Project CV 3: Self-Driving Car

Kelompok CV A & C

- Amar Ma'ruf
- Bayuzen Ahmad
- Benedictus Dikha Arianda
- Dika Mahendra
- Elsa Nurul Hidayah
- Fariz Rachman Hadi
- Fatah Abdul Jalil
- Haris Raharjo Putro
- Kio Sato
- M. Andhika Dwiki Nugraha
- Mahdia Aliyya Nuha Kiswanto
- Michael Andrian
- Moch Ardhi Kurniawan
- Muhammad Bramansyah
- Muhammad Fatih Zuhdy

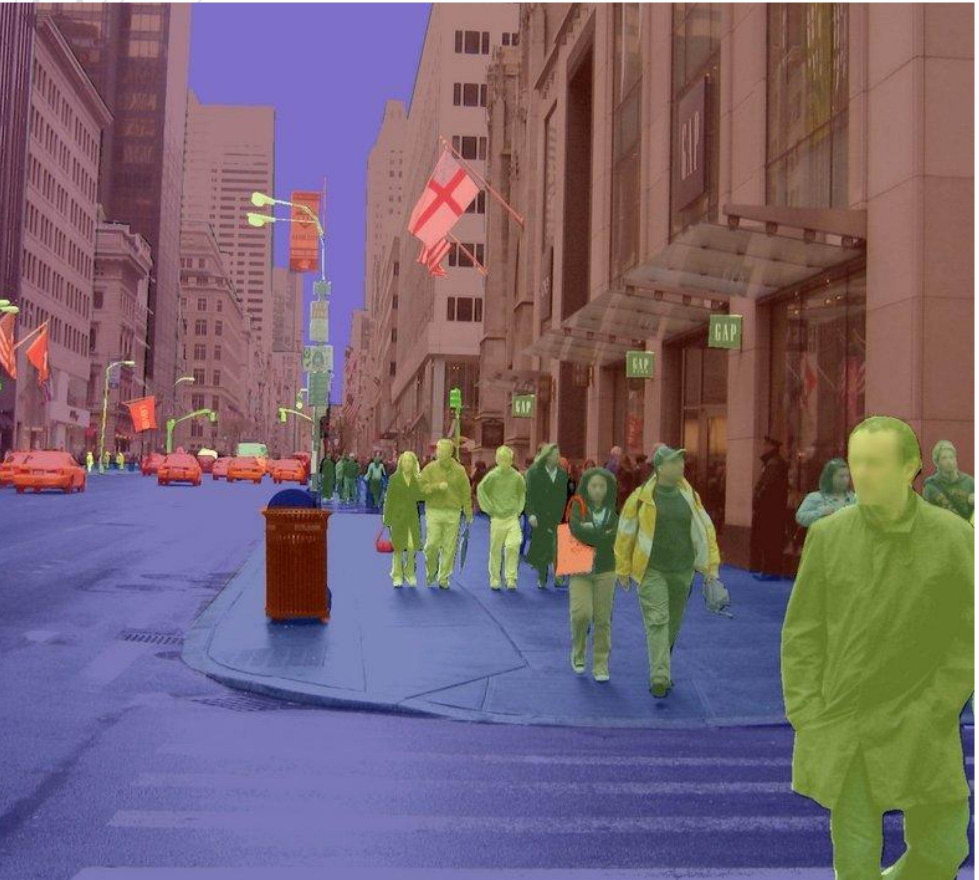
Background: Self-Driving Car

One of the big problems in urban areas is traffic congestion which can cause accidents. One of the factors that played a big role in this incident was the driver's negligence.

According to WHO, the traffic system is one of the most complex and dangerous systems that we face almost every day.

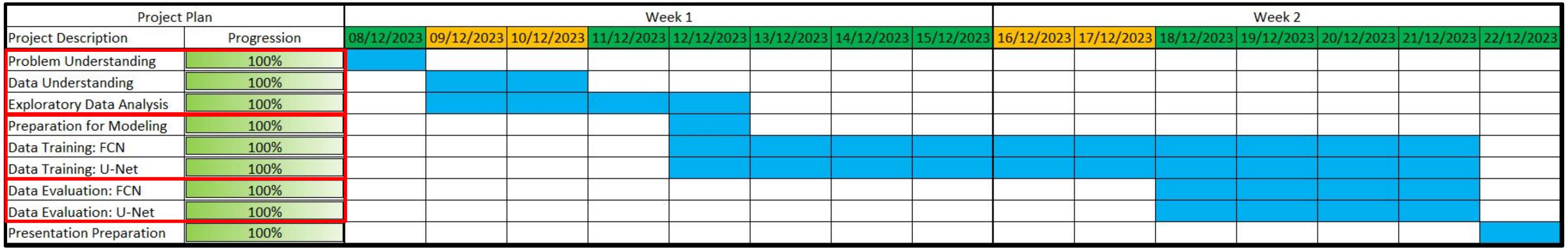
For this reason, it is hoped that the existence of a self-driving car using AI technology can be a solution to improve driving safety.

Objectives



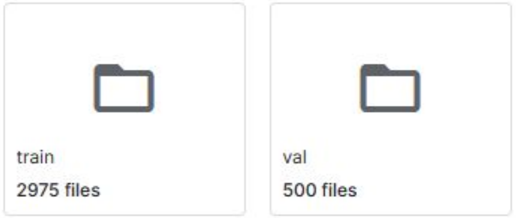
1. Utilizes AI technology such as object segmentation to identify and avoid objects in the streets so as to avoid traffic congestion and accidents.
2. Implement FCN8s and U-Net models for object segmentation.

Timeline



Project Plan		Week 1								Week 2						
Project Description	Progression	08/12/2023	09/12/2023	10/12/2023	11/12/2023	12/12/2023	13/12/2023	14/12/2023	15/12/2023	16/12/2023	17/12/2023	18/12/2023	19/12/2023	20/12/2023	21/12/2023	22/12/2023
Problem Understanding	100%	🟦														
Data Understanding	100%		🟦	🟦												
Exploratory Data Analysis	100%		🟦	🟦	🟦	🟦										
Preparation for Modeling	100%					🟦										
Data Training: FCN	100%					🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	
Data Training: U-Net	100%					🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	
Data Evaluation: FCN	100%											🟦	🟦	🟦	🟦	
Data Evaluation: U-Net	100%											🟦	🟦	🟦	🟦	
Presentation Preparation	100%															🟦

Dataset: CITYSCAPES DATASET


train
2975 files

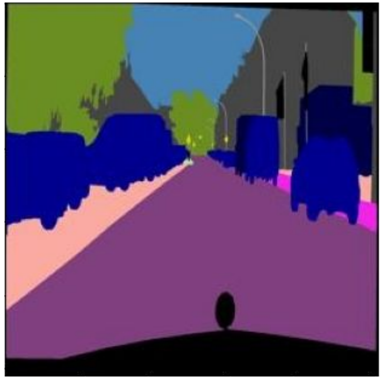
val
500 files

Source Data:

- ➔ Link1 (Cityscapes Dataset - Original)
- ➔ Link2 (Cityscapes Image Pairs - Kaggle)
- ➔ Link3 (Cityscapes Dataset - Dashboard)

Image

Segmented Image



Segmented Image (Grey Scale)

FCN8s Architecture

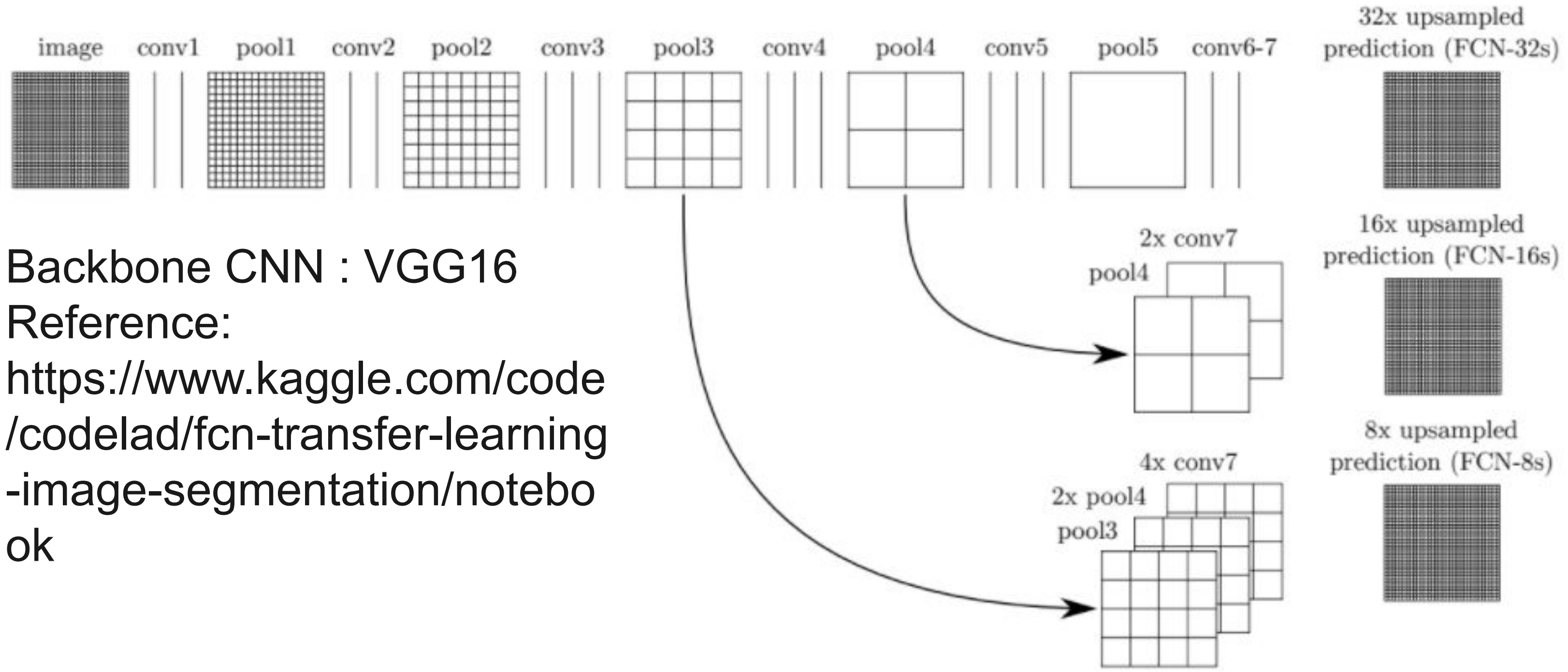


image
conv1
pool1
conv2
pool2
conv3
pool3
conv4
pool4
conv5
pool5
conv6-7
32x upsampled prediction (FCN-32s)
2x conv7
pool4
16x upsampled prediction (FCN-16s)
4x conv7
2x pool4
pool3
8x upsampled prediction (FCN-8s)

Backbone CNN : VGG16
Reference:
https://www.kaggle.com/code/codelad/fcn-transfer-learning-image-segmentation/notebook

Setting Parameter FCN8s

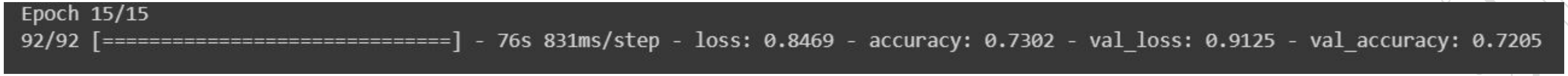
1. Width Image : 256
2. Height Image : 256
3. Classes : 12
4. Batch Size : 32

Setting Training Experiment FCN8s

1. Optimizer : Adam
2. Loss : Categorical Crossentropy
3. Metrics : Accuracy
4. Learning Rate : 0.0001
5. Weight Decay : 0.01; 0.001; 0.0001; 0.00001
6. Epochs : 15

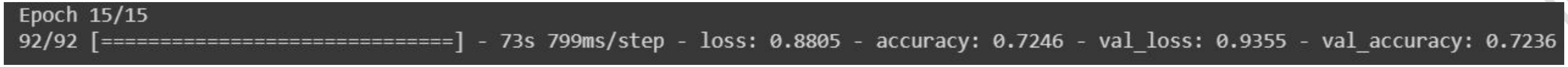
FCN8s Training Result

Weight Decay : 0.01



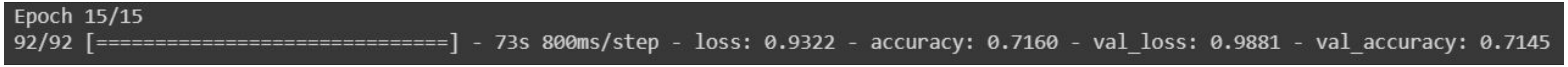
```
Epoch 15/15
92/92 [==============================] - 76s 831ms/step - loss: 0.8469 - accuracy: 0.7302 - val_loss: 0.9125 - val_accuracy: 0.7205
```

Weight Decay : 0.001



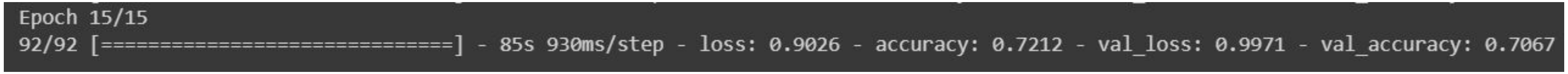
```
Epoch 15/15
92/92 [==============================] - 73s 799ms/step - loss: 0.8805 - accuracy: 0.7246 - val_loss: 0.9355 - val_accuracy: 0.7236
```

Weight Decay : 0.0001



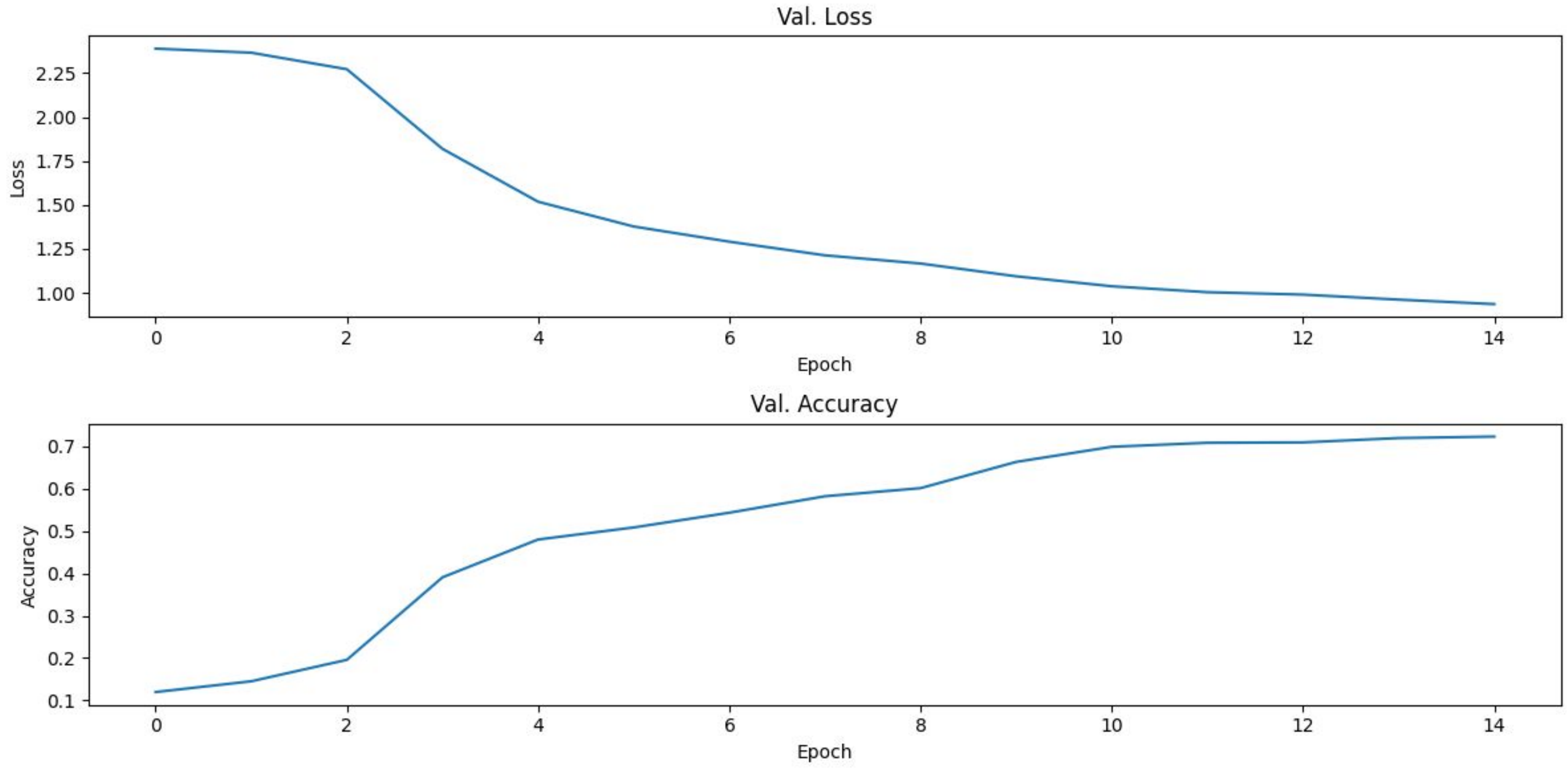
```
Epoch 15/15
92/92 [==============================] - 73s 800ms/step - loss: 0.9322 - accuracy: 0.7160 - val_loss: 0.9881 - val_accuracy: 0.7145
```

Weight Decay : 0.00001

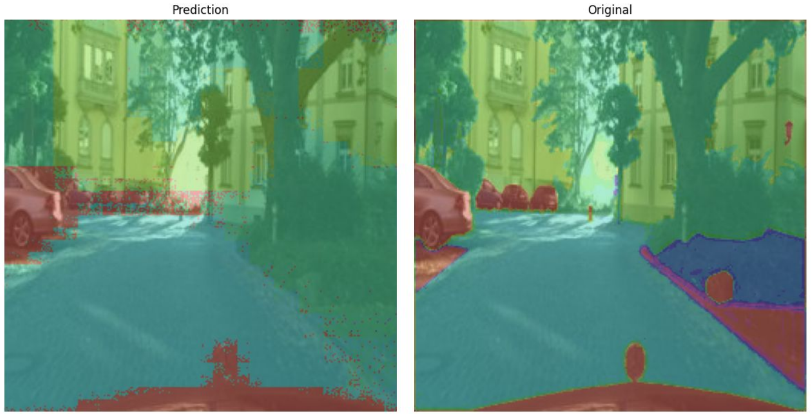


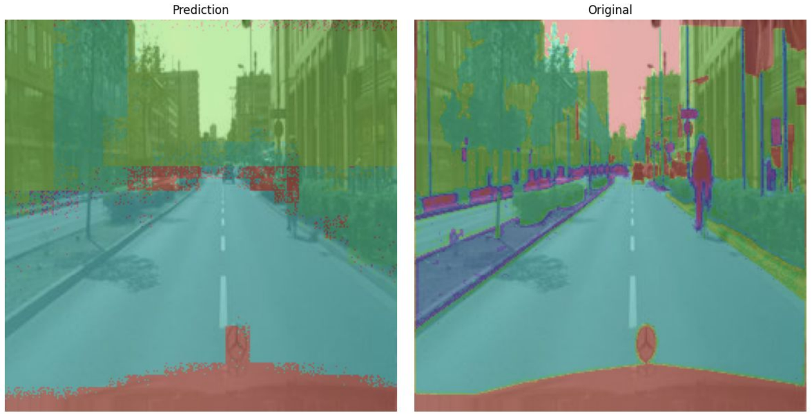
```
Epoch 15/15
92/92 [==============================] - 85s 930ms/step - loss: 0.9026 - accuracy: 0.7212 - val_loss: 0.9971 - val_accuracy: 0.7067
```

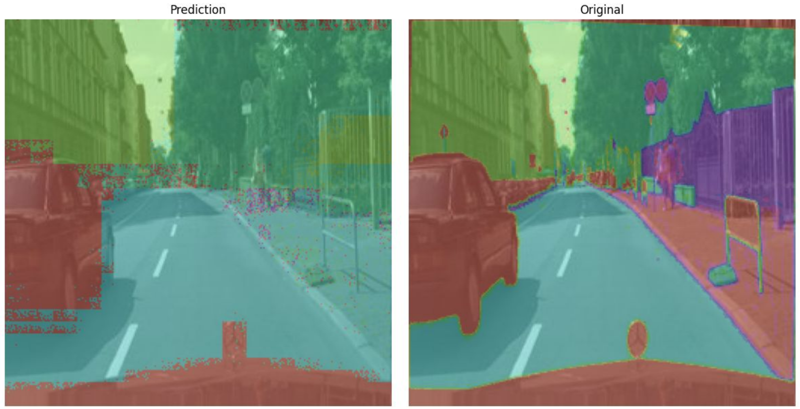
FCN8s Training Evaluation

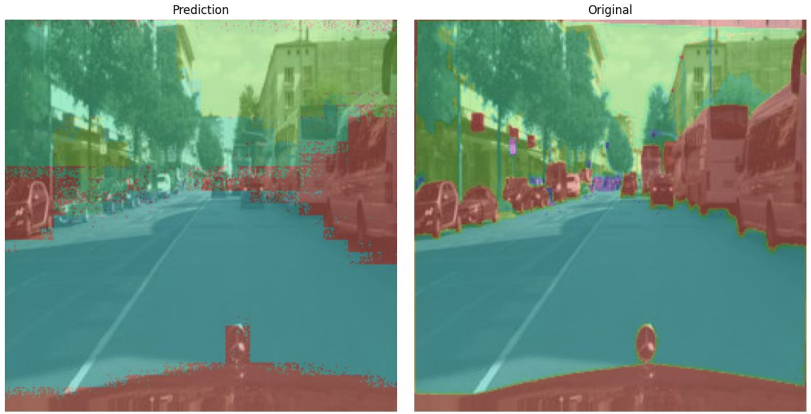


FCN8s Training Evaluation

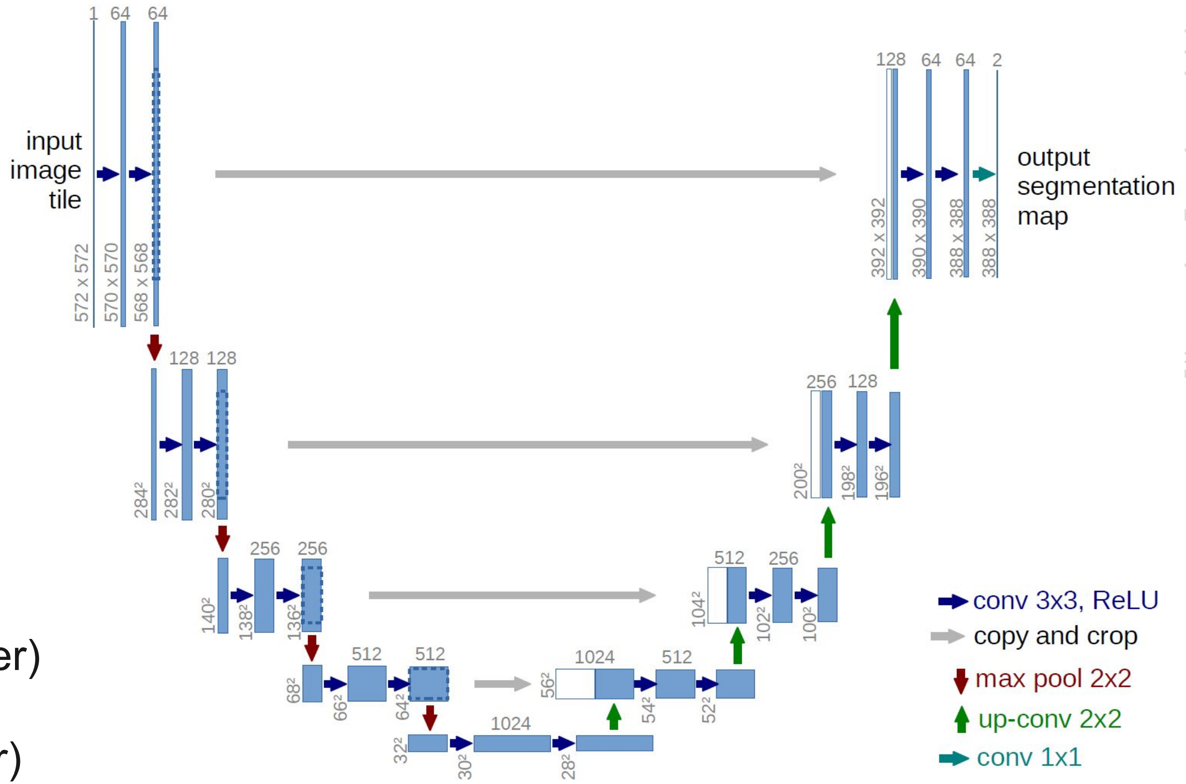








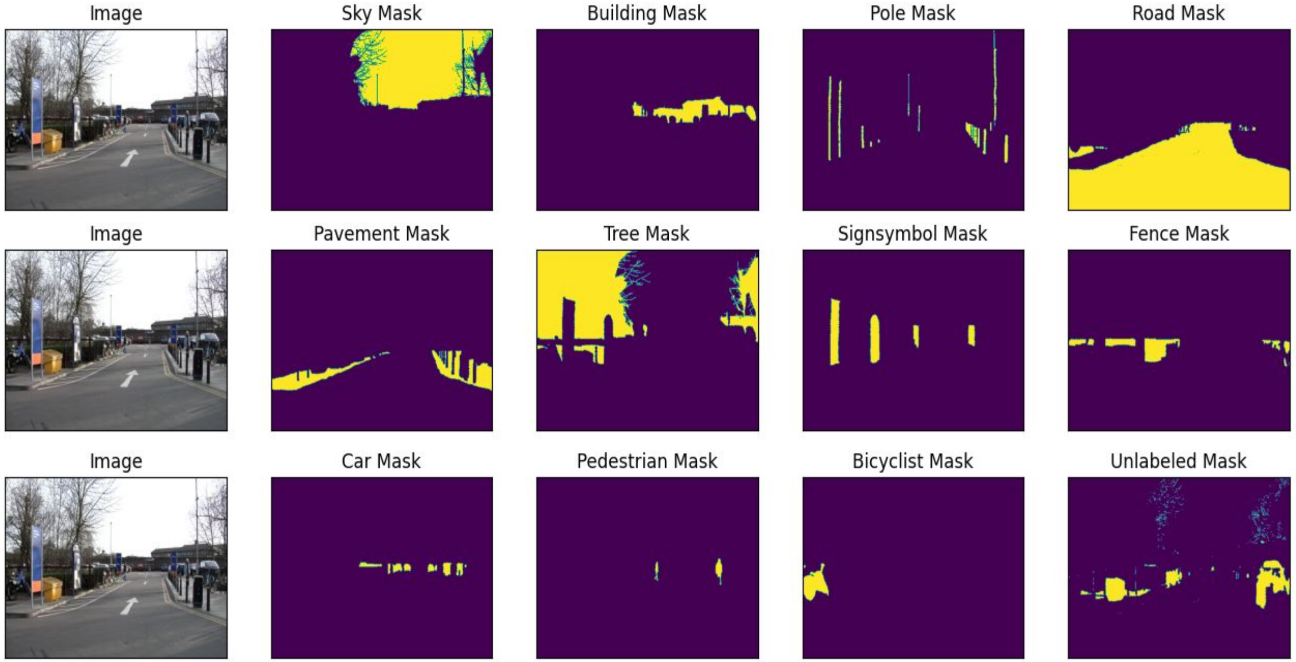
U-Net



Tiga bagian utama:

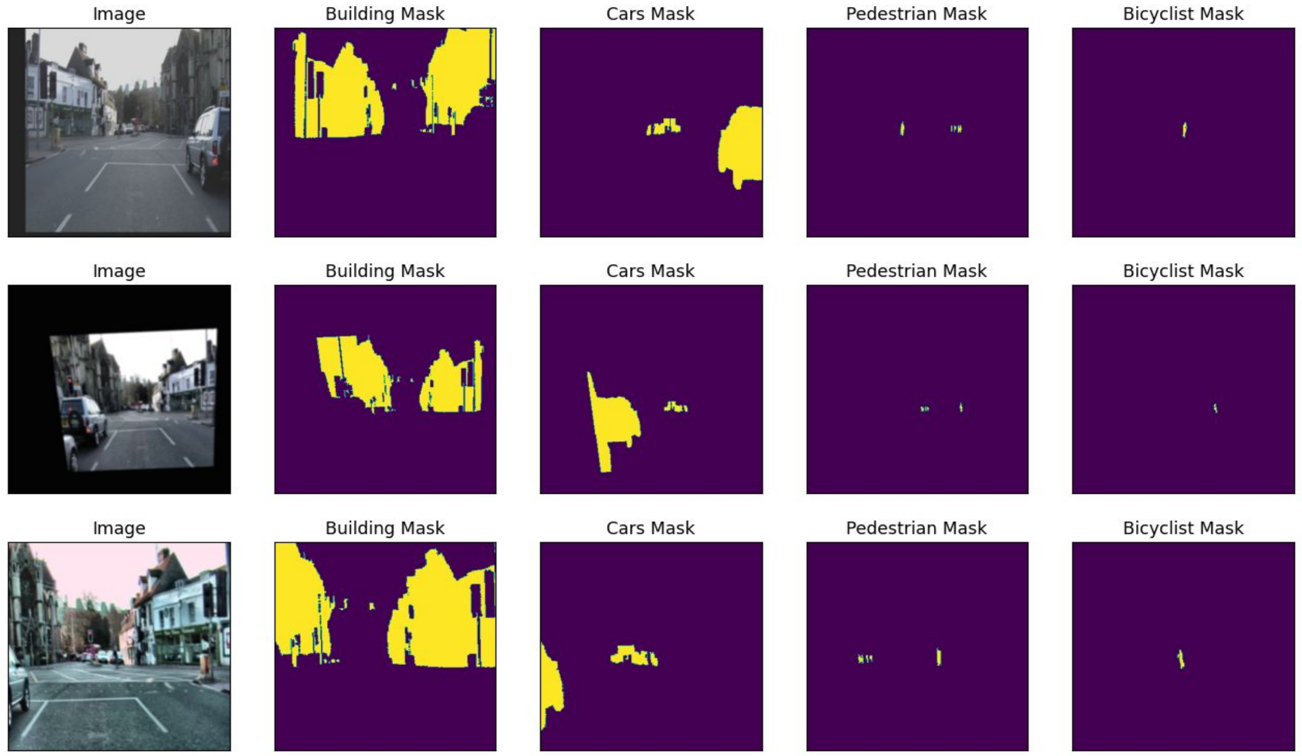
1. Contraction (encoder)
2. Bottleneck
3. Expansion (decoder)

Dataset



- 367 training data (split to 80/20 training and validation)
- 101 testing data

Data Augmentations (Use albumentations):



Training:

Model	UNet (libary: *https://github.com/qubvel/segmentation_models.pytorch*)
Decoder	resnet34, resnet50, resnet101, efficientnet-b4
Decoder weight	imagenet
Activation	softmax 2d
Learning Rate	1e-4 (for 0-24 epoch), 1e-5 (for 25-49 epoch), 5e-6 (for 50-74 epoch), 1e-6 (for next epochs)
Metrics	● DiceLoss ● IoU(threshold=0.5) ● Accuracy
Optimizers	Adam

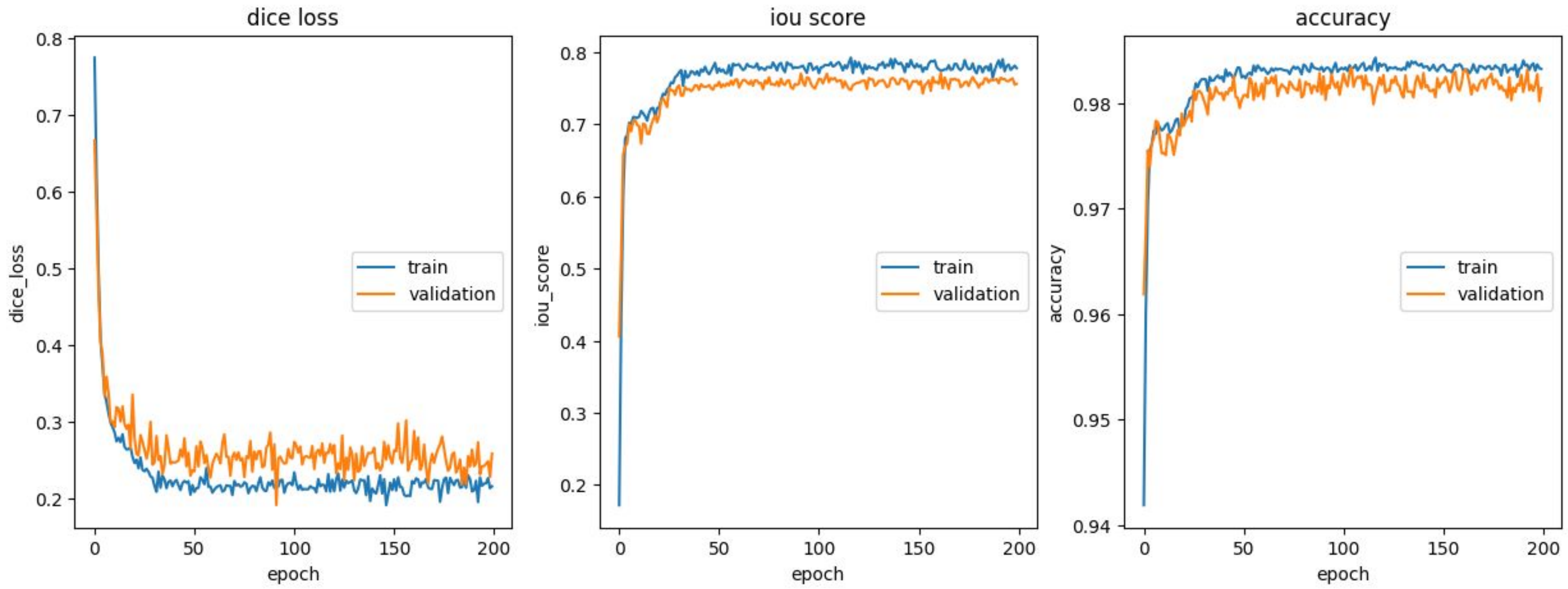
Training:

classes	encoder	pretrained	epoch	augmention	dice_loss	iou_score	accuracy	training time
12	resnet34	no	50	yes	0.2034	0.6702	*	*
12	resnet34	yes	50	yes	0.1247	0.7869	*	*
12	resnet50	no	50	yes	0.3523	0.4832	*	*
12	resnet50	yes	50	yes	0.1197	0.8043	*	*
12	efficientnet-b4	yes	50	yes	0.1903	0.7236	*	*

classes	encoder	pretrained	epoch	augmention	dice_loss	iou_score	accuracy	training time
3^	resnet50	yes	50	yes	0.9328	0.06404	*	*
12	resnet34	yes	200	yes	0.1164	0.7969	*	*
12	resnet101	yes	50	no	0.1941	0.7617	0.9792	45.67 min
12	resnet101	yes	50	yes	0.1869	0.7299	0.9749	48.64 min
12	resnet101	yes	200	yes	0.1154	0.8121	0.9828	192.26 min

* not documented
^3 classes:car, pedestrian, bicyclist

Training:


dice loss
dice_loss
0.8
0.7
0.6
0.5
0.4
0.3
0.2
train
validation
epoch
0
50
100
150
200
iou score
iou_score
accuracy
0.98
0.97
0.96
0.95
0.94

Evaluation:

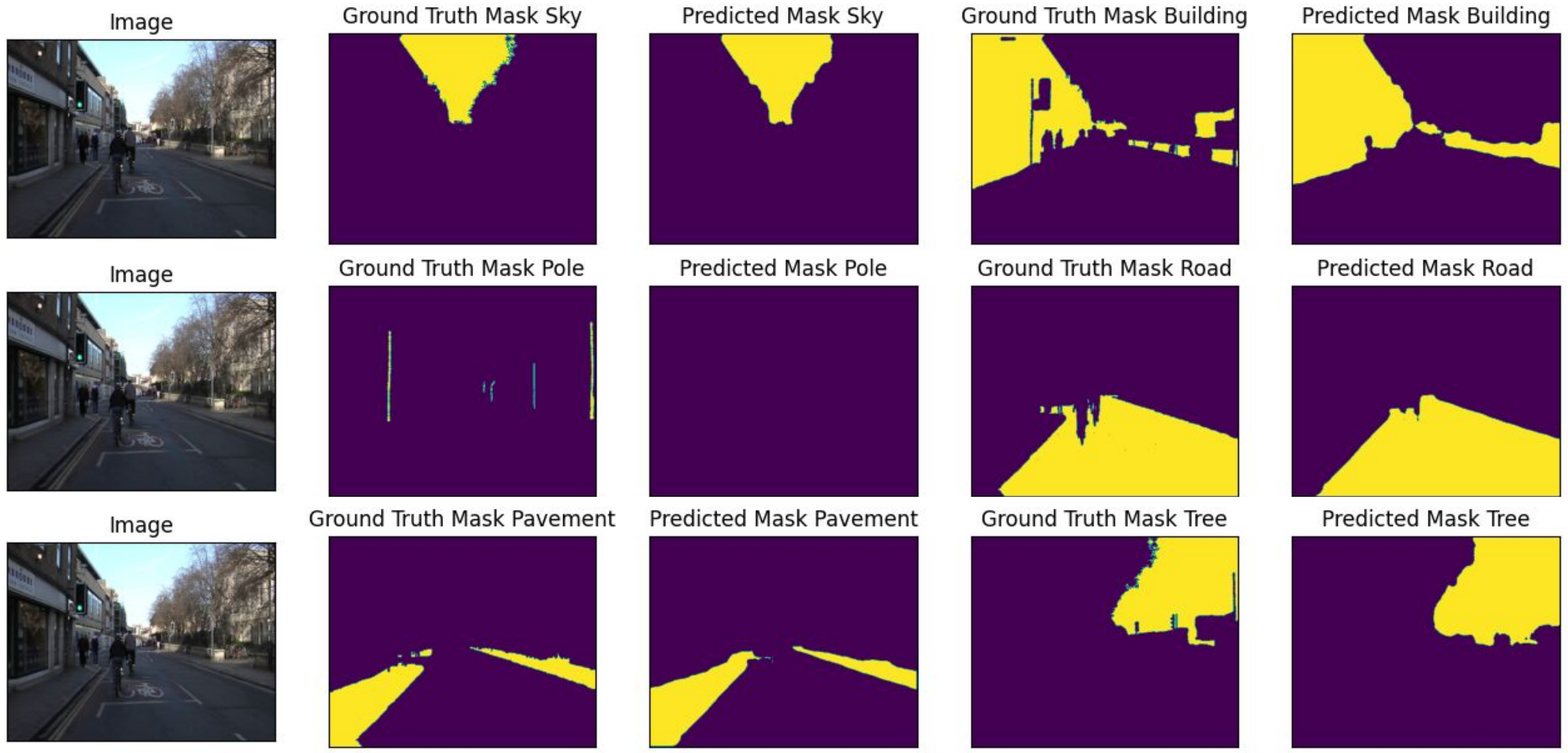


Image
Ground Truth Mask Sky
Predicted Mask Sky
Ground Truth Mask Building
Predicted Mask Building
Image
Ground Truth Mask Pole
Predicted Mask Pole
Ground Truth Mask Road
Predicted Mask Road
Image
Ground Truth Mask Pavement
Predicted Mask Pavement
Ground Truth Mask Tree
Predicted Mask Tree

Evaluation:

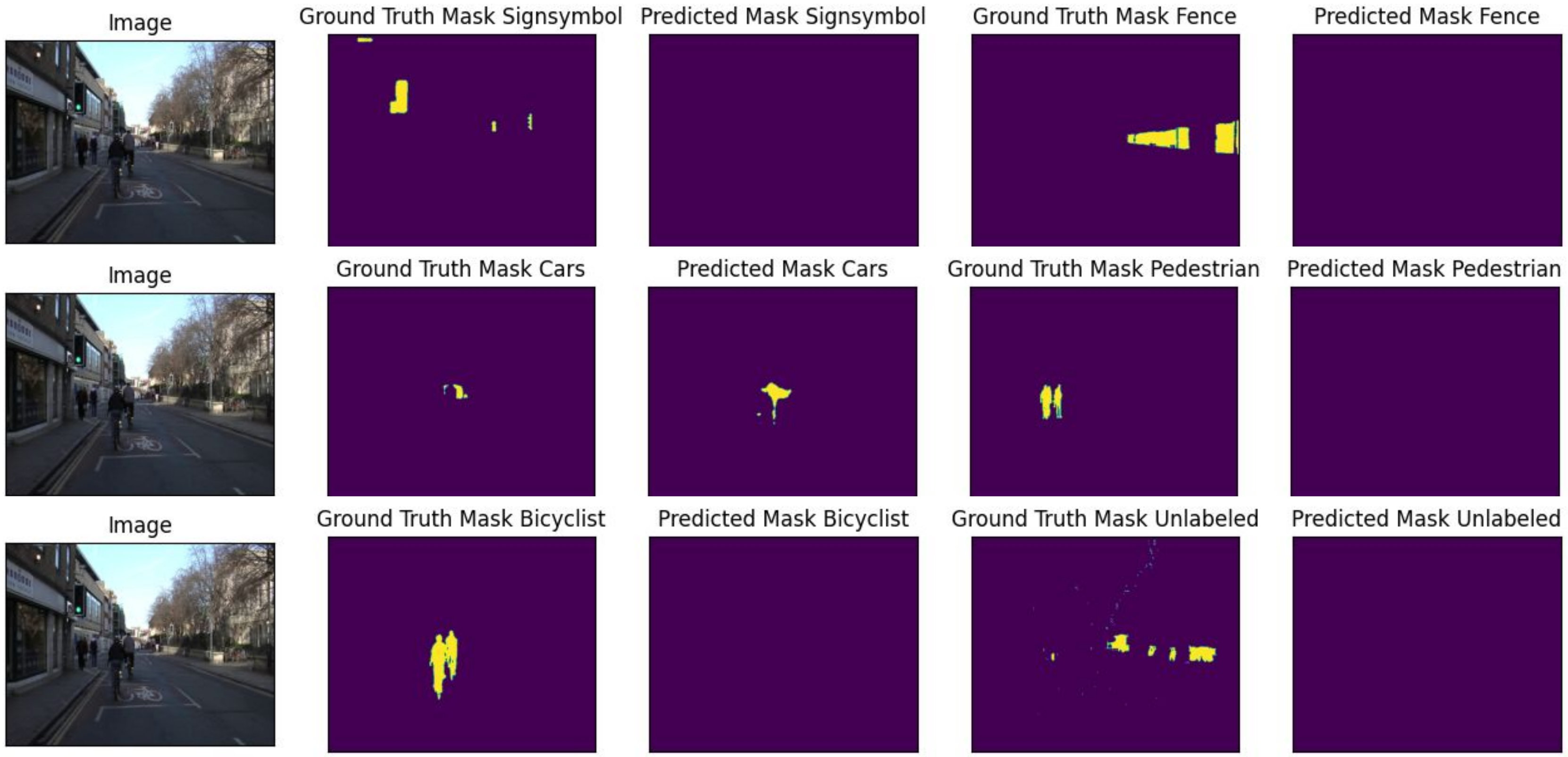
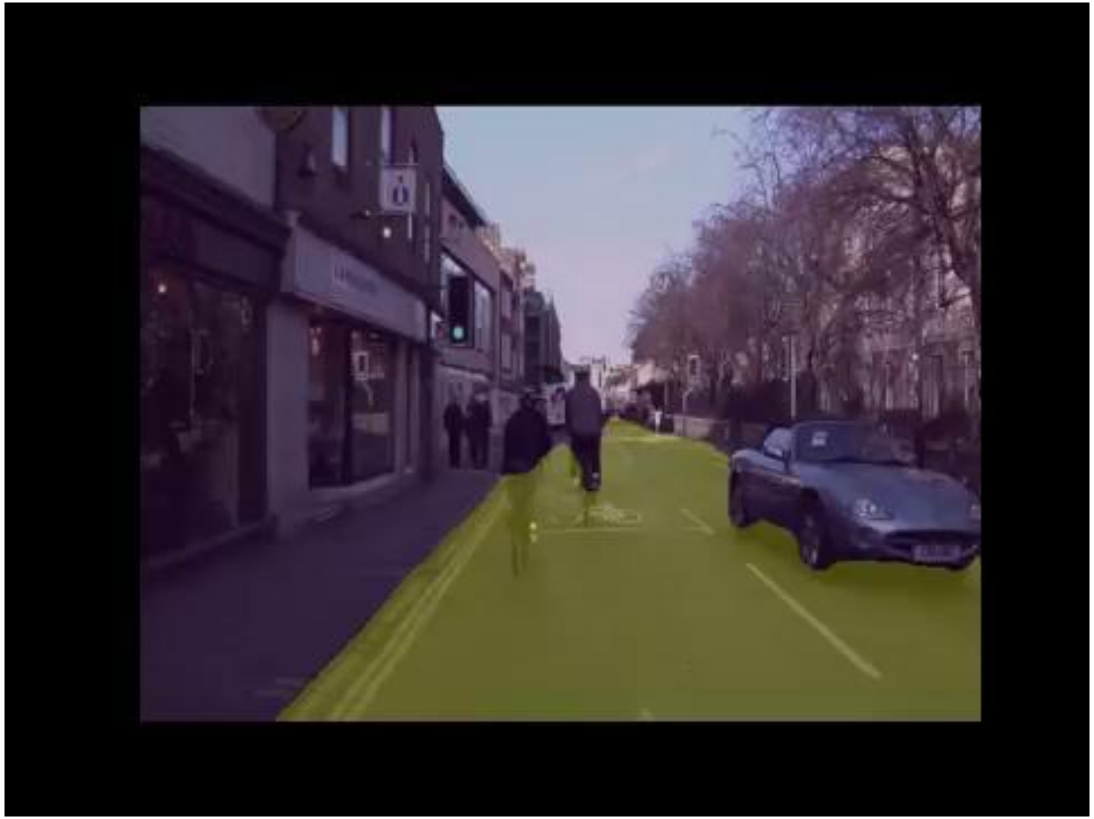


Image
Ground Truth Mask Signsymbol
Predicted Mask Signsymbol
Ground Truth Mask Fence
Predicted Mask Fence
Image
Ground Truth Mask Cars
Predicted Mask Cars
Ground Truth Mask Pedestrian
Predicted Mask Pedestrian
Image
Ground Truth Mask Bicyclist
Predicted Mask Bicyclist
Ground Truth Mask Unlabeled
Predicted Mask Unlabeled

Evaluation:



Comparison: FCN vs U-Net

Model	Classes	Encoder	Pretrained	Epoch	Augmentation	Accuracy	IoU
U-Net	12	Restnet101	Yes	50	No	0.9792	0.7617
FCN-8	12	VGG-16	Yes	15	No	0.7236	*

Conclusion

FCN8s:

- Eksperimen awal didapatkan nilai learning rate yang optimal 0.0001 dan jumlah epochs 15.
- Eksperimen selanjutnya memvariasikan weight decay dan optimal pada 0.001.
- Dari eksperimen mendapatkan akurasi yang optimal 0.7236.
- Hasil evaluasi prediksi gambar menunjukkan bahwa model ini dapat memprediksi segmen object gambar, tetapi hasilnya masih kurang halus, dan masih belum dapat memprediksi object yang kecil, jauh, serta rapat.
- Untuk improvement selanjutnya, sebaiknya melakukan eksperimen pada model arsitektur, hyperparameter, dan menerapkan metrik evaluasi yang lebih tepat seperti IoU score dan Dice Loss.

UNet:

- Di atas epoch 30 nilai metric sudah tidak banyak berubah (dice loss, IoU score, accuracy)
- Setting hyperparameter yang lain (encoder, pretrained, augmentation) juga tidak meningkatkan akurasi segmentasi secara signifikan
- Model memiliki nilai accuracy yang tinggi tapi tidak dapat membuat segmentasi untuk objek-objek dengan ukuran kecil
- Ketidakmampuan membuat segmentasi untuk objek-objek kecil ini disebabkan kasus imbalance classes (object berukuran kecil memiliki jumlah pixel yang relatif lebih sedikit dibandingkan dengan background-nya)
- Saran untuk eksperimen berikutnya: Jumlah data untuk training harus diperbanyak (saat ini hanya mempergunakan 367 data) dan variasinya diperbesar