Semantic 2D-Segmentation using VGG16-FCN

Adithya Suresh, Prajval Vaskar
Department of Automotive Engineering, Clemson University
International Center for Automotive Research (CU-ICAR)
2020 Spring Course Report: AuE 8930 Machine Perception and Intelligence, Instructor: Dr. Bing Li

Abstract

In the world of autonomous driving, perception plays an important role for autonomous vehicle to visualize the surrounding so it can take the necessary actions. The vehicle should be capable sense-think- act paradigm. Sensing the surrounding is not enough for autonomous vehicle unless data is interpretable for vehicle to act on it. On the collected sensing data, various deep learning tools can be applied such as object detection, segmentation etc. In this paper, we will discuss the semantic segmentation, its necessities in autonomous vehicle, its challenges and results we got from implementing on two different datasets. Lots of benchmark datasets are released for researchers to verify their algorithms. Semantic segmentation has been studied for many years. Since the emergence of Deep Neural Network (DNN), segmentation has made a tremendous progress. In this project, we will elaborate on the importance of Deep Learning in various autonomous vehicle and Advanced Driver Assisting Systems (ADAS) features that would help in making the vehicle futuristic and sophisticated.

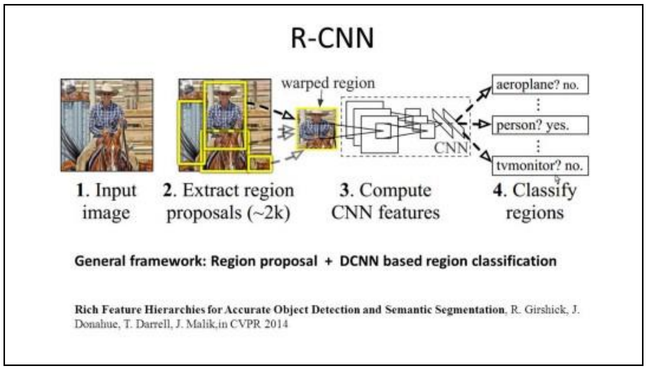
1. Introduction

Few years ago, semantic segmentation task was a complex task in computer vision. Now due to deep learning it is easy to do semantic segmentation. Semantic segmentation is different from image classification. In image classification, it will only classify objects that it has specific labels like car pedestrian, road etc. where in image segmentation algorithm will also segment unknown objects. Image classification is the task of classifying what appears in an image into one out of set of predefined classes whereas sematic segmentation is task of classifying each pixel in an image into one out of set of predefined class. Before the development of deep learning in computer vision, other machine learning approaches such as random forest were used to do segmentation.

There are three type of semantic segmentation that can be done using deep learning architecture.

1.1. Region-Based Sematic Segmentation

The region-based methods generally follow the "segmentation using recognition" pipeline, which first extracts free-form regions from an image and describes them, followed by region-based classification. At test time, the region-based predictions are transformed to pixel predictions, usually by labeling a pixel according to the highest scoring region that contains it.



Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation, R. Girshick, J. Donahue, T. Darrell, J. Malik,in CVPR 2014

Figure 1 Framework of R-CNN

R-CNN (Regions with CNN feature) is one representative work for the region-based methods. It performs the semantic segmentation based on the object detection results. To be specific, R-CNN first utilizes selective search to extract a large quantity of object proposals and then computes CNN features for each of them. Finally, it classifies each region using the class specific linear SVMs. Compared with traditional CNN structures which are mainly intended for image classification, R-CNN can address more complicated tasks, such as object detection and image segmentation, and it even becomes one important basis for both fields. Moreover, R-CNN can be built on top of any CNN benchmark structures, such as AlexNet, VGG, GoogLeNet, and ResNet [1].

However, there are several drawbacks of this method,

- The feature is not compatible with the segmentation task.
- The feature does not contain enough spatial information

for precise boundary generation.

- Generating segment-based proposals takes time and would greatly affect the final performance.

1.2. Fully Convolutional Network Based Semantic Segmentation.

The original Fully Convolutional Network (FCN) learns a mapping from pixels to pixels, without extracting the region proposals. The FCN network pipeline is an extension of the classical CNN. The main idea is to make the classical CNN take as input arbitrary-sized images. The restriction of CNNs to accept and produce labels only for specific sized inputs comes from the fully connected layers which are fixed. Contrary to them, FCNs only have convolutional and pooling layers which give them the ability to make predictions on arbitrary-sized inputs.

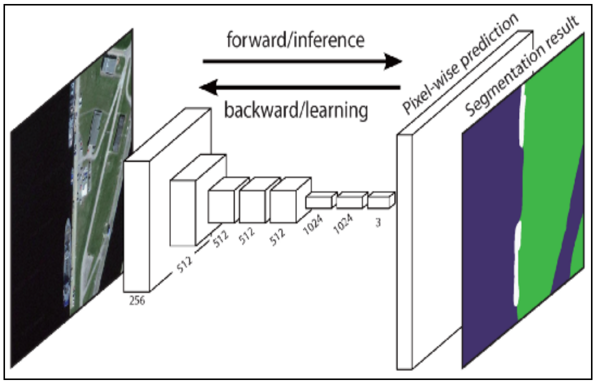


Figure 2 Framework of FCN

One issue in this specific FCN is that by propagating through several alternated convolutional and pooling layers, the resolution of the output feature maps is down sampled. Therefore, the direct predictions of FCN are typically in low resolution, resulting in relatively fuzzy object boundaries. A variety of more advanced FCN-based approaches have been proposed to address this issue, including SegNet, DeepLab-CRF, and Dilated Convolutions [1].

1.3. Weakly Supervised Semantic Segmentation.

Most of the relevant methods in semantic segmentation rely on many images with pixel-wise segmentation masks. However, manually annotating these masks is quite time-consuming, frustrating and commercially expensive. Therefore, some weakly supervised methods have recently been proposed, which are dedicated to fulfilling the semantic segmentation by utilizing annotated bounding boxes [1].

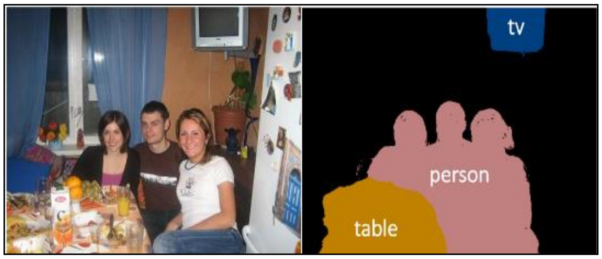


Figure 3 Weakly Supervised Semantic Segmentation

2. Datasets

PASCAL VOC 2012 [10] has been the most tested upon benchmark for semantic segmentation, although that is bound to change since state of the art has reached 89.0 mIOU on its test set. MS COCO is another large-scale, popular segmentation dataset. Other significant datasets include SiftFlow, NYUDv2, Stanford Background. DAVIS is a significant video object segmentation Dataset [2].

2.1. MS COCO

MS COCO is a popular and very challenging large-scale detection data, which includes pixel segmentations. State of the art mIoU on MS COCO is around 48.0. The dataset contains high-resolution images and features 80 object categories and 91 stuff categories. The semantic segmentation challenge provides 82000 train images, 40500 validation images, and the evaluation is carried out on subsets of 80000 test images. MS COCO has received continued attention due to its quality annotations and large size. Several detection challenges are hosted at ECCV each year [2].

2.2. PASCAL VOC 2012

The dataset is composed of images from the image hosting website Flickr. The images are hand annotated.
There are several detection related challenges for the PASCAL VOC 2012, one of which is a semantic segmentation challenge. There are 21 classes (airplane, bicycle, background). The public dataset has 1464 training and 1449 validation images. The test set is privately held. Although formally no new competition has been held since 2012, algorithms continue to be evaluated on the 2012 challenge for segmentation (for example DeepLab v3, top of the leaderboard, is a 2018 submission) [2].

3. Fully Convolutional Neural Network

Different from the classical CNN, after using the convolutional layers to obtain a fixed-length eigenvector for classification, the FCN can accept input images of any size, and use the deconvolution layer to feature the feature map of the last convolutional layer. Perform up sampling to restore it to the same size of the input image, so that a prediction can be generated for each pixel, while retaining the spatial information in the original input image, and finally pixel-by-pixel classification on the up sampled feature map. Each layer of data in the convolutional network is a three-dimensional array of h*w*d, where h and w are spatial dimensions and d is the feature or channel dimension. The first layer is an image with a pixel size of h*w and a color channel number of d. The coordinates in the upper layer correspond to the coordinates of their connections in the image and are called the receiving domain. Convolutional networks are based on translational distortion, its basic components (convolution, pooling, and excitation functions) act on the local input domain and rely only on relative spatial coordinates. Typical identification networks, including LeNet, AlexNet, etc., use a fixed-size input to produce a non-spatial output. The fully connected layers of these networks have a certain number of bits and discard the spatial coordinates. However, these fully connected layers are also considered to be nuclear convolutions covering all input domains. They need to be added to a full convolutional network that can be input in any size and output a classification map. This conversion is to convert the fully connected layer in the convolutional neural network into a convolutional layer [3].

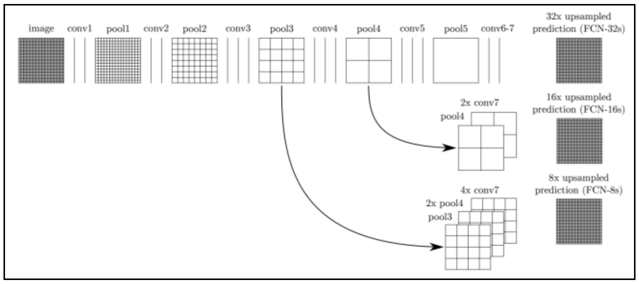


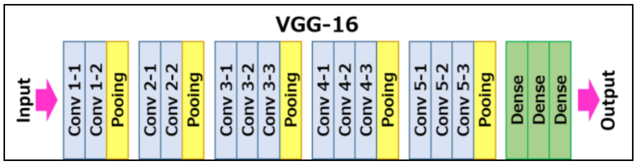
image
conv1
pool1
conv2
pool2
conv3
pool3
conv4
pool4
conv5
pool5
conv6-7
32x upsampled prediction (FCN-32s)
2x conv7
pool4
16x upsampled prediction (FCN-16s)
4x conv7
2x pool4
pool3
8x upsampled prediction (FCN-8s)

Figure 4 Detailed Structure Fully Convolutional Network

After the input image is processed by multiple convolution layers and pooling layers, the feature map is gradually reduced, and the resolution is gradually reduced. The FCN network contains five convolutional layers and five pooling layers, and the resolution of the images is reduced by 2, 4, 8, 16, and 32 times. After 32-times scaling, the perceived field of the feature map is 32×32 (that is, each output pixel corresponds to a 32×32 size image block of the original image). In order to achieve the end-to-end result of each pixel, the FCN restores the resulting feature map to the resolution of the original image by up sampling/deconvolution. Deconvolution and convolution are similar, they are operations of multiplication and multiplication, but the latter is many-to-one, the former is one to-many. The forward and backward propagation of the deconvolution can be propagated by reversing the convolution. Because the results obtained by directly up sampling the results after convolution will be rough, FCN uses a skip layer to combine the coarse high-level information with the fine underlying information to optimize the results. The specific structure is shown in Figure 2. Since the size of the feature map generated by the non-unique stride convolutional layer and the pooled layer has some downward rounding operations, the final feature map size is not strictly a multiple relationship with the original image size, so will have a crop layer that crop the result of the up sampling to exactly the same size as the input image[3].

4. VGG16 CNN

VGG16 is a convolutional neural network model proposed by K. Simonyan and A. Zisserman from the University of Oxford in the paper "Very Deep Convolutional Networks for Large-Scale Image Recognition". The model achieves 92.7% top-5 test accuracy in ImageNet, which is a dataset of over 14 million images belonging to 1000 classes. It was one of the famous models submitted to ILSVRC-2014. It makes the improvement over AlexNet by replacing large kernel-sized filters (11 and 5 in the first and second convolutional layer, respectively) with multiple 3×3 kernel-sized filters one after another [4].


VGG-16

Input → Conv 1-1, Conv 1-2, Pooling, Conv 2-1, Conv 2-2, Pooling, Conv 3-1, Conv 3-2, Conv 3-3, Pooling, Conv 4-1, Conv 4-2, Conv 4-3, Pooling, Conv 5-1, Conv 5-2, Conv 5-3, Pooling, Dense, Dense, Dense → Output

Figure 5 Layers in VGG16

The architecture depicted below is VGG16.

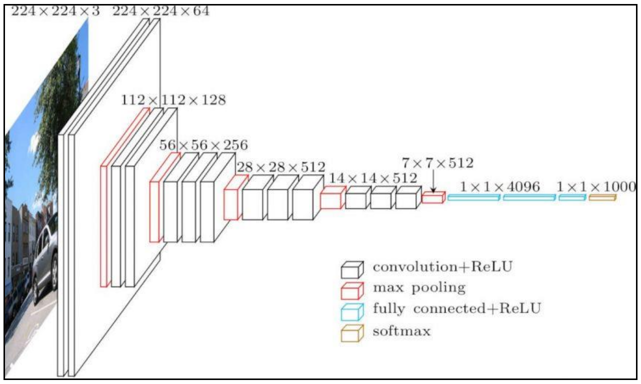


Figure 6 Architecture of VGG16

The input to cov1 layer is of fixed size 224 x 224 RGB image. The image is passed through a stack of convolutional (conv.) layers, where the filters were used with a very small receptive field: 3×3 (which is the smallest size to capture the notion of left/right, up/down, center). In one of the configurations, it also utilizes 1×1 convolution filter, which is a linear transformation of the input channels (followed by non-linearity). The convolution stride is fixed to 1 pixel; the spatial padding of conv. layer input is such that the spatial resolution is preserved after convolution, i.e. the padding is 1-pixel for 3×3 conv. layers. Spatial pooling is carried out by five max-pooling layers, which follow some of the conv. layers (not all the conv. layers are followed by max-pooling). Max-pooling is performed over a 2×2-pixel window, with stride 2 [4].

5. Implementation of VGG16 to FCN

VGG16 is mainly used for classification task due to its fully connected layers. For segmentation task, spatial information to be stored to make pixel wise classification. FCN allows this by making all layers of VGG to convolutional layers. Fully convolutional indicates that the neural network is composed of convolutional layers without any fully connected layers usually found at the end of the network. Fully Convolutional Networks for Semantic Segmentation motivates the use of fully convolutional networks by "creating convolutions" popular CNN architectures e.g. VGG can also be viewed as FCN. The model used for semantic segmentation is FCN8. It duplicates VGG16 net by discarding the final classifier layer and convert all fully connected layers to convolutions. Fully Convolutional Networks for Semantic Segmentation appends a 1 x 1 convolution with channel dimension the same as the number of segmentation classes (in our case, this is 12) to predict scores at each of the coarse output locations, followed by up-sampling deconvolution layers which brings back low resolution image to the output image size. These up-sampling layers do not have weights/parameters, so the model is not flexible. Instead, FCN8 uses up sampling procedure called backwards convolution (sometimes called deconvolution) with some output stride. This method simply reverses the forward and backward passes of convolution and implemented in Keras's Conv2DTranspose.The up-sampling layer brings low resolution image to high resolution. There are various up sampling methods. We have used VGG16 weights to avoid training the network from scratch [5].

6. Dataset used

6.1. Indian Driving Dataset

While several datasets for autonomous navigation have become available in recent years, they tend to focus on structured driving environments. This usually corresponds to well-delineated infrastructure such as lanes, a small number of well-defined categories for traffic participants, low variation in object or background appearance and strict adherence to traffic rules. IDD, a novel dataset for road scene understanding in unstructured environments where the above assumptions are largely not satisfied. It consists of 10,004 images, finely annotated with 34 classes collected from 182 drive sequences on Indian roads. The label set is expanded in comparison to popular benchmarks such as Cityscapes, to account for new classes. It also reflects label distributions of road scenes significantly different from existing datasets, with most classes displaying greater within-class diversity. Consistent with real driving behaviors, it also identifies new classes such as drivable areas besides the road [6].

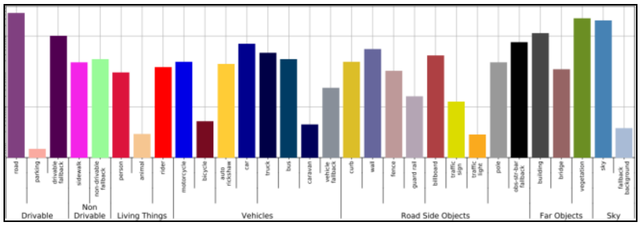


Figure 7 Labels used in IDD

The following information is shown here in the above table: (i) pixel counts of individual labels on the y-axis (ii) four-level label hierarchy used by the dataset at the bottom, (iv) the color legend for the predicted and ground truth masks shown in the paper is used for the corresponding bars [6].

6.2. France Dataset

The dataset we used is mini version of the France dataset which contains 101 annotations for testing, 367 annotation for training, 101 images for testing and 367 images for training. The dataset has 12 classes or labels for semantic segmentation like road, car, sky, trees, pedestrian etc.

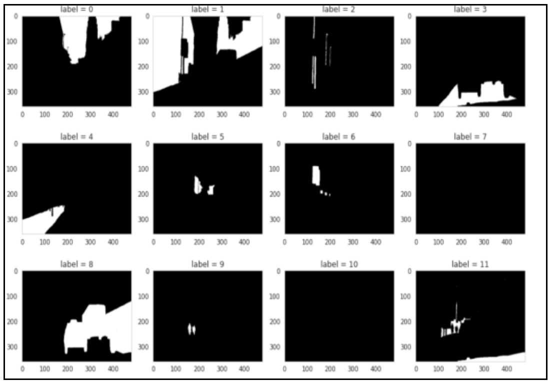


Figure 8 Visualization of labels in dataset

7. Experimental Environment:

The experiment uses the TensorFlow pipeline and the CUDA-GPU acceleration scheme under the Windows operating system. The computation required for running the VGG-16 network was high, and various cloud computing platforms have been referred to for receiving good results. Initially, the experiment used the CUDA-GPU acceleration scheme under the Windows Operating System. The graphics card used NVIDIA's GTX 1650Ti (11G memory) for GPU acceleration. This proved be relatively less power intensive in terms of computational complexity and the subsequent platform called the Palmetto Cluster by Clemson University was approached. The GPU cores utilized was 2 cores with V100 GPU. This helped with training the network, yet the computational complexity was faced and finally we went with the Google's cloud computing platform called the Colab Notebooks. Using this method, we were able to train a network with relatively fast speed. The Google Colab uses Tesla K80 GPU with 12 Gb RAM and a job time allocation for 12 hours.

8. Experimental data:

The experimental data uses image data of high-resolution urban environment, and the spatial resolution of the image is 2 meters. The images include urban driving condition targets: vehicles, drivable spaces, curbs road, and building. The number of data is 500, and 367 samples are randomly selected from the data set as training samples, and the remaining 101 are used as test samples. The training samples are not duplicated with the test samples, and the size of all images is set to 224 × 224 pixels.

9. Analysis of Results:

After training, the optimal network model is obtained. In order to quantitatively evaluate the accuracy of segmentation results, this paper uses Per Pixel Accuracy (PPA), Mean Class Accuracy (MCA) and Mean Interview Over Union (MIOU) as evaluation criteria. Assuming that there are $n_{c,}$ feature types in the image data to be processed, n_{ij} is the number of pixels of class i predicted to belongs to class j, and the $=\sum_{i\ j\ ij} t\ n$ is the total number of pixels of class i. Then there are:

Average accuracy per pixel (PPA):

$$\frac{\Sigma_i n_{ii}}{\Sigma_i t_i}$$

Average Class Accuracy (MCA):

$$\frac{1}{n_c}\Sigma_i \frac{n_{ii}}{t_i}$$

Average IOU (MIU):

$$\frac{1}{n_c}\Sigma_i \frac{n_i}{t_i + \Sigma_j n_{ji} - n_{ii}}$$

Applying the above methodology in our experimental data while training the network using VGG-16 architecture as base model and FCN-8 as segmentation model, the values for accuracy, losses are as follows:

Epochs	Training Accuracy	Training Losses	Validation Accuracy	Validation Losses
1	0.0849	2.5930	0.0856	2.4851
200	0.8805	0.4250	0.8594	0.4945

IoU for each class is as follows:

Sr No	Class	IoU
1	00	0.871
2	01	0.772
3	02	0.000
4	03	0.920
5	04	0.494
6	05	0.668
7	06	0.018
8	07	0.241
9	08	0.663
10	09	0.007
11	10	0.008
12	11	0.387

Comparing the above experimental results can lead to the following conclusions:

a) It can be seen from the segmentation results that the method of this paper can effectively segment all kinds of ground objects, and the segmentation results are accurate

b) It can be seen from the comparison in above table that the segmentation accuracy of the proposed method is higher than that of other methods in the evaluation criteria for different segmentation classes in a particular dataset and MIOU, which can significantly improve the automatic segmentation effect of urban sensing images.

10. Visualization of results

10.1. Indian Driving Dataset

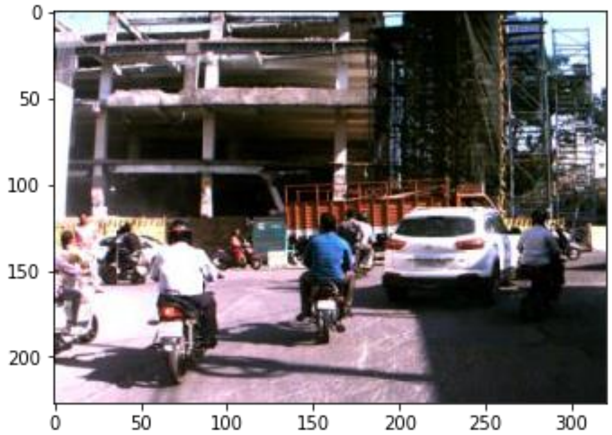


Figure 9 Visualization of image

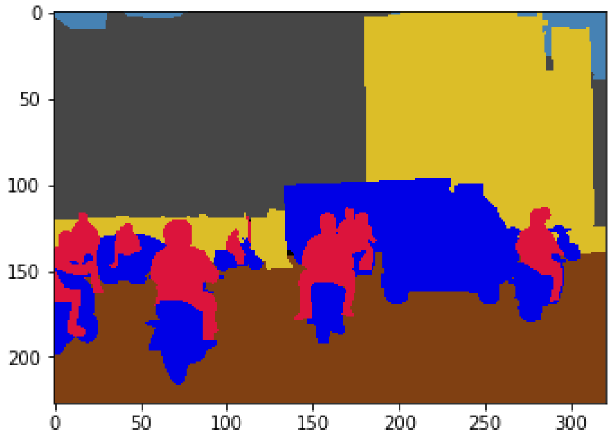


Figure 10 Visualization of image using given color coding

Figure 11 Results after semantic segmentation

10.2. France Dataset

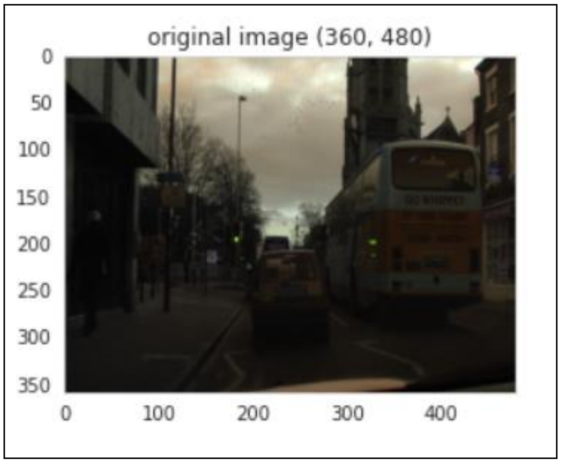


Figure 12 Visualization of Original Image

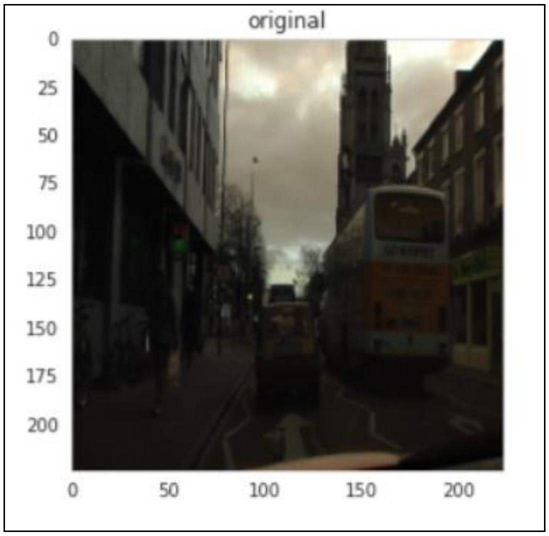


Figure 13 Visualization of Resized image

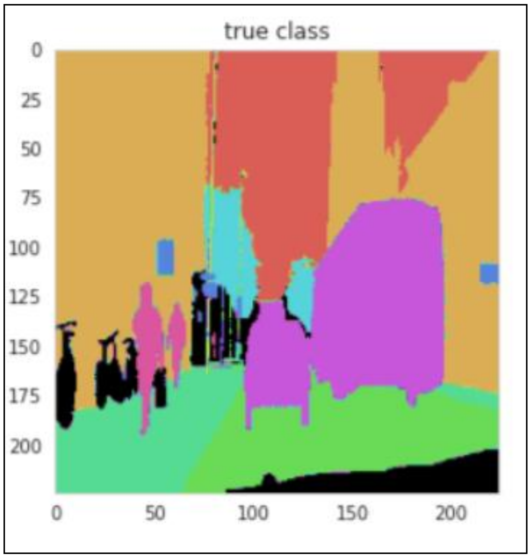


Figure 14 Visualization of image using color coding

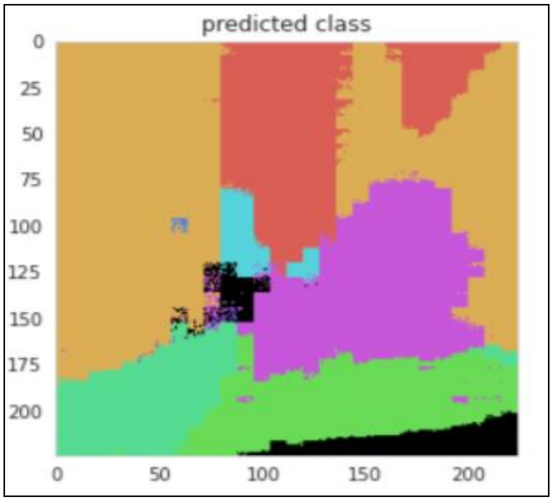


Figure 15 Visualization of image after semantic segmentation using our trained network

11. Conclusion

In this paper, the FCN network is applied to the semantic segmentation of high-resolution remote sensing of vehicles and free space data, and the matrix expansion technique is combined to optimize the convolution operation and improve the computational efficiency. The classification of road, vehicles, buildings, drivable spaces in urban sensing images are segmented. It shows that FCN can achieve automatic segmentation of urban sensing images. On the other hand, the semantic segmentation of the Indian Driving Dataset did not result in accurate result because of the presence of the color channel in the dataset which had 256 channels in one image. These segmentation classes channels resulted in a noise which gave inaccurate results with the accuracy of 0.1058 in training set. Thus, the segmentation was performed on the France dataset to get an accurate result with the accuracy 0.8805. Further explanation that the deep learning network can extract feature features better and can more accurately mine the spatial distribution law of high-resolution urban sensing image data from massive urban sensing data and improve segmentation accuracy and efficiency.

References

1. J. Le, "How to do Semantic Segmentation using Deep learning," AI & Machine Learning Blog, 23-Oct-2019. [Online]. Available: https://nanonets.com/blog/how-to-do-semantic-segmentation-using-deep-learning/. [Accessed: 30-Apr-2020]

2. "Semantic Segmentation, Urban Navigation, and Research..." [Online]. Available: https://www.cs.princeton.edu/courses/archive/spring18/cos598B/public/projects/LiteratureReview/COS598B_spr2018_SemanticSegmentationNavigation.pdf. [Accessed: 30-Apr-2020].

3. X. Fu and H. Qu, "Research on Semantic Segmentation of High-resolution Remote Sensing Image Based on Full Convolutional Neural Network," 2018 12th International Symposium on Antennas, Propagation and EM Theory (ISAPE), 2018.

4. "VGG16 - Convolutional Network for Classification and Detection," VGG16 - Convolutional Network for Classification and Detection, 21-Nov-2018. [Online]. Available: https://neurohive.io/en/popular-networks/vgg16/. [Accessed: 30-Apr-2020].

5. "Learn about Fully Convolutional Networks for semantic ..." [Online]. Available: https://fairyonice.github.io/Learn-about-Fully-Convolutional-Networks-for-semantic-segmentation.html. [Accessed: 30-Apr-2020].

6. G. Varma, A. Subramanian, A. Namboodiri, M. Chandraker, and C. Jawahar, "IDD: A Dataset for Exploring Problems of Autonomous Navigation in Unconstrained Environments," 2019 IEEE Winter Conference on Applications of Computer Vision (WACV), 2019.