1score	test_time
8	0.0001	DeepLabV3	0.361826	0.764084	0.958398465	0.85439463	0.8543946	0.85439463	58.56128
8	0.0001	UnetPlusPl	0.322544	0.775158	0.960480819	0.86168285	0.8616828	0.86168285	55.99933
8	0.001	DeepLabV3	0.421823	0.688691	0.941493165	0.79522608	0.7952261	0.79522608	58.81569
8	0.001	UnetPlusPl	0.357296	0.74454	0.954129225	0.83945228	0.8394523	0.83945228	58.62923
16	0.0001	DeepLabV3	0.362944	0.764666	0.958534395	0.85487038	0.8548704	0.85487038	57.71321
16	0.0001	UnetPlusPl	0.327609	0.780157	0.961536485	0.8653777	0.8653777	0.8653777	55.80176
16	0.001	DeepLabV3	0.414928	0.719694	0.94853768	0.81988187	0.8198819	0.81988187	57.05049
16	0.001	UnetPlusPl	0.372046	0.750247	0.955418154	0.84396354	0.8439635	0.84396354	58.4261

DeepLabV3+ IoU

batch size	Adam LR = 0.0001	Adam LR = 0.001
8	0.764	0.689
16	**0.765**	0.720

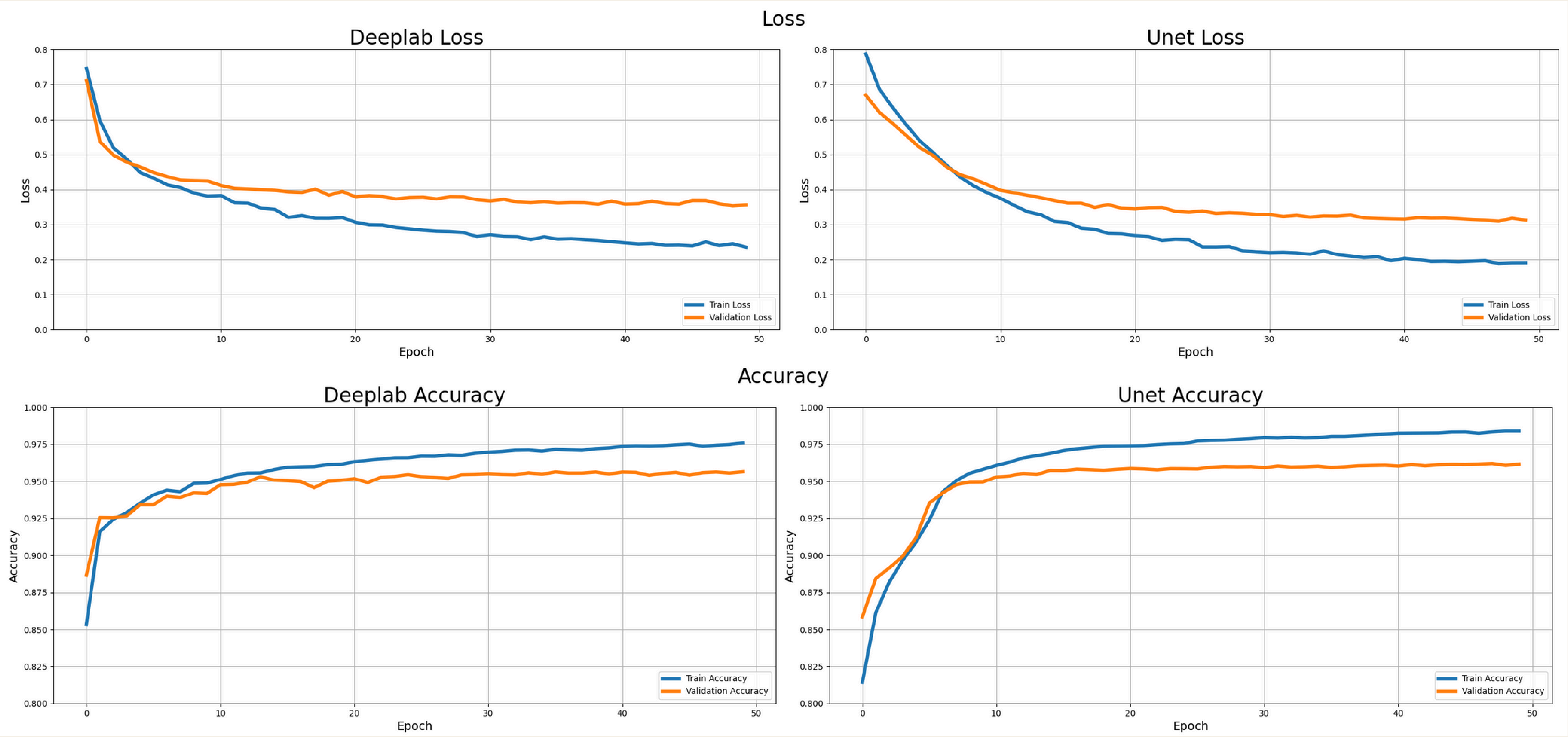
UNet++ IoU

batch size	Adam LR = 0.0001	Adam LR = 0.001
8	0.775	0.745
16	**0.780**	0.750

EXPERIMENTAL RESULTS

Comparison of Best Models’ Acc, Loss

Avg of Train-Valid Loss (5 experiment)

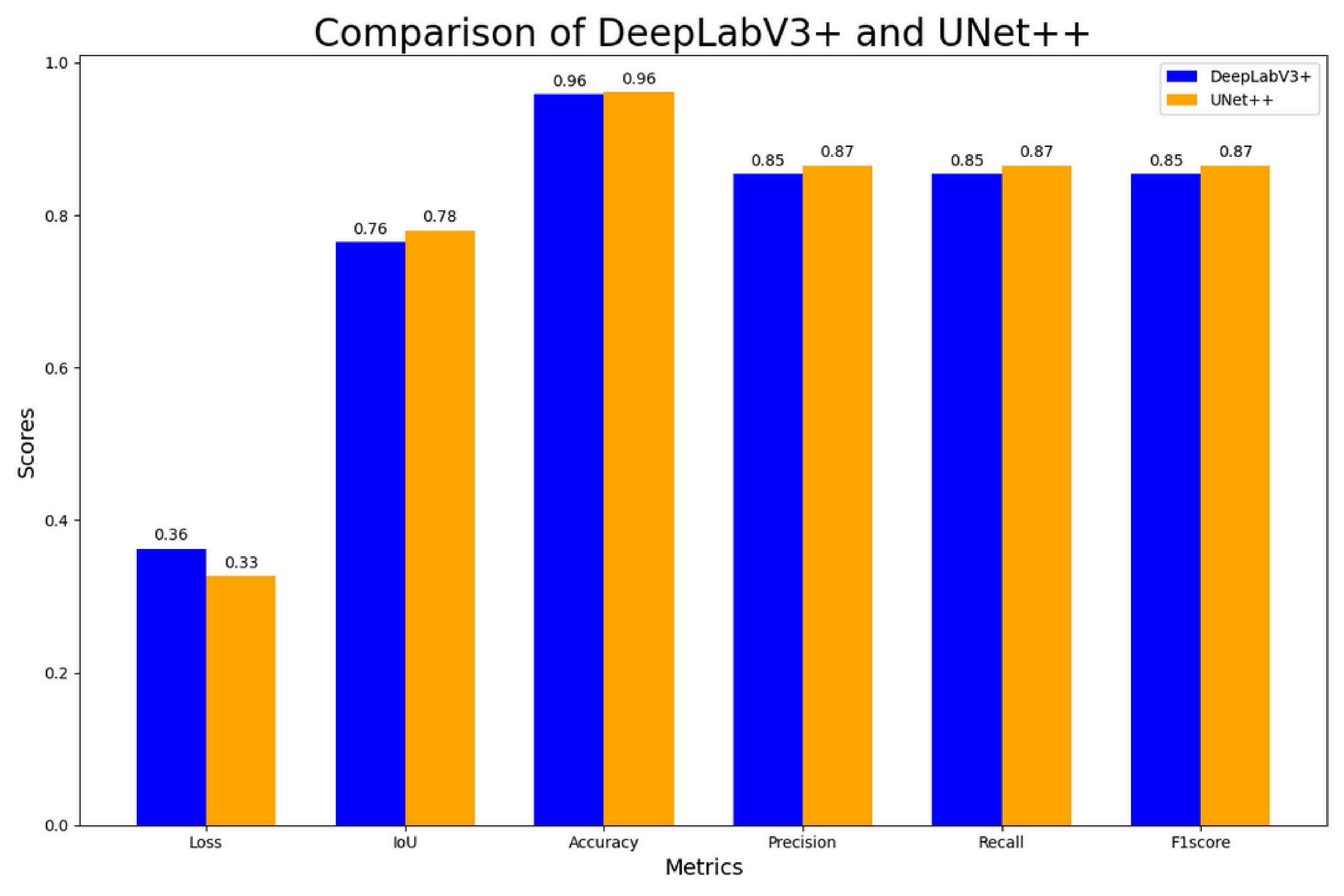


Performance Measure

Dice Loss	F1-Score
IoU	Precision
Accuracy	Recall

Comparison of Performance

Metrics	DeepLabV3+	U-Net++
Dice Loss	0.363	0.328
IoU	0.765	0.780
Accuracy	0.959	0.962
Precision	0.855	0.865
Recall	0.855	0.865
F1-Score	0.855	0.865



Comparison of Efficiency

Unet++

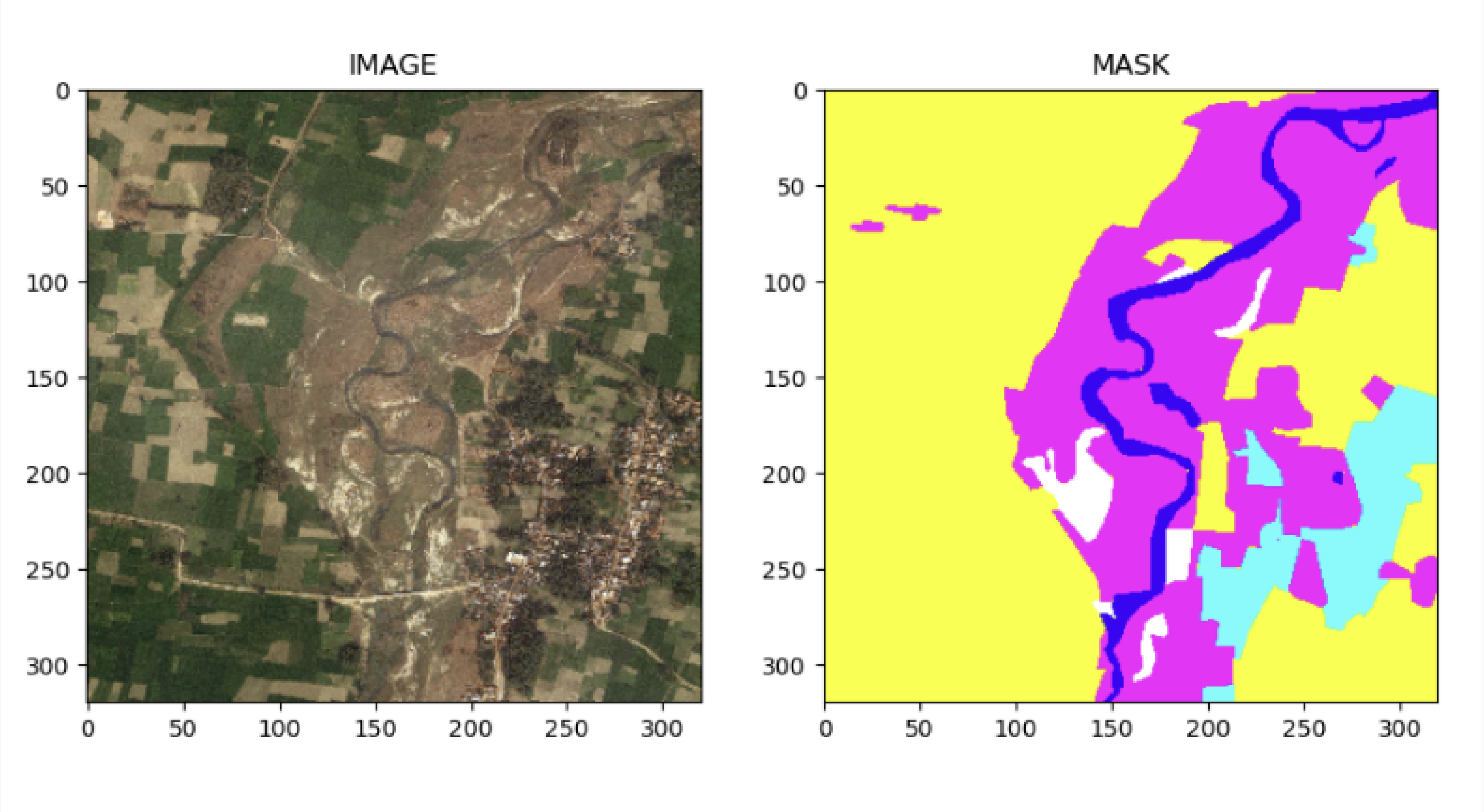
0.348 Sec

DeepLabV3+

0.361 Sec

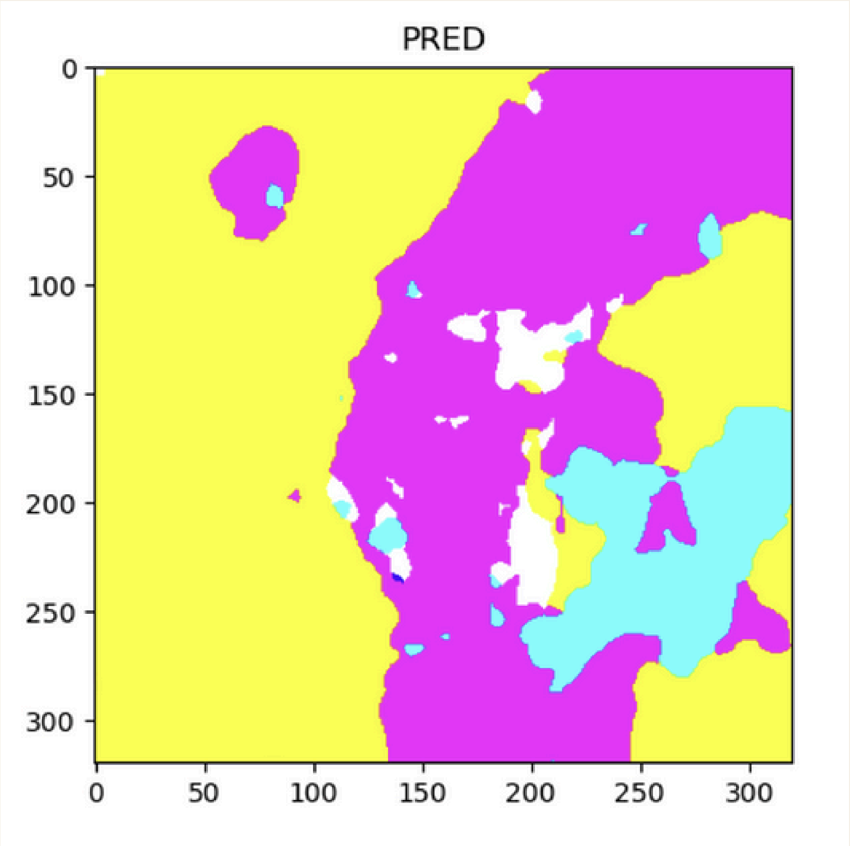
Inference Time for 1 Image on GPU A30

Predicted Result

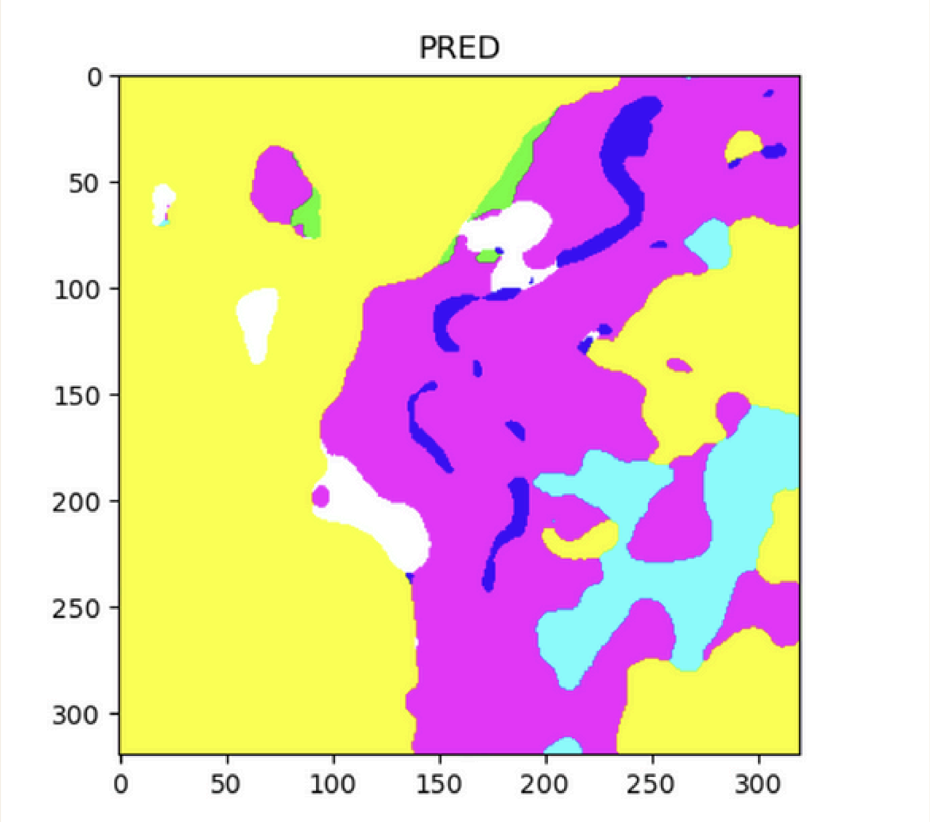


Original Image

Masked Image
(Ground Truth)

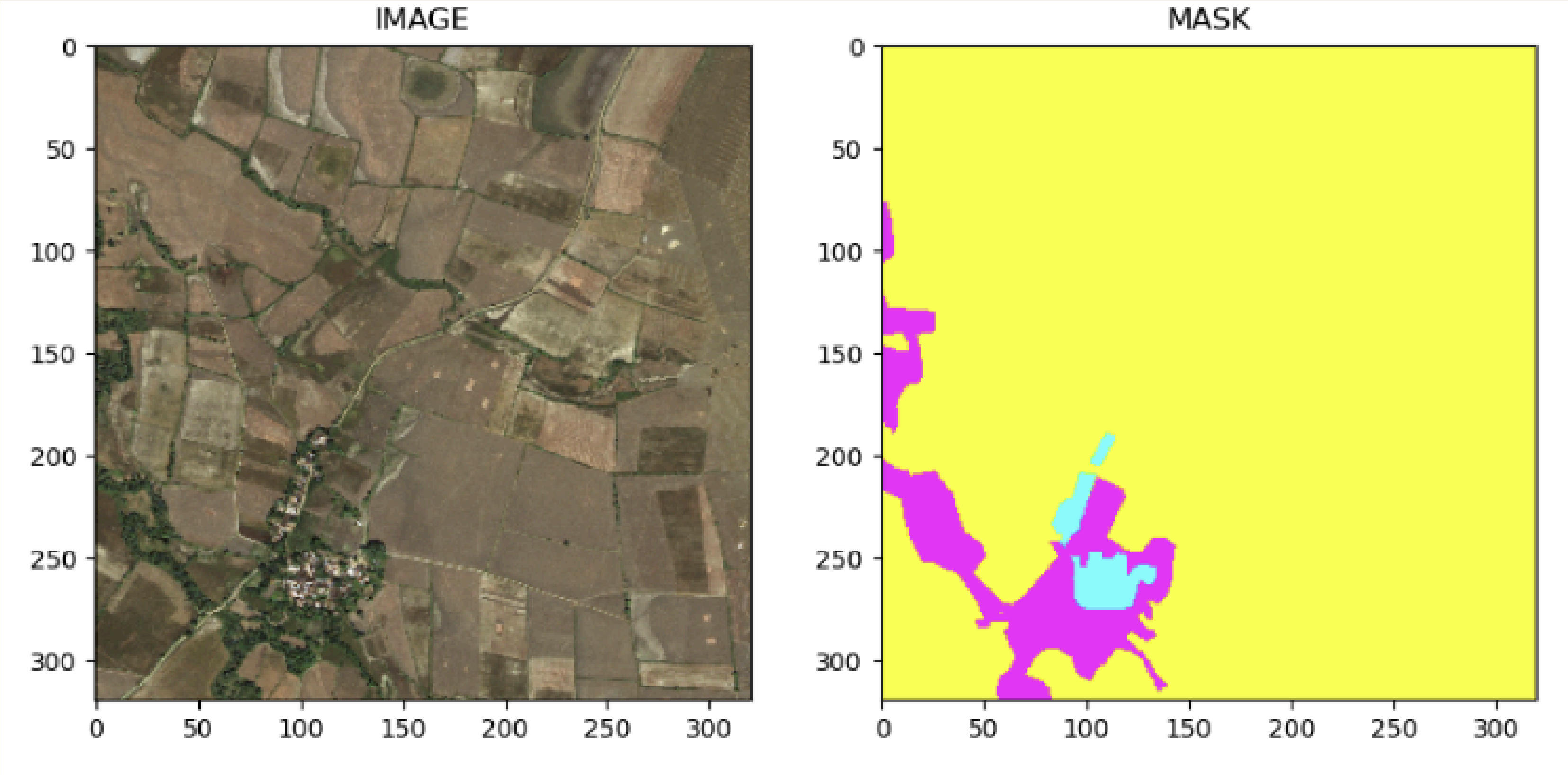


DeeplabV3+



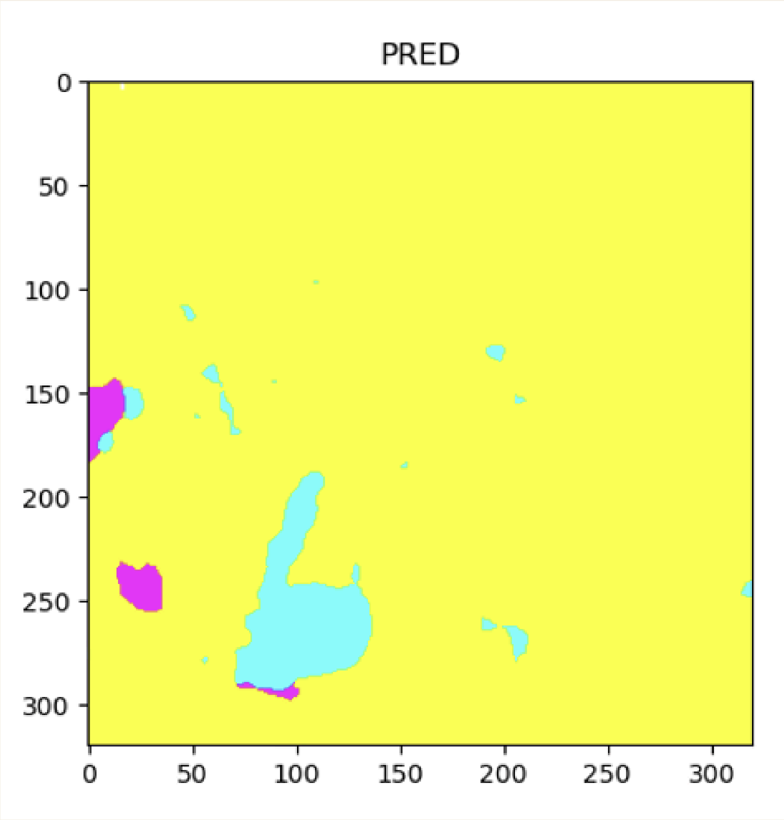
Unet++

Predicted Result

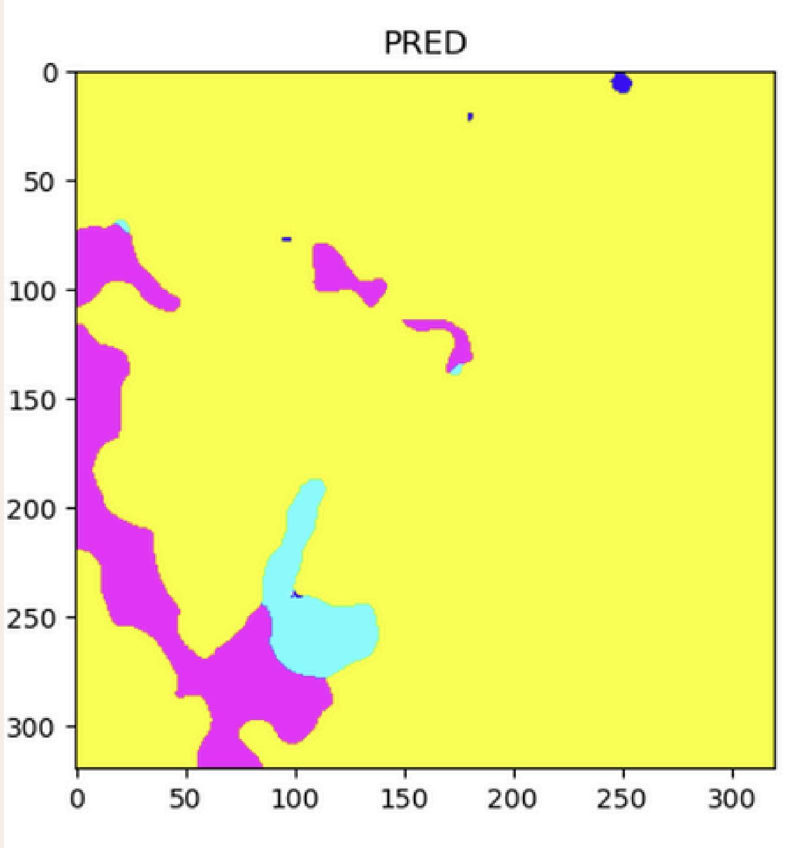


Original Image

Masked Image (Ground Truth)



DeeplabV3+



Unet++

CONCLUSION & FURTHER STUDY

Conclusion

Stratified Sampling

Higher Performance

Best Hyperparameters

LR = 0.0001
BatchSize = 16

Performance

Unet > DeeplabV3

Efficiency

Unet > DeeplabV3

Discussion

Features	DeepLabV3+	U-Net++
Encoder	ResNet-50	RegNetY-120
Architecture	Atrous Convolution Atrous Spatial Pyramid Pooling	Nested Skip Connections Deep Supervision
Strength	Multi Scale Features Complex Spatial Contexts	**Fine Boundary** **Precise Localization**
Application	Road Traffic Image	**Satellite Image** Biomedical Image

Limitation & Further Study

Lack of Models

Try **Different Models** to Compare
e.g. SegNet, CENet

Is it Really Optimal?

Consider **Architecture Hyperparameters**
e.g. Encoder, Layer

Apply to Other Fields

Biomedical Images
Autonomous Driving

References

Research Papers

- Ahmed et al. (2019) : Comparison of Deep-Learning-Based Segmentation Models: Using Top View Person
- Appl. Sci. (2023) : Multiclass Segmentation of Concrete Surface Damages Using U-Net and DeepLabV3+

Dataset

- https://www.kaggle.com/datasets/balraj98/deepglobe-land-cover-classification-dataset/data
- https://medium.com/gumgum-tech/creating-balanced-multi-label-datasets-for-model-training-and-evaluation-16b6a3a2d912

Models

- https://github.com/srsawant34/land_cover_classification/tree/main
- https://www.kaggle.com/code/balraj98/deepglobe-land-cover-classification-deeplabv3
- https://www.kaggle.com/code/nisaneretva/multiclass-segmentation-deepglobe-with-unet

THANK YOU

Designer	*Huiseong Cho*
Director	*Hyongjoon Park*
Modeler	*Jaeyoung Jung*
Dataset Engineer	*Yutak Seo*
Analyst	*Taehyun KIm*
PM	*Daniel Witoslawski*