International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878, Volume-7 Issue-5S2, January 2019

Image Segmentation and Semantic Labeling using Machine Learning

Abhishek Thakur, Rajeev Ranjan

Abstract: In this paper image color segmentation is performed using machine learning and semantic labeling is performed using deep learning. This process is divided into two algorithms. In the first algorithm machine learning is used to detect super pixels. These super pixels are segmented on the basis of colors. In the second algorithm deep learning is used to train color categories. This algorithm classify each object into semantic labels. Experiment is performed on BSDS300, CASIA v1.0, CASIA v2.0, DVMM and SegNetVGG16CamVid.

Keywords: Feature Extraction; Machine Learning; Deep Learning; Convolution Neural Network; Image Forensic.

I. INTRODUCTION

The objective of this paper is to find out objects in an image/video with the help of segmentation algorithm. Object detection is the most difficult problem in image processing. In image segmentation it is not necessary to know initially what the visual objects are present in image/video. Object segmentation is different from image classification. Ideal image segmentation algorithm segment unknown objects but in image classification only known categories are classified [1]. There are so many applications [2] where segmentation is used to find out image forgery detection. In image forgery detection, segmentation is performed on each image and it is added to the database. In copy move forgery detection similar color pixel patch are extracted and matched with the help of color segmentation. If forged region query is passed to the algorithm it suggest expected regions of the forgery in the database. Another application in unattended baggage detection. In this task same query is asked by security officer in bus stand, the proposed algorithm segment objects on the basis of that query. When a new image is given to segmentation algorithm, it should segment each pixels of the image into such categories. For example, in Fig. 1, the input image consists of a natural image of plants. In Fig. 2, the segmentation of the input image consists of a semantic objects which clusters the pixels of plant and colored them green. Similarly with the background grass and rocks are colored with dark green and light green. Segmenting an image includes a deep semantic understanding of the world and which things are parts of a whole [3].

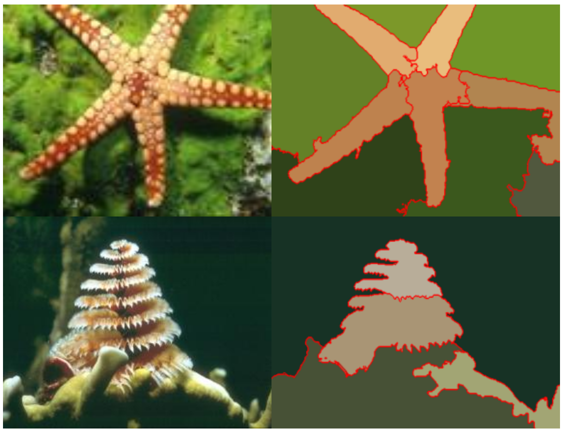

Fig. 1 Image segmentation (right column) of the input image (left column).

Fig. 2 Image semantic labeling (bottom image) of the input image (top image).

In order to understand how object segmentation and semantic labeling are used by machine learning and deep learning architectures. The first step is classification, which predict objects from input image. Localization of the object is the second step in which spatial location (i.e. centroids or bounding boxes) of those classes are provide. In this paper,

Revised Manuscript Received on January 25, 2019
Abhishek Thakur, Electronics & Communication Engineering, Thapar Institute of Engineering and Technology, Patiala, Punjab, India
Rajeev Ranjan, Electronics & Communication Engineering, Thapar Institute of Engineering and Technology, Patiala,Punjab, India

we will mainly focus on image segmentation and semantic labeling, i.e., per-pixel class segmentation. In order to understand deep image segmentation and semantic segmentation systems concept we will discuss some common networks, methods, and design. In addition, we will discuss data pre-processing techniques and learning technique.

1.1 Common Deep Network Architectures

Machine learning and deep learning algorithms have made important contributions in image processing and computer vision field. AlexNet, VGG-16, GoogLeNet, and ResNet used as building blocks for many segmentation architectures.

AlexNet: Krizhevsky et al. [4] proposed AlexNet which won the ILSVRC-2012 with test accuracy of 84:6% over traditional techniques of 73:8% accuracy in the same challenge. It consists of five convolutional layers, max-pooling ones, Rectified Linear Units (ReLUs) as non-linearities, three fully-connected layers, and dropout.

VGG: Visual Geometry Group (VGG) developed by University of Oxford submitted to the ImageNet Challenge (ILSVRC)-2013 achieved 92:7% accuracy. VGG-16 consists of 16 weight layers [5].

GoogLeNet: Szegedy et al. [6] introduced GoogLeNet which won the ILSVRC-2014 challenge with a accuracy of 93:3%. This CNN architecture composed by 22 layers.

ResNet:Microsoft's ResNet [7] won ILSVRC-2016 with 96:4% accuracy. This architecture consists of 152 layers.

Transfer Learning: It is very hard to train deep neural network from scratch because large datasets are not usually available and it take long time to achieve desired results [8].

Data Preprocessing and Augmentation:Data augmentation and preprocessing are common technique to solve overfitting and under fitting problems in machine learning and deep learning. Data augmentation use transformations such as translation, crops, rotation, scaling, warping, color space shifts, etc. [9]. The transformations are used to create larger dataset.

1.2 Datasets and Challenges

Data is one of the most important part of any machine learning system. In machine learning data is everything, if anyone want to develop deep learning model under fit and over fit are the major problems. When data is not sufficient DCNN model struggle with under fit problem. When data is more for particular category, DCNN model struggle with overfitting problem. To overcome this problem dataset should be defined with standard size. For that reason, collecting acceptable data into a dataset is acute for any segmentation system based on deep learning techniques. In this paper BSDS300, CASIA v1.0, CASIA v2.0, DVMM, SegNetVGG16CamVid datasets are used.

The rest of the paper is arranges as follow: In section 2 image segmentation algorithm is described, in section 3 image semantic labeling algorithm is described, in section 4 all the results of image segmentation and semantic labeling are performed, and finally conclusion is given in section 5.

IMAGE SEGMENTATION

Fig. 3 is the block diagram of image segmentation. It is divided in to four parts as 1) set parameters for bipartite graph and over segmentation, 2) extract super pixels & set parameter for mean shift, 3) extract regions with red boundaries and get the overall number of super pixels, 4) Build the super pixel graph and compute Ncut. Eigenvectors, k-means clustering 5) save the results.

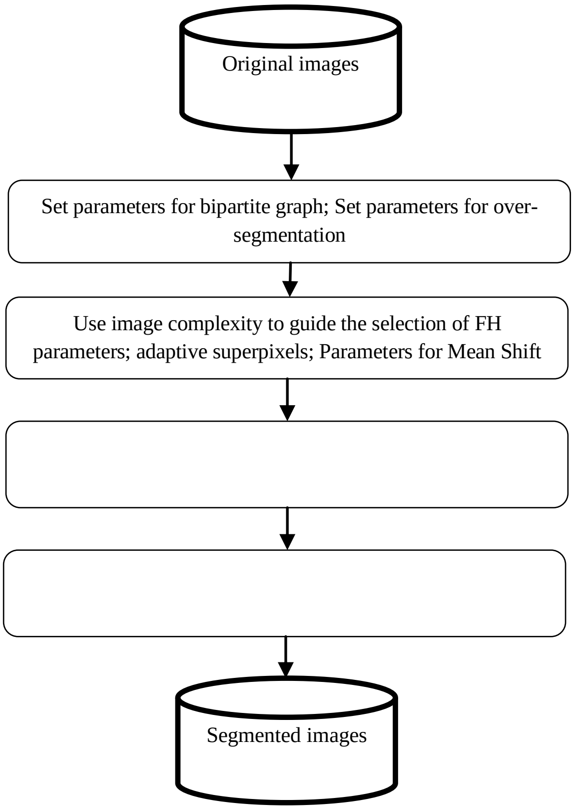

Fig 3: Block diagram of feature extraction

2.1 Set parameters for bipartite graph and over segmentation

To find out correct semantic object segmentation proposed algorithm clusters the pixels of object and colored them into their categories. The value of alpha is 0.001%, beta is 20% and neighbor is 1% between pixels and super pixels. After that read numbers of segments used in the image. Set all values to zero for PRI, VoI, GCE and BDE.

2.2 Extract super pixels & set parameter for mean shift

To extract super pixel, calculate the image dataset length and read number of segments in the image. Generate super pixels using over segmentation by extracting the parameters i.e. segment, labels of image, segment values, segment labels, segment edges, segment image pixels. These parameters are extracted by passing image location, parameter MS, parameter FH.

2.3 Extract regions with red boundaries and get the overall number of super pixels

The graph over pixels and super pixels collectively created by bipartite is given in Fig. 4. To apply super pixel signs, we join a picture element to a super pixel. If the picture element is included in that super pixel. To impose flatness signs, we could simply join adjoining picture elements weighs by similarity, but this would finish up with termination because the flatness regarding adjoining

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

picture element with in super pixels were combined. When imposing super pixel signs. It may also acquire complex graph splitting due to deeper links on the graph. To reward for the flatness on the adjoining picture elements across super pixels, we join adjoining super pixels that are nearby in feature space.

Image Over Segmentation O1 Over Segmentation O2

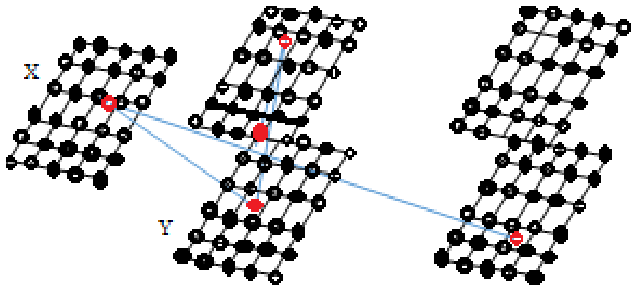

Fig 4: k over segmentation with bipartite graph prototype of an image. A red dot denotes a super pixel whereas black dot denotes a pixel.

2.4 Build the super pixel graph and compute Ncut. Eigenvectors, k-means clustering

The above methods do not exploit the structure that the bipartite graph [7-8] can be unbalanced, i.e., NX = NY. In our case, NX = NY + I, and I NY in general. Thus we have NX NY. This unbalance can be interpreted into the effectiveness of partial SVDs without dropping accuracy. One way is by using the dual property of SVD that a left singular vector can be derived from its right counterpart, and vice versa [10, 13]. In this paper, we follow a “sophisticated” pathway which not only exploits such unbalance but also sheds light on spectral clustering when operated on bipartite graphs. Finally save the results.

II. IMAGE SEMANTIC LABELING

This section, illustrate the steps to perform semantic labeling as shown in Fig. 5. Image segmentation labeling consist of two data folders. The first data folder contain original images and second data folder contain segmented images [11]. These segmentation images are considered as labels for original images. Semantic segmentation using deep learning import Cam-Vid pixel labeled images from vgg16 network [12]. Define classes in the image dataset as sky, building, pole, road, pavement, tree, sign symbol, fence, car, pedestrian and water. To reduce 32 classes into 11, multiple classes from the original dataset are considered.

3.1 Preprocess image Data

Normalize images between (0 1) and resize them to (360 ×480). Save these images into disk. Resize pixel label data to (360×480) and convert them from categorical to uint8. Use 'nearest' interpolation to preserve label IDs. Save these images into disk. Partition Cam-Vid data by randomly selecting 60% of the data for training. The rest is used for testing. Set initial random state for example reproducibility. Use 60% of the images for training. Use the rest for testing. Create image data stores for training and test. Extract class and label IDs info. Create pixel label data stores for training and test.

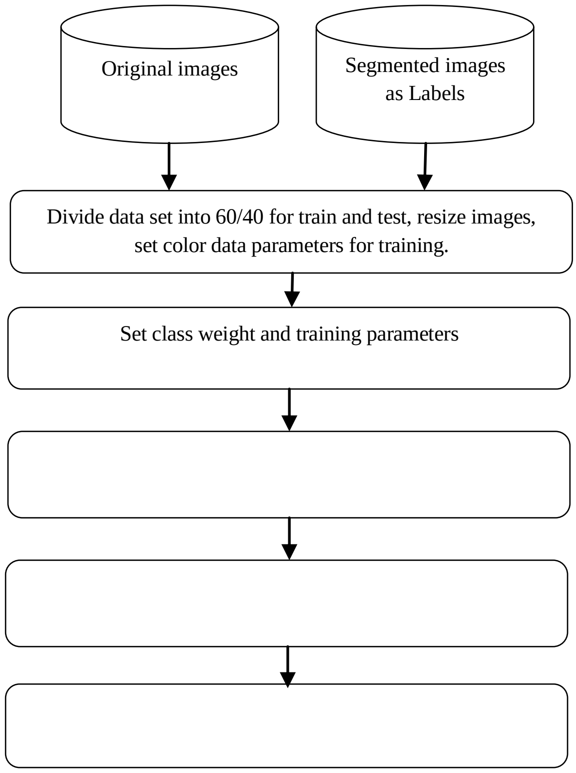

Fig 5: Flow chart of semantic labeling.

Use the classes and label IDs to create the pixel label data set. Read and display one of the pixel labeled images by overlaying it on top of an image. Analyze data set statistics. To see the distribution of class labels in the Cam-Vid dataset, use count each label. This function counts the number of pixels by class label. Resize Cam-Vid data. The 60/40 split results in the following number of training and test images. Create the network image size of 360×480×3. Balance classes using class weighting. As shown earlier, the classes in Cam-Vid are not balanced. To improve training, you can use class weighting to balance the classes. Use the pixel label counts computed earlier with count each label and calculate the median frequency class weights [1]. Specify the class weights using a pixel classification Layer. Update the Seg-Net network with the new pixel classification layer.

3.2 Select training Options

Select training Options as stochastic gradient descent, momentum is 0.9, initial learn rate is 1e-3, L2 regularization is 0.0005, max epochs are 100, Mini batch size is 2, data input strategy is shuffle, execution environment is multi GPU, and plot the training progress bar. The Cam Vid dataset has 32 classes. Group them into 11 classes following. The original Seg-Net training methodology [1]. The 11 classes are: sky, building, pole, road, pavement, tree, sign symbol, fence, car, pedestrian, and bicyclist. Cam-Vid pixel label IDs are provided as RGB color values. Group them into 11 classes and return them as a cell array of M-by-3 matrices. The original Cam-Vid class names are listed alongside each RGB value. Note that the other class are excluded below.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

3.3 Define Label IDs

Label IDs are defined as: (128 128 128) as sky, (000 128 064) as bridge, (128 000 000) as building, (064 192 000) wall, (064 000 064) tunnel, (192 000 128) as archway, (192 192 128) as column pole, (000 000 064) as traffic cone, (128 064 128) as road, (128 000 192) as lane mkgs driv, (192 000 064) as lane mkgs non driv Pavement, (000 000 192) as side walk, (064 192 128) as parking block, (128 128 192) as road shoulder, (128 128 000) as tree, (192 192 000) as vegetation misc, (192 128 128) as sign symbol, (128 128 064) as misc text, (000 064 064) as traffic light, (064 064 128) as fence, (064 000 128) as car, (064 128 192) as SUV pickup truck, (192 128 192) as truck bus, (192 064 128) as train, (128 064 064) as Other Moving, (064 064 000) as pedestrian, (192 128 064) as child, (064 000 192) as cart luggage pram, (064 128 064) as animal, (000 128 192) as bicyclist and (192 000 192) as motor cycle scooter.

3.4 Define the color map

Add a color bar to the current axis. The color bar is formatted to display the class names with the color. Define the color map used by CamVid dataset. Color map as (128 128 128 Sky), (128 0 0 Building), (192 192 192 Pole), (128 64 128 Road), (60 40 222 Pavement), (128 128 0 Tree), (192 128 128 Sign Symbol), (64 64 128 Fence), (64 0 128 Car), (64 64 0 Pedestrian) and (0 128 192 Bicyclist).

3.5 Train and Test deep neural network

Image data is trained with 60% of images from BSDS300, CASIA v1.0, CASIA v2.0, DVMM and SegNetVGG16CamVid with defined training options. One simple command trainnetwork perform this task in matlab 18b. After training test unknown 40% images. This algorithm provide good accuracy to predict each object label as shown in Fig. 7.

III. RESULT

This section, illustrate the results of segmentation in Fig. 6 and semantic labeling in Fig. 7.

Fig. 6 Image segmentation (right column) of the input image (left column).

Fig. 7 shows the semantic segmentation of the input image.All the input images from different dataset are segmented and stored as a labels in a disk. Fig. 8 shows the semantic output labels of the input original and segmented label image. These semantic labels are generated after training a deep neural network with defined color maps. These color maps show the categories label.

Fig. 7 Image segmentation (right column) of the input image (left column).

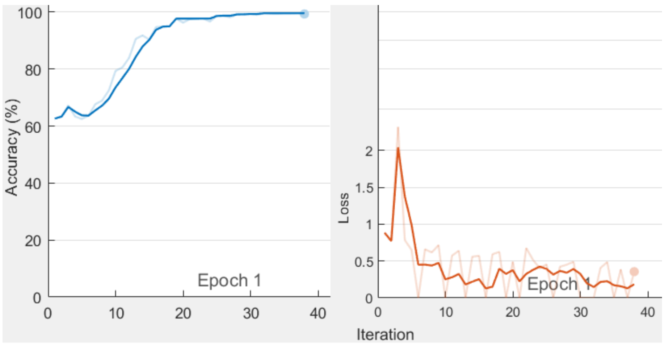

Fig. 8 Training accuracy (left) and training loss (right).

Fig. 8 shows the training accuracy vs loss graph. The training accuracy was 60% initially but after some iteration it approaches to 99%. The training loss was 2% initially but after some iteration it reduced to 0.1%. If we train our deep neural network for long time its accuracy reach to 99.99% and loss decrease to 0.001%.

IV. CONCLUSION

There are so many segmentation technique available nowadays. The color segmentation technique is very important technique that are used for image segmentation and also used for semantic labeling. Here we concluded that the color segmentation for semantic segmentation gives good results to differentiate each object. In semantic labeling deep neural network predict each object label correctly with accuracy of 99.99% in the training phase. So it is a best color segmentation technique in compare to other.

REFERENCE

1. Badrinarayanan, Vijay, Alex Kendall, and Roberto Cipolla. "Segnet: A deep convolutional encoder-decoder architecture for image segmentation." arXiv preprint arXiv:1511.00561 (2015).
2. Brostow, Gabriel J., Julien Fauqueur, and Roberto Cipolla. "Semantic object classes in video: A high-definition ground truth database." Pattern Recognition Letters 30.2 (2009): 88-97.
3. Li, Zhenguo, Xiao-Ming Wu, and Shih-Fu Chang. "Segmentation using superpixels: A bipartite graph partitioning approach." Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on. IEEE, 2012.
4. Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." Advances in neural information processing systems. 2012.
5. Simonyan, Karen, and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." arXiv preprint arXiv:1409.1556 (2014).
6. Szegedy, Christian, et al. "Going deeper with convolutions." Proceedings of the IEEE conference on computer vision and pattern recognition. 2015.
7. Visin, Francesco, et al. "Renet: A recurrent neural network based alternative to convolutional networks." arXiv preprint arXiv:1505.00393 (2015).
8. Yosinski, Jason, et al. "How transferable are features in deep neural networks?." Advances in neural information processing systems. 2014.
9. Garcia-Garcia, Alberto, et al. "A survey on deep learning techniques for image and video semantic segmentation." Applied Soft Computing 70 (2018): 41-65.
10. Guo, Yanming, et al. "A review of semantic segmentation using deep neural networks." International Journal of Multimedia Information Retrieval 7.2 (2018): 87-93.
11. Thakur, Abhishek, and Neeru Jindal. "Image forensics using color illumination, block and key point based approach." Multimedia Tools and Applications (2018): 1-21.
12. Kim, Tae Hoon, Kyoung Mu Lee, and Sang Uk Lee. "Learning full pairwise affinities for spectral segmentation." IEEE transactions on pattern analysis and machine intelligence 35.7 (2013): 1690-1703.
13. Cour, Timothee, Florence Benezit, and Jianbo Shi. "Spectral segmentation with multiscale graph decomposition." Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on. Vol. 2. IEEE, 2005.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication