Documentation

The goal of the project is to create semantic segmentation model which could segment humans from the webcamera in real-time setting. We have experimented with the two architetcures.

Dataset

Link: https://github.com/VikramShenoy97/Human-Segmentation-Dataset
Examples: 300
Resolution: Variable resolution

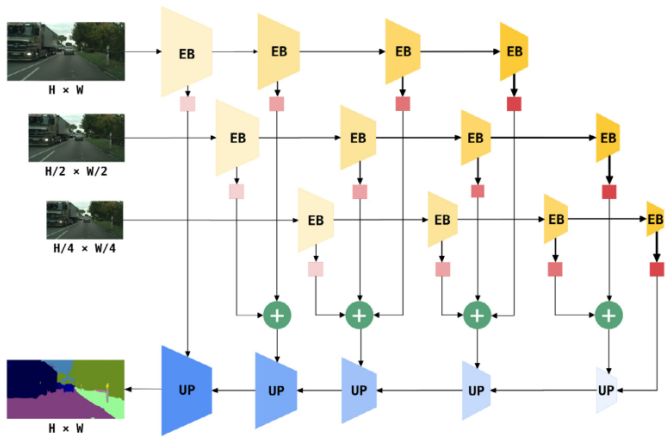





```
{
        "src.transforms.augmentations.AffineJitter": {
                "rotation": [-10,10],
                "scale": [0.7, 2],
                "shear": [-2 ,2]
        }
},
{
        "src.transforms.augmentations.ColorJitter": {
                "brightness": 0.3,
                "contrast": 0.8,
                "saturation": 0.3,
                "hue": 0.1
        }
}
```

Pyramid Swiftnet

Schema:



Description

The original image is being interpolated on 3 different resolutions. These resolutions are then passed to the shared encoder. The encoder is composed of encoder blocks that downsample the image and feature maps. We then upsample this extracted feature map to the original resolution. We do that by continuously upsampling the feature map with the decoder block and with features brought by lateral connections from encoder features. These connections bring the information which was lost during the encoding process and help us to reconstruct a realistic segmentation map.

The benefit of this approach is that model is invariant to the image resoultion and to the scale of the objects in the image. By blending feature maps from different resolutions we are increasing the receptive field of the model and thus increase the performance on large scale objects.

Speed

Semantic segmentation performance on full resolution images from Cityscapes val. Column fps shows the inference speed (frames per second) on GTX 1080 Ti:

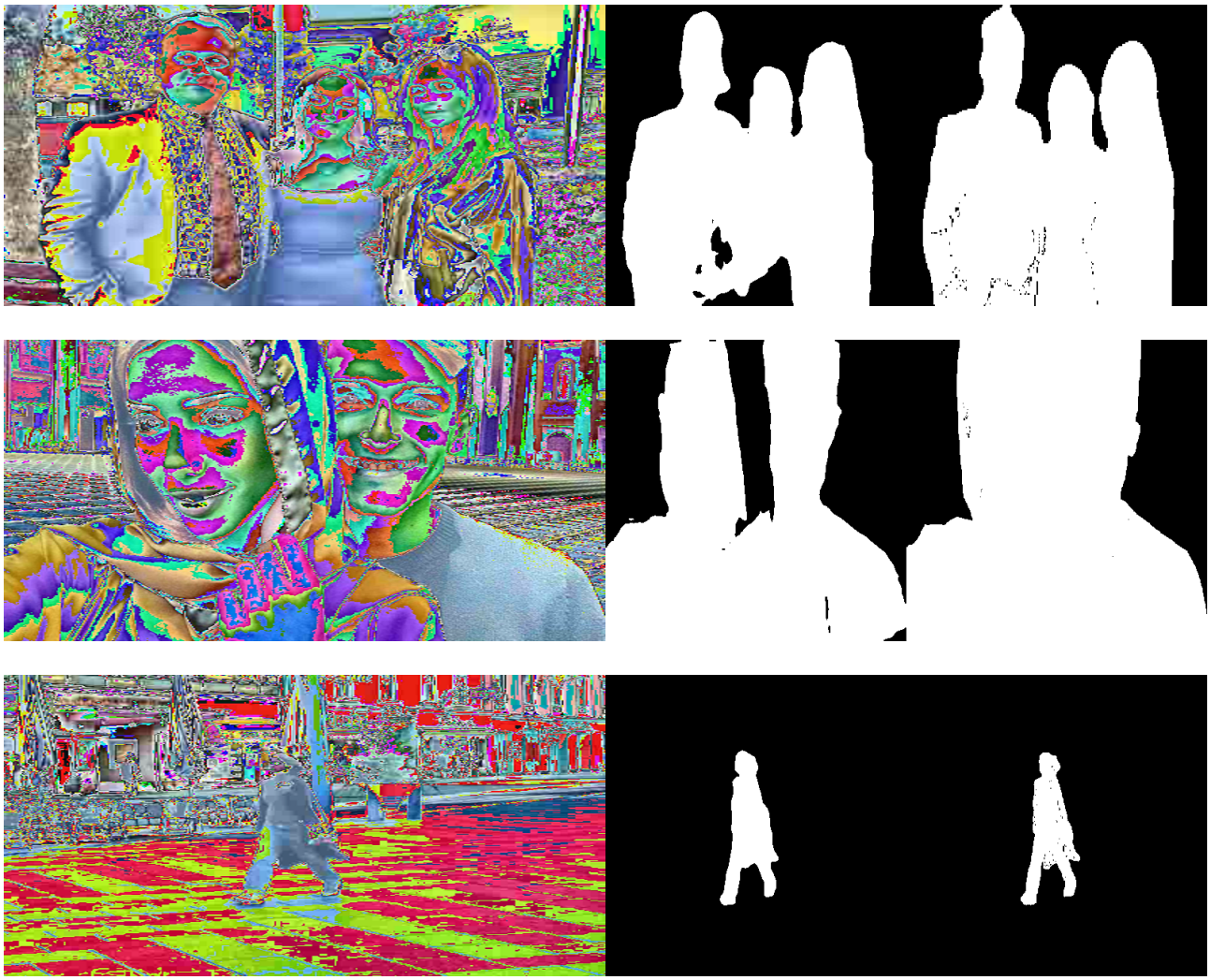
backbone	method	mIoU	fps	GFLOP	parameters
ResNet-18	pyramid	76.4	34.0	128	12.0 M
MobileNet V2	pyramid	77.4	29.7	42	2.7 M

Dataset split:

- Train = 171
- Valid = 29
- Test = 90

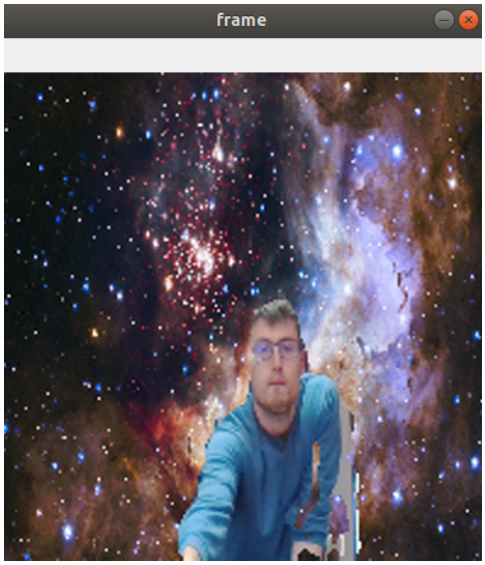
Training=100 epochs

Evaluation on test set:
accuracy: 96.9%
precission: 93.6%
recall: 95.6%
mIoU: 89.7%
ce loss: 0.085



Inference throughput on my cpu on 200x200 resolution: 7.16 FPS

One of the main cons of this model is that it doesn't work so well on small scaled objects. Reason for this is increased receptive field of neurons which are then more activated when they notice larger object.



The largets distance form the webcam where the model works reasonably well.

Reg Seg

https://arxiv.org/pdf/2111.09957v2.pdf

Proposes a novel dilated block structure which contain dilated convolution operations. The aim is to increase the receptive field while maintaining real-time performance.

Speed: 30 FPS on Cityscapes

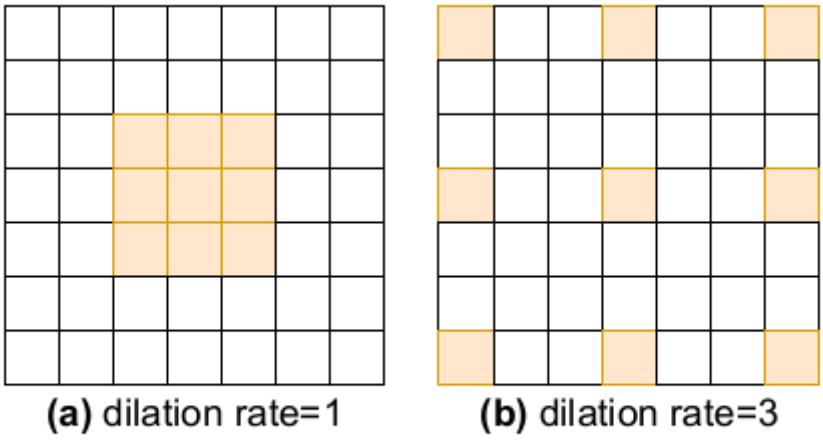
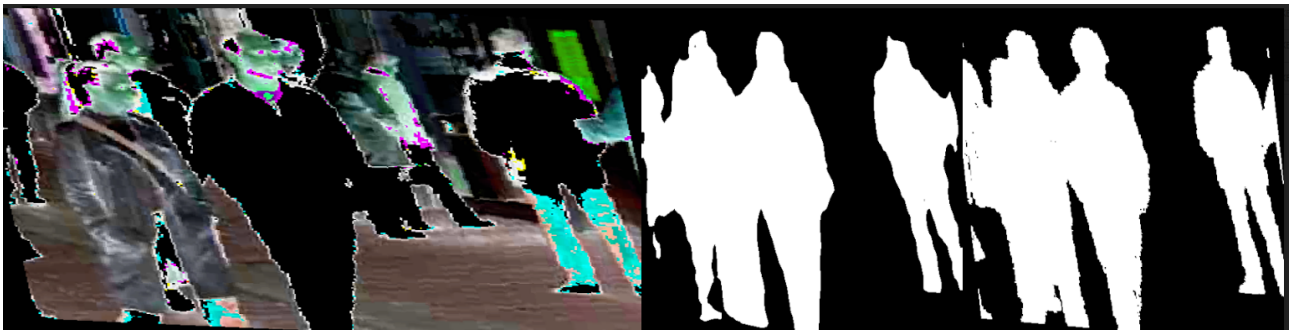
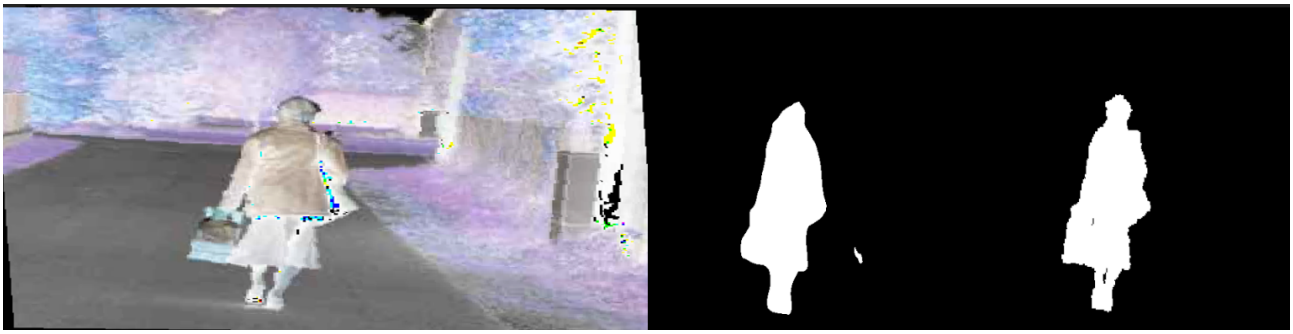


Figure 2. Dilated Convolution.

Evaluation on test set:
accuracy: 90.6%
mIoU: 77,17%
precission: 85.59%
recall: 88.78%
test loss: 0.2611

Inference throughput on my cpu on 200x200 resolution: 16 FPS





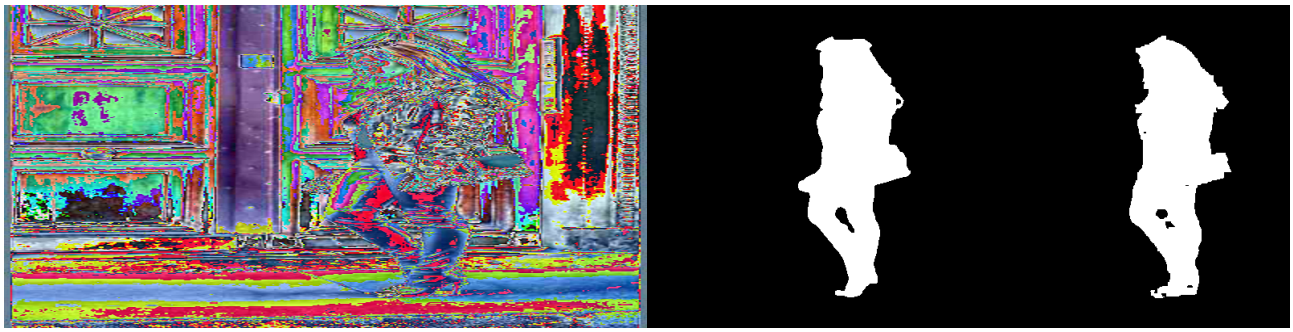
New Dataset

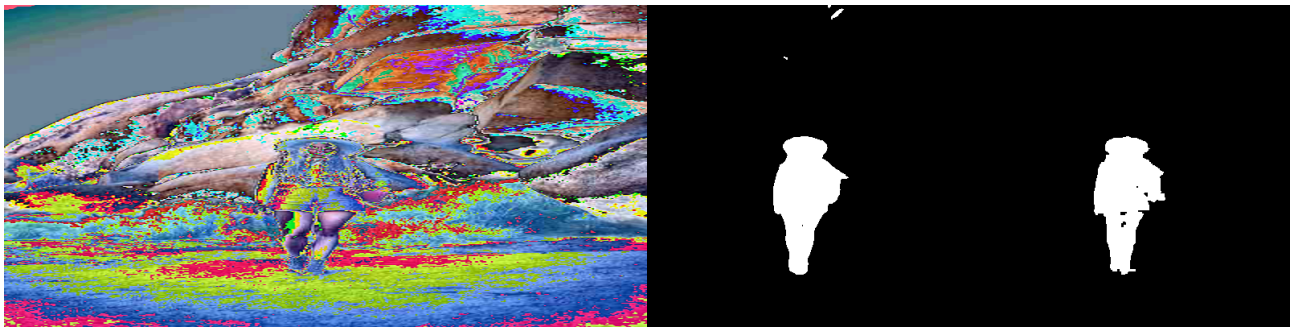
New dataset: https://www.kaggle.com/soumikrakshit/human-segmentation

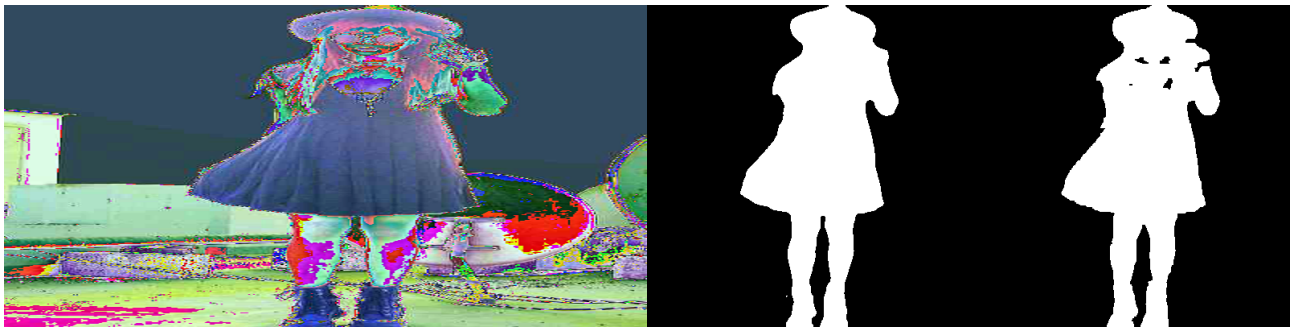
Split:

- **Train = 10588**
- **Test = 5351**
- **Valid = 1767**

Evaluation on test set:
accuracy: 98.4%
mIoU: 93.56%
precission: 95.63%
recall: 97.73%
test loss: 0.0445







FPS = 1.28