Segmentation of High Resolution Aerial Images

By

Maher Thakkar
18BCE104
&
Jaineet Shah
18BCE083

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
Ahmedabad 382481

CERTIFICATE

This is to certify that the minor project entitled “Segmentation of High Resolution Aerial Images” submitted by Maher Thakkar (18BCE104) , towards the partial fulfillment of the requirements for the degree of Bachelor of Technology in Computer Science and Engineering of Nirma University is the record of work carried out by him/her under my supervision and guidance. In my opinion, the submitted work has reached a level required for being accepted for examination.

ANITHA MODI
Assistant Professor
Computer Science and Engineering Dept.,
Institute of Technology,
Nirma University,
Ahmedabad

Dr. Madhuri Bhavsar,
Professor and HOD,
Computer Science and Engineering Dept.,
Institute of Technology,
Nirma University,
Ahmedabad

ACKNOWLEDGEMENT

We would like to thank the Institute of Technology, Nirma University for giving us the opportunity to develop a project in our desired domain. We would also like to extend our gratitude to Prof. Anitha Modi (Institute of Technology, Nirma University) for guiding us throughout the duration of the semester and motivating us to improve our procedure. We thank Prof. Tejal Upadhyay (Institute of Technology, Nirma University) for providing us with insights on our implementation and presentation reviews.

ABSTRACT

In computer vision and image processing, pixel-wise picture segmentation is a difficult and time-consuming procedure. Building segmentation from aerial (satellite/drone) photos is the topic of this blog. The availability of high-resolution remote sensing data has opened the door to new applications, such as more detailed per-pixel classification of specific objects. Segmentation and categorization of pictures have become much more efficient and intelligent because of the usage of Convolution Neural Networks (CNN). We used photographs with a very high quality and size in this paper; as a result, the images were cropped and their sizes were lowered so that processing could be done on a moderately powerful computer with standard specs. Convolutional networks are sophisticated visual models that produce feature hierarchies. We show that convolutional networks can outperform the state-of-the-art in semantic segmentation when trained end-to-end, pixel-by-pixel. Our key insight is to create "totally convolutional" networks that can take any size input and produce output of the same size with efficient inference and learning.

CONTENTS

Certificate		3
Acknowledgement		4
Abstract		5
List of figures		iv
List of tables		v
Chapter 1	**Introduction**	7
	1.1 **General**	7
	1.2 **Topic title**	7
	1.3 **Objectives**	7
	1.4 **Problem Statement**	
Chapter 2	**Literature Survey**	8
Chapter 3	**Methodology**	9
	3.1 Dataset	9
	3.2 VGG-16 Encoder and FCN8 Decoder	9
	3.3 U-Net	12
	3.4 SegNet	13
Chapter 4	**Result Analysis**	15
Chapter 5	**Conclusion**	17
	References	17

CHAPTER 1 Introduction

1.1 General

Semantic image segmentation is a fundamental task in computer vision. It predicts dense labels for all pixels in the image, and is regarded as a very important task that can help deep understanding of scene, objects, and human[3]. Development of recent deep convolutional neural networks (CNNs) makes remarkable progress on semantic segmentation. The effectiveness of these networks largely depends on the sophisticated model design regarding depth and width, which has to involve many operations and parameters. CNN-based semantic segmentation mainly exploits fully convolutional networks (FCNs). It is common wisdom now that increase of result accuracy almost means more operations, especially for pixel-level prediction tasks like semantic segmentation.

1.2 Topic Title

The title is “Segmentation of High Resolution Aerial Images”. The categorization of Aerial Images is a fantastic way to study a wide range of land cover in remotely sensed aerial pictures.

1.3 Objective

The objective of the study is to compare different models used to perform classification on aerial images given in the Aerial Semantic Segmentation Drone Dataset gathered by Graz University of Technology.

1.4 Problem Statement

We have made 3 models- 1) VGG-16 Encoder and FCN8 Decoder 2)U-NET and 3)SegNet for Segmentation of High Resolution Images. All the 3 models will work on the Aerial Semantic Segmentation Drone Dataset. We will be comparing these 3 models to determine which model will have the most accurate and efficient result for the Segmentation of High Resolution Images.

CHAPTER 2 Literature Review

In the last few years neural networks, which had fallen out of favour in machine learning for some time, have made a spectacular return. Driven by a number of methodological advances, but especially by the availability of much larger image databases and fast computers, deep learning methods, in particular CNNs, have outperformed all competing methods on several visual learning tasks. With deep learning, the division into feature ex- traction, per-pixel classification, and context modelling becomes largely meaningless. Rather, a typical deep network will take as input a raw image[5]. The intensity values are passed through multiple layers of processing, which transform them and aggregate them over progressively larger contextual neighborhoods, in such a way that the information becomes explicit which is required to discriminate different object categories[8]. The entire set of network parameters is learned from raw data and labels, including lower layers that can be interpreted as features, middle layers that can be seen as the layout and context knowledge for the specific domain, and deep layers that perform the actual classification.

Semantic Image Segmentation has an important role in Computer Vision problems. Semantic image segmentation is the process of understanding the role of each pixel in an image. Since Fully Convolutional Networks (FCN) which popularized the Convolutional Neural Networks (CNN) architecture in predicting densities without fully connected layers were introduced, semantic image segmentation has become famous[16].

Over time, the rapid growth of the technological world has produced various architectural models that have emerged to solve the Semantic Image Segmentation problem. In addition, semantic image segmentation has also been used or applied in many domains such as medical areas and intelligent transportation[21]. In medical areas, semantic image segmentation is used to detect brains and tumors, and detect and track medical instruments in operations. Whereas in intelligent transportation, semantic image segmentation is used to detect road signs, colon crypts segmentation, land use and land cover classification.

CHAPTER 3 Methodology

3.1 Dataset

Aerial Semantic Segmentation Drone Dataset: The Semantic Drone Dataset focuses on semantic understanding of urban scenes for increasing the safety of autonomous drone flight and landing procedures. The imagery depicts more than 20 houses from nadir (bird's eye) view acquired at an altitude of 5 to 30 meters above ground. A high resolution camera was used to acquire images at a size of 6000x4000px (24Mpx). The training set contains 400 publicly available images and the test set is made up of 200 private images.

PERSON DETECTION

For the task of person detection the dataset contains bounding box annotations of the training and test set.

SEMANTIC SEGMENTATION

We prepared pixel-accurate annotations for the same training and test set. The complexity of the dataset is limited to 20 classes as listed below:

Tree, gras, other vegetation, dirt, gravel, rocks, water, paved area, pool, person, dog, car, bicycle, roof, wall, fence, fence-pole, window, door, obstacle.

3.2 VGG-16 Encoder and FCN8 Decoder

VGG-16 Encoder

VGG-16 is a convolutional neural network that is 16 layers deep. The 16 in VGG16 refers to it has 16 layers that have weights.You can load a pretrained version of the network trained on more than a million images from the ImageNet database. The pretrained network can classify images into 1000 object categories, such as keyboard, mouse, pencil, and many animals. As a result, the network has learned rich feature representations for a wide range of images. The network has an image input size of 224-by-224.

The input to the 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 filters, which can be seen as 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.

Three Fully-Connected (FC) layers follow a stack of convolutional layers (which has a different depth in different architectures): the first two have 4096 channels each, the third performs 1000-way ILSVRC classification and thus contains 1000 channels (one for each class). The final layer is the soft-max layer. The configuration of the fully connected layers is the same in all networks.

All hidden layers are equipped with the rectification (ReLU) non-linearity. It is also noted that none of the networks (except for one) contain Local Response Normalisation (LRN), such normalization does not improve the performance on the ILSVRC dataset, but leads to increased memory consumption and computation time.

FCN-8 Decoder

The fully convolutional neural network. The architecture diagram from the original paper describing fully convolutional networks or FCNs are shown here. The model will learn the key features of the image using a CNN feature extractor, which is considered the encoder part of the model. As the image passes through convolutional layers, it gets downsampled. Then the output is passed to the decoder section of the model, which are additional convolutional layers. The decoder layers upsamples in the image step-by-step to its original dimensions so that we get a pixelwise labeling, also called pixel mask or segmentation mask of the original image. The encoder can use the convolutional layers of a traditional CNN architecture. Note that the fully connected layers of these traditional CNN models are used for classification in object detection tasks, so the encoder of the image segmentation models won't reuse those fully connected layers.} {hat allows you to

take the CNN from the encoder and turn it into an architecture that gives you image segmentation is the decoder. One of the most popular decoder is FCN-8.}

{ As an example, let us take a tiny image that has eight pixels and two columns and four rows. If you perform pooling with a window size of 2 by 2, such as average pooling, the first application of the pooling window applies to the top four cells of the image, and it pools the four values into a single value. If you choose a stride of 2 by 2, the pooling window will slide two cells down. Then the pooling is applied to the bottom four cells of the image, and it pools the image into a single value that you can see here. Notice that the input image has four rows, but the pooling result has two rows. Also notice that the input image had two columns, but the pooling result has one column. If you have a pooling layer with a 2 by 2 pooling window on a stride of 2 by 2, the result of your pooling will reduce the height and width by half. }

{FCN-8 decoder works very similar with the same first two steps, but instead of upsampling the summation of the pool 4 and 5 predictions by 16, it will 2x upsample it, and then add that to the pool 3 prediction. This is then upsampled by eight, and hence the decoder is named FCN-8. Going back to this image, we can see the impact of this by factoring in the results from pools earlier in the architecture, when the image is at a higher resolution, our segments are better defined. Thus, the FCN-8 looks better than the FCN-16, and better than the FCN-32. Of course, depending on your scenario, the FCN-32 might be enough, but it might not be worth the extra processing required to do FCN-16 or FCN-8.

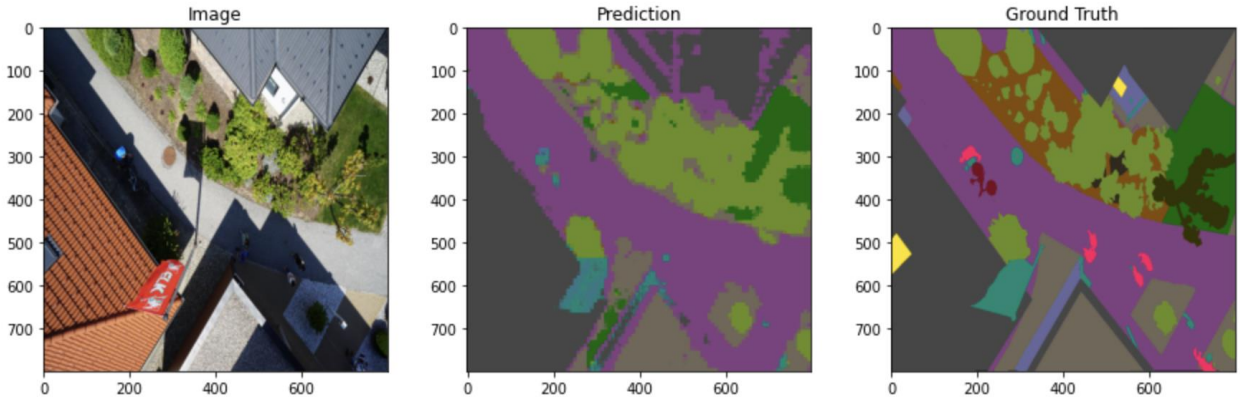

Figure 1: The actual image, ground truth and the prediction using VGG-16 & FCN8.

3.3 U-NET

UNet, evolved from the traditional convolutional neural network, was first designed and applied in 2015 to process biomedical images. As a general convolutional neural network focuses its task on image classification, where input is an image and output is one label, but in biomedical cases, it requires us not only to distinguish whether there is a disease, but also to localise the area of abnormality.

UNet is dedicated to solving this problem. The reason it is able to localise and distinguish borders is by doing classification on every pixel, so the input and output share the same size. For example, for an input image of size 2x2: [[255, 230], [128, 12]] where each number is a pixel the output will have the same size of 2x2: [[1, 0], [1, 1]] this could be any number between [0, 1].

First sight, it has a “U” shape. The architecture is symmetric and consists of two major parts — the left part is called contracting path, which is constituted by the general convolutional process; the right part is expansive path, which is constituted by transposed 2d convolutional layers(you can think it as an up-sampling technique for now). Each process constitutes two convolutional layers, and the number of channels changes from 1 → 64, as the convolution process will increase the depth of the image. The red arrow pointing down is the max pooling process which halves down size of image(the size reduced from 572x572 → 568x568 is due to padding issues, but the implementation here uses padding= “same”). The image at this moment has been resized to 28x28x1024. Now let’s get to the expansive path.In the expansive path, the image is going to be up sized to its original size. Transposed convolution is a sampling technique that expands the size of images. There is a visualised demo here and an explanation here. Basically, it does some padding on the original image followed by a convolution operation.
After the transposed convolution, the image is up sized from 28x28x1024 → 56x56x512, and then, this image is concatenated with the corresponding image from the contracting path and together makes an image of size 56x56x1024. The reason here is to combine the information from the previous layers in order to get a more precise prediction.
Now we’ve reached the uppermost part of the architecture, the last step is to reshape the image to satisfy our prediction requirements. The last layer is a convolution layer with 1 filter of size 1x1(notice that there is no dense layer in the whole network). And the rest left is the same for neural

network training. In conclusion, U-Net is able to do image localisation by predicting the image pixel by pixel and the author of U-Net claims in this paper that the network is strong enough to do good prediction based on even few data sets by using excessive data augmentation techniques. There are many applications of image segmentation using U-Net and it also occurs in lots of competitions.

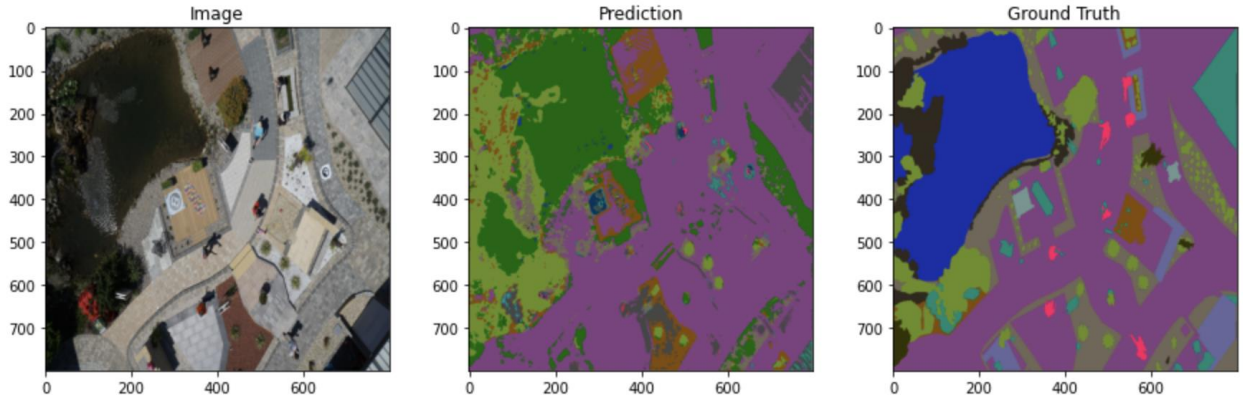

Figure 2: The actual image, ground truth and the prediction using U-Net.

3.3 Segnet

SegNet is a semantic segmentation model. This core trainable segmentation architecture consists of an encoder network, a corresponding decoder network followed by a pixel-wise classification layer. The architecture of the encoder network is topologically identical to the 13 convolutional layers in the VGG16 network. The role of the decoder network is to map the low resolution encoder feature maps to full input resolution feature maps for pixel-wise classification. The novelty of SegNet lies is in the manner in which the decoder upsamples its lower resolution input feature maps. Specifically, the decoder uses pooling indices computed in the max-pooling step of the corresponding encoder to perform non-linear upsampling. This eliminates the need for learning to upsample. The upsampled maps are sparse and are then convolved with trainable filters to produce dense feature maps. The comparison with the other models reveals the memory versus accuracy trade-off involved in achieving good segmentation performance.

SegNet was primarily motivated by scene understanding applications. Hence, it is designed to be efficient both in terms of memory and computational time during inference. It is also significantly smaller in the number of trainable parameters than other competing architectures. We show that

SegNet provides good performance with competitive inference time and more efficient inference memory-wise as compared to other architectures.

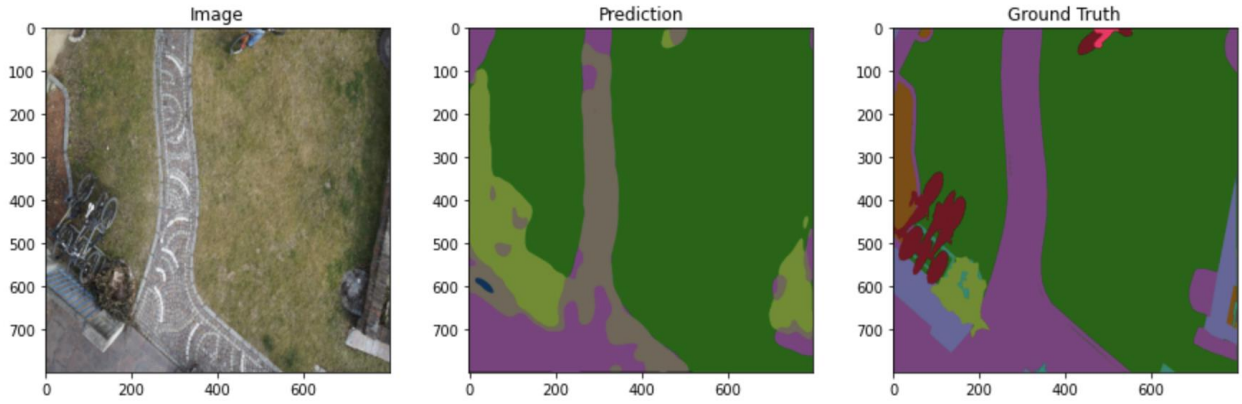

Figure 3: The actual image, ground truth and the prediction using SegNet.

CHAPTER 4 Result Analysis

Dice Score: The Dice score is used to gauge model performance, ranging from 0 to 1. Dice score is used as one of the evaluation metrics. The dice score is twice the area of overlap divided by the combined area. It can be used in similar circumstances to the intersection over union score, and they're often both used. The subtle difference between them is that the dice score tends to veer towards the average performance. f1 is the dice score coefficient.

IOU: IOU is used as one of the evaluation metrics. IoU is the area of overlap between the predicted segmentation and the ground truth divided by the area of union between the predicted segmentation and the ground truth.

VGG16 ENCODER AND FCN 8 DECODER shows the following:
{Train Loss:1.1114}
{Train Accuracy: 0.6767}
{Train IOU Score: 0.1063}
{Train f1 score: 0.1351}
{Val loss: 1.1182}
{Val accuracy: 0.6786}
{Val IOU score: 0.1104}
{Val f1 score: 0.1388}
{Test loss: 0.9814}
{Test accuracy: 0.7197}
{Test IOU score: 0.1273}
{Test f1 score: 0.1514}

U-NET shows the following:
{Loss: 1.0826}
{Accuracy: 0.6796}
{IOU Score: 0.1090}
{f1 score: 0.1473}
{Val loss: 1.1125}

{Val accuracy: 0.6746}
{Val IOU score: 0.1039}
{Val f1 score: 0.1405}
{Test loss: 1.0615}
{Test accuracy: 0.6816}
{Test IOU score: 0.0940}
{Test f1 score: 0.1353}

SegNet shows the following:
{Loss: 1.6372}
{Accuracy: 0.6016}
{IOU Score: 0.0688}
{f1 score: 0.0967}
{Val loss: 2.7133}
{Val accuracy: 0.5570}
{Val IOU score: 0.0823}
{Val f1 score: 0.1000}
{Test loss: 2.5067}
{Test accuracy: 0.5828}
{Test IOU score: 0.0786}
{Test f1 score: 0.0985}

CHAPTER 5 Conclusion

From our results we got to see that VGG-16 Encoder & FCN-8 Decoder has the highest accuracy and IOU score showing that it is the most efficient model for segmentation of high resolution images using the Aerial Semantic Segmentation Drone Dataset.

REFERENCES

1. J. Carreira, R. Caseiro, J. Batista, and C. Sminchisescu. Semantic segmentation with second-order pooling. In ECCV, 2012.
2. D. C. Ciresan, A. Giusti, L. M. Gambardella, and J. Schmidhuber. Deep neural networks segment neuronal membranes in electron microscopy images. In NIPS, pages 2852–2860, 2012.
3. J. Dai, K. He, and J. Sun. Convolutional feature masking for joint object and stuff segmentation. arXiv preprintarXiv:1412.1283, 2014.
4. J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. DeCAF: A deep convolutional activation feature for generic visual recognition. In ICML, 2014.
5. D. Eigen, D. Krishnan, and R. Fergus. Restoring an image taken through a window covered with dirt or rain. In Computer Vision (ICCV), 2013 IEEE International Conference on, pages 633–640. IEEE, 2013.
6. Kodirov, E.; Xiang, T.; Gong, S. Semantic autoencoder for zero-shot learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4447–4456.
7. Mei, X.; Pan, E.; Ma, Y.; Dai, X.; Huang, J.; Fan, F.; Du, Q.; Zheng, H.; Ma, J. Spectral-Spatial Attention Networks for Hyperspectral Image Classification. Remote Sens. 2019, 11, 963.
8. Zhou, B., Zhao, H., Puig, X., Fidler, S., Barriuso, A., Torralba, A.: Semantic understanding of scenes through the ADE20K dataset. arXiv:1608.05442 (2016)
9. Iandola, F.N., Moskewicz, M.W., Ashraf, K., Han, S., Dally, W.J., Keutzer, K.: Squeezenet: Alexnet-level accuracy with 50x fewer parameters and <1mb model size. arXiv:1602.07360 (2016)
10. Han, S., Pool, J., Narang, S., Mao, H., Tang, S., Elsen, E., Catanzaro, B., Tran, J., Dally, W.J.: DSD: regularizing deep neural networks with dense-sparse-dense training flow. In: ICLR. (2017)
11. . Li, H., Kadav, A., Durdanovic, I., Samet, H., Graf, H.P.: Pruning filters for efficient convnets. In: ICLR. (2017)
12. Zhu, X., Xiong, Y., Dai, J., Yuan, L., Wei, Y.: Deep feature flow for video recognition. In: CVPR. (2017)
13. Redmon, J., Farhadi, A.: YOLO9000: better, faster, stronger. In: CVPR. (2017)

14. F. Feng, S. Wang, C. Wang, and J. Zhang, “Learning deep hierarchical spatial–spectral features for hyperspec- tral image classification based on residual 3d-2d cnn,” Sensors, vol. 19, no. 23, p. 5276, 2019.
15. X. Wang, “Moving window-based double haar wavelet transform for image processing,” IEEE Transactions on image processing, vol. 15, no. 9, pp. 2771–2779, 2006.
16. Chen, L., Yang, Y., Wang, J., Xu, W., Yuille, A.L.: Attention to scale: Scale-aware semantic image segmentation. In: CVPR. (2016)
17. Hariharan, B., Arbel´aez, P.A., Girshick, R.B., Malik, J.: Hypercolumns for object segmentation and fine-grained localization. In: CVPR. (2015)
18. P. Liu, H. Zhang, K. Zhang, L. Lin, and W. Zuo, “Multi- level wavelet-cnn for image restoration,” in Proceedings of the IEEE conference on computer vision and pattern recognition workshops, pp. 773–782, 2018.
19. F. Melgani and L. Bruzzone, “Classification of hyperspectral remote sensing images with support vector machines,” IEEE Transactions on geoscience and remote sensing, vol. 42, no. 8, pp. 1778–1790, 2004.
20. M. He, B. Li, and H. Chen, “Multi-scale 3d deep convolutional neural network for hyperspectral image classification,” in IEEE International Conference on Image Processing (ICIP), 2017, pp. 3904–3908.
21. B. Tu, X. Zhang, X. Kang, G. Zhang, J. Wang, and J. Wu, “Hyperspectral image classification via fusing correlation coefficient and joint sparse representation,” IEEE Geoscience and Remote Sensing Letters, vol. 15, no. 3, pp. 340–344, 2018
22. T. Dundar and T. Ince, “Sparse representation-based hyperspectral image classification using multiscale superpixels and guided filter,” IEEE Geoscience and Remote Sensing Letters, 2018.
23. Kundu, A., Vineet, V., Koltun, V.: Feature space optimization for semantic video segmentation. In: CVPR. (2016)