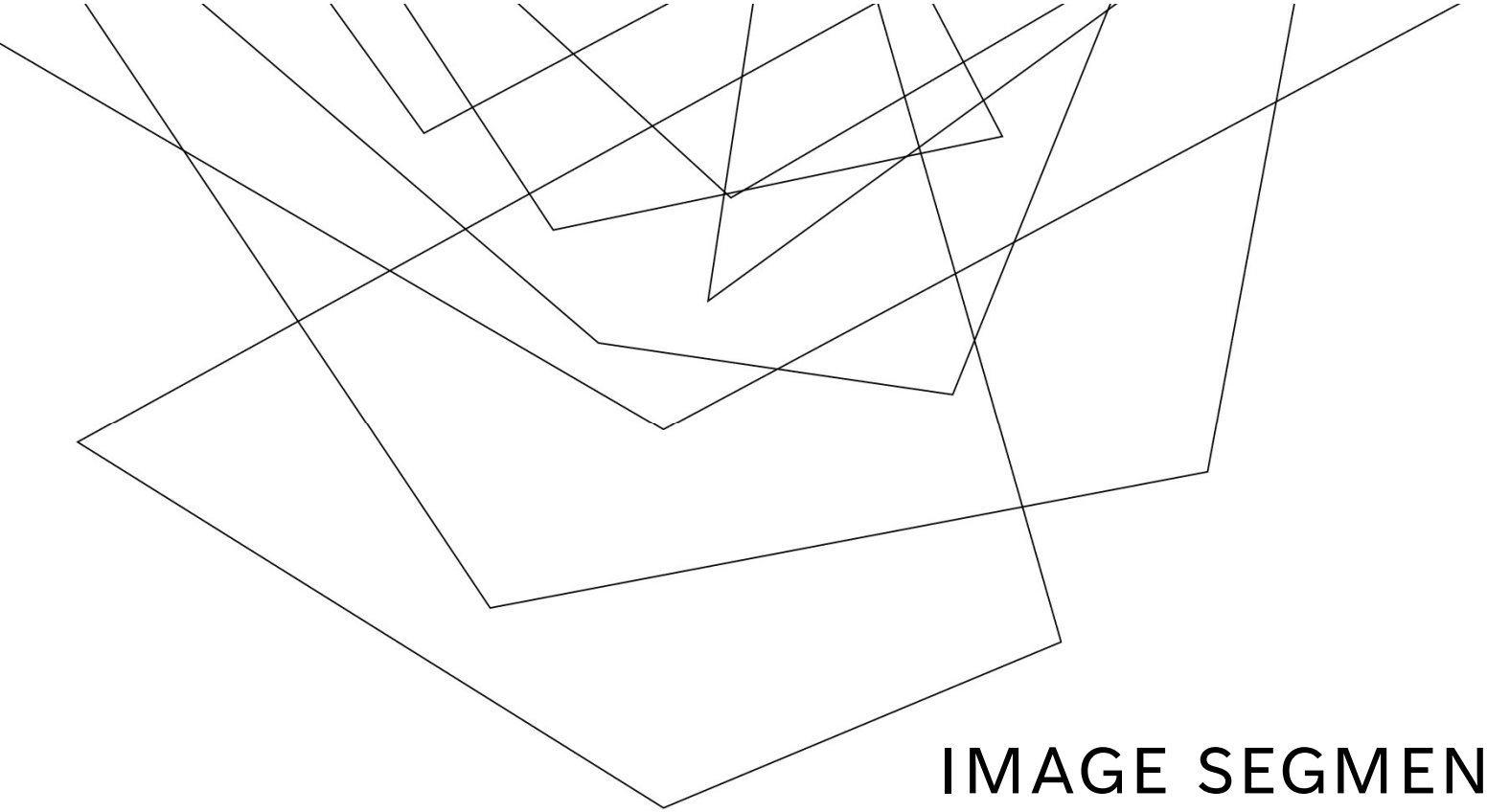


IMAGE SEGMENTATION

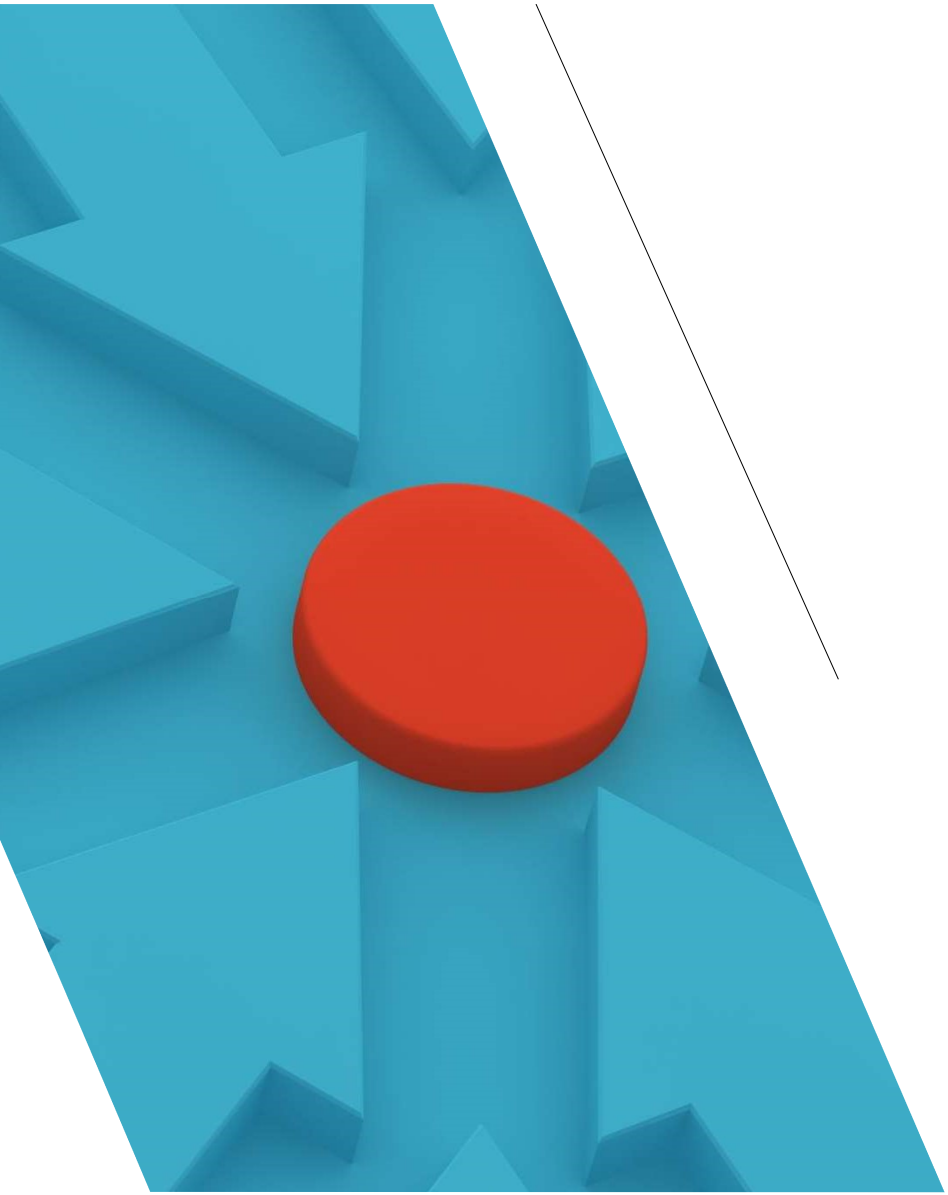
SHEETAL MUNJEWAR
BELLEVUE UNIVERSITY

HTTPS://SMUNJEWAR.GITHUB.IO/PORTFOLIO/

INTRODUCTION AND PROBLEM STATEMENT

What is Image Segmentation?
A technique to divide an image into meaningful regions based on object boundaries.

Why is it important?
Enhances image analysis for various applications such as medical imaging, autonomous driving, and object detection.



APPLICATIONS OF IMAGE SEGMENTATION

Medical Imaging: Tumor detection, organ segmentation.

Autonomous Vehicles: Pedestrian and traffic sign recognition.

Satellite Imaging: Land use classification, environmental monitoring.

Security & Surveillance: Facial recognition, object tracking.

TYPES OF SEGMENTATION

Semantic Segmentation: Labels each pixel based on object class.
Instance Segmentation: Identifies individual object instances.
Panoptic Segmentation: Combines semantic and instance segmentation.

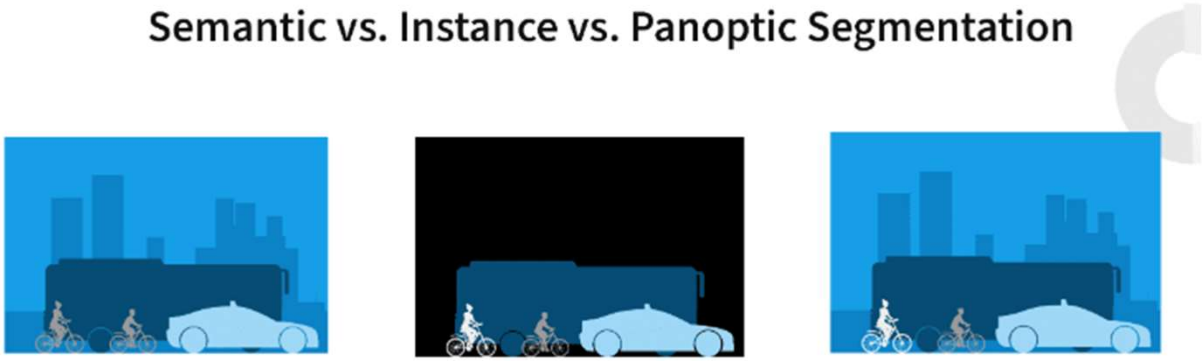


IMAGE SEGMENTATION ARCHITECTURES

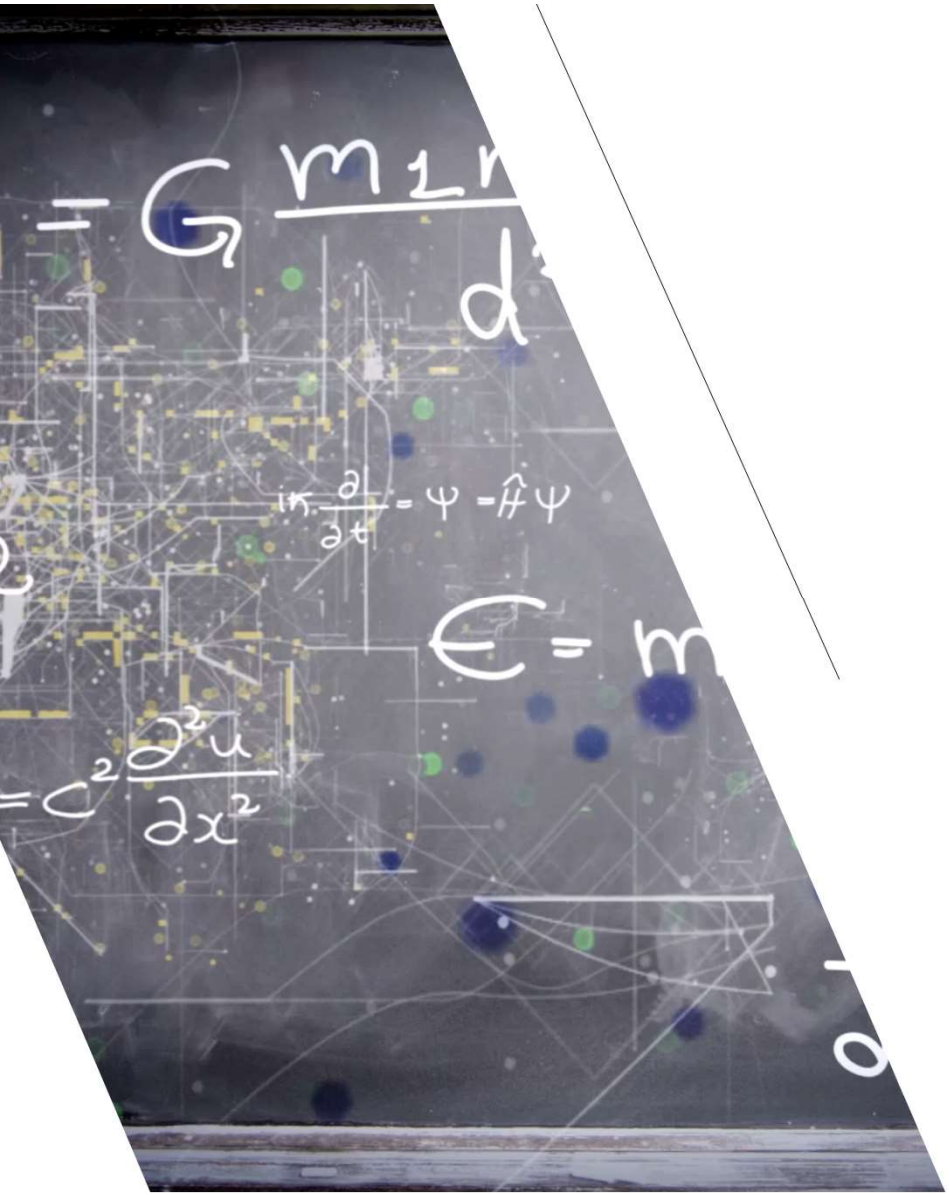
U-Net – Designed for biomedical image segmentation.

FastFCN – Fully connected network optimized for speed.

Gated-SCNN – Incorporates edge detection for improved segmentation.

DeepLab – Uses atrous convolutions for capturing contextual information.

Mask R-CNN – Extends Faster R-CNN for instance segmentation.



DATASET OVERVIEW

Visual Object Classes Challenge 2012 (VOC2012)

Training Images: 1,464

Validation Images: 1,449

Components:

Original Image

Segmentation Mask

DATA UNDERSTANDING

- **Input Images:** RGB images with three channels.
- **Mask Images:** Single-channel grayscale images where each pixel represents an object class.
- **Label Categories:**

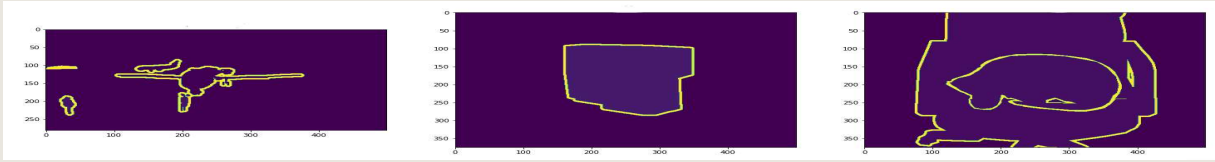
  20 Object Classes (e.g., airplane, bicycle, car, person, etc.)

  0 - Background

  255 - Unknown

Input images

Mask Images

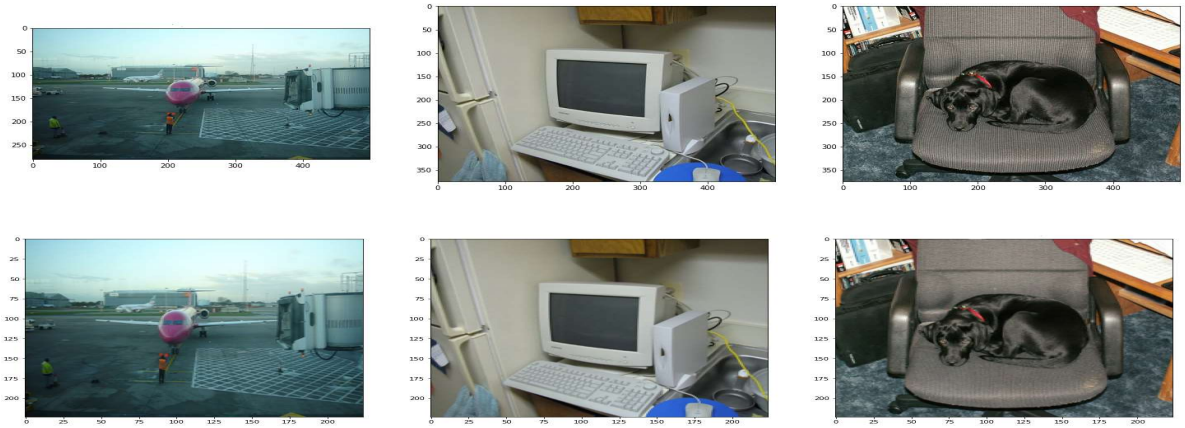


DATA PREPARATION

Image Resizing: Standardized to 256×256 pixels.

Data Augmentation:
Horizontal and vertical flipping.
Enhances dataset diversity and prevents overfitting.

Image Resizing



DATA PREPARATION

Augmentation

DATA PREPARATION

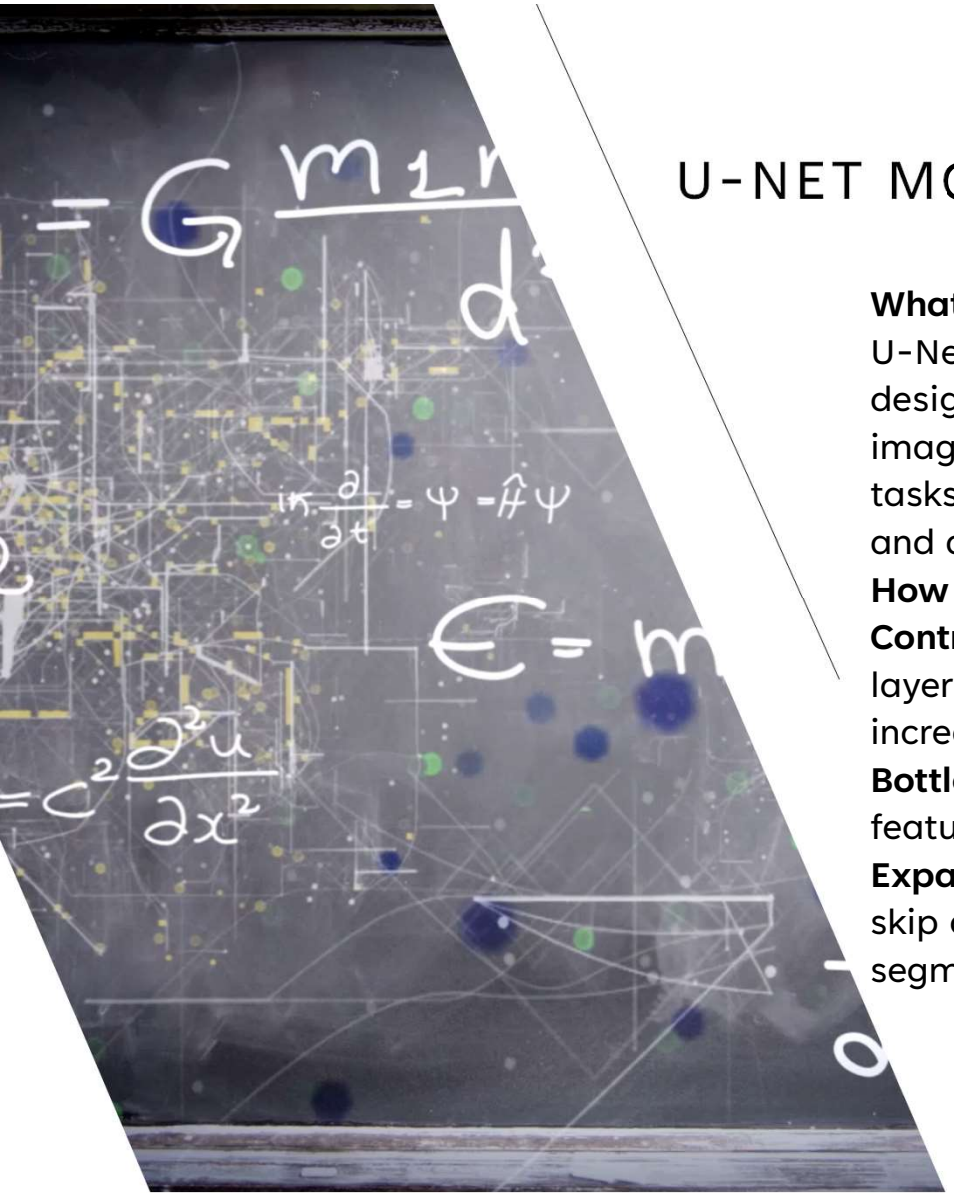
Train, Test, and validation dataset Split

Original Dataset

1464 Training images
1449 Validation images

After Split
2164 Training images
300 Test images

449 Validation images



U-NET MODEL ARCHITECTURE

What is U-Net?
U-Net is a convolutional neural network (CNN) architecture designed for image segmentation, particularly in biomedical image processing. It was introduced in 2015 and is widely used for tasks like medical image analysis, satellite imagery segmentation, and autonomous driving.

How U-Net Works:

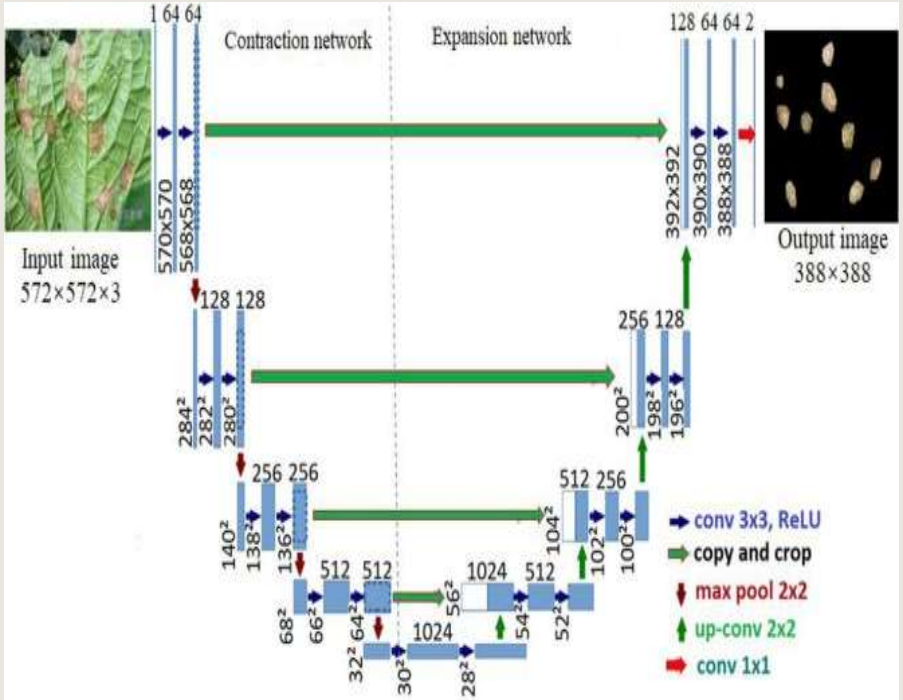
Contracting Path (Encoder) – Extracts features using convolutional layers and max pooling, reducing spatial dimensions while increasing depth.

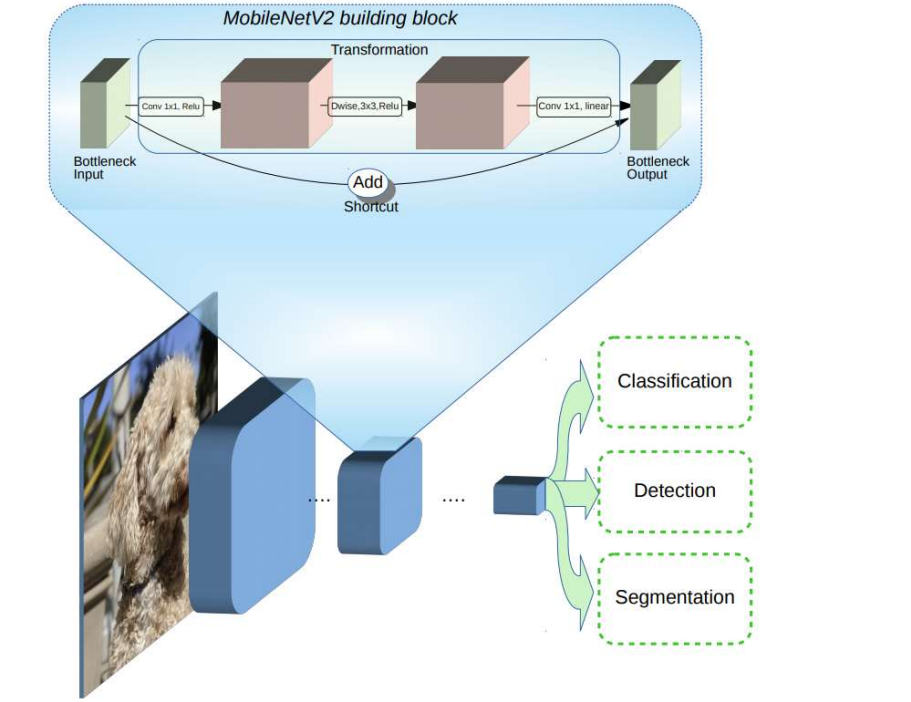
Bottleneck – Connects the encoder and decoder with high-level feature representations.

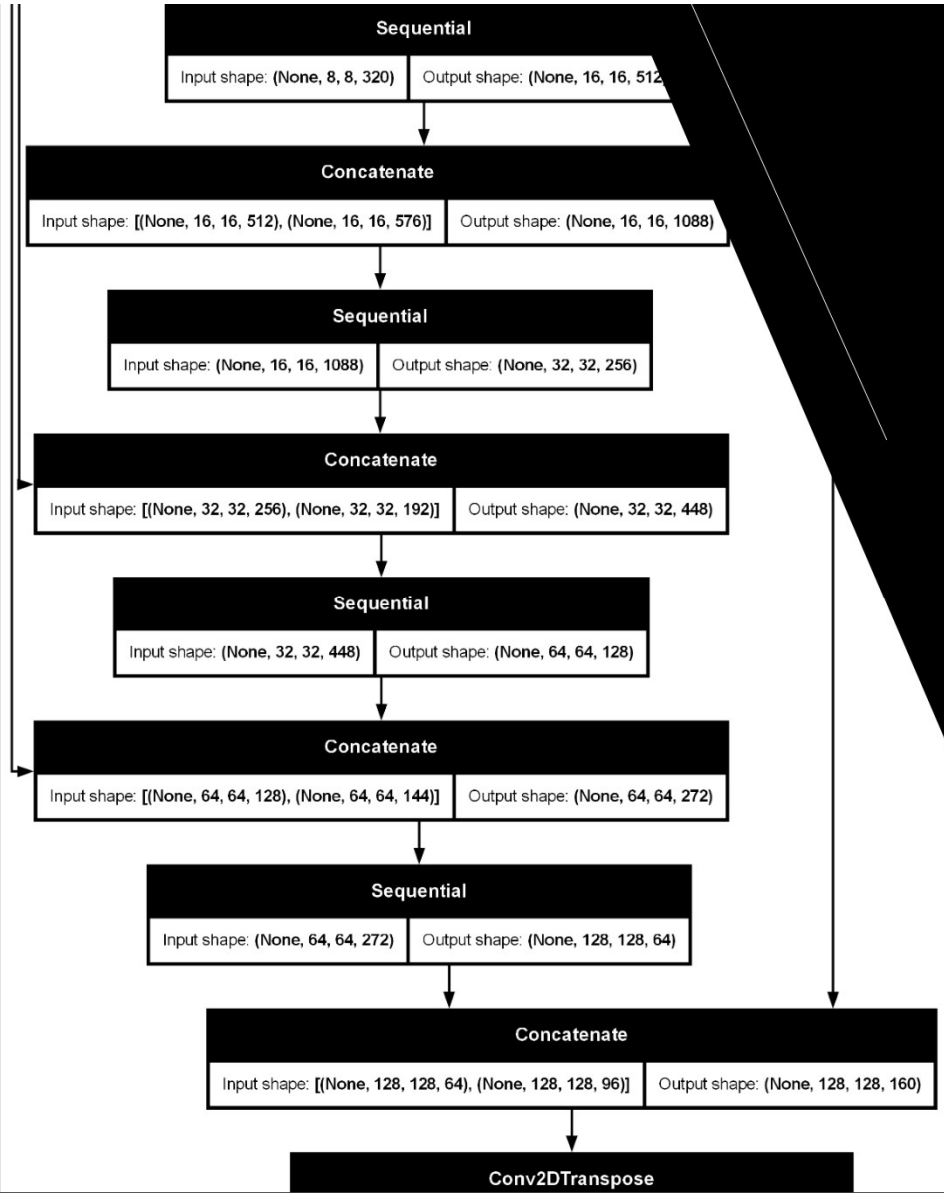
Expanding Path (Decoder) – Uses transposed convolutions and skip connections to restore spatial resolution and refine segmentation.

U-NET MODEL ARCHITECTURE

Encoder: MobileNetV2 – Extracts essential features while maintaining efficiency.
Decoder: Pix2Pix – Enhances the resolution of segmented masks.





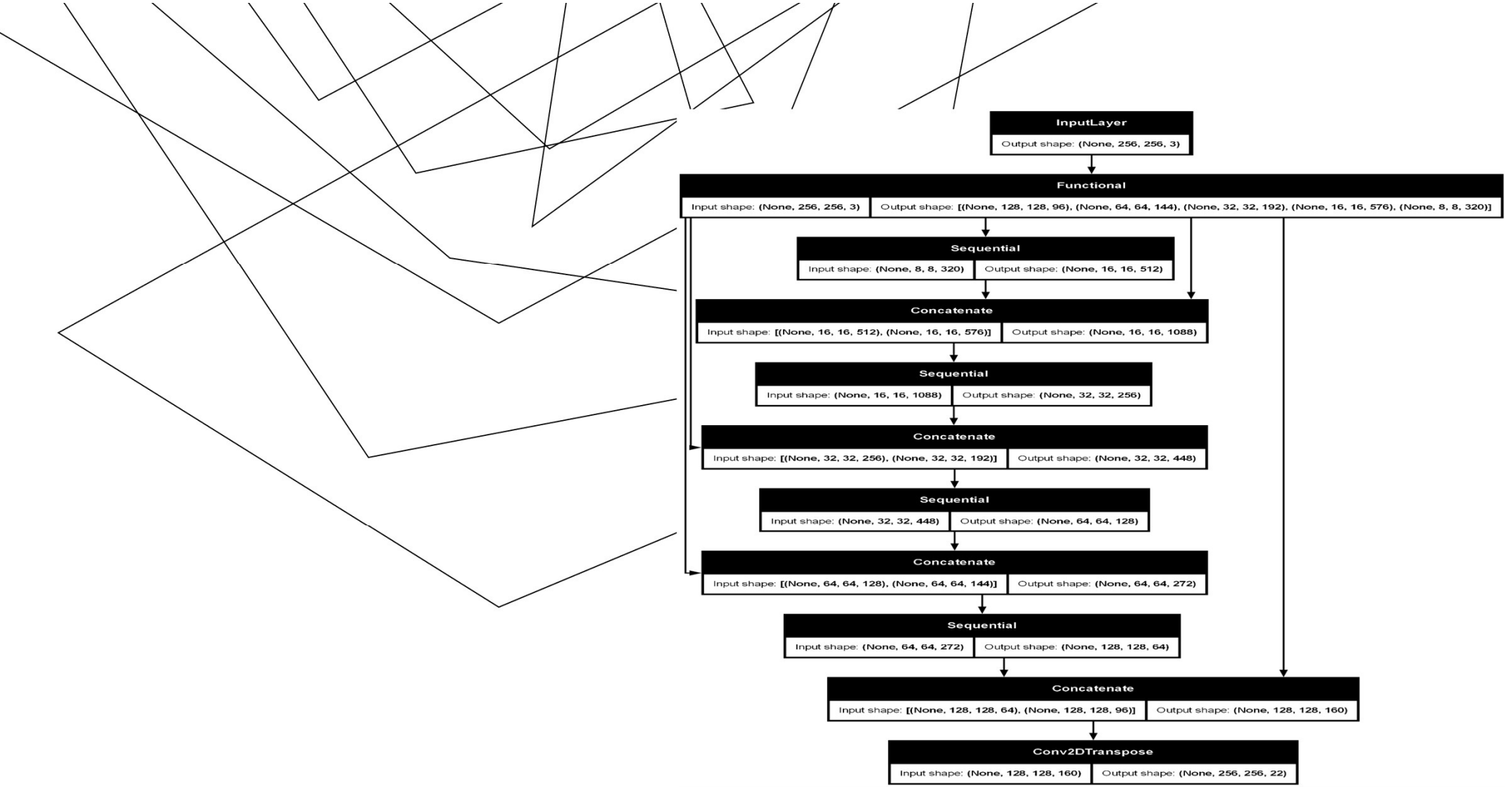


MODEL TRAINING PROCESS

TRAINING PROGRESS:
EXAMPLE FROM **EPOCH 1** TO **EPOCH 11** SHOWING IMPROVED ACCURACY.

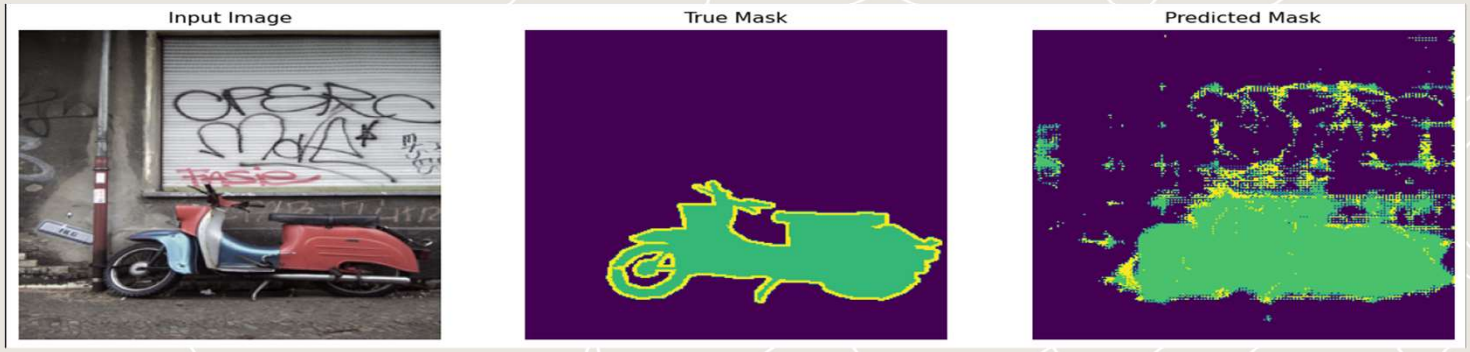
OPTIMIZATION TECHNIQUES:
EARLY STOPPING BASED ON VALIDATION LOSS.

LEARNING RATE ADJUSTMENTS FOR STABLE CONVERGENCE.

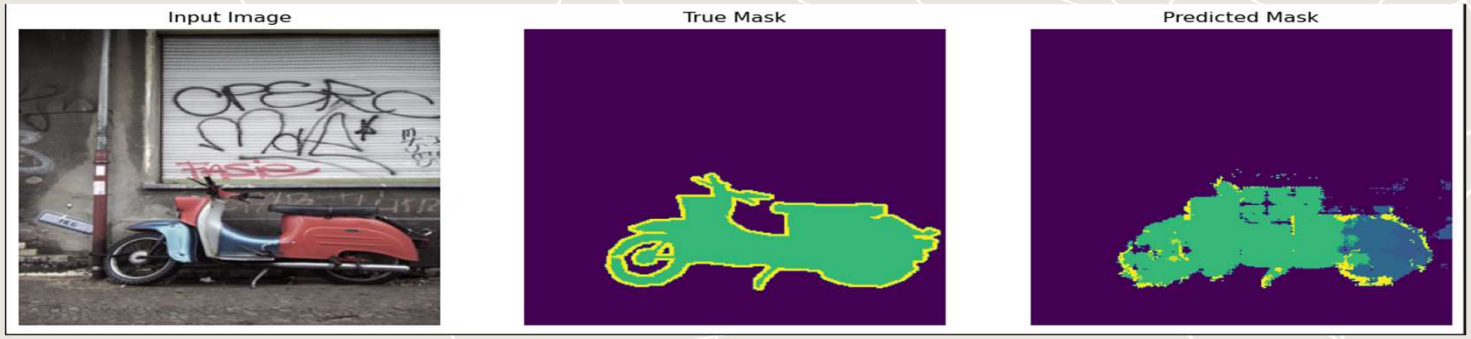

InputLayer
Output shape: (None, 256, 256, 3)
Functional
Input shape: (None, 256, 256, 3)
Output shape: [(None, 128, 128, 96), (None, 64, 64, 144), (None, 32, 32, 192), (None, 16, 16, 576), (None, 8, 8, 320)]
Sequential
Input shape: (None, 8, 8, 320)
Output shape: (None, 16, 16, 512)
Concatenate
Input shape: [(None, 16, 16, 512), (None, 16, 16, 576)]
Output shape: (None, 16, 16, 1088)
Sequential
Input shape: (None, 16, 16, 1088)
Output shape: (None, 32, 32, 256)
Concatenate
Input shape: [(None, 32, 32, 256), (None, 32, 32, 192)]
Output shape: (None, 32, 32, 448)
Sequential
Input shape: (None, 32, 32, 448)
Output shape: (None, 64, 64, 128)
Concatenate
Input shape: [(None, 64, 64, 128), (None, 64, 64, 144)]
Output shape: (None, 64, 64, 272)
Sequential
Input shape: (None, 64, 64, 272)
Output shape: (None, 128, 128, 64)
Concatenate
Input shape: [(None, 128, 128, 64), (None, 128, 128, 96)]
Output shape: (None, 128, 128, 160)
Conv2DTranspose
Input shape: (None, 128, 128, 160)
Output shape: (None, 256, 256, 22)

TRAINING PROGRESS

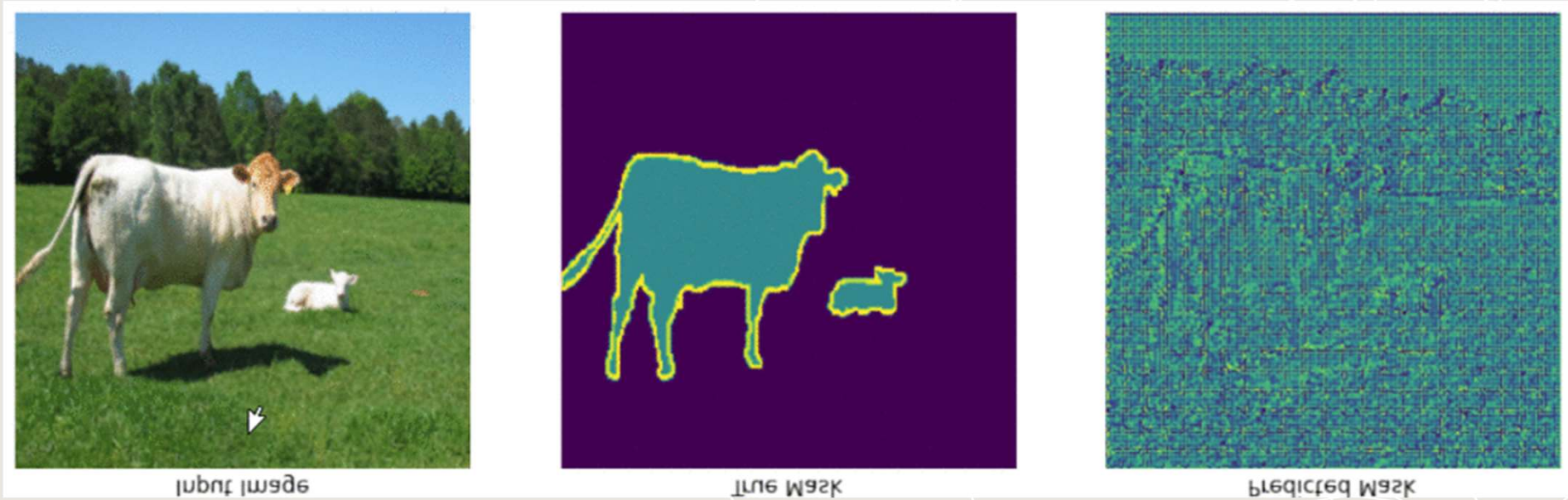
Epoch 1



Epoch 11

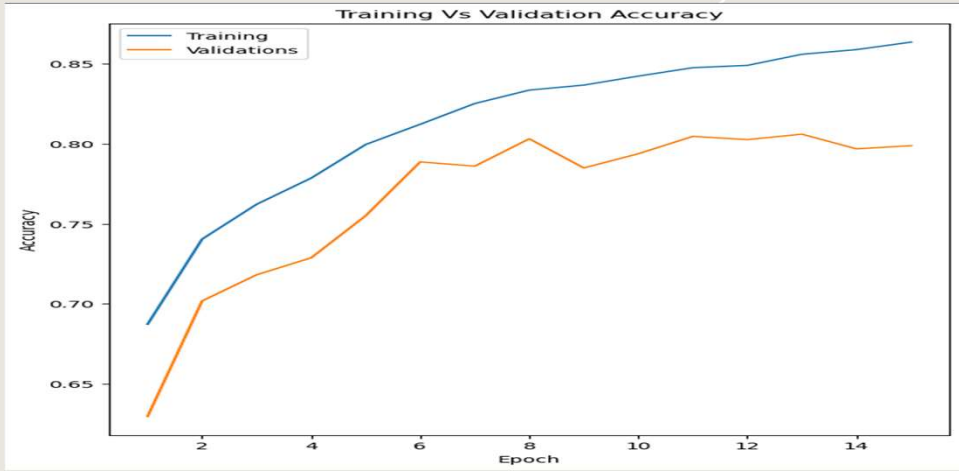


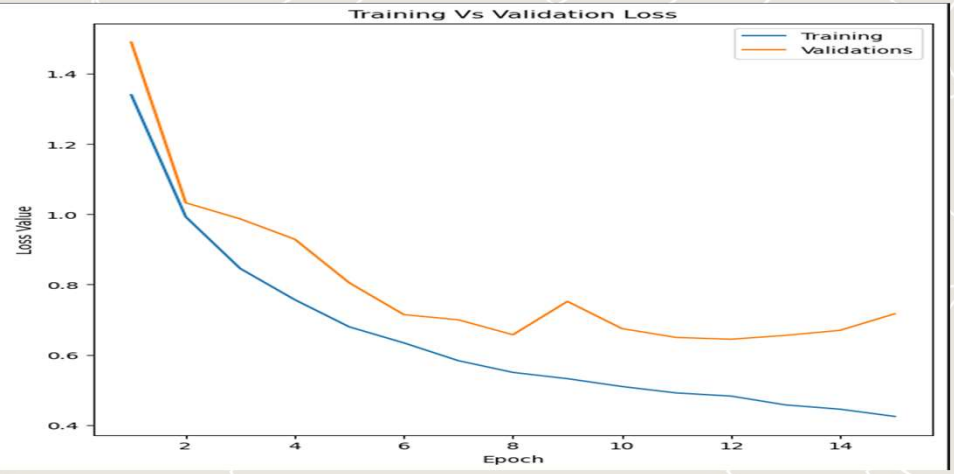
MODEL TRAINING PROGRESS



LEARNING RATE

Performance Metrics:
Training Accuracy: 86%
Validation Accuracy: 79%
Segmentation Predictions:
Examples of model-generated segmentation masks.

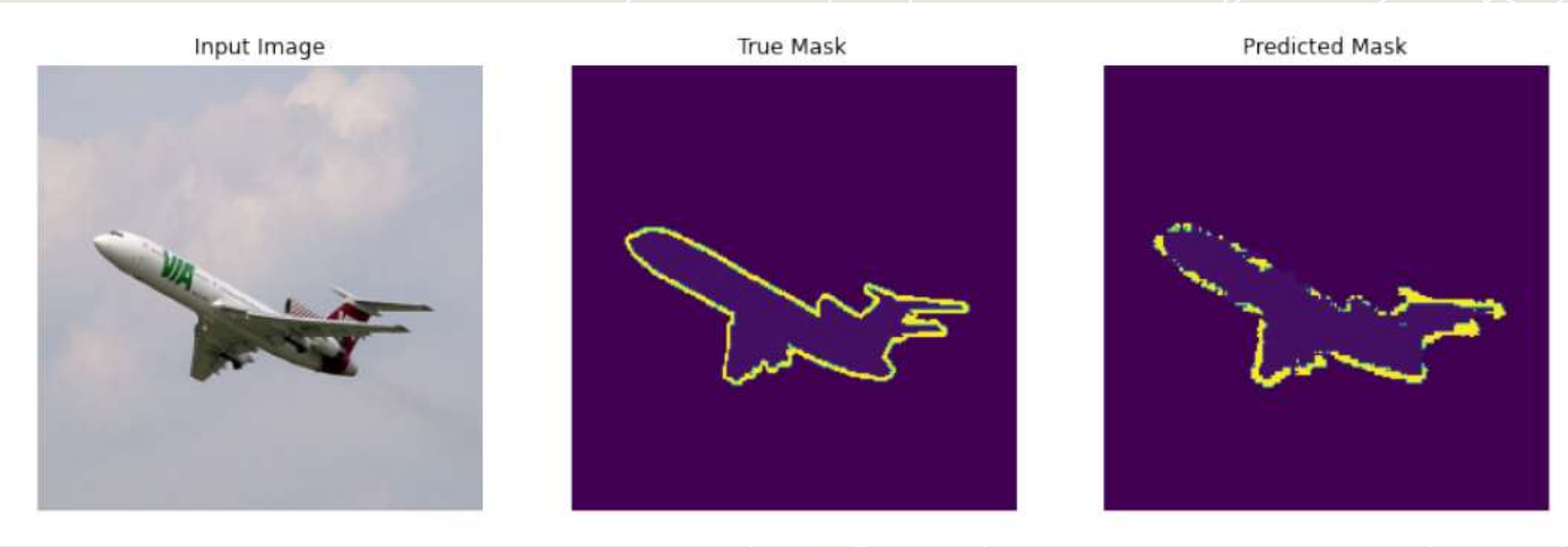




MODEL RESULTS

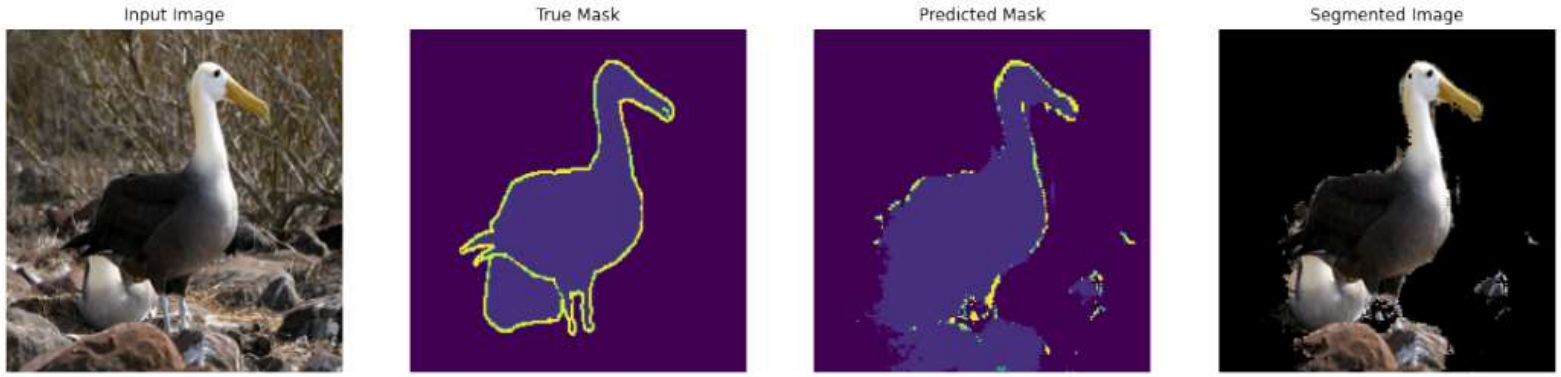
Dataset	Accuracy	Loss
Training	86.53%	0.41
Validation	79.88%	0.71
Test	84.71%	0.51

SAMPLE PREDICTION



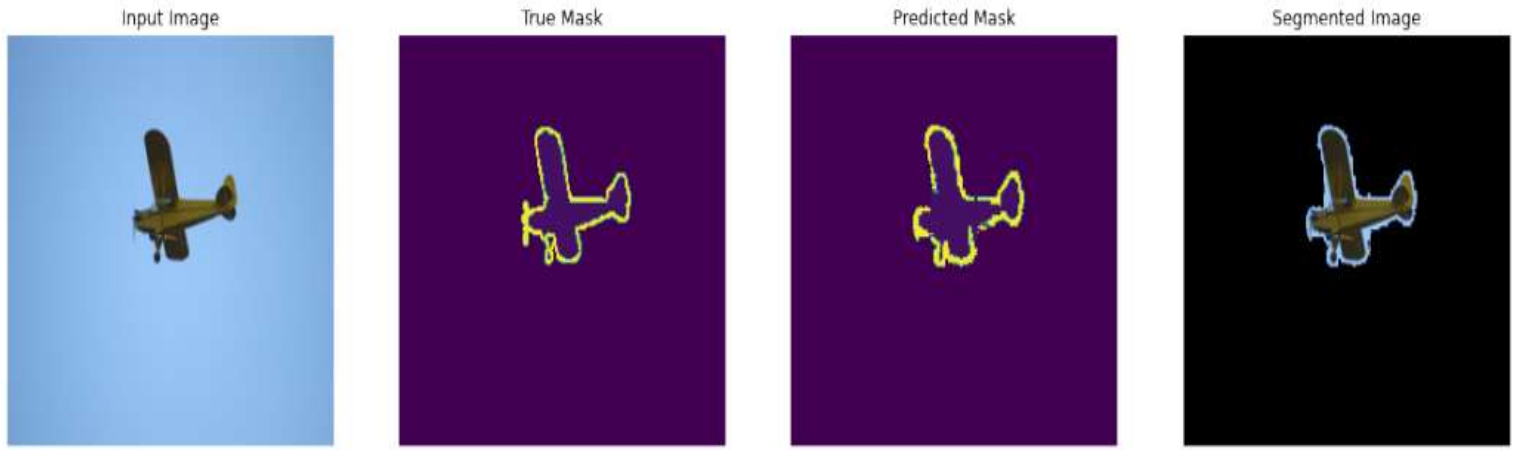
MODEL PREDICTIONS AND SEGMENTATION

```
Classes detected:
0 background
3 bird
13 horse
17 sheep
21 Unknown
```



MODEL PREDICTIONS AND SEGMENTATION

```
Classes detected:
0 background
1 aeroplane
21 Unknown
```



Current Achievements:

Efficient segmentation with U-Net and MobileNetV2.

Applicable to real-world use cases with limited computational resources.

Future Improvements:

Exploring MobileNetV3 for enhanced accuracy.

Expanding the dataset for better generalization.

Optimizing for real-time segmentation applications.

THANK YOU!