National University
Of Computer and Emerging Sciences

Department of Computer Science

Student Name: Tazmeen Afroz
Roll No:22P-9252 **Date:** Dec 3, 2025
Semester: 7th

Computer Vision

Lab 10: Semantic Segmentation with TensorFlow – Implementing U-Net on Cityscapes Dataset

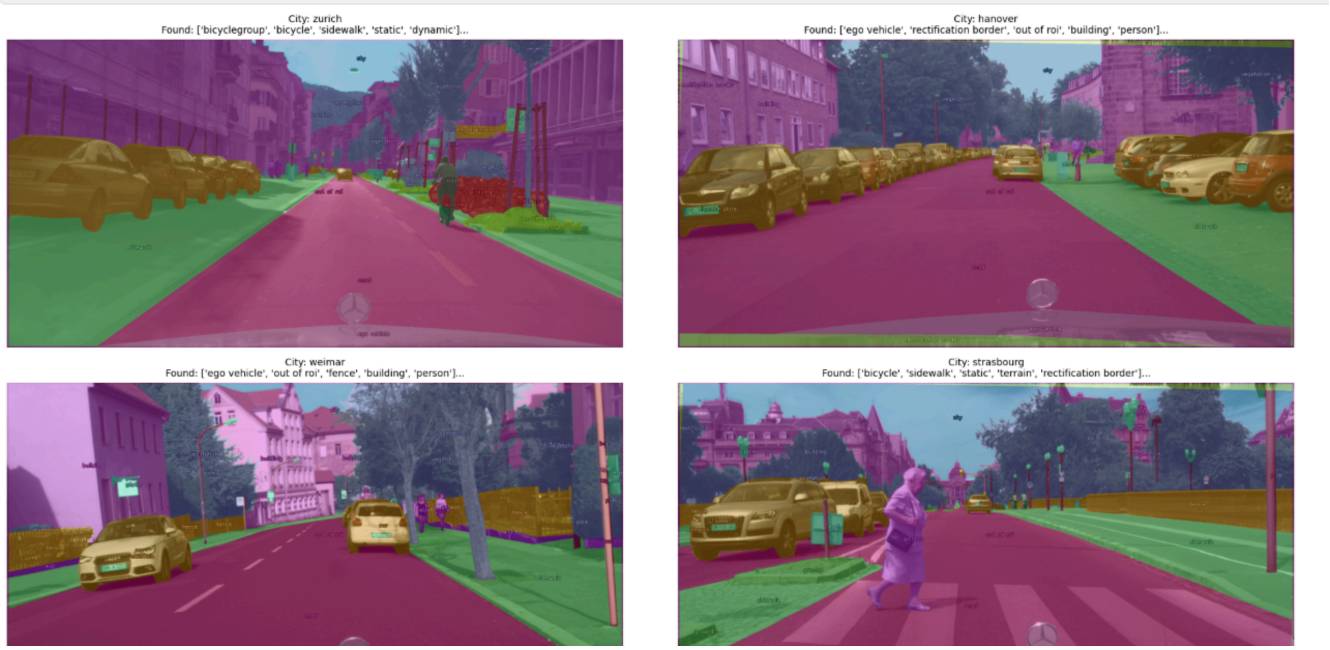
Task

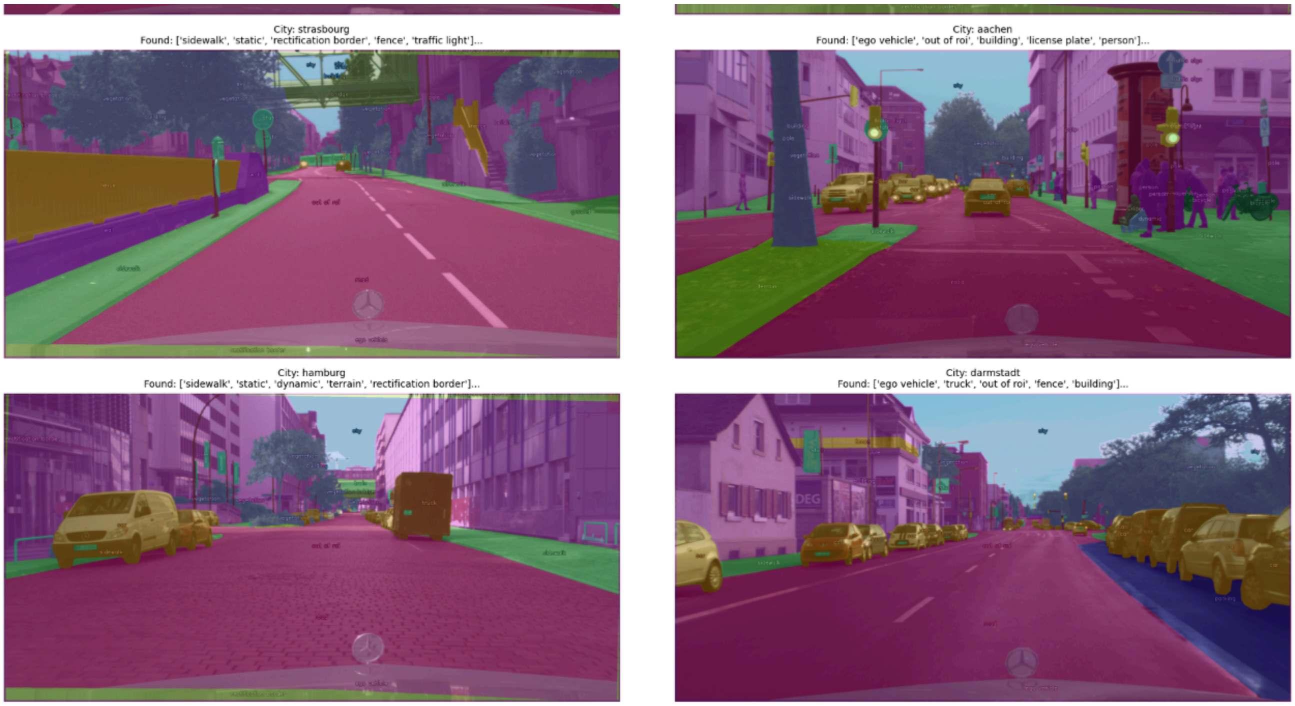
Implement a complete U-Net architecture from scratch using TensorFlow/Keras, building every component manually without relying on any pretrained or high-level segmentation APIs. Your model should follow the classical encoder–decoder structure: the encoder (contracting path) extracts features through repeated blocks of two convolution layers with ReLU activation followed by max-pooling, while the decoder (expanding path) progressively upsamples the feature maps using transposed convolutions and concatenates them with corresponding encoder layers through skip connections to recover spatial information. After reconstructing the full-resolution feature map, apply a final 1×1 convolution to produce pixel-wise class predictions using either softmax (multi-class) output. The goal is to understand how U-Net performs pixel-level segmentation by combining feature extraction, upsampling, and skip connections in a unified architecture.

Dataset: https://www.kaggle.com/datasets/electraawais/cityscape-dataset/data

Kaggle Notebook Link :
https://www.kaggle.com/code/tazmeenafroz/u-net-implementation

Visualization:


City: zurich
Found: ['bicyclegroup', 'bicycle', 'sidewalk', 'static', 'dynamic']...
City: hanover
Found: ['ego vehicle', 'rectification border', 'out of roi', 'building', 'person']...
City: weimar
Found: ['ego vehicle', 'out of roi', 'fence', 'building', 'person']...
City: strasbourg
Found: ['bicycle', 'sidewalk', 'static', 'terrain', 'rectification border']...



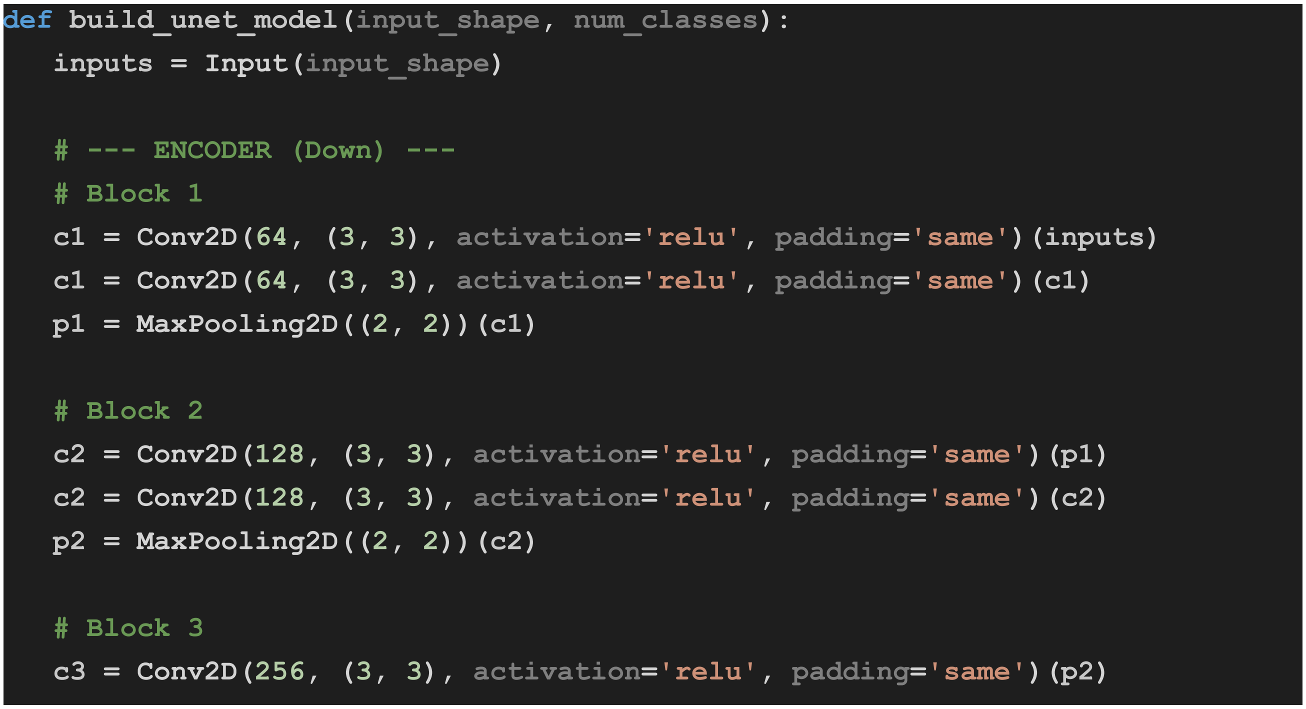
City: strasbourg
Found: ['sidewalk', 'static', 'rectification border', 'fence', 'traffic light']...

City: aachen
Found: ['ego vehicle', 'out of roi', 'building', 'license plate', 'person']...

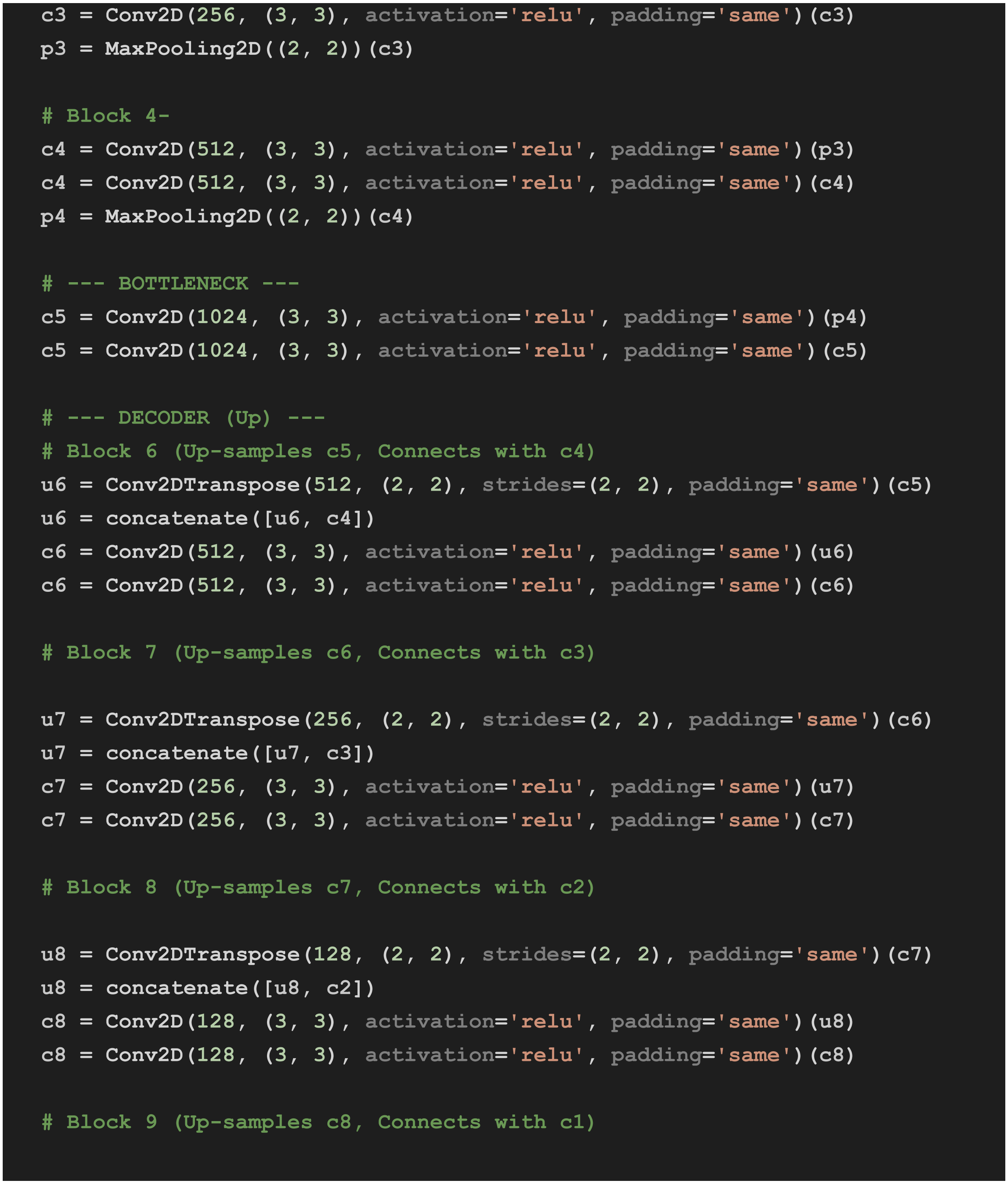
City: hamburg
Found: ['sidewalk', 'static', 'dynamic', 'terrain', 'rectification border']...

City: darmstadt
Found: ['ego vehicle', 'truck', 'out of roi', 'fence', 'building']...

U-Net Code:



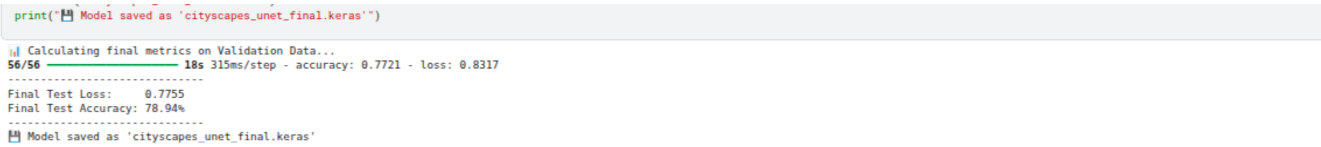
```
def build_unet_model(input_shape, num_classes):
   inputs = Input(input_shape)

   # --- ENCODER (Down) ---
   # Block 1
   c1 = Conv2D(64, (3, 3), activation='relu', padding='same')(inputs)
   c1 = Conv2D(64, (3, 3), activation='relu', padding='same')(c1)
   p1 = MaxPooling2D((2, 2))(c1)

   # Block 2
   c2 = Conv2D(128, (3, 3), activation='relu', padding='same')(p1)
   c2 = Conv2D(128, (3, 3), activation='relu', padding='same')(c2)
   p2 = MaxPooling2D((2, 2))(c2)

   # Block 3
   c3 = Conv2D(256, (3, 3), activation='relu', padding='same')(p2)
```



```
c3 = Conv2D(256, (3, 3), activation='relu', padding='same')(c3)
p3 = MaxPooling2D((2, 2))(c3)

# Block 4-
c4 = Conv2D(512, (3, 3), activation='relu', padding='same')(p3)
c4 = Conv2D(512, (3, 3), activation='relu', padding='same')(c4)
p4 = MaxPooling2D((2, 2))(c4)

# --- BOTTLENECK ---
c5 = Conv2D(1024, (3, 3), activation='relu', padding='same')(p4)
c5 = Conv2D(1024, (3, 3), activation='relu', padding='same')(c5)

# --- DECODER (Up) ---
# Block 6 (Up-samples c5, Connects with c4)
u6 = Conv2DTranspose(512, (2, 2), strides=(2, 2), padding='same')(c5)
u6 = concatenate([u6, c4])
c6 = Conv2D(512, (3, 3), activation='relu', padding='same')(u6)
c6 = Conv2D(512, (3, 3), activation='relu', padding='same')(c6)

# Block 7 (Up-samples c6, Connects with c3)

u7 = Conv2DTranspose(256, (2, 2), strides=(2, 2), padding='same')(c6)
u7 = concatenate([u7, c3])
c7 = Conv2D(256, (3, 3), activation='relu', padding='same')(u7)
c7 = Conv2D(256, (3, 3), activation='relu', padding='same')(c7)

# Block 8 (Up-samples c7, Connects with c2)

u8 = Conv2DTranspose(128, (2, 2), strides=(2, 2), padding='same')(c7)
u8 = concatenate([u8, c2])
c8 = Conv2D(128, (3, 3), activation='relu', padding='same')(u8)
c8 = Conv2D(128, (3, 3), activation='relu', padding='same')(c8)

# Block 9 (Up-samples c8, Connects with c1)
```

```
    u9 = Conv2DTranspose(64, (2, 2), strides=(2, 2), padding='same')(c8)
    u9 = concatenate([u9, c1])
    c9 = Conv2D(64, (3, 3), activation='relu', padding='same')(u9)
    c9 = Conv2D(64, (3, 3), activation='relu', padding='same')(c9)

    # --- OUTPUT ---
    outputs = Conv2D(num_classes, (1, 1), activation='softmax')(c9)

    model = Model(inputs=[inputs], outputs=[outputs])
    return model
```

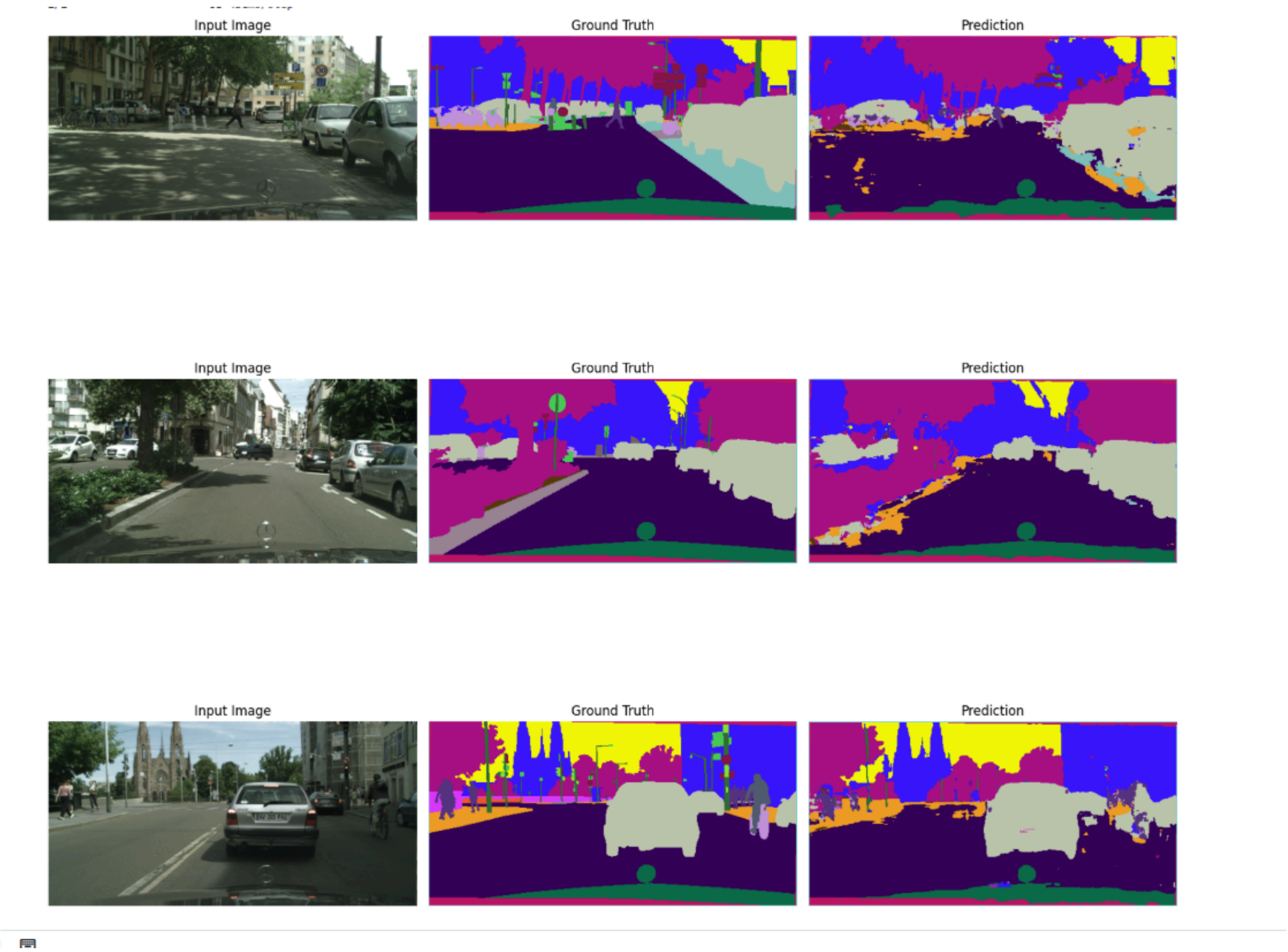
Test accuracy on 50 epochs ->78%



```
print("💾 Model saved as 'cityscapes_unet_final.keras'")
```

```
📊 Calculating final metrics on Validation Data...
56/56 ━━━━━━━━━━━━━━━━━━━━ 18s 315ms/step - accuracy: 0.7721 - loss: 0.8317
------------------------------
Final Test Loss:     0.7755
Final Test Accuracy: 78.94%
------------------------------
💾 Model saved as 'cityscapes_unet_final.keras'
```

Results


Input Image
Ground Truth
Prediction
Input Image
Ground Truth
Prediction
Input Image
Ground Truth
Prediction