Siddhant Kumar

B.Tech Undergrad, IIT Mandi | ML Enthusiast | Coder | Photographer

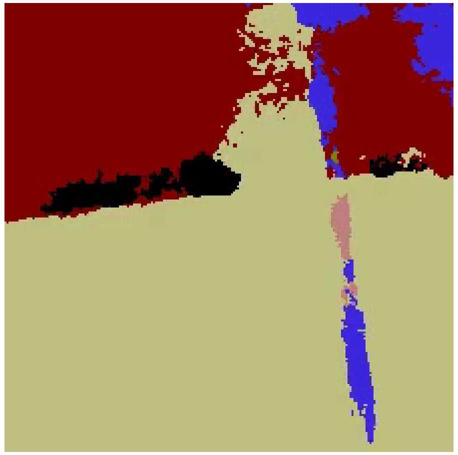
Blog Web Resume About

Summary of - SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation

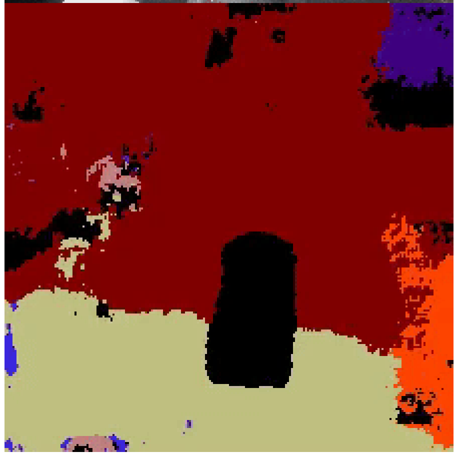
Paper published by: Vijay Badrinarayanan, Alex Kendall, Roberto Cipolla at CVPR ’15

Some videos I used to play around with SegNet

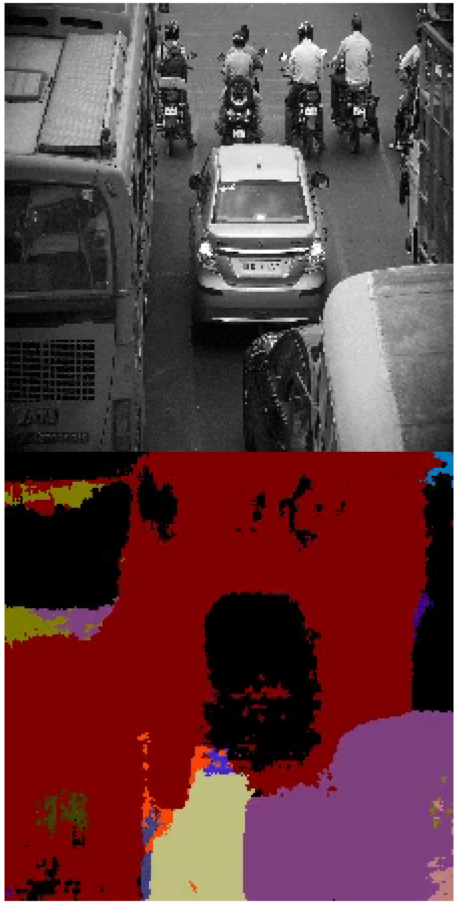
1.



2.



3\.



For more info on replicating this visit my repo here

TLDR:

- Uses a novel technique to upsample encoder output which involves storing the max-pooling indices used in pooling layer. This gives reasonably good performance and is space efficient
- VGG16 with only forward connections and non trainable layers is used as ÷encoder. This leads to very less parameters.

Problem

- Semantic pixel-wise labelling i.e. labelling each pixel of an image to belong to some class(tree, road, sky, etc) as shown in the image.

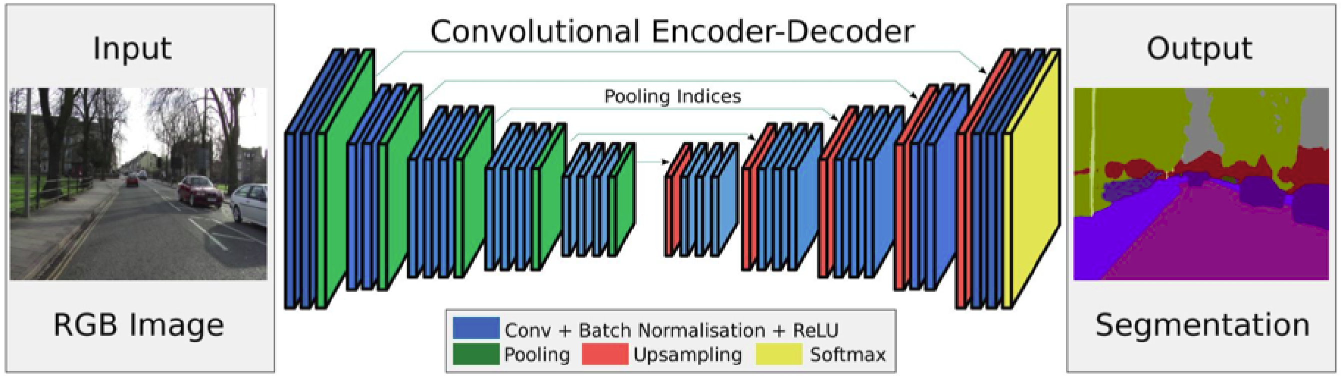


Fig 1: Segmentation of a road scene imagess

Some applications include autonomous driving, scene understanding, etc. Direct adoption of classification networks for pixel wise segmentation yields poor results mainly because *max-pooling* and *subsampling* reduce feature map resolution and hence output resolution is reduced. Even if extrapolated to original resolution, lossy image is generated.

SegNet - Challenges

- Trained on road scene datasets hence, classes represent macro objects, hence segmentations are desired to be smooth
- Boundary information is critical for objects like road markings and other small objects. (*Boundary delineation*)
- Major use cases will be embedded systems hence it must be *Computationally Efficient*

SegNet- Architecture

Encoder-Decoder pairs are used to create feature maps for classifications of different resolutions.

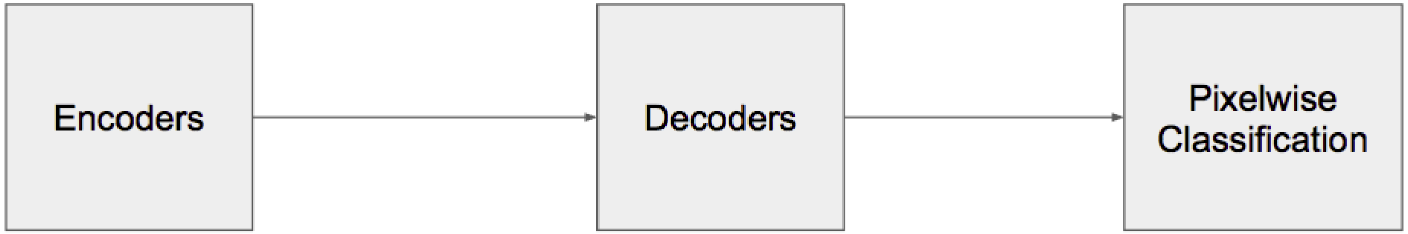


Fig 2: Nut-shell architecture

Encoder

- 13 VGG16 Conv layers
- Not fully connected, this reduces parameters from 134M to 14.7M
- Good initial weights are available hence these layers are made non trainable

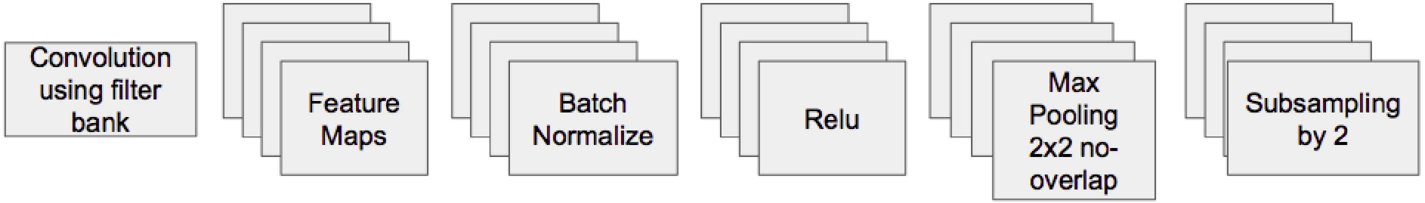


Fig 3: Encoder architecture

Each encoder is like Fig 3. The novelty is in the subsampling stage, Max-pooling is used to achieve translation invariance over small spatial shifts in the image, combine that with

Subsampling and it leads to each pixel governing a *larger input image context* (spatial window). These methods achieve better classification accuracy but reduce the feature map size, this leads to lossy image representation with blurred boundaries which is not ideal for segmentation purpose. It is desired that output image resolution is same as input image, to achieve this SegNet does Upsampling in its decoder, to do that it needs to store some information. It is necessary to *capture and store* boundary information in the encoder feature maps before sub-sampling. In order to to that space efficiently, SegNet stores only the *max-pooling indices* i.e. the locations of maximum feature value in each pooling window is memorised for each encoder map. Only 2 bits are needed for each window of 2x2, slight loss of precision, but *tradeoff*.

- Advantages
  - Improved boundary delineation
  - Less number of parameters

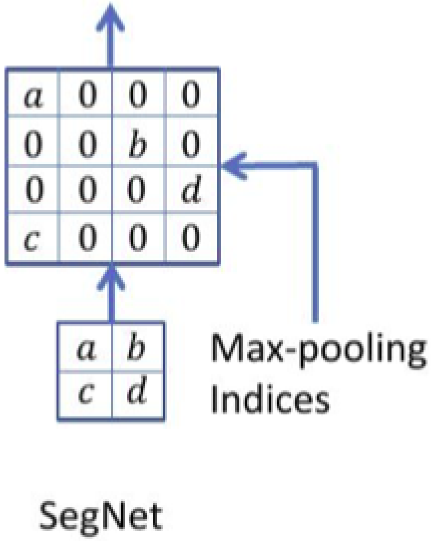


Fig 4: Upsamplig in SegNet

This form of upsampling can be incorporated in any encoder-decoder architecture

Decoder

- For each of the 13 encoders there is a corresponding decoder which upsamples the feature map using memorised *max-pooling indices*
- Sparse feature maps of higher resolutions produced
- Sparse maps are fed through a *trainable filter bank* to produce dense feature maps
- The last decoder is connected to a *softmax classifier* which classifies each pixel

SegNet paper compares its technique with several other decoders as shown in Fig 5.

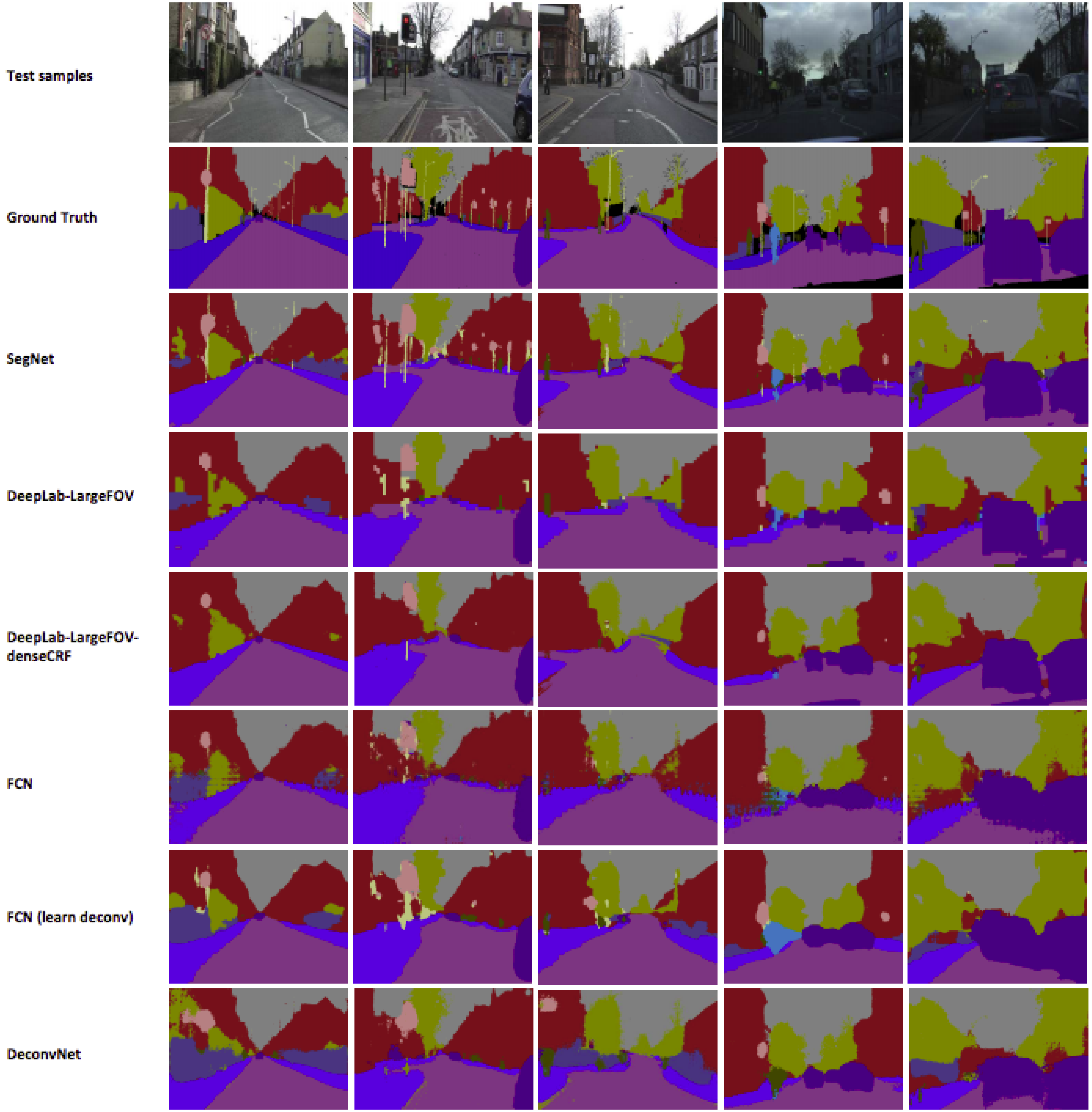

Test samples
Ground Truth
SegNet
DeepLab-LargeFOV
DeepLab-LargeFOV-denseCRF
FCN
FCN (learn deconv)
DeconvNet

Fig 5: Several decoders compared

Written on July 13, 2017