Semantic Segmentation using UNET & Transformer

- Soham Shinde

Introduction

Semantic Segmentation

Task of assigning a label to each pixel in an image based on its semantic meaning.

Output: Pixel-wise Segmentation Map which partitions the given image into different regions, each corresponding to a particular object or background class.

Applications:

- Medical Image Analysis
- Video Surveillance
- **Autonomous Driving**

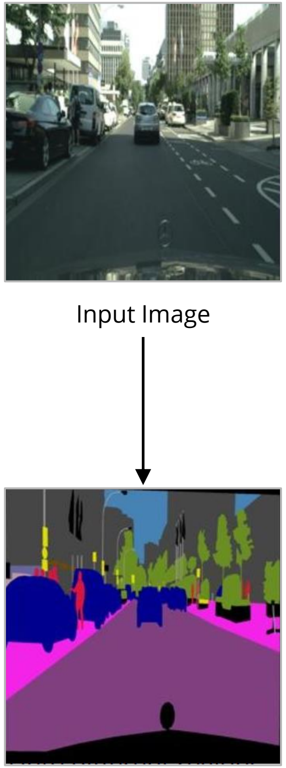


Fig1. Semantic Segmentation

Dataset and Preprocessing

Dataset: **CityScapes**

It has 256x256px images of Urban Street Scenes with 31 semantic classes.

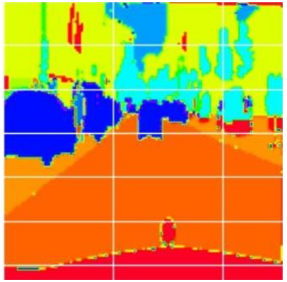
The data was splitted into Training - Test - Validation as 2975 - 300 - 200.

Actual Image

Preprocessing: preprocess()

1. Resize all the images, normalize the input image and create black mask of the same size as target

2. Map the color to class for each pixel of the target image by matching the given class id.



Original Mask

Fig2. Preprocessed Mask

Baseline: FCN

FCN model is easy to implement and flexible to size of input image size.

Fully connected layers in a traditional CNN are replaced with convolutional layers.

Architecture:

1. Convolution layers are used in the encoding for downsampling
2. Transposed Convolution layers for upsampling in the Decoder
3. Final Layer for classification

Problems:

1. Loss of spatial details during downsample which can lead to inaccurate model performance

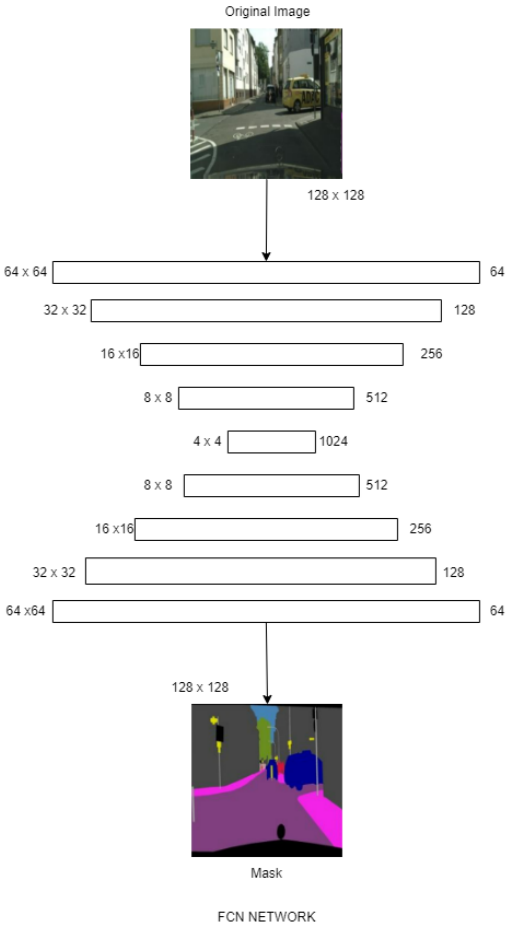


Fig3. FCN

UNET

The architecture follows a encoder-decoder network with skip connections.

1. **Contracting path:** convolutional and pooling operations to extract the features
2. **Expansive path:** transposed convolutions to upsample the feature maps and generate a segmentation mask
3. **Skip-Connections:** allow information from earlier layers to be used directly in the later layers which would have lost in pooling-downsampling. Captures both, the global context as well as fine-grained details

Problems:

1. Struggles to capture objects of small size or having complex shapes

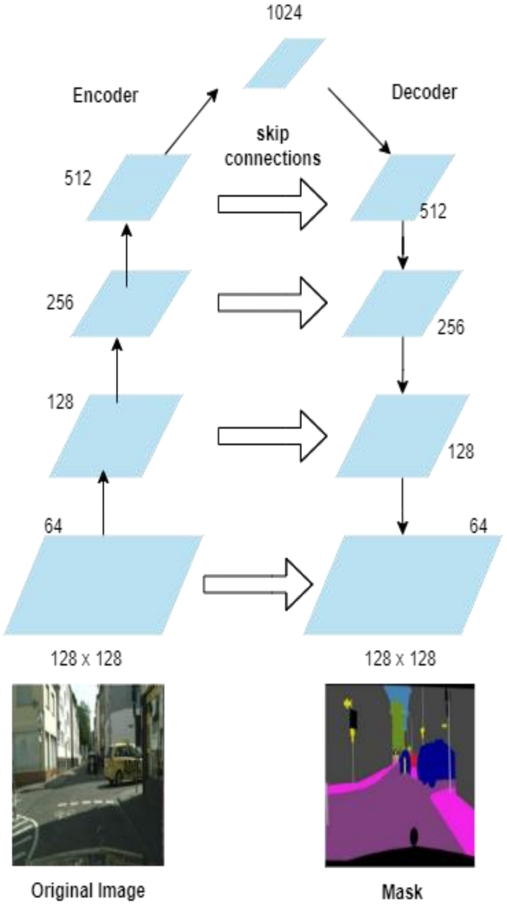


Fig4. UNET

EXPERIMENTAL MODIFICATIONS ON UNET

Residual Connections:

These shortcut connections reduce the problem of vanishing gradients during training by allowing the gradients to flow directly to a later layer, thus learning more complex features.

Attention Mechanism:

Selectively focus on relevant parts of the input data, enhancing the network's ability to attend to important regions using attention mechanism.

Residual + Attention:

Combine the two methods together to collectively improve the performance of prior models which only used either of the methods

EXPERIMENTAL MODIFICATIONS ON UNET

UNET + Atrous Spatial Pyramid Pooling

ASPP: Extracts features at different scales and concatenate them to get a robust representation of the image.

U-Net+ ASPP model:

1. Uses ASPP module after the encoder network to increase its receptive field
2. Captures global and local context
3. Efficiently identifies objects of varied shapes and sizes belonging to different classes

The model achieved the best results among the U-Net variations due to its multi scale feature extraction capability.

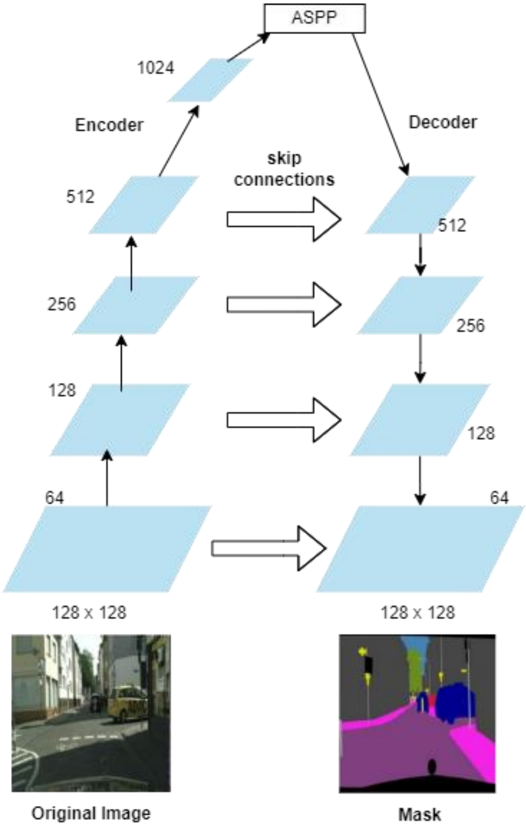


Fig5. UNET ASPP

SWIN TRANSFORMER OVERVIEW

SWIN T: A type of ViT that uses patch merging approach and window based Multi head Self Attention (W-MSA) unlike ViT.

W-MSA: Like MSA but computes self-attention only within the patch/ partition thereby reducing complexity.

GELU (Gaussian Error Linear Unit) non-linear activation function is used.

Patch Merging: Processing patches into groups and then merging these patches into larger partitions and repeating the process.

Shifted Windows Approach: shifting the position of the partitioning scheme by a small amount (e.g., by half the size of the partition), such that adjacent partitions overlap with each other.

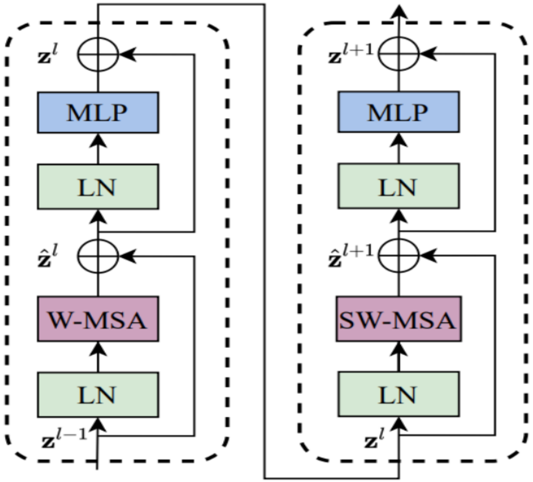


Figure 1: Two Consecutive Swin-T blocks

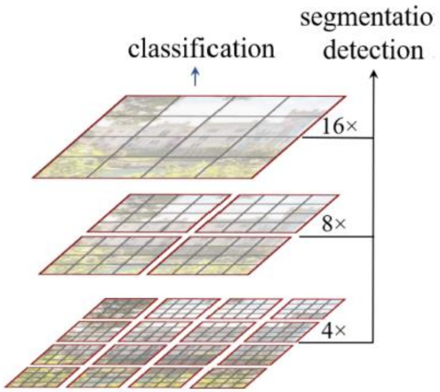


Figure 2: Hierarchical Patch Merging Approach

Reference: [Swin Transformer: Hierarchical Vision Transformer using Shifted Windows]

SWIN-UNET

Architecture:

1. **3** Swin-T blocks in the encoder
2. **3** Swin- T blocks in the decoder
3. **1** Bottleneck block
4. **Patch Merging** and **Patch Expanding** is applied after every transformer block in encoder and decoder respectively

Advantages:

1. **Improved performance** -> Transformer based approach
2. **Better multi-scale features** ->Patch Merging -Shifted Windows
3. **Fine-grained details** -> Patch Expanding approach

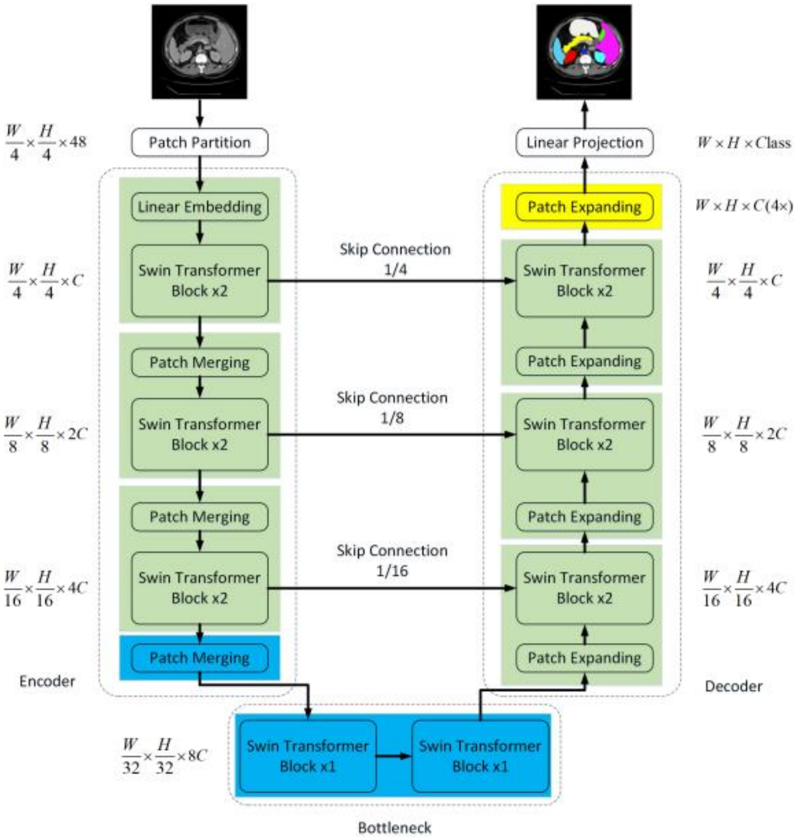


Fig8. SWIN UNET

Reference: Swin-Unet: Unet-like Pure Transformer for Medical Image Segmentation

HYPER PARAMETER TUNING and RESULTS

Hyperparameters tuned:

1. **Batch Sizes:** 4, 8, 16, 32
2. **Learning Rate**: 0.01 - 0.001
3. **Image Sizes:** 128, 256
4. **Drop out rate**: 0.2, 0.5
5. **Number of epochs** - 100
    a. Later introduced Early stopping to avoid overfitting

Other specifications:

1. All models were trained from scratch
2. Loss function used: Sparse Categorical Cross Entropy
3. All models were evaluated using Mean Intersection over Union and Pixel Accuracy
4. Weights were initialized using Xavier initialization

RESULTS and OUTPUTS

- **Results for 128 x 128-pixel resolution:-**

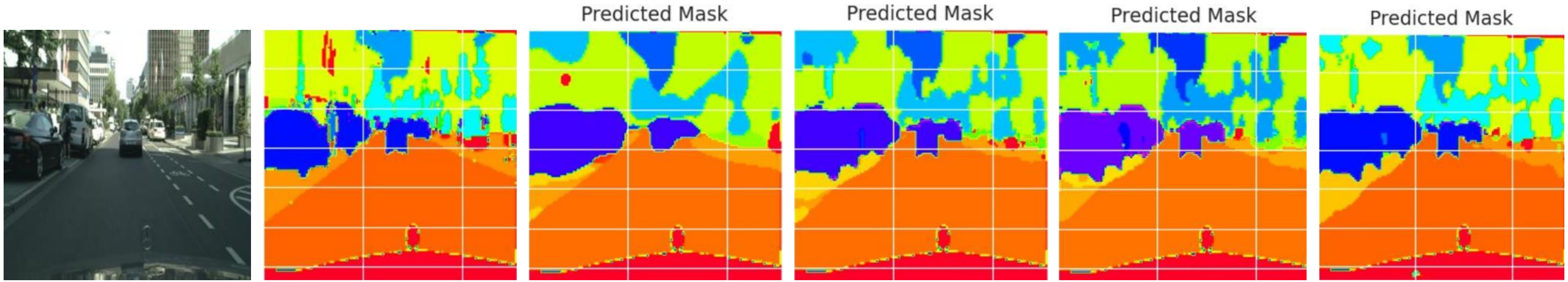
	Valid Accuracy	**Valid mIOU**	**Test Accuracy**	**Test mIOU**
FCN	0.810	0.264	0.810	0.266
UNET	0.846	0.339	0.841	0.322
UNET + RES	0.843	0.344	0.841	0.326
UNET + ATT	0.842	0.343	0.841	0.330
UNET + RES +ATT	0.838	0.337	0.838	0.325
UNET + ASPP	0.846	0.342	0.838	0.356
SWIN UNET	0.834	0.370	0.821	0.363

- **Results for 256 x 256-pixel resolution:-**

	Valid Accuracy	**Valid mIOU**	**Test Accuracy**	**Test mIOU**
FCN	0.824	0.288	0.812	0.281
UNET	0.847	0.343	0.838	0.341
UNET + ASPP	0.854	0.361	0.846	0.354
SWIN UNET	0.861	0.395	0.859	0.393

RESULTS and OUTPUTS

Inference Results:



Actual Image **Original Mask** **FCN** **UNET** **UNET + ASPP** **SWIN UNET**

Fig 9. Predictions

CONCLUSION

- Utilizing Unet for Road Semantic Segmentation
- Implemented different mechanisms within the unet model to see how they affect the performance of our model
- Additionally, we chose to implement a transformer based UNet approach to leverage its powerful self-attention mechanism for capturing long-range dependencies between pixels

Future Work:

- Using pre training to further improve the model performance
- Improving the preprocessing method
- Utilizing Data augmentation methods to improve the model robustness and generalization.
- Lot of scope to tune hyperparameters to further improve model performance

Thank You!