



# Self Driving Car

AUGUST 2024

Indonesia AI Project 3

# Table of Content

1

Team  
Member

2

Object  
Segmentation

3

Implementation

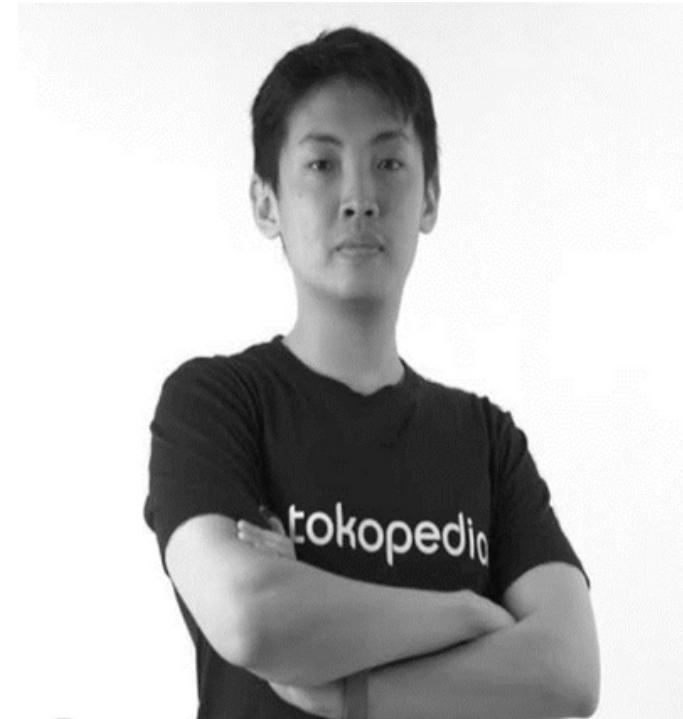
4

Matrices

+



# OUR TEAM



**Cahyadi Hartanto**  
AI Engineer & Leader



**Yosabad T. S.**  
AI Engineer Support

# OBJECT SEGMENTATION



# OBJECT SEGMENTATION

Object segmentation adalah proses mengidentifikasi dan memisahkan objek-objek individual dalam sebuah gambar atau video yang mana setiap piksel dalam gambar ditandai dengan label sesuai dengan objek yang diwakilinya.

- **Semantic Segmentation:** setiap objek dibuat sama
- **Instance Segmentation:** setiap objek dibuat berbeda

## Self Driving Car



Proyek ini bertujuan untuk mengembangkan sistem yang mampu mendekripsi dan mengklasifikasikan berbagai objek di lingkungan sekitar kendaraan sehingga bisa membantu dalam meningkatkan keselamatan dan navigasi kendaraan serta untuk mengambil keputusan yang lebih baik dalam berbagai situasi di jalan.

# IMPLEMENTATION



# IMPLEMENTATION

# Preprocessing

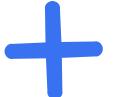
31 Duplicate Data

367 Cleaned Train Image

101 Test Image

```
'0006R0_f03600(1).png' '0016E5_07680(2).png'  
'0006R0_f03630(1).png' '0016E5_07710(1).png'  
'0006R0_f03660(1).png' '0016E5_07710(2).png'  
'0006R0_f03690(1).png' '0016E5_07740(1).png'
```

.....



# IMPLEMENTATION

# Preprocessing

**Make image Grayscale**

```
img_gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
```

**Blur Image**

```
img_blur = cv2.GaussianBlur(img_gray, (3,3), 0)
```

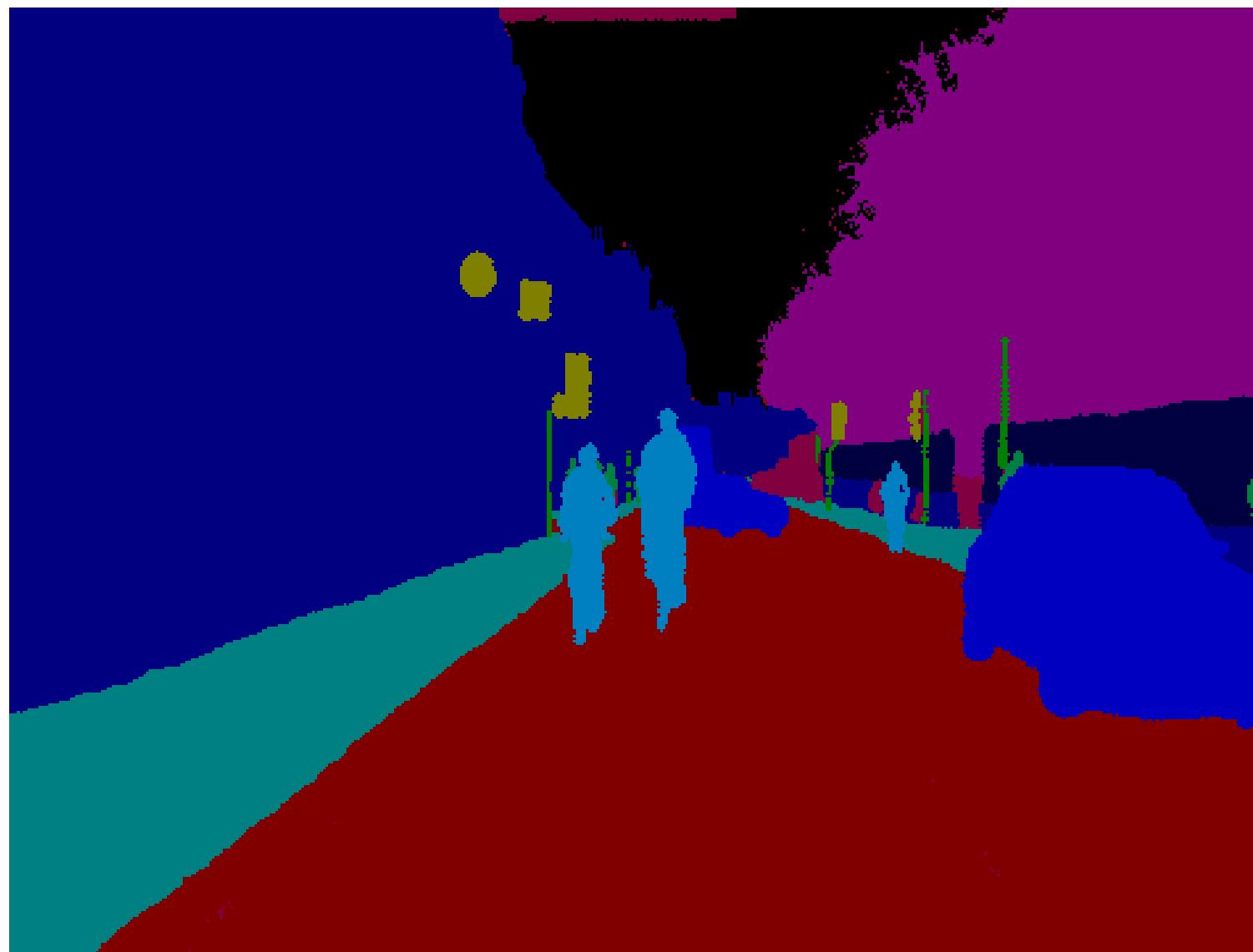
**Process image to show edge only**

```
edges = cv2.Canny(image=img_blur, threshold1=100, threshold2=200)
```



# IMPLEMENTATION

# Preprocessing

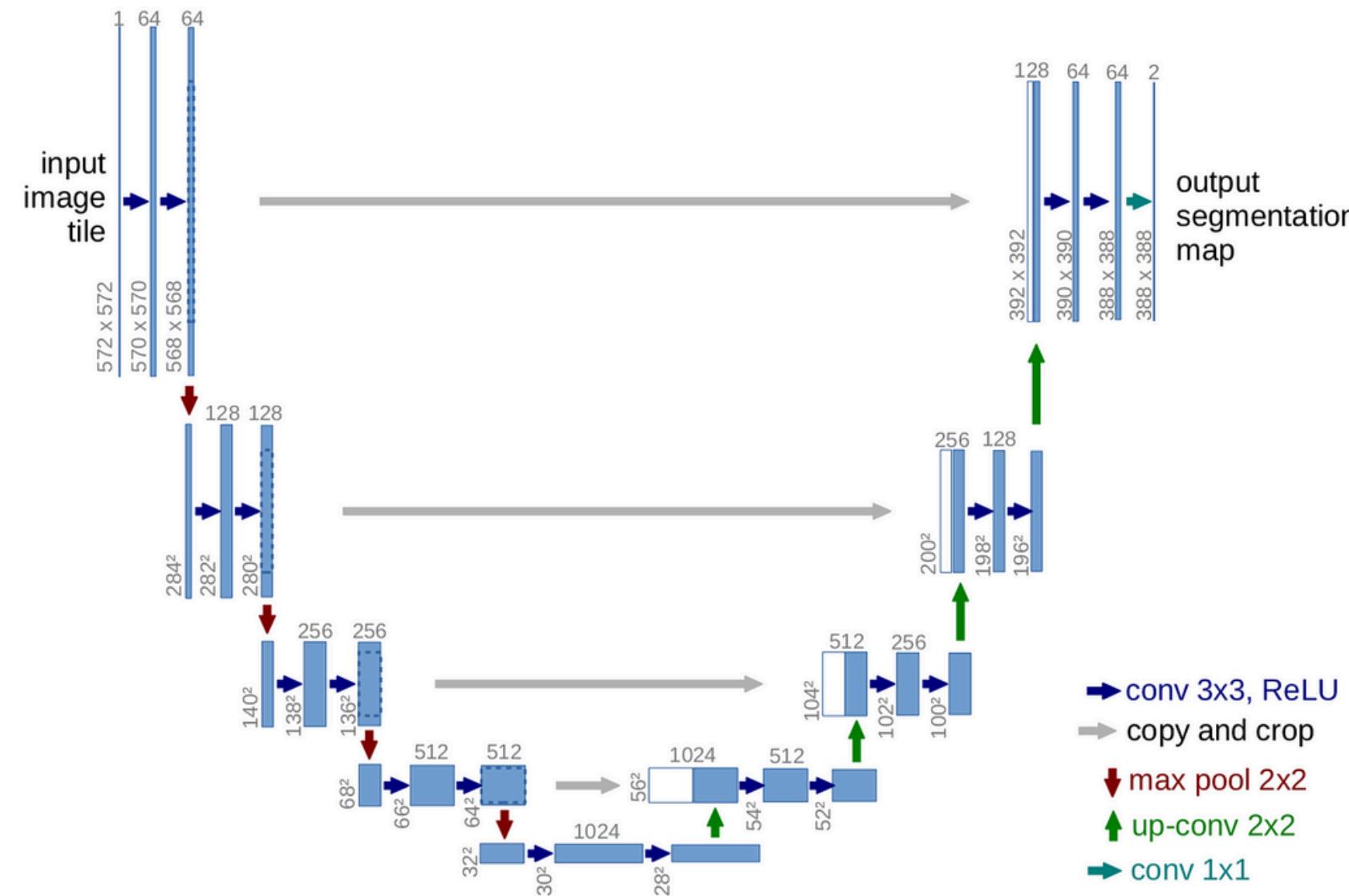


GROUND TRUTH



PREPROCESS

# IMPLEMENTATION Model



**U-Net** adalah arsitektur yang dikembangkan untuk Segmentasi Citra Biomedis pada tahun 2015 di Universitas Freiburg, Jerman. Saat ini, U-Net merupakan salah salah satu pendekatan yang paling populer untuk tugas segmentasi objek.

Komponen arsitektur pada U-Net:

- **Encoder (Contracting Path)**
- **Bottleneck**
- **Decoder (Expansive Path)**



# IMPLEMENTATION

## Model

### VGG U-Net

**VGG U-Net** adalah variasi dari model U-Net yang menggantikan bagian kontraksi dengan arsitektur VGG, yang terkenal karena struktur konvolusinya yang dalam dan kemampuannya untuk mengekstrak fitur.

### Resnet U-Net

**ResNet** dapat digunakan dalam bagian kontraksi untuk mengekstrak fitur dari citra dengan menambahkan koneksi residual sehingga mengurangi masalah *vanishing gradient* dan bisa meningkatkan performa dalam pengenalan fitur.

### YOLO v8 Segmentation

**YOLO Segmentation** adalah pengembangan dari framework YOLO yang bertujuan untuk menghasilkan peta segmentasi dari sebuah object melalui metode *masking*.



## IMPLEMENTATION

# Color Classification

### YOLO v8 Object Classification

**YOLO Object Classification** adalah pengembangan dari framework YOLO yang bertujuan untuk mengklasifikasikan suatu gambar. Dalam hal ini digunakan untuk **mendeteksi Warna lampu lalu lintas (Merah, Hijau, Kuning)**.



# MATRICES



# IMPLEMENTATION

# RESNET50 UNET

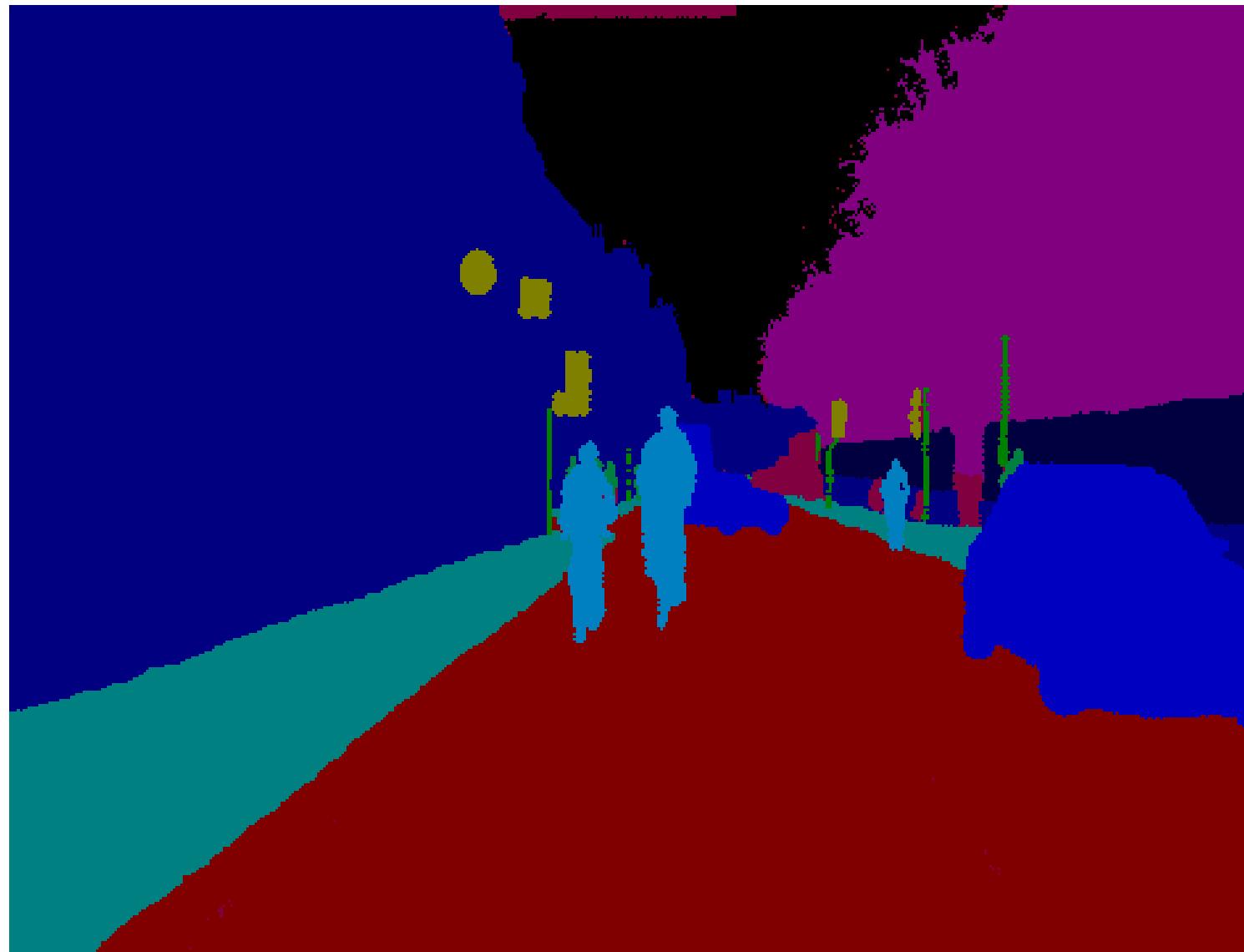
**EPOCH: 5 | n\_classes=50**  
**height=320**  
**width=640**

```
512/512 [=====] - 43s 73ms/step - loss: 3.8597 - accuracy: 0.1408
512/512 [=====] - 39s 75ms/step - loss: 3.1928 - accuracy: 0.4229
512/512 [=====] - 36s 71ms/step - loss: 2.7541 - accuracy: 0.4796
512/512 [=====] - 36s 71ms/step - loss: 2.5042 - accuracy: 0.5074
512/512 [=====] - 36s 71ms/step - loss: 2.3562 - accuracy: 0.5242
```

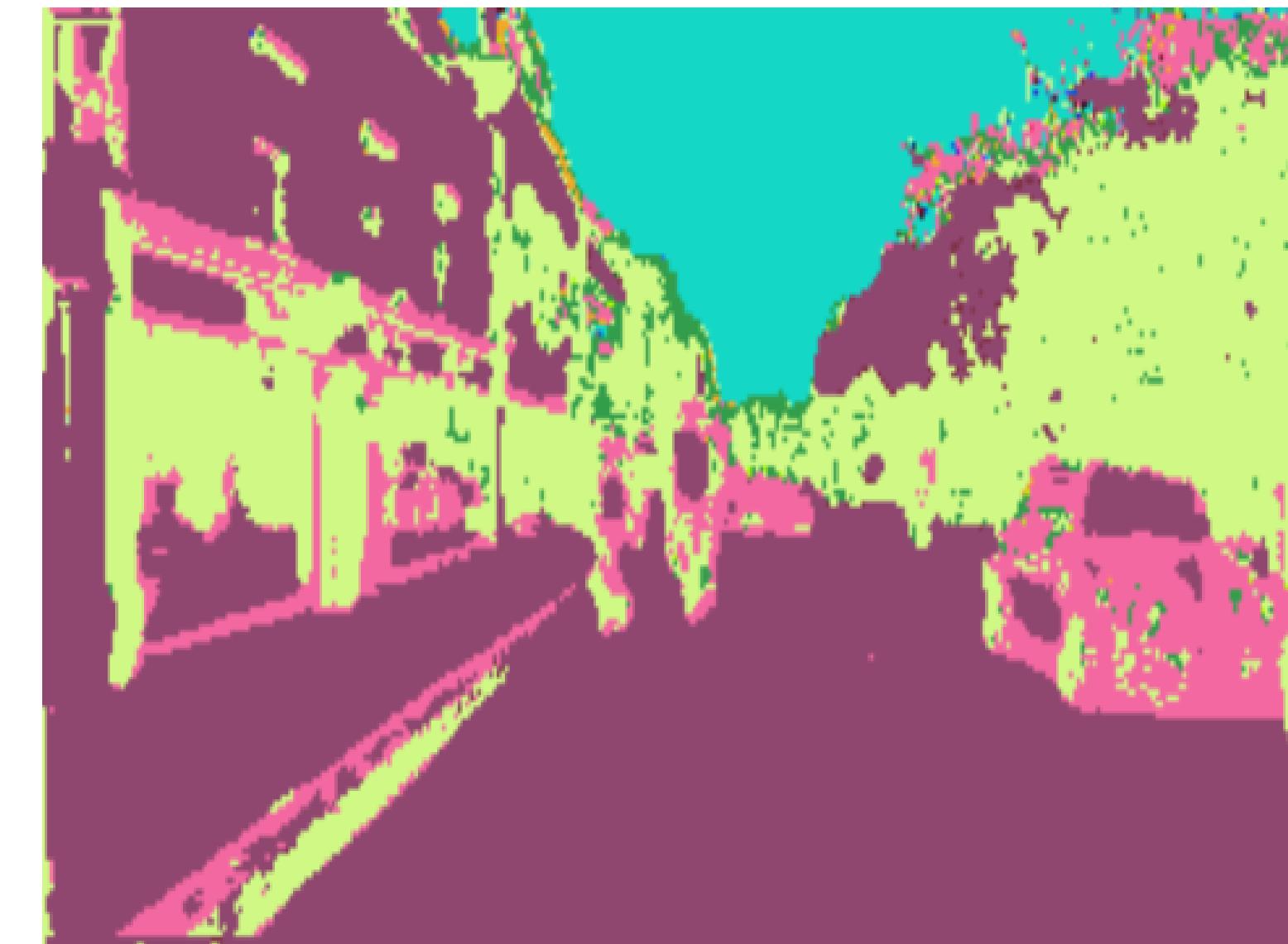
**Inferences : 71ms**  
**Loss : 2.3562**  
**Accuracy : 52.42%**



# IMPLEMENTATION RESNET50 UNET



GROUND TRUTH



HASIL

# IMPLEMENTATION

# VGG UNET

**EPOCH: 5 | n\_classes=50**  
**height=320**  
**width=640**

```
512/512 [=====] - 41s 76ms/step - loss: 3.5823 - accuracy: 0.2825  
512/512 [=====] - 38s 74ms/step - loss: 3.0316 - accuracy: 0.4244  
512/512 [=====] - 36s 71ms/step - loss: 2.6780 - accuracy: 0.4708  
512/512 [=====] - 37s 72ms/step - loss: 2.4740 - accuracy: 0.4976  
512/512 [=====] - 36s 71ms/step - loss: 2.3337 - accuracy: 0.5189
```

**Inferences : 71~76ms**

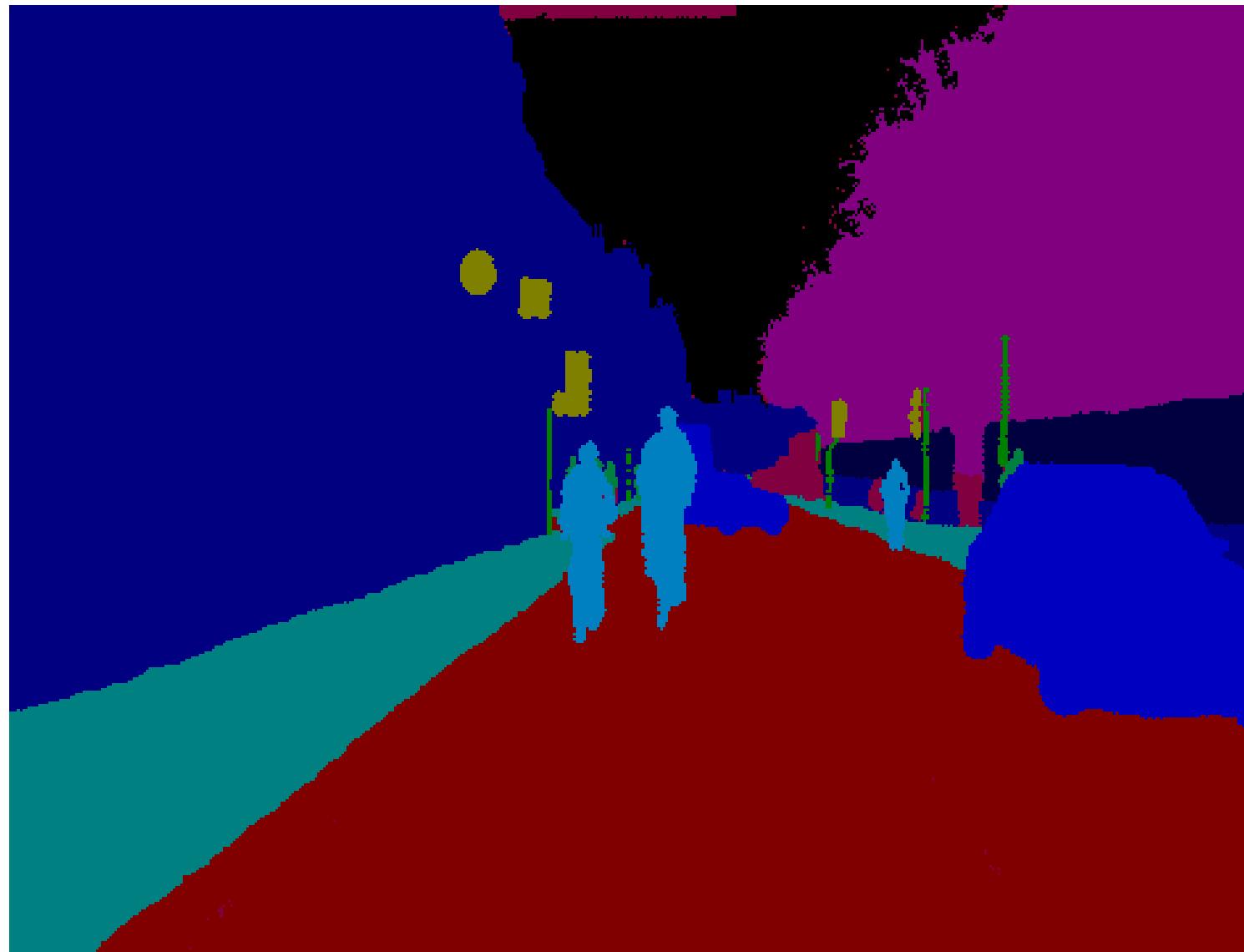
**Loss : 2.3337**

**Accuracy : 51.89%**

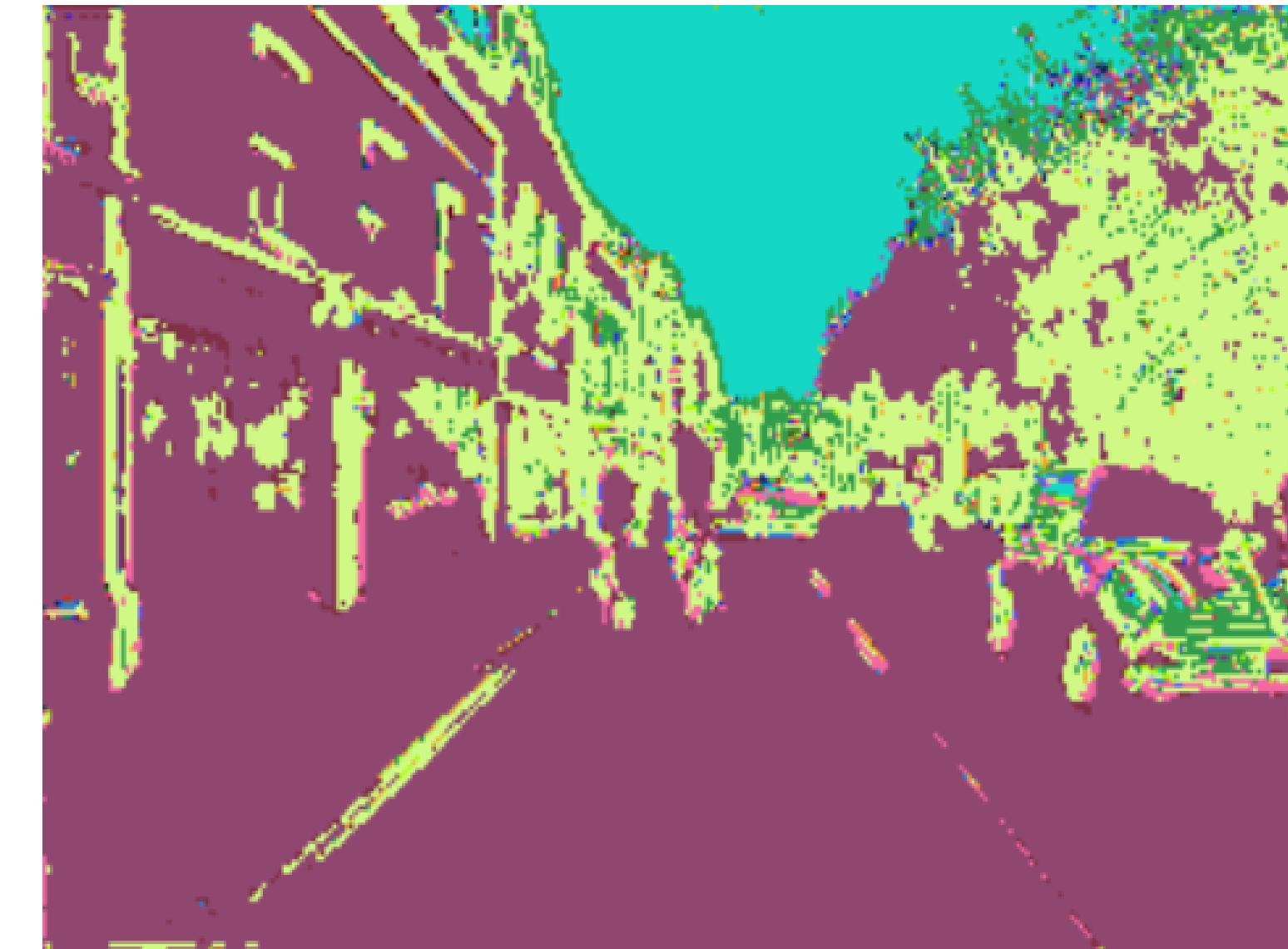


# IMPLEMENTATION

# VGG UNET



GROUND TRUTH

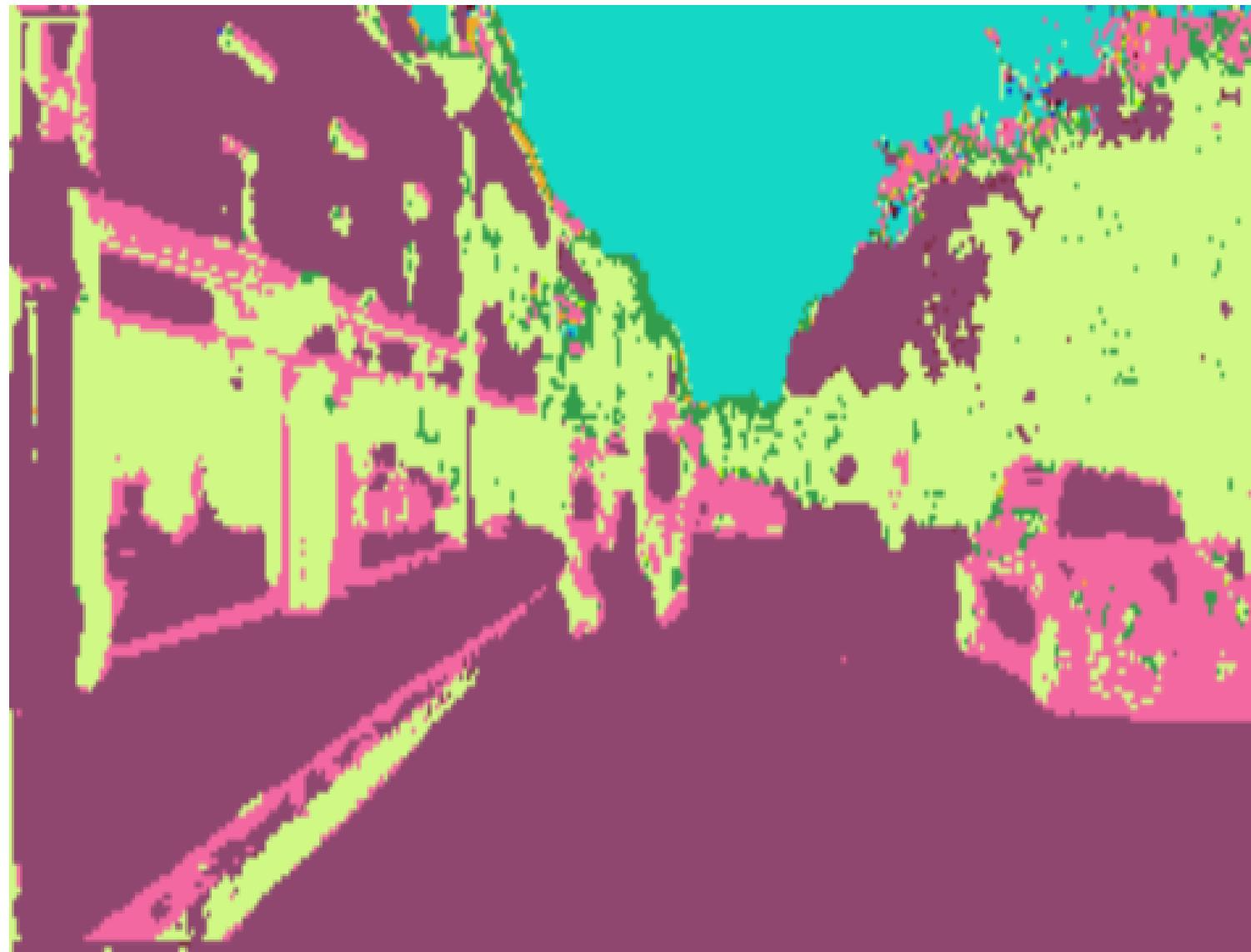


HASIL

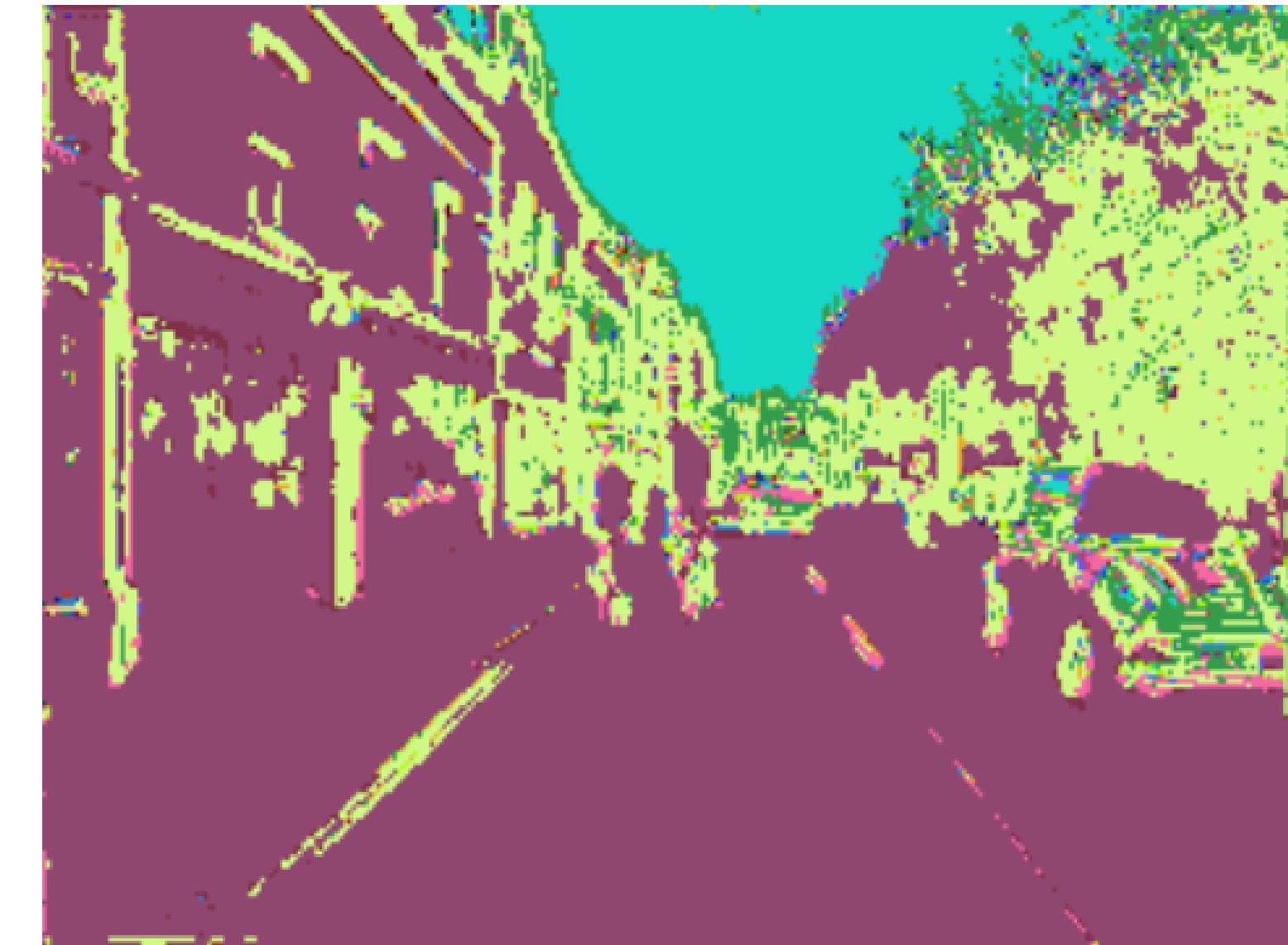


# IMPLEMENTATION

## VGG VS RESNET



RESNET



VGG



## IMPLEMENTATION

# YOLOv8 Segmentation + Classification

**Inferences : 7.71 ms**

**IMGsize: 640**

**MAP50: 94.6%**

Epoch	GPU_mem	box_loss	cls_loss	dfl_loss	Instances	Size
1/5	11.6G	1.567	1.649	1.704	1	640:
	Class	Images	Instances	Box(P	R	mAP50 mAP50-95):
	all	200	272	0.944	0.833	0.917 0.368

Epoch	GPU_mem	box_loss	cls_loss	dfl_loss	Instances	Size
5/5	11.3G	1.315	0.9893	1.502	4	640:
	Class	Images	Instances	Box(P	R	mAP50 mAP50-95):
	all	200	272	0.927	0.854	0.946 0.46



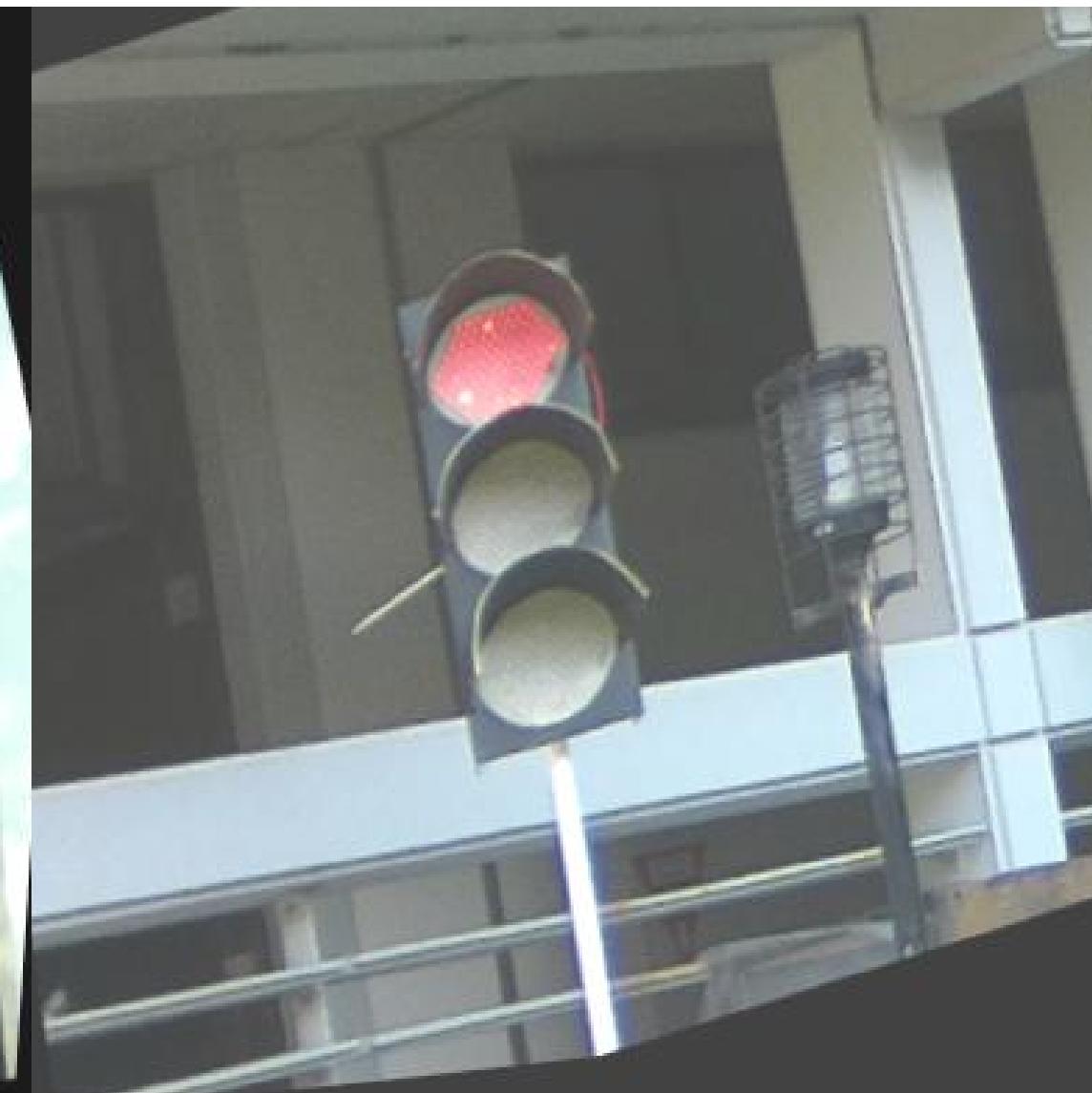
IMPLEMENTATION

# YOLOv8 Classification

Dataset : **cinTA\_v2-1**

Train: **2097**

Test: **100**



IMPLEMENTATION

# YOLOv8 Classification

Dataset : **cinTA\_v2-1**

Train: 2097

Test: 100



# IMPLEMENTATION YOLOv8



Base img



HASIL

# DEMO



# Thank You

