Dissertation Report
on

Semantic Segmentation Of Images Using Convolutional Neural Networks

Submitted By:
Manpreet Kaur
Enrollment No. 16535022

Under the guidance of
Dr. Biplab Banerjee
Assistant Professor

Department of Computer Science and Engineering
Indian Institute of Technology
Roorkee
May, 2018

Declaration

I declare that the work presented in this dissertation with title **"Semantic Segmentation of Images Using Convolutional Neural Network"** towards fulfillment of the requirement for the award of the degree of **Master of Technology** in **Computer Science & Engineering** submitted in the **Department of Computer Science & Engineering, Indian Institute of Technology Roorkee, India** is an authentic record of my own work carried out during the period of **August 2017 to May 2018** under the supervision of **Dr. Biplab Banerjee**, Assistant Professor, Department of Computer Science and Engineering, Indian Institute of Technology Roorkee, Roorkee, India. The content of this dissertation has not been submitted by me for the award of any other degree of this or any other institute.

Date: .....
Place: ROORKEE

Manpreet Kaur
M.Tech (CSE)
16535022

Certificate

This is to certify that the statement made by the candidate is correct to the best of my Knowledge and belief.

Date: .....
Place: .....

Sign: .....
Dr. Biplab Banerjee
Assistant Professor
Indian Institute of Technology
Roorkee

Until you spread your wings, you will have no idea how far you can fly.

- Napolean Bonaparte

Acknowledgement

Dedicated to my family and friends, for standing by me through thick and thin, without whom I would not have gotten this far. I would like to express my sincere gratitude to my advisor **Dr. Biplab Banerjee** for the continuous support of my study and research, for his patience, motivation, enthusiasm and immense knowledge. His guidance helped me in all time of research and writing of this thesis. I could not have imagined having a better advisor and mentor for my study.

I am also grateful to the Department of Computer Science and Engineering and Tinkering Lab, IIT Roorkee for providing valuable resources to aid my research.

MANPREET KAUR

Abstract

In this report, a different approach for the semantic segmentation of images is presented. It mainly focuses on the semantic analysis of images that lessen the gap between semantics and low level visual features of images. Here Convolution Neural Network(CNN) with deconvolution layers and Conditional Random Fields (CRFs) is used to get semantic meaning of images. Convolution neural networks are very powerful visual models which produces the hierarchy of features. Fundamental principle of these networks is to aggregate the information over larger image region and collect fine contextual information as the receptive field of the image represent. Main motive is to design a model that takes input image of a fixed size and produces output with efficient activation map of the objects appearing in the image. The model is trained on PASCAL VOC 2007 dataset and getting 70.89% accuracy on training data.

Keyword: Convolutional-Deconvolutional Neural Network, Superpixels, Conditional Random Fields.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Process of combining pixels of different objects appearing in the image together which belong to same object class is called Segmentation[2] and giving them class labels that what they are interpreting in a particular image along with segmentation is called *Semantic Segmentation*. As the humans can quickly recognize patterns in the images and group them into meaningful parts which is possible because of visual perception, i. e. humans have visual ability to abstract low level image features without knowing even the context of image. Semantic image segmentation has many applications such as detecting road signs, face detection, locate objects in satellite images,for diagnosis and detecting instruments in medical imaging etc.

It is different from the object detection. In object detection the task is to distinguish the different instances of same object in the image. Semantic segmentation has some advantages over object detection, it detects the instances of different objects in a particular image. Semantic Segmentation helps to resolve many problems such as, in an image there is possibility of some pixels which are more near to some object but those are not part of that object, that might belongs to some other object [2]. As in object detection these can be declared as part of the object that actually are not part of that. In figure 1.1, it's clear that in semantic segmentation this is not the case. There might be some very far pixels which are part of same object.

Many algorithms works only on a predefined set of classes. Models for binary classification i.e. foreground and background. The procedure is called Supervised Segmentation. Unsupervised Segmentation algorithms don't know the labels of

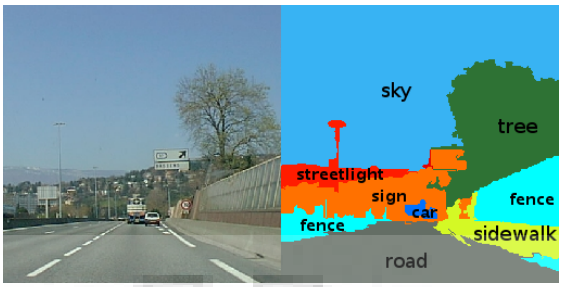

FIGURE 1.1: Semantic segmentation of roadside image

the images in prior. Instead of this fact, these algorithms still able to recognize the classes when they don't know the labels of the input data.

As segmentation means grouping the pixels, Superpixels can also be used in place of pixels. It is the method of grouping similar pixels in the image into some predefined number of regions or patches on the basis of their 'similarity'. Similarity here refers to the RGB values of the pixels or their texture, the cut points of the objects appearing in the image, their potential values etc. By grouping pixels into regions reduces the overhead to process each and every pixel. Talking about the segmentation algorithm, it must divide the image into semantically meaningful parts or objects. But there are chances of high ambiguity because here it's not clear that what an "*object*" is? It can be referred to any "*thing*", any "*texture*" or "*part of another object*" [2]. So task of segmenting an image becomes difficult merely on the basis these low level features, because the objects having similar texture or properties may have different context. So in order to classify objects in the image it is required to have some more initial information about the image context like "shape and size of object" etc. In image segmentation a natural image is divided into K non-overlapping parts. It's main application in computer vision is Image Annotation.

In this study Deep Convolutional Neural Network with Deconvolution layers has been used. Convolutional Neural Networks(CNNs) are intelligent enough to collect contextual features in image categorization problem. CNN consist of stack of learned filters that extract hierarchical image features and are formed of deep learning networks[3]. For feature representation CNN uses the output of last layer. It is good in case of bounding box problems. Output of last layer is very much rough to get precise output in terms of shape.To this contrast, Output of previous layers are precise in localization but they are not good enough as much so that we

can get semantic meanings from them. The useful information is dispersed over all levels of CNN. From the output given by the CNN, the hypercolumn based features are extracted on the basis of per pixel in the image, these features gives the activation map of the input image. This activation map highlights the edges of the objects appearing in the image, thus segmenting the objects in the image.

The proposed model in this report consists of mainly two layers : convolution layer and deconvolution layer. Convolution layer consists of set of filters called "kernels", these filters are having small receptive field, instead of this they can be extended through full depth of input volume. In convolution step each filter is convolved across the height and width of input image. This step computes the dot product of filter and input with bias values added which produce a 2-D activation map of that image. Thus the network learns weights of the kernel, these weights helps to activate the potential when some specific features appears at some spatial position in the input. The second main layer is deconvolutional layer. As Convolution layer works like a down-sampling layer similarly deconvolution layer works like an up-sampling layer. The convolution operation reduces the size of the input based on various parameters like padding, stride etc whereas deconvolution give the size of input back prior to applying convolution.

From the output of CNN, the hypercolumn features are extracted on per pixel basis which were activated/calculated at each layer. Hypercolumns at an input location is an output of all unit lying above that location in all layers of the CNN, stacked as one vector[1]. This term is derived from the "neuroscience". It is used to describe a set of neurons sensitive to edges which can be at multiple orientations.

In the study of this project he network is first trained on the superpixels. The superpixesl are sematically more precise to the context. Since the neighboring pixels have very less variability, by grouping them using superpixel reduces the count of pixels to consider because it groups the pixels into regions. Use of this technique not only decreases the time to process each and every image but also lessen the number of trainable parameters. In this project SLIC has been used to generate the superpixels . SLIC works on KNN algorithm. And it is considered as fastest algorithm to generate superpixels.

Conditional Random fields(CRF) models are composed of unary potential and pairwise potential. The unary potential means the probability of a pixel to belong to particular class. It's value depends upon that pixel only. The pairwise potential

signifies that what is effect of the neighboring pixel of a pixel on that pixel so that it can belong to particular class.

CRF is characterized by Gibbs distribution[4]. It is given as follows:

$$P(x,y) = \frac{1}{Z} exp(-E(x,y)) \tag{1.1}$$

Here Z is *Normalization Factor* and E is *Gibbs Energy*.

This Energy function is given by:

$$E(x,y) = \sum_{c \in \mathcal{C}} \psi(x,y) \tag{1.2}$$

ψ is called '*Clique Potential*'.The above equation shows the energy of labeling $x \in L^{\mathrm{N}}$.

The CRFs, basically the models which are composed of unary potential and pairwise potentials. In CRFs all the clique potentials conditionally depend upon input features. The CRFs just doesn't learn distribution, it learn the *joint probability distribution* as follows[2]:

$$P(y|x) = \frac{1}{Z} \Pi_{c \in C} \psi(y_{\mathrm{c}}|X) \tag{1.3}$$

CRF take limited parameters, as we don't have to estimate for it's value and we also don't have to make any distribution assumption about x[2]. We have entire input available to scan in CRF for the class labels.

There are many proposed and implemented models in which Convolution neural network has been used with CRF or MRF to make the output smoother. Novelty of our approach is that we are using Convolution Neural Network with Deconvolution layers and using CRF to take input image's class lables. CRF gives better performance in giving standard prediction problem.

Chapter 2

Related Work

2.1 Literature Survey

[2]Discuss traditional techniques used for semantic segmentation at pixel-level and metrics used to evaluate those techniques. Some traditional algorithms and their related papers are given such as SVM and Decision Forests. Problems which comes in semantic segmentation algorithms are explained and examined. Classification of such algorithms and different approaches used for semantic segmentation with convolutional neural network are given.

Encyclopedic review of recent advancements in the field of semantic segmentation is given in [5]. There has been reviewed around 190 papers. Analysis of broad image segmentation techniques comes under unsupervised segmentation, semi-supervised segmentation and supervised segmentation is given. Here also evaluation metrics and persuasive datasets are given. They have suggested some ways to design algorithms according to the particular choice and to proceed in the direction of image segmentation research area.

The Convolutional neural network which trained from end-to-end produces better results than many state-of-art methods in image segmentation is given here[6]. The intuition is to make a fully-convolutional neural network, which give segmentation results on arbitrary sized images. They have used the recent classification models such as AlexNet, GoogLeNet[6] and VGGNet and alter their functionality into image-segmentation by converting them into fully-convolutional neural network.

Their approach provides good results on PASCAL VOC dataset and time taken for the segmentation of regular image is also very less.

To solve the issue of narrow capacity of deep learning networks to depict the visible objects in the image, they suggested a new model. This model combines CNN and CRF(Conditional Random Fields) and results in combined benefits of both models. They developed CRF with mean-field approximate inference and Gaussian pairwise potential as RNN(Recurrent Neural Network). Being a single entity, the resulting model(CRF-RNN)[7] is trained end-to-end with conventional back-propagation technique. By doing this they eliminated the need of any post processing technique.

The combination of classic VGG network and deconvolution network has been proposed in[8]. The deconvolution network is composed of deconvolution layers and unpooling layers as same of VGG network which is composed of convolution and pooling layers. The another different thing used here in object proposal. The segmentation has been performed by suggesting different proposals of objects and then the result is computed for each proposal, after that all results are combined to get final result. The good thing about this object proposal technique is that the image segmentation task can be performed at different scales, thus making the resulting model very flexible and robust.

In [1] the *Hyperoclumns* have been introduced. The hypercolumn at each pixel of image contains information of all the layers of convolutional neural network. It overcome the limitation of getting no semantic information from initial layers of the network and no structure information of the objects from last layers of the network. The hypercolumn is just a vector of activation information of all the CNN layers above a particular pixel of input image.

The methods in which image features of multiple low-level and contextual features of high-level are ensembled, performs better than other state-of-art methods. The main focus of this paper [9] is on two major tasks: first is to localize and segment objects in the image, they proposed a bottom to up region proposal technique and second is that in case of deficiency of labeled data, pre-training of data is performed and then fined-tuned depending upon the specific domain. They call this method R-CNN i.e. region with CNN features.

Due to invariance property of deep convolutional neural network, the last layer of the models don't contain sufficient information for image segmentation tasks.

They overcome this problem by combining the CNN with probabilistic graphical model [10] for pixel-level image segmentation. For this the responses of the last layer are combined with fully connected random fields. This model performs best and become a new state-of-art model by reaching 71.6% accuracy on PASCAL VOC dataset.

The big thing about this paper[11] is that they have trained a deep convolutional neural network a 1.2 million high resolution images for classification task. It is done as a part of ImageNet LSVRC-2010 contest. The number of classes in this task is 1000. And they got top-5 and top-1 error rates. The network contains 5 convolutional layers followed by max-poling layer, fully connected layers and at the end they used 1000 way softmax. The trainable parameters and neurons are around 60 million and 650k million respectively. To overcome overfitting problem the dropout layer has been used. And to make the training fast, non-saturating neurons have been used with efficient GPU implementation of convolution operation.

Semantic Segmentation is done by taking two module approach in [12]. The first module is graphical model which produces multiple semantic segmentation proposals, and the second module called *segnet* analyse these proposals and rank them resulting in final prediction. The aim of the paper is to take build a model which give advantages of massive level PASCAL IOU loss function. The MIOU of the resulting model is 52.5%

A feed-forward image segmentation approach has been proposed in [13], which works on statistical structure exploited in image and label space. The small image regions are mapped to rich feature representation. The sequences of nested regions from increasing image extent are build and from them rich feature representation is extracted. These regions are infact extracted by zooming out the superpixels from scene level resolution. This approach is labled as new state-of-art method after obtaining accuracy of 64.4% on PASCAL VOC 2012 dataset.

To extract the dense feature vectors, corresponding to each pixel, to encode all regions of multiple sizes which are centered at one pixel, a multiscale convolutional network is trained from image pixels[14]. This approach produces texture representation, shape and context information. Multiple post processing methods are combined to produce final labels. The technique to get optimal set of components from a collection of segmentation components. The optimal set of components is

that which describe the scene at it's best. The response of the algorithm is less than a second including feature extraction and labels extraction.

Pixel-level segmentations are generally costly in terms of size of model and parameters. Pixel level graph structure are very sparse as a result of these approaches while the region based segmentation gives dense graph[4]. Here in this paper they considered CRF over all set of pixels in image is defined which results in billions of edges and that's very tedious task. They proposed a technique to use pairwise edge potential with fully connected crf to build a efficient inference algorithm. As a result they got dense pairwise edge connectivity which improves the segmentation and accuracy.

The comparison of five state-of-art algorithms used to generate superpixels has been shown on some parameters such as speed, memory efficiency and impact on segmentation. This paper introduces the algorithm Simple Linear Iterative Clustering[15] to generate the superpixels. This algorithm take the k-mean clustering approach. This algorithm is fast as well as memory efficient, thus improves segmentation performance.

Till now the image segmentation has been done on the basis of the idea that what's likely to be belong to particular class. In is paper [16] they are considering global features as well as local features of the image. This is as inter-class spatial relationships. For example to classify some pixels to belong to a tree identify and how's it like to belong to the class 'sky' for the pixels above that tree and how's it like to belong to class 'grass' for the pixels below that tree. It's not an easy task to compute global information corresponding to each pixel. The proposed model is two-stage model. For first part local features are calculated from relative location maps. In the second stage these are combined with appearance based features to get final final segmentation.

2.2 Related Work

In this section various approaches are reviewed that made use of deep learning and CNNs for various low level computer vision tasks, while having a focus on semantic image segmentation. There are number of ways, which can be used for segmentation of images. These can be divided into two categories: First is that in which feature extraction and segmentation tasks are performed individually.

Another approach in this category is to use CRF formulated as RNN(Recurrent Neural Network) to form a part of deep learning for end-to end training[7]

In second category, learn a non-linear model directly from the images to the label map[17]. The authors replaced last fully connected layer of CNN by convolutional layer to keep spatial information. [6] used the concept of fully convolutional layer, they used the notion that most meaningful features are obtained by top layers while lower layers keep information about the structure of image. They trained the FCN end-to-end, pixels-to-pixels and this doesn't require any supervised pre-training and any post-processing step and approach towards superpixels has been used. In [10] they used CRF to refine segmentation produced by CNN. The convolution is used with up-sampled filter. It helps in controlling explicitly the resolution of image and compute the features within deep network.And they got 64.18 MIOU in Pascal VOC2012 data with the deep network combined with CRF.

The combination of two models, Convolutional neural network and CRF has been presented in [7]. They used CRF based probabilistic graphical modelling. The Conditional Random Fields is formulated with mean field approximation inference and Gaussian pairwise potentials as RCNN(Recurrent Neural networks). Using this model they are getting MIOU 74.7 on PASCAL VOC 2012 data set.

For feature extraction, In [1] and [8], they have used deconvolution for semantic segmentation and hypercolumns for object segmentation. The model is being trained on parts of the objects,and for each different classifier is trained. Different results are given for all different categories.

[18] they have proposed a method to identify the location as well as some other features of the objects in the input image. In[12] they have used the CRFs as a proposal mechanism for DCNN based re-ranking system, while in[14], they treated superpixels as nodes for locally pairing CRF and used graph-cuts for discrete speculation.

Chapter 3

Problem Statement

Semantic segmentation has many applications in real world. As discussed in the introduction part of this report, these might be related to medical field (to analyze the anatomy of human body etc.) as well as to the technical field (building self-driving cars etc.). The motivation for semantic segmentation is the capability of self-visualizing applications or devices to visualize real-world scenes as visualized by humans. As we want the accuracy of the applications or devices as close to humans's perception, we must have a system that can visualize the real world scene as close as the humans see and what they interpret. And algorithms on that data should be applied in a way so that we incur with minimum loss of meaningful information. Suppose we have a data set $\{X_1,X_2,X_3,.....,X_N\}$. We will divide this data into two sets training ($\{X_1,X_2,X_3,.....,X_N^{tr}\}$)and testing($\{X_1,X_2,X_3,.....,X_N^{ts}\}$). The task is to segment objects and assign semantic meanings y_{i} to each object o_{j} such that $j \in k$, where k is number of objects appearing in the image X_i, where $i \in N$, the total number of classes. These semantic meanings are class labels which will be there in the training set as well as testing set in the leaning phase. Accuracy of the system will be calculated by the proficiency of model to accurately assign the labels to the objects in the image during testing phase.

There are couple of problems like neighboring pixels of same class might belong to different object instances and disconnected regions belong to same object instance or the boundaries might not be clear to segment. So method to segment the images should be like it can perform edge-detection. Direct implementation of

edge-detecting algorithms is sparse and to some extent it's difficult. Some edge-detection methods are Canny edge detection, Harris corner detection and SUSAN detector.

So our approach is to use Convolution Neural Network(CNN) with Deconvolution layers. The trained network will be then passed to the Conditional Random Fileds (CRFs), it makes the results smoother and maximizes the probability to assign a semantic meaning to particular object.

3.1 Challenges

One might infer from our ability to solve the object detection task in fraction of seconds with smaller error rate. There are some challenges that are in the way of automatic system, that are handled by humans very well. They are, as objects can vary in pose and orientation especially in natural images as we have to take the context of the image into account as the humans do. If the objects can be distinguished without difficulty then the consideration of context is not so important. Here are some challenges described due to appearance variations of objects that can be divided into two main classes:

- **Object Variation:** Stemming from the object itself such as pose, size, articulation which affect the intra-class variations directly but can also influence the inter-class differences.
- **Image Variation:** such as lightening conditions, viewpoint, background clutter etc.

3.1.1 Object Variation

- Intra-class variations refers to the variations that occur between the objects of the same class for example the different breeds of animals in one category say dogs.
- Inter-class variations are those variations which occur between objects of different classes. There are many object classes which might have very less difference among them. So it becomes challenging to distinguish those classes. For example the birds in the sky and aeroplanes flying high in the sky.

3.1.2 Image Variation

- Scale variations and Resolution variation in the image can directly be caused by the imaging devices or can be due to perspective transformations.
- Lighting conditions have a great influence on the appearance of objects. Due to different lighting and the occurrence of shadows, objects can appear completely different and some even difficult to recognize on first site, even for humans.
- The viewpoint of the image can entirely change the appearance of the objects in the image.
- Clutter in images can create confusion between foreground and background and make it difficult to segment the image

In next section we will discuss the details of this problem.

Chapter 4

Proposed Solution

For the semantic segmentation of images, the proposed approach is divided into following three parts Preprocessing, Learning, Inference. These are described as follows:

1. For preprocessing phase, Superpixels are generated of training dataset and SLIC(Simple Linear Iterative Clustering) is used for this process.
2. Convolution Neural Ntwork consists of convolution and deconvolution layers is trained on training data. It will adjust the weights associated with neurons using back- propagation.
3. The post processing module: CRF with unary and pairwise potentials are used for post processing step.

Detailed description is given below:

To prepare Superpixels of the images Simple Linear Iterative Clustering(SLIC) is used. Superpixels provide an easy approach to process the images for segmentation and image annotation tasks. They capture the redundancy in the image and make the further image processing tasks easier. SLIC used KNN algorithm. And it is library in ski-learn library. We just need to pass the input image, number of segments, sigma which is the smoothing Gaussian kernel applied prior to segmentation. And it results in that number of segmented regions of image. In the network according to proposed idea, consists of convolution layers as well as deconvolution layers. The convolution neural network works as a learning network, it will learn

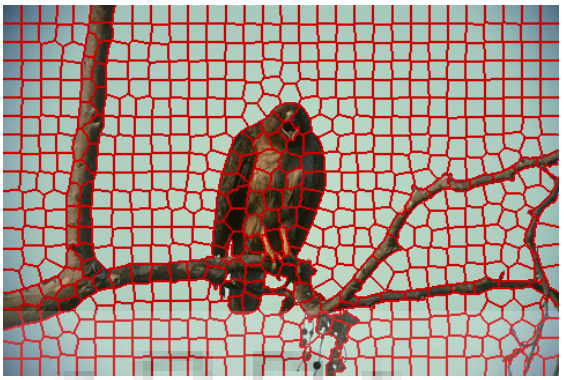

FIGURE 4.1: Superpixels.

according to the input image, optimize the weights of the neurons at each layer corresponding to input. The Deconvolution network works as a shape generator of images. The convolution layers produces the visual cortex of image processed by all DCNN layers. The final output will be the feature map, also known as probability map of the same size as of input image. It indicates the probability of each pixel to belong to one of predefined classes of objects in training data set. The deconvolution layer works

opposite to convolution layer. As the convolution layer decreases the size of the image with auto-pooling method, deconvolution layer's work is to make the activations sparse and enlarge the activations. Then after this. This model preserves the features of initial layer which are related to size and shape of the objects and the features of the last layer which gives the semantic meaning.

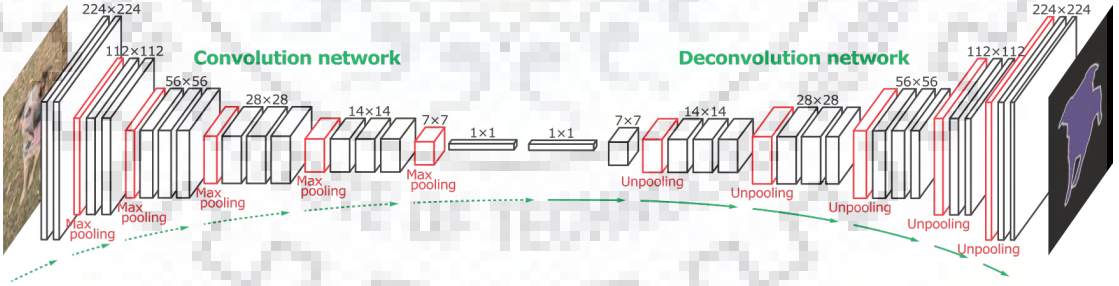

FIGURE 4.2: Convolution Deconvolution network generate segmentation map of the image.

4.1 Convolution-Deconvolution Neural Network

It is a feed-forward artificial neural network. It consists of multiple layers of receptive field. There are collections of small neurons that help in processing

the image. To obtain high-resolution representation of input, output of the these collections are arranged as tiles and it is repeated for every layer. Tiling plays a big role in translating input image. CNN also consists of other layers, for example activation, dense layer etc.

1. Convolution Layer

   It is base of CNN. It consists of a set of learnable parameters also known as "kernels", they don't have large receptive field. Each filter is convolved across height and width of the image. It gives dot product of terms in the filter and the input and it gives a 2-D activation map or feature map of that filter[6]. When Network is trained, network learns the filters when it detect some particular feature at some location in the input image. These filters are learnt during forward pass. During each epoch of training filter is convolved from top-left corner to right-bottom corner of the image. It produces a 2-D activation map of the image. The network learns these filter in a way that they activate whenever they incur a particular type of feature at a specific location in the input image. These activations maps when combined in form of stack, they form the full output volume of the this layer. Every entity in the output can be visualized as a neuron, which is very small region in the input but it share some parameters with neurons in that activation map.
   Convolution Equation : $O = W \cdot I$
   That means the result is the dot product of the input and weight matrix or filter matrix.

   Parameter Sharing:
   In convolution layer if so many neurons would be there then there would be a number of parameters to train for each neurons, as a result the complexity of network would be very high, but it's not the case. There is term called *Parameter Sharing*. There is one assumption that if a patch feature is used to compute spatial information then it can be used to compute spatial information at other location. So all the neurons the single depth slice shares the same parameter. Parameter sharing helps to cope up with translation invariance property of convolution neural network architecture.

   But not always it helps, sometimes if we want to make the network learn images specific to some structure or related to some particular features. For example in dataset we are having face images and we want the network to learn hair structure, color and shapeof eys etc. then parameter sharing

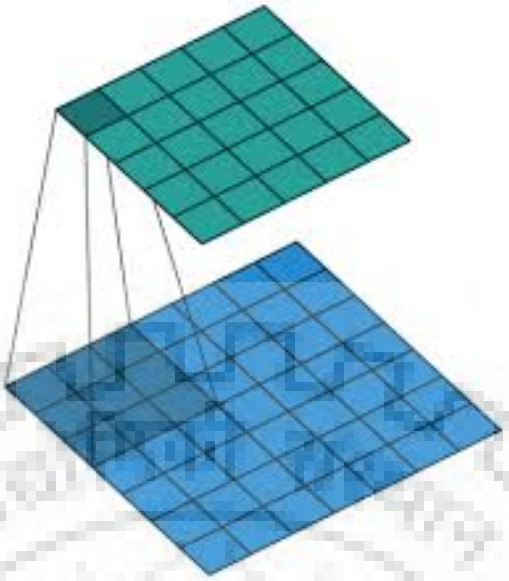

FIGURE 4.3: Convolution Operation.

.

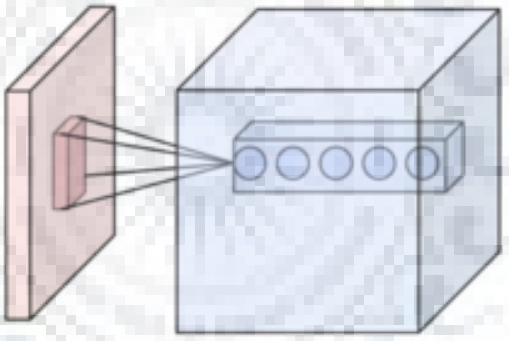

FIGURE 4.4: Parameter Sharing:Neurons are activated from their receptive field.

.

shouldn't be used.

Hyperparameters

Hyperparameters are the entities which control size of the output volume of the convolution layer, these are as:

(a) Depth: Number of neurons in a layer connected to same region in the input volume are controlled by the depth of the output volume. All the neurons learn activation map for different features in the input.

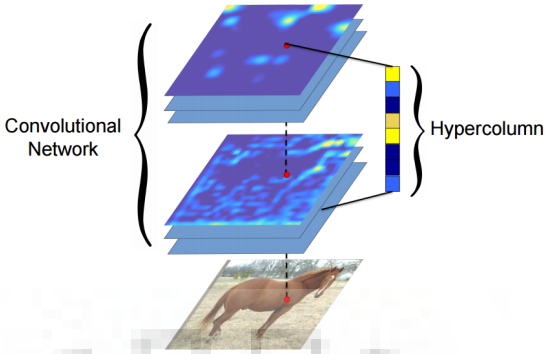

FIGURE 4.5: Hypercolumn Representation. The image at bottom represents input image and feature maps are shown at different layers in the CNN. The hypercolumns at a pixel represents vector of activations of all the units that are lying above that particular pixel. [1]

.

(b) Stride: Stride controls that how the filter will convolve around the input volume. The amount by which the filter shifts is the stride for example if we have stride 1, then it will convolve around the input volume by shifting one unit at a time.

(c) Zero-padding: We know that every time we convolve around the input volume, it reduces it's spatial dimensions by some factor. Suppose if we want to keep the size constant in order to preserve the information about the original input volume for the extraction of some low level features, input volume is padded with zeros. With following formula we can fix the size of padding:

$$zeropadding = \frac{(K-1)}{2} \tag{5.1.1}$$

where K is size of the filter

The output size can be calculated by following formula :

$$O = \frac{(W-K+2P)}{S} + 1 \tag{5.1.2}$$

where O: output height/length, W :input height/length, K:filter size, P:Padding, S: stride

2. Activation layer

After each convolution layer, there is a non-linear activation layer in the proposed model, because Convolution involves only linear operations and in order to make the computation faster without affecting the accuracy there must be non-linearity in the system and to avoid overfitting too. I have used the ReLU activation function in the activation layer, being the most powerful function ReLU uses the function $f(x) = max(0, x)$ to all the values in the input. There are some other functions such as : hyperbolic tangent and sigmoid function. But the ReLU performs better than all of these because it increases the speed of training several times without affecting it's accuracy. This layer is used to insert some nonlinear properties of the model and in overall network without making any effect in receptive fields of the convolution layers[6].

3. Deconvolution layer

   Deconvolution layer map a single activation to large number of outputs. The learned filter helps to reconstruct the shape of the input image. Here deconvolution layers is used similar to convolutional network to capture the shape details at different levels[8] but it performs transpose of convolution. The overall shape of the object is obtained by the lower level filters and the higher level filters describe the class-specific fine-details. Therefore to get the shape features of the input back, we need some operations which can perform oppsosite operation of previous operation. Deconvolution just do like the same. In contrast to unpooling operation, It is not actually a reverse operation of convolution, it can be though of as an upsampling the image which has been downsampled by some operations such as convolution. As in convolution operation dot product of kernel and image is performed in order to get the output kernel result. And similarly in deconvolution operation the weighted kernel when convolved with the output the resulting input image can produced back. The problems are also the same as that of convolutional network, the small activation maps can provide good semantic information while large activation maps can provide good shape information. Therefore combination of both can provide goof results.

4. Backpropagation

   As we learn from our mistakes, similarly the neural network learns from the error it incur. It compute the derivative of the error w.r.t. network

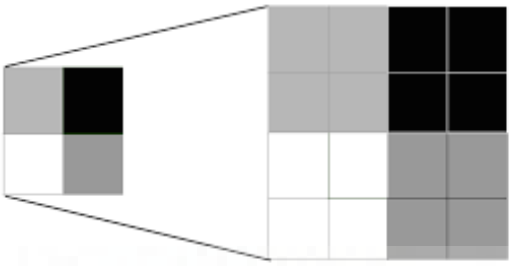

FIGURE 4.6: Deconvolution Operation.

.

parameters, this is achieved using algorithm '*backpropagation*', it can be described as following formulae:

$$E = \frac{1}{P}\sum_{i=1} P(T_{\mathrm{i}} - O_{\mathrm{i}})^2 \tag{5.3.1}$$

Here P is the total no. of predictions, O_{i} is predicted output and T_{i} is targeted output. The error describes the difference in the predicted output and corresponding targeted output, here in case of semantic segmentation these outputs will be labels i.e. the predicted label by the model and the actual label of the object respectively. The weights are updated in order to fit the model over data-set. The derivative of this error function is computed w.r.t. weights and corresponding delta value is computed then weights are either increased or decreased depending upon the difference. These weights are changed according to gradient descent direction of error surface E [19]. It is as follows :

$$\frac{\partial E}{\partial w} = \frac{\partial E}{\partial O}\frac{\partial O}{\partial w} \tag{5.3.2}$$

With the above chain rule the weights also called kernels are learnt for each layer. The term $\frac{\partial E}{\partial O}$ is called as 'delta', which controls the changes in weights of neurons of each layer. w is set of weights associated with neurons and x is x is the given input to the model. O is equivalent to $(w{\cdot}x + b)$ i.e. output of the model. It is described for single perceptron, similarly can be used for multi-perceptron training.

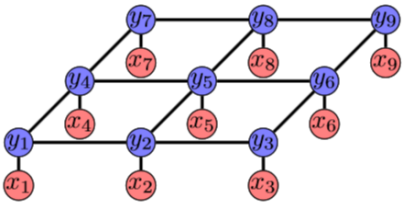

FIGURE 4.7: Pairwise potential representation.

4.2 Conditional Random Fields

Unary CRFs: First we will discuss the pixel-based CRFs also called Unary CRFs, used to formulate the problem of object class segmentation. Here each image pixel is associated with a discrete random variable, each of it takes value from set of labels $L = \{l_1, l_2, l_3, ..., l_k\}$. The set of random variables $X = \{X_1, X_2, X_3, ..., X_N\}$ corresponding to image pixels $i \in \mathcal{I} = \{1,2,3,...N\}$. $\mathcal{N}$ is defined as neighborhood system i.e. set of all neighbors of a variable X_i. Set of random variables X_c which depends on each other are defined as clique c. Labeling is defined as assigning the labels to random variables. Labels are assigned from $L' = L^N$ [20]. The *posterior distribution* Pr(x—D) over labeling of CRF is *Gibbs distribution* and is written as following:

$$Pr(x|D) = \frac{1}{Z} exp(-\sum_{c \in C} \psi(x_c)) \tag{5.5.1}$$

where Z: *partition function* and it is normalizing constant, C: set of all cliques, $\psi(x_c)$: potential function of clique $c \subset \mathcal{I}$ here $x_c = \{x_i : i \in c\}$

The Gibbs' Energy is given as :

$$E(x) = -logPr(x|D) - logZ = \sum_{c \in C} \psi(x_c) \tag{5.5.2}$$

Pairwise CRFs: Pairwise CRFs are mostly used to formulate pixel-labeling problems. In pairwise CRFs Gibbs Energy is sum of unary and pairwise potentials:

$$E(x) = \sum_{i \in \mathcal{I}} \psi_{\mathrm{i}}(x_{\mathrm{i}}) + \sum_{i \in \mathcal{I}, j \in \mathcal{N}_{\mathrm{i}}} \psi_{\mathrm{ij}}(x_{\mathrm{i}}, x_{\mathrm{j}}) \tag{5.5.3}$$

The unary potentials $\psi(x_{\mathrm{i}})$ of the CRFs are defineds as the negative log likelihood variable $\mathrm{x_i}$, while the pairwise potentials encourages neighboring pixels/superpixels appearing in the image to take same label.

There is a limitation of pairwise CRF, they are unable to define high-level dependencies among pixels. Instead it is used widely and considered one of the effective methods for segmentation. In [21] they have used pairwise CRF in object class segmentation.

$\boldsymbol{P^N}$ **model**: Pixels interpret less information as compared to segments. Pixels lying in the same segment are more likely to have same label. First the pairwise CRF was extended by [22] by incorporating with higher order potentials defined over segments. The Energy of higher order method proposed by [22] was of the form :

$$E(x) = \sum_{i \in \mathcal{I}} \psi_{\mathrm{i}}(x_{\mathrm{i}}) + \sum_{i \in \mathcal{I}, j \in \mathcal{N}_{\mathrm{i}}} \psi_{\mathrm{ij}}(x_{\mathrm{c}}, x_{\mathrm{j}}) + \sum_{c \in \mathcal{S}} \psi_c^h(x_{\mathrm{c}}) \tag{5.5.4}$$

here $\mathcal{S}$ is set of segments, ψ_{c} are higher order potentials defined for segments. Higher order potentials take the following form:

$$\psi_c^h = \min_{l \in L}(\gamma_c^{max}, \gamma_c^l, k_c^l N_c^l) \tag{5.5.5}$$

It satisfied $\gamma_c^l \leq \gamma_c^{max}, \forall l \in L$, where $N_c^i = \sum_{i \in c} \delta(x_{\mathrm{i}} \neq (l))$ is no. of pixels labeled wrongly with label l. Labeling of each segment is associated with a cost which is γ_c^l if all pixels in segment are taking label l and there is penalty cost that each inconsistent pixel may incur of k_c^l. The maximum truncated cost of potential is γ_c^{max}. By setting $\gamma_c^l = 0 \forall L$, inconsistent segments are penalised and label consistency in segments is encouraged.

Chapter 5

Results

5.1 Data Set Used and Experimental Setup

Data Set

For this project work Pascal VOC2007 data set has been used. In data set, annotations have been given for the segmented class for all images, which contains source of the image and pixel positions of the major objects appearing in the image. But I am reading the training images directly from their parent folder using a train.txt file. And similarly for the validation and testing of model. There are 209 training, 213 validation images. Their ground-truth images are given in .png format. Ground truth images are given for class as well as object segmentation. From ground truth values, we can predict two values, one is class of object appearing in the image and other is boundaries i e. segmented objects appearing in the image. The testing dataset for PASCAL VOC 2007 is separate from the training data. It consists of 210 images. All the images having objects in them belonging to set of 21 predefined classes.

Experimental Setup

The code for this project is written in python(version2.7) language. For the Convolution neural network, I have used the keras framework. Which is very user friendly and can be customized easily. The system configuration is 8GB RAM, 1 TB hard disk and 3.50 GHZ processor.

In the implementation part of the project, convolution neural network consists of 2 blocks, each consists of one convolution layer, activation layer and deconvolution

layer. Activation function used in the second layer is ReLU. Kernel values for the experiment are initialized to random uniform with a min and max norm, so that they don't become too much high and also not too much low(so that the loss shouldn't dropped to 'nan'). The optimizer I have used is 'adam' and for the loss function the 'categorical cross entropy' function have been used to calculate the training loss, validation loss as well as testing loss. The last layers of the whole network are dense, activation and dense again. Dense layer has been used in order to get result consists of 21-dimensional vector. So that each index value shows the probability to belong to each of 21 classes. The learning rate of the network is kept 0.01 The kernel values are normalized before the start of each epoch using L2 norm function. The biases are initialized to zeros. This network is trained for the 1000 epochs.

The superpixels of all training and validation images are produced separately. All the training and validation are converted into images consists of superpixels. Which is a separate module implemented using SLIC. I have used RAG(region adjacency graph) to get centroid of each region. Then I replaced the pixel value of each pixel in the image with the RGB pixel value of the centroid of the region. The resulting image consists of less number of regions to consider than the original training images.

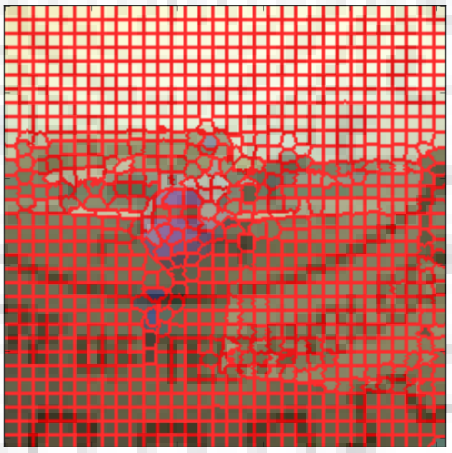

FIGURE 5.1: Superpixels of image generated by SLIC

The third part is CRF part. There is library of the CRF in python 'pydensecrf', I made use of that to implement it. The unary potentials are calculated from a simple neural network. For this other model is made separately which consists of only one convolution, one deconvolution and a last layer 'softmax' as classification layer. This gives the probability value of each pixel to belong to the one of 21

classes(20 for object class + 1 background). And for the Pairwise potential, I have used the 4-neighbor technique. In this approach first the input image is padded with 0's by giving padding of 1x1. And then for each location of original image the difference of RGB pixel value is subtracted from each neighbor's pixel value and those are summed up and to this the index difference of each location is also added. Let's say to calculate pairwise potential value for pixel at location (i,j) is $\sum$ (4 + 4 x I(i,j) - I(i-1,j) - I(i+1,j) - I(i,j-1) - I(i,j+1)). 4 is added to add the index difference between the location of source pixel and all the neighboring pixels. These two values(unary potential and pairwise potential values) then added to the CRF.

Results

The model contains 7 layers. The model is trained for 1000 iterations and it gave 69.97% accuracy on the training dataset, 70.84% on validation data and 68.91% accuracy on testing dataset. The loss values for training , validation and testing are 90.34% ,87.32%, 91.76% respectively.

Data set	Accuracy	Loss
Training	69.97	90.34
Validation	70.84	87.32
Testing	68.91	91.76

TABLE 5.1: Accuracy and Loss table

The above table showing the loss and accuracy values of the training, validation and tsting of the model trained over 1000 iterations. As the training loss and validation loss are near around, the nearness of validation and training loss signifies for the goodness of the model. In earlier trained model I was getting testing accuracy and validation accuracy 1.00 which was the sign of overfitting. After adding the dropout layer, it got decreased from 1.00 to 0.7084 for validation data. Dropout layer drops some value of the data to avoid overfitting.

During training of the model, there is a term called history of the model. It's avialable in keras as well tensorflow. What history does, it store the accuracy and loss values after every epoch of the model and at the end of training of the model, it plot it's graph. For the trained model the accuracy and loss graph are shown below.

Accuracy and Loss Graphs

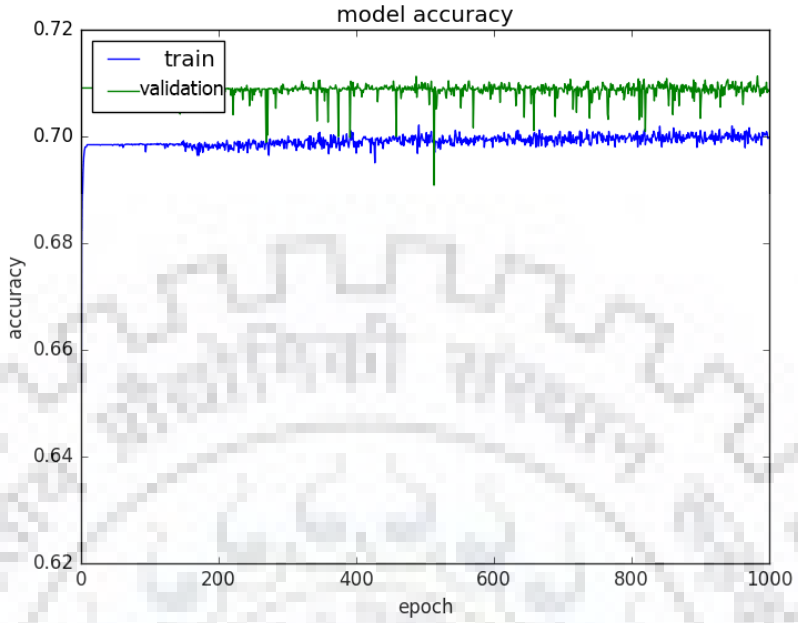

FIGURE 5.2: Accuracy

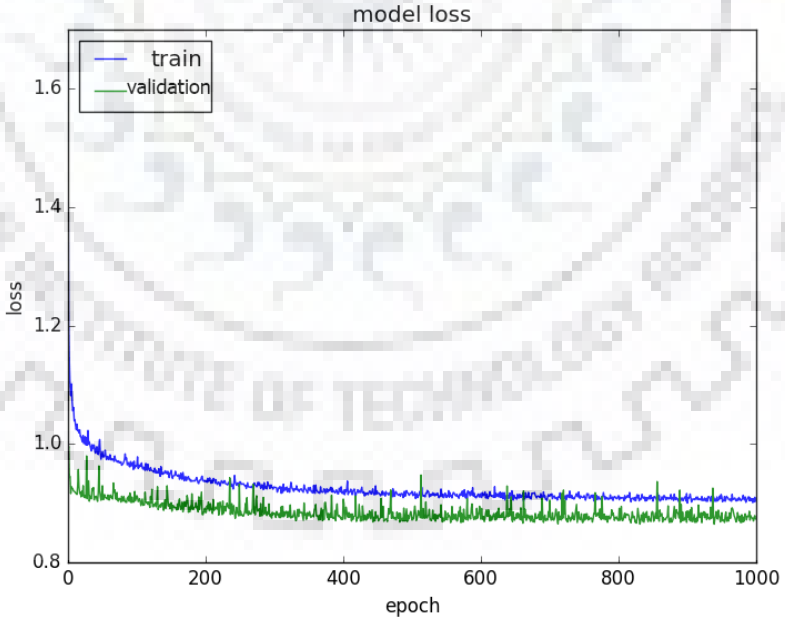

FIGURE 5.3: Loss

Conclusion

The proposed solution for semantic segmentation in this project is to train Convolution Neural Network with deconvolution layers by using CRF as post processing step. The unary potential and pairwise potentials are added to dense crf to get inference of the input image while working on superpixels rather than working pixels of images. The deconvolution layer is added in order to get back the output of the size of original image. It alleviate the need to add maxpooling layer and then add unpooling. The optimizer and loss functions which are used in this model are known to perform best in their domains. Because the maxpooling layers are not added in the model, the resulting training time is high. The accuracy of the model is 70.84%. But it's giving quite good results on PASCAL VOC 2007 Dataset on training, validation as well as testing data in 1000 iterations. But due to less number of training iterations the segmentation output is not too much identifiable. There are three future scopes : One is to make changes in the current model and add some more things(one is to train the model for large number if iterations and train on larger dataset i.e. which contains a large number of input data) which I have missed in the current model to make the segmentation more identifiable and fine and second is to extend the model to be able to produce the semantics of the segmented objects in the resulting image. The third is to make the three models train and work end-to-end. Till now all three are implemented separately and working separately. Benefit of training the model end-to-end is that the overall complexity of problem can be find. The number of trainable parameters will get reduced by number of times.

References

[1] Bharath Hariharan, Pablo Arbeláez, Ross Girshick, and Jitendra Malik. Hypercolumns for object segmentation and fine-grained localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 447–456, 2015.

[2] Martin Thoma. A survey of semantic segmentation. *arXiv preprint arXiv:1602.06541*, 2016.

[3] Emmanuel Maggiori, Yuliya Tarabalka, Guillaume Charpiat, and Pierre Alliez. Convolutional neural networks for large-scale remote-sensing image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 55(2): 645–657, 2017.

[4] Philipp Krähenbühl and Vladlen Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In *Advances in neural information processing systems*, pages 109–117, 2011.

[5] Hongyuan Zhu, Fanman Meng, Jianfei Cai, and Shijian Lu. Beyond pixels: A comprehensive survey from bottom-up to semantic image segmentation and cosegmentation. *Journal of Visual Communication and Image Representation*, 34:12–27, 2016.

[6] Evan Shelhamer, Jonathon Long, and Trevor Darrell. Fully convolutional networks for semantic segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 2016.

[7] Shuai Zheng, Sadeep Jayasumana, Bernardino Romera-Paredes, Vibhav Vineet, Zhizhong Su, Dalong Du, Chang Huang, and Philip HS Torr. Conditional random fields as recurrent neural networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1529–1537, 2015.

[8] Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. Learning deconvolution network for semantic segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1520–1528, 2015.

[9] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[10] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. *arXiv preprint arXiv:1412.7062*, 2014.

[11] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[12] Michael Cogswell, Xiao Lin, Senthil Purushwalkam, and Dhruv Batra. Combining the best of graphical models and convnets for semantic segmentation. *arXiv preprint arXiv:1412.4313*, 2014.

[13] Mohammadreza Mostajabi, Payman Yadollahpour, and Gregory Shakhnarovich. Feedforward semantic segmentation with zoom-out features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3376–3385, 2015.

[14] Clement Farabet, Camille Couprie, Laurent Najman, and Yann LeCun. Learning hierarchical features for scene labeling. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1915–1929, 2013.

[15] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Süsstrunk. Slic superpixels. Technical report, 2010.

[16] Stephen Gould, Jim Rodgers, David Cohen, Gal Elidan, and Daphne Koller. Multi-class segmentation with relative location prior. *International Journal of Computer Vision*, 80(3):300–316, 2008.

[17] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. In *Advances in neural information processing systems*, pages 2366–2374, 2014.

[18] Yi Yang and Deva Ramanan. Articulated human detection with flexible mixtures of parts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(12):2878–2890, 2013.

[19] Andrea Vedaldi and Karel Lenc. Matconvnet: Convolutional neural networks for matlab. In *Proceedings of the 23rd ACM international conference on Multimedia*, pages 689–692. ACM, 2015.

[20] Chris Russell, Pushmeet Kohli, Philip HS Torr, et al. Associative hierarchical crfs for object class image segmentation. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 739–746. IEEE, 2009.

[21] Jamie Shotton, John Winn, Carsten Rother, and Antonio Criminisi. Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In *European conference on computer vision*, pages 1–15. Springer, 2006.

[22] Pushmeet Kohli, Philip HS Torr, et al. Robust higher order potentials for enforcing label consistency. *International Journal of Computer Vision*, 82(3): 302–324, 2009.

plagiarism report

ORIGINALITY REPORT

5%	1%	4%	2%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	L'ubor Ladicky, Chris Russell, Pushmeet Kohli, Philip H.S. Torr. "Associative hierarchical CRFs for object class image segmentation", 2009 IEEE 12th International Conference on Computer Vision, 2009 Publication	1%
2	Submitted to Siddaganga Institute of Technology Student Paper	1%
3	Hariharan, Bharath, Pablo Arbelaez, Ross Girshick, and Jitendra Malik. "Object Instance Segmentation and Fine-Grained Localization using Hypercolumns", IEEE Transactions on Pattern Analysis and Machine Intelligence, 2016. Publication	1%
4	Shuai Zheng, Sadeep Jayasumana, Bernardino Romera-Paredes, Vibhav Vineet et al. "Conditional Random Fields as Recurrent Neural Networks", 2015 IEEE International Conference on Computer Vision (ICCV), 2015	1%

	Publication	
5	Submitted to Banaras Hindu University Student Paper	<1%
6	"Intelligent Systems Design and Applications", Springer Nature, 2018 Publication	<1%
7	Farabet, Clement, Camille Couprie, Laurent Najman, and Yann LeCun. "Learning Hierarchical Features for Scene Labeling", IEEE Transactions on Pattern Analysis and Machine Intelligence, 2012. Publication	<1%
8	Ľubor Ladický. "What, Where and How Many? Combining Object Detectors and CRFs", Lecture Notes in Computer Science, 2010 Publication	<1%
9	Lecture Notes in Computer Science, 2010. Publication	<1%
10	www.k2.t.u-tokyo.ac.jp Internet Source	<1%

Exclude quotes	On	Exclude matches	< 20 words
Exclude bibliography	On		