Learning context with factorization techniques in semantic segmentation

Nageswara Rao Gurram
University of Rochester
ngurram@ur.rochester.edu

Abstract

In the latest research on semantic segmentation problem, there has been a pattern going on to achieve better accuracy i.e Encoder-Context Retrieval-Decoder. In many networks, the Encoder is any Deep CNN such as ResNet [8] and Decoder is built with few layers of up-sampling layers by leaving most of the novelty to context retrieval module. To mention few, Deeplabv3+ [10] and [5] used Atrous Spatial Pyramid Pooling (ASPP) module, EMANet [15] relies on EM algorithm based attention module, EncNet [4] uses context encoding module [3] and A^2Net uses double attention module for the same purpose. All these methods require heavy computation with huge number of parameters. In this work, I tried to get almost same accuracy by reducing the parameter space to very minimal by using matrix factorization module to get the same global context. The detailed training and evaluation of the network is done on PASCAL VOC 2012 dataset. The complete code is at https://github.com/nageshgurram12/generic-semantic-segmentation/tree/mfnet

1. Introduction

Semantic Segmentation is one of the important computer vision problems today. Contrast to classification problem where the task is identifying the correct label/class for the whole image, in semantic segmentation every pixel will get classified to a label/class. There are numerous applications of solving this problem, which may include:

- Detecting objects in autonomous driving
- Medical Imaging Analysis
- Satellite Imagery Analysis
- Industrial inspection

Figure 1. Semantic Segmentation

This problem can also be viewed as denoising the image from very high dimensional space to output space where dimension is limited by number labels in dataset. So the essential part of the problem lies in mapping of high level semantics to the ground truth labeled images. Over the years, numerous architectures have been proposed to solve this problem. One can trace back all these solutions from fully convolutional networks (FCN) [6] where they applied fully convolution layers first to-do down-sampling and then up-sampling to match the ground truth. One more network that comes under this blanket is U-Net [13] where they added skip connections from backbone layers to decoder. As fully convolutional networks depend wholly on convolution operations, they failed to capture long range dependencies. For this reason, various networks have been proposed with separate module in-between encoder and decoder. In [5], they've used spatial pyramid pooling module (SPP) with increasing sizes of pooling layers and then aggregation over them to get global context over the entire spatial region. The another kind of architectures that used same principles for capturing overall spatial context are DeeplabV3 [9], DeeplabV3+ [10]. Although in these networks, they've used atrous convolution layers instead of pooling layers, the module serves same purpose as 'SPP' in PSPNet [5].

In recent years, there have been many approaches that used attention mechanism for the same purpose. In non-local networks [16], correlation between every pair of pixels in feature maps is calculated and in A^2Net [17] they proposed a double attention block to do same. But as the computation complexity increases to do this full kernel attention, Xia. Li *et al.* in EMANet [15] obtained the reduced bases and responsibilities vectors using EM algorithm. In similar way Hang. Zang *et al.* in [4] proposed a context encoding module to learn this latent vectors that represents a code word dictionary to weight the correct objects more in image.

In this work, I've explored factorization techniques to learn the global context from feature maps similar to [15], [4]. In previous works, other kind of machine learning algorithms (not part of the network itself) like EM, attention and kernel blocks were introduced to learn spatial correspondence between pixels that are farther away from each other. However, this task can also be done with factorization to learn the latent components that represent object categories. So in this work, most of the focus is going to be in:

- Learning the global context in feature maps using factorization techniques.
- Evaluate the end-end model performance, efficiency and size on PASCAL VOC 2012 dataset and compare with other state-of-the-art models.

2. Background

There had been lot of research in 'semantic segmentation' in past few years, its good to know some of the important papers that kick-started the problem and approached in different directions.

2.1. FCN

In the last few years there had been tremendous growth to solve segmentation problem using deep learning. FCN [6] networks attempted to solve the problem by arranging fully convolution layers first by down-sampling and then up-sampling to match the ground truth. This down-sampling using convolution and max-pooling operations and then up-sampling using the any aforementioned techniques referred as encoder-decoder pattern.

2.2. U-Net

To segment bio-medical scanned images, Olaf Ronneberger *et al.* in [13] proposed a 'U' shaped network with skip connections from encoder layers. Like in FCN, first input image is down-sampled with traditional convolution, max-pooling layers and then up-sampled again to get original spatial size. However, in FCN Jonathan Long *et al.* [6] couldn't capture localization information as up-sampling only low spatial size semantic features will lose the local context.For this reason they added skip connections from low level feature maps in encoder layers to up-sampled features in decoder.

2.3. PSPNet

In PSPNet [5], Hengshuang Zhao *et al.* used average pooling over the sub regions for the semantic features obtained through encoder to get **multi-scale context information**. They called this as spatial pyramid pooling module (SPP) [7] as pooling is applied over sub-regions at different rates. In PSPNet, once the pooling is applied, a 1x1 convolution is applied to reduce the channels and finally all the features from different pooling modules are aggregated with low level features of encoder to up-sample to the original size.

2.4. Deeplab v3+

Liang-Chieh Chen *et al.* proposed multiple architectures [12], [11], [9], [10] over the past years for semantic segmentation task. In their recent work Deeplab v3+ [10], the input image will first go through any of the backbone like ResNet [8] or Xception [2] to get rich semantic features. Like in PSPNet [5], these features are applied to spatial pyramid pooling module to get multi-scale context information. However to do this, in [5], [7] average pooling over sub regions is used but in Deeplab v3+ [10] they have used atrous convolutions with different dilation rates and one global average pooling module. In the case if Xception is used as backbone network then, they have used atrous separable convolutions in ASPP module to increase the speed and efficiency. In the next step, these features from different pooling modules are aggregated to give as input to decoder. In decoder they have concatenated the these rich semantic features after applying bi-linear interpolation with low level features (output features of second convolution layer in backbone) from encoder. After that few more 3x3 convolutions are applied before up-sampling again to match with ground truth.

2.5. EMANet

Attention based models are extensively used in recent times in machine translation, VQA etc. The self-attention methods calculate the context coding at one position by a weighted summation of embeddings at all positions in sentences. Non-local models [16] first used self-attention as kernel applied on entire image. As the computation complexity is huge for these models, Xia Li *et al.* proposed a self-attention module based on EM algorithm in [15]. In this network, the image is first fed into a backbone architecture such as ResNet and then a iterate EM Attention unit is imposed on it to learn bases and responsibilities which represent the aggregate information over entire dataset.Once these are iteratively computed they re-estimate the original feature maps as the inner product.

Responsibility Estimation:

$$\mathbf{Z}^t = softmax(\lambda \mathbf{X}(\boldsymbol{\mu}^{t-1})^T)$$

Liklihood Maximization:

$$\boldsymbol{\mu}_k^t = \frac{z_{nk}^t \boldsymbol{x}_n}{\sum_{m=1}^N z_{mk}^t}$$

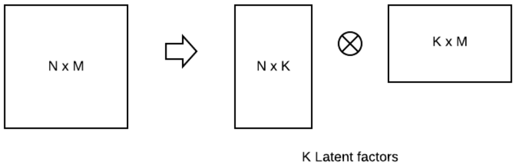


Figure 2. Matrix Decomposition

Data Restimation:

$$\widetilde{\boldsymbol{X}} = \boldsymbol{Z}^T \boldsymbol{\mu}^T$$

2.6. Factorization

Matrix Factorization is classic problem in many fields like optimization, linear algebra and machine learning etc. These problems are well studied in recommendation systems where some user-item ratings are given and filling out the rest of the matrix is the ultimate goal. There had already been an extensive research done in factorization algorithms. One standard technique, that can be applied to decompose any kind of matrix is by Singular Value Decomposition (SVD). In SVD, the original matrix A is factorized by two orthogonal matrices U, V and with a diagonal matrix of singular values $\boldsymbol{\lambda}$.

$$\boldsymbol{A} = \boldsymbol{U \lambda V^T}$$

There has been various another kind of decomposition techniques like UR, neural matrix decomposition etc. In all these methods, the common goal to achieve a latent representation of original matrix.

2.7. Attention

Ever since attention is proposed in [1], it's widely used in many research areas to gather relevant context. In computer vision tasks such as segmentation problems, to retrieve the entire spatial context over entire feature map, many kinds of attention mechanisms are used. Non-local methods [16] first used self-attention module for object detection and instance segmentation. A^2 Net [17] proposes the double attention block to distribute and gather informative global features from the entire spatio-temporal space of the images.

3. My Work

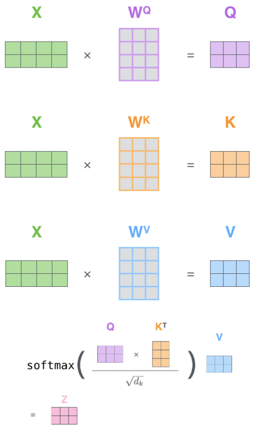


Figure 3. Self Attention

In this work, I focused on reshaping the self-attention module using factorization techniques. If one sees a self-attention block that has been used in [1] in Fig. 3, it can be seen as reconstruction of X as the weighted representation of all other embeddings/featuremaps as Z. For this re-construction, the weight matrices W, Q, V are learnt as part of network training. Likewise, in [15] also reconstruction weights which are called as bases and responsibilities are learnt separately using EM algorithm. As part of this work, I've simplified this module by just employing a decomposition module as a self-attention block and allowing the network to learn these latent representations. The complete architecture can be divided into three modules :

- The input image is fed into a backbone network (ResNet50) and extracted the feature maps with high level semantics.

- Then these feature maps X are first flattened to $C \times N$ from $C \times H \times W$. Then X' is reconstructed by doing matrix multiplication of latent vectors Y and Z which loosely represent the bases(Y) and responsibilities (Z).

  $$X' = Y * Z \approx X$$

  Unlike [15], there is no another explicit algorithm like EM or self-attention module here but just simple factorization matrices as parameters to learn.

- Finally, the reconstructed feature maps X' have low-rank representation with more weightage to features that represent plausible objects. These X' are fed to a decoder [similar to [10]] to reconstruct the gold image.

Initially, these tensors Y and Z are initialized as white noise from standard normal. As the network learns, slowly these represent the relaxed form of bases and responsibilities. The reason why they're called 'relaxed' representations is we'are not enforcing any softmax layer on weights to represent them as probabilities, so Y and Z are two decomposed units that are learnt to represent the global semantics well in entire dataset. As X' represents the low-rank representation of X for every image, I've added a reconstruction loss between X, X' with $\lambda = 0.1$ weight added to cross entropy loss.

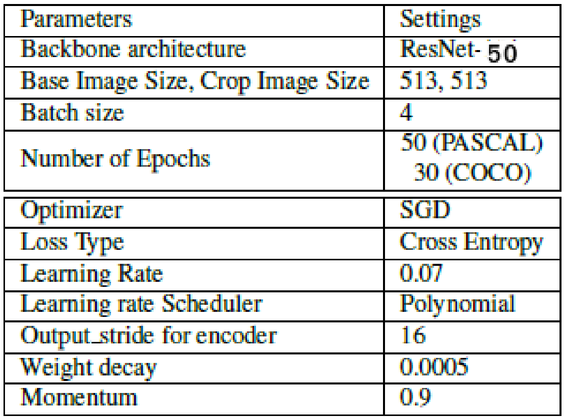
$$L = L_{ce} + \lambda L_{mse}$$

The detailed illustration of network is in Fig.4.

4. Experiments & Results

4.1. Training

The backbone network is built with pretrained ResNet-50 [8] on ImageNet[14] with output stride as 16 to minimize the computation requirement. For the first few epochs (10), Y and Z are simply learned to reconstruct the X without any reconstruction loss and after that $\lambda = 0.1$ of it is added to final CE loss. For the decoder part, the architecture is borrowed from Deeplabv3+ [10], so the X' feature-maps are up-sampled then merged with low-level features from encoder to match with ground truths. The network is trained with initial learning rate 0.07 and slowly decayed using poly LR scheduler. The step function for each batch (size=4) is simple stochastic gradient algorithm to update the gradients.There was no augmentation strategies used for input image of size 513x513 and as the output_stride parameter is set to 16, after the encoder spatial size for feature maps will be reduced to 33x33. I've used PASCAL VOC-2012 dataset to train the network for 50-epochs with cross-entropy loss criteria. The detailed setting of all hyper-parameters are listed in Fig. 5. For computation, Ive have used Google Cloud Platform Free-Tier 4CPU, 26G Memory as host machine and 1 GPU with 2096 CUDA cores as accelerator device.

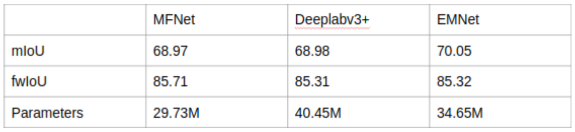


Parameters	Settings
Backbone architecture	ResNet- 50
Base Image Size, Crop Image Size	513, 513
Batch size	4
Number of Epochs	50 (PASCAL) 30 (COCO)
Optimizer	SGD
Loss Type	Cross Entropy
Learning Rate	0.07
Learning rate Scheduler	Polynomial
Output_stride for encoder	16
Weight decay	0.0005
Momentum	0.9

Figure 5. Hyper-parameters

4.2. Results

To evaluate the efficiency and accuracy of my work, I've compared the results with Deeplabv3+ [10] and EMANet [15] as these are closely related to my work. I've considered mean IoU and frequency weighted IoU over each class metrics. For PASCAL VOC 2012 validation data set, results are summarized in Fig. 6. I've also attached the mIoU and fwIoU progression graph over the epochs for all architectures in **??**, 9, 10



	MFNet	Deeplabv3+	EMNet
mIoU	68.97	68.98	70.05
fwIoU	85.71	85.31	85.32
Parameters	29.73M	40.45M	34.65M

Figure 6. Comparison of size and accuracy between different models.

As its very clear from the results, the proposed architecture (MFNet) achieved almost same mIoU and better fwIoU than state-of-the-art models with far less parameters. I believe the number of FLOPs are also substantially lesser than other two networks in comparison as they have complex module between encoder and decoder to retrieve global context.

I've done some ablation studies by increasing and decreasing the λ value but the results are better at 0.1.

A sample set of segmented images with corresponding original ground truths can be seen Fig. 7

References

[1] N. P. J. U. L. J. A. N. G. . K. Ashish Vaswani, Noam Shazeer and I. Polosukhin. Attention is all you need.

[2] F. Chollet. Xception: Deep learning with depthwise separable convolutions.

[3] J. X. H. Zhang and K. Dana. Deep ten: Texture encoding network.

[4] J. S. Z. Z. X. W. A. T. A. A. Hang Zhang, Kristin Dana. Context encoding for semantic segmentation.

[5] X. Q. X. W. J. J. Hengshuang Zhao, Jianping Shi. Pyramid scene parsing network.

[6] T. D. Jonathan Long, Evan Shelhamer. Fully convolutional networks for semantic segmentation.

[7] S. R. Kaiming He, Xiangyu Zhang and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition.

[8] S. R. J. S. Kaiming He, Xiangyu Zhang. Deep residual learning for image recognition.

[9] F. S. H. A. Liang-Chieh Chen, George Papandreou. Rethinking atrous convolution for semantic image segmentation.

[10] G. P. F. S. Liang-Chieh Chen, Yukun Zhu and H. Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation, 2018.

[11] I. K. K. M. Liang-Chieh Chen+, George Papandreou+ and A. L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs.

[12] I. K. K. M. A. L. Y. Liang-Chieh Chen+, George Papandreou+. Semantic image segmentation with deep convolutional nets and fully connected crfs.

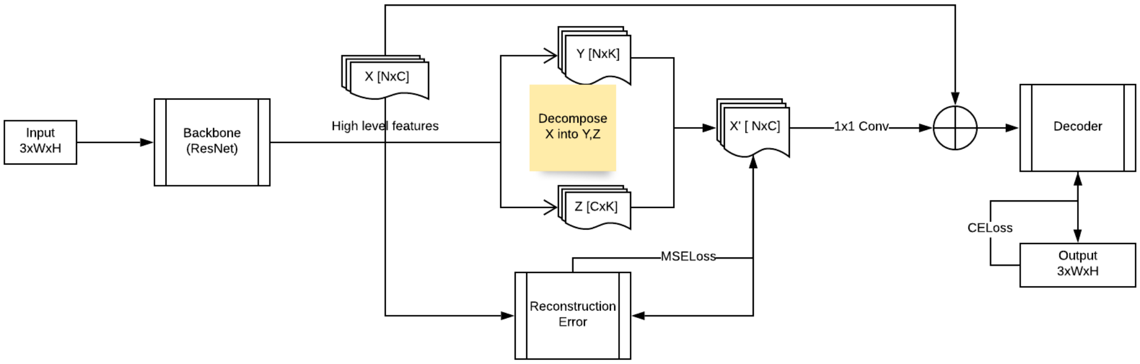


Figure 4. The self-attention module is built on simple factorization technique to learn the global context from all spatial features

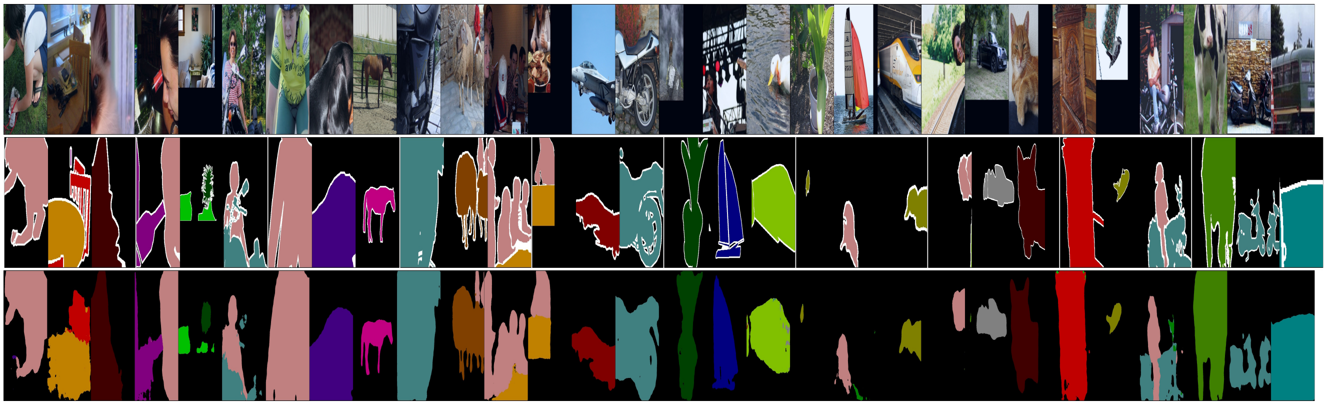


Figure 7. The original image with segmented truths and predicted in column-wise order.

[13] P. F. Olaf Ronneberger and T. Brox. U-net: Convolutional networks for biomedical image segmentation.

[14] H. S. J. K. S. S. Olga Russakovsky, Jia Deng. Imagenet large scale visual recognition challenge.

[15] Y. Y. Z. L. H. L. Xia Li, Zhisheng Zhong. Expectation-maximization attention networks for semantic segmentation.

[16] A. G. Xiaolong Wang, Ross Girshick and K. He. Non-local neural networks.

[17] J. L. S. Y. Yunpeng Chen, Yannis Kalantidis and J. Feng. A2-nets: Double attention networks.

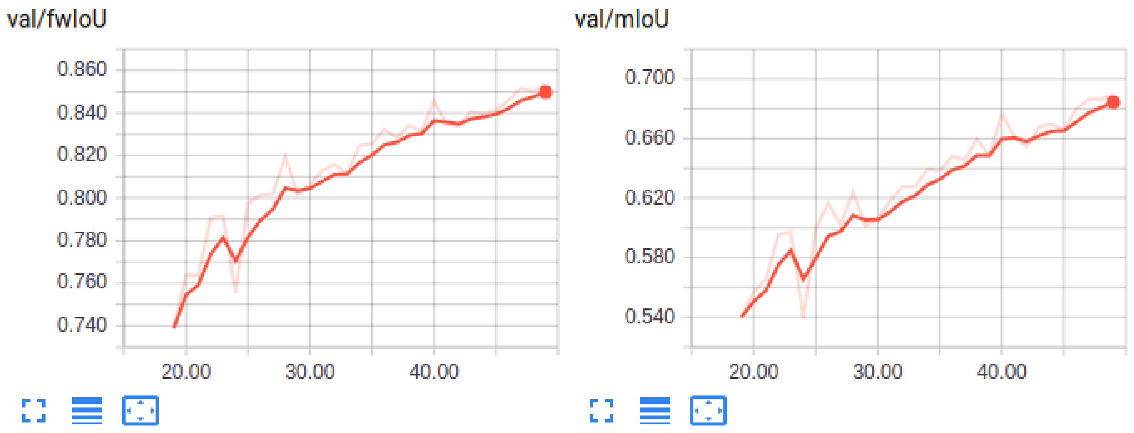


Figure 8. Progression of mIoU and fwIoU in Deeplabv3+

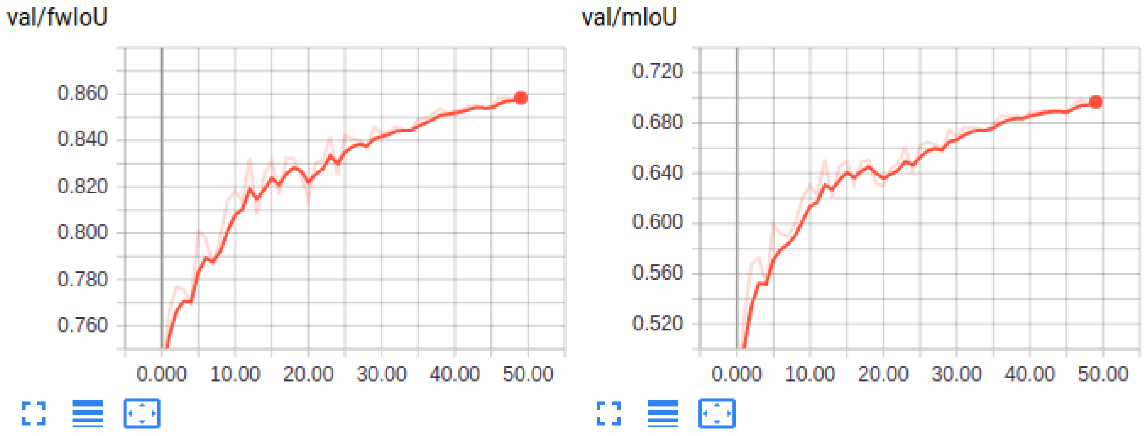


Figure 9. Progression of mIoU and fwIoU in EMANet

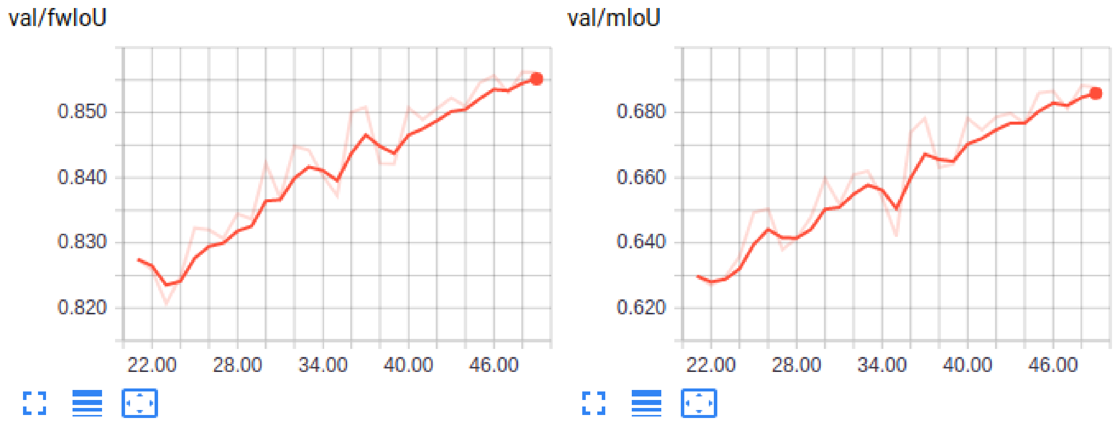


Figure 10. Progression of mIoU and fwIoU in MFNet