Semantic Image Segmentation On Indian Traffic Data

Chirag Tagadiya
Khoury College of Computer Sciences
Northeastern University
Boston, MA 02115
tagadiya.c@northeastern.edu

April 21, 2021

1 Introduction Original Paper

In this project My objective is to perform Semantic segmentation on Indian Traffic Data set. Original Paper I Implemented are

- U-Net: Convolutional Networks for Biomedical Image Segmentation
- Attention-guided Chained Context Aggregation for Semantic Segmentation

2 Context Information

Image segmentation is a dense prediction task, as we want to classify every pixel in the image as a corresponding class of what it represents in the image. It has very interesting applications in computer vision domains like Autonomous vehicles, Medical Image Analysis, Snap chat/ Instagram filters, and many more. In Image Segmentation, we are interested in two main task

1. What is in the image? (classification)
2. where in the image it is located? (localization)

In this project, I have used the Indian Traffic dataset. The dataset consists of images obtained from a front-facing camera attached to a car. The car was driven around Hyderabad, Bangalore cities, and their outskirts. The dataset consists of 41 different kinds of objects, and these objects further sub-grouped into 21 different classes. I have used 4000 images in which 3200 images were used for training and 800 images were used for testing. sample classes: Derivable, Non-Derivable, Living things, vehicles, roadside objects, sky, etc

3 Related Work and My Approach

3.1 Related Work

A naive approach to solve the image segmentation task is to use series of convolution neural networks with the same padding to preserve input dimension and then predict the output segmentation mask, this approach is not good because it is computationally very expensive to preserve the high resolution throughout the network. to remedy this problem I choose Encoder-Decoder base architecture for my project. In Encoder-Decoder architecture, Encoder down-samples the input image via a stack of convolution neural network (increasing feature map in each layer) to learn high-level semantic information, decoder then slowly improve the resolution to get back the same resolution as an input image to generate segmentation mask. In Encoder-Decoder architecture, the Earlier layer learns low-level information, while the later layer learns task-specific information / higher-level feature. In Encoder, we gradually reduce the resolution but to maintain expressiveness increase the feature maps.

3.2 U-net

U-net was originally invented for segmentation tasks in biomedical images, but this architecture is widely used for other datasets. In the Encoder part of the architecture I used a pre-trained ResNet-34 classification network trained on Image net dataset, basically in the encoder part, you can use any of the pre-trained models on Image Net like VGG, Inception, Alex. In the Encoder part we down-sample the spatial resolution of the image, and increasing feature maps, by doing this we will learn features that are highly discriminating in nature which is important for the image segmentation task (**classification**) . Now in a decoder, we need to up-samples the feature so that we can generate the higher resolution images, I tried to stack the discriminative feature (lower resolution) learned in encoder architecture (**localization**) with features learned from the previous layer of the decoder to generate higher resolution image, and gradually recovered the original resolution. The problem with vanilla Encoder-Decoder (U-net) architecture is that due to continuous down-sampling, we lose the spatial information that results in poor object delineation and small spurious region. Simple U-net is not utilizing the information learned in Encoded layers effectively.

3.3 Chained Context Aggregation for Semantic Segmentation (CANET)

To address the issue with U-net, I tried to implement other encoder-decoder-based architecture which improves the segmentation task by aggregating multiple-scale context information. This architecture is known as Chained Context Aggregation Network (CANET). The encoder part of the network is series of 4 dilated ResNet Convolution layers, in the first layer we decrease the size by 2, then by 4, and for the next 3 layers by 8.

After Encoder I have implemented the Context Aggregation Module (CAM) which consists of context flow and global flow. Global flow uses the encoded feature learned from the backbone network to get a global receptive field, which is really important to understand the overall global context of the image, this reduced the error of classification for similar objects. Global features is obtained by applying global average pooling on the shared encoded features. Inputs of the context flow are shared features from the encoder network and features we get from upper information flow. This flow aggregates the information of different spatial scales by using convolution and pooling layers, this can eliminate the aliasing effect and will increase the receptive field. In the end, CAM combines the features of various spatial scales learned from global flow and context flow using the residual connection, this will further enhance the context information. Then input will pass through the Feature selection module which assigns different attention weights to different parts of the input.

up-sampled features from feature selection module, which represents higher-level semantic features. initial layer in encoder network represents lower level features. The decoder will combine the feature from the C1 layer from encoded with feature selection input and then apply series of up-sampling convolution to create the final output which has the same resolution as input.

4 Experiment, Loss Function, Results

4.1 Loss Function

4.1.1 Pixel wise cross entropy loss

$$\sum_{i=1}^{M} y_{true} \log(y_{pred})$$

This loss treats each pixel individually and compares the depth-wise true label with predicted one hot encoder class label, then it averages over all pixel, If we have a class unbalanced problem, this loss will be

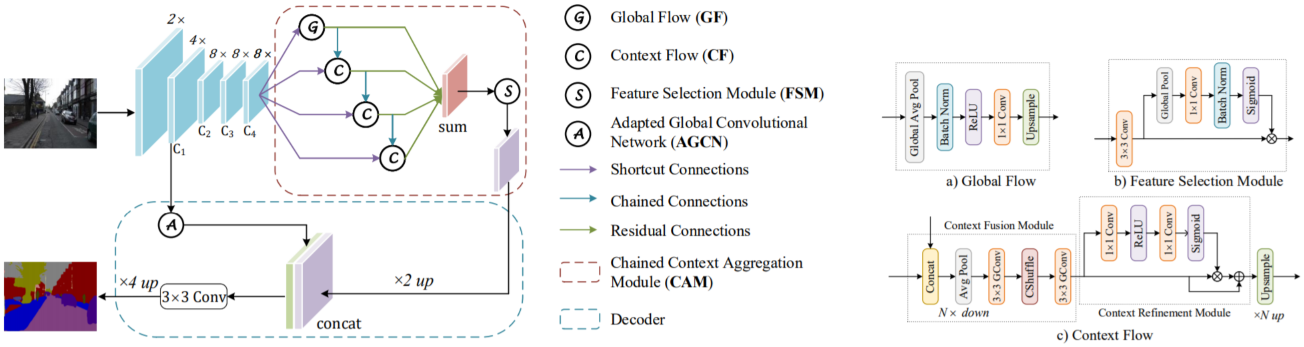

Figure 1: CANET

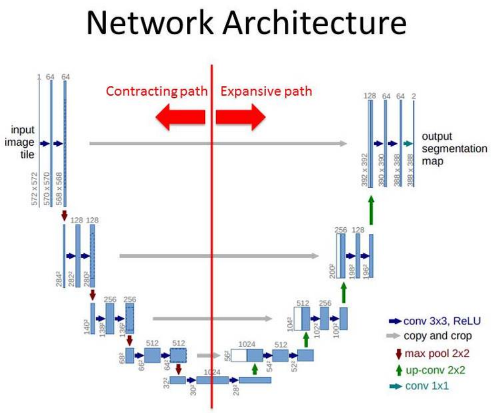

Figure 2: U-net

dominated by the prevalent class, which is the background pixel in most cases. M is the total number of classes in image, we are calculating the above loss for each pixel in each channel and then aggregating the result.

4.1.2 Dice Coefficient

$$dice = \frac{TP + TP}{TP + TP + FP + TN}$$

dice coefficient is very similar to F1 score which is nothing but the harmonic mean of precision and recall, Harmonic mean is biased towards the lowest value between precision and recall, penalize the worst score while optimizing (1 - dice). This is useful in unbalanced class problem.

4.2 My Experiment and Results

I have trained both the network using two different loss function. for both, the network Dice loss has higher validation IOU (Intersection Over Union) score than the IOU score for pixel-wise cross-entropy loss. This is an expected result as we have lots of background pixel and a few foreground class pixel in each channel. CANET accuracy is slightly better than U-NET architecture. I have not used any augmentation technique during the training. This is one more experiment I can try in a future iteration. I used Keras Tensorflow for implementing the model, used Adam Optimizer, due to limited GPU in my system, I have not trained the

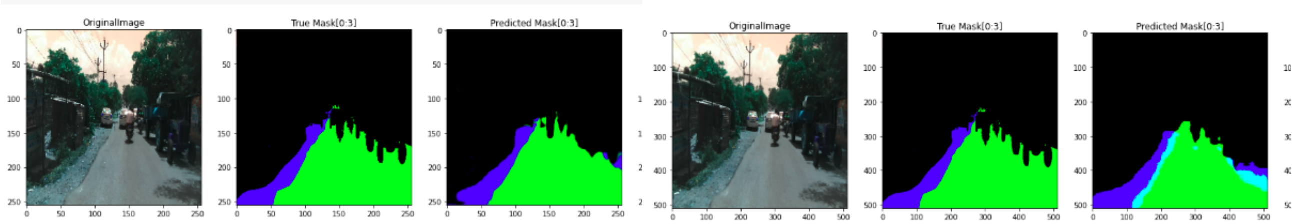

Figure 3: left Dice Loss[original image, true mask, predicted mask], right CCE loss for CANET

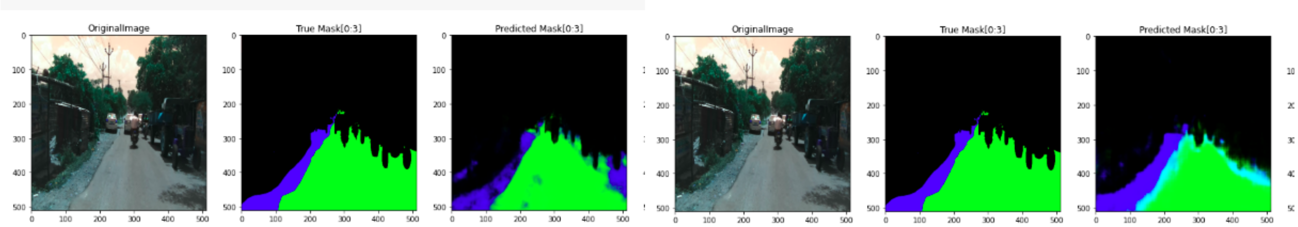

Figure 4: left Dice loss [original image, true mask, predicted mask], right CCE loss for U-net

network more than 25 epochs. Plot shows the output of first 3 channels of predicted mask, and true mask for both the network, I have plotted the results for Dice loss and Cross Entropy loss.

Model	CCE IOU	Dice IOU
U-net	.36	.49
CANET	.47	.57

5 Learning and Future Work

- Goal of this project was to familiar my self with computer vision problems, image preprocessing, building complex architectures and implementing research papers from scratch. I personally believe that I achieved my goal by implementing this task
- I want to extend this project by implementing other similar network, I want to compare different architecture (SegNet, PSPNet, MobileNet) performance on different loss function (weighted cross entropy loss, focal loss etc.) and diverse dataset (satellite dataset, Biomedical dataset).

References

[1] Olaf Ronneberger and Philipp Fischer and Thomas Brox U-Net: Convolutional Networks for Biomedical Image Segmentation networks with existing applications. *arXiv 1505.04597*, 2015.

[2] Girish Varma, Anbumani Subramanian, Anoop Namboodiri, Manmohan Chandraker , C V Jawahar IDD: A Dataset for Exploring Problems of Autonomous Navigation in Unconstrained Environments *IEEE Winter Conf. on Applications of Computer Vision https://idd.insaan.iiit.ac.in/media/publications/idd-650.pdf*, 2019.

[3] Tang, Quan and Liu, Fagui and Jiang, Jun and Zhang, Yu Attention-guided Chained Context Aggregation for Semantic Segmentation In *arXiv preprint arXiv:2002.12041*, 2020.

[4] Jadon, Shruti. A survey of loss functions for semantic segmentation In *2020 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)*, pages 1–7. IEEE, 2020.

[5] JEREMY JORDAN : An overview of semantic image segmentation.
`https://www.jeremyjordan.me/semantic-segmentation/`