Object Co-segmentation using Deep Learning

Dual Degree Project Report

Submitted in partial fulfillment of the requirements

for the degree of

Dual Degree (B.Tech + M.Tech)

by

Richa

(Roll No. 13D070063)

Under the guidance of

Prof. Rajbabu Velmurugan

Department of Electrical Engineering
Indian Institute of Technology Bombay

June 2018

Declaration of Academic Ethics

I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I adequately cited and referenced the original sources. I declare that I have properly and accurately acknowledged all sources used in the production of this thesis.

I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/ data/ fact/ source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the Institute and can also evoke penal action from the sources which have not been properly cited or from whom proper permission has not been taken when needed.

Date: 03/07/18

Richa

(Roll No. 13D070063)

Dissertation Approval Sheet

This is to certify that the dissertation titled

Object Co-segmentation using Deep Learning

By

Richa
(13D070063)

is approved for the degree of **B.Tech + M.Tech (Dual Degree)**.

V. Rajbabu

Prof. Rajbabu Velmurugan
(Guide)

Examiner-1

Examiner-2

Chairperson

Date : 03/07/18

Acknowledgement

I would like to thank my PhD supervisor, Mr. Sayan Banerjee for the constant help he provided in clearing my project related doubts during my DDP work. He used to be available in VIP lab whenever I needed help in clearing any of the concepts. I would also like to thank Mr. Avik Hati for helping me out with matlab problems and Mr. Amit More for giving me futher insights on Deep Learning. I am thankful to my Prof. Rajbabu V. for choosing me and considering me capable of solving the challenging Object Co-segmentation problem.

Richa
Electrical Engineering
IIT Bombay

Abstract

Object Co-segmentation aims to segment common objects across multiple images simultaneously using some shared information. Object co-segmentation has a variety of real life applications in image retrieval, generating automatic image annotations and object tracking. Aiming to build an end-to-end Supervised Image Co-segmentation as the final goal, this report includes the implementation of the topics that we have looked at along with improvising a state-of-art co-segmentation technique. It begins with analyzing Supervised foreground object segmentation using a simple Convolutional Neural Network as well as using variety of color invariant features for better image segmentation on iCoseg dataset. The later part of this work involves analyzing the Segnet architecture and how it can be used to get better semantic segmentation results on CamVid dataset. This was followed by implementation of a state-of-art object co-segmenting technique.

Contents

List of Figures

Chapter 1

Introduction

In this dissertation work we tackle the problem of Object Co-segmentation. This task is particularly challenging when objects involve substantial appearance variations due to changes in pose, scale and illumination, or objects boundaries are distracted by occlusion and background clutter. High similarity between the foreground and the background makes the task difficult. When we have a large number of images for common object extraction, we further have a large variations in pose, appearance and illumination. Multiple common foreground makes the task of co-segmentation more difficult. Some studies claim that Deep Neural Networks (DNNs) can be used to tackle this problem [6]. Hence in our first approach for image segmentation we have used a CNN architecture to learn the features of the foreground from the given images as the training images and use the trained model to segment the test images into foreground and background.

Apart from the DNN approach to extract features from the foreground, other intensity based descriptors are also used for feature extraction. But these intensity based color descriptors are not illumination invariant. Hence to obtain better segmentation results illumination-invariant descriptor such as color descriptors can be used.

SegNet[1] has proved to be an efficient architecture for pixel-wise semantic segmentation. It consumes less memory during inference as it only stores the max-pooling indices of the feature maps and use them in the decoder network.
In order to overcome the challenges of co-segmentation we need to develop a robust method to share information across images. Deep-dense Conditional Random Fields for Object Co-segmentation [6] proposes an object co-segmentation system that learns to share common information for segmentation. The images are broken into image proposals which are then labelled using as containing common object or not using some criteria. These are then passed through a deep dense network for estimating the unary as well their pairwise potentials. The unary potential signifies the potential of a proposal containing potential common object whereas the pairwise potential signifies the similarity of two very similar proposals. This DDCRF [6] is an extention of dense CRF in [Krahenbuhl and Koltun, 2012].

1.1 Related Work

CNN has demonstrated a good performance in Computer Vision tasks such as image classification. Exploitation of the analysis capacity of CNN for image analysis can be found in literature: feature extraction and classification. In feature extraction approach, pre-trained CNN models are used to automatically extract image features that later are analyzed by traditional machine learning methods. In classification approach, a CNN is trained from scratch using a large set of images.[2] However to perform pixel-based classification it is necessary to break the image down into overlapping patches. This method is proved to be time consuming. To avoid this superpixels are used during labeling process. Since the pixels belonging to a superpixel are similar in color and belong to the same object these properties allow them to be completely contained by a window, maintaining the characteristics of the reduced environment to be analyzed with a CNN. This labeling approach has reduced the number of windows to those centered at the centroid of the superpixel. Such superpixel labeling has proven to give results as good as pixel-wise classification as shown in Figure 1.1.

W size	Image	Superpixel size											
		20			30			40			50		
		Pixel	SP	Diff. (%)	Pixel	SP	Diff. (%)	Pixel	SP	Diff. (%)	Pixel	SP	Diff. (%)
32	15	0.7202	0.7187	0.1435	0.7256	0.7233	0.2294	0.7248	0.7219	0.2854	0.7204	0.7166	0.3830
	28	0.7312	0.7297	0.1541	0.7346	0.7326	0.1985	0.7306	0.7269	0.3684	0.7384	0.7335	0.4908
	34	0.7468	0.7449	0.1858	0.7531	0.7511	0.2041	0.7530	0.7496	0.3386	0.7574	0.7530	0.4344
	37	0.7748	0.7732	0.1635	0.7703	0.7678	0.2475	0.7765	0.7732	0.3328	0.7629	0.7585	0.4417
48	15	0.7555	0.7539	0.1643	0.7584	0.7567	0.1770	0.7515	0.7486	0.2865	0.7515	0.7474	0.4093
	28	0.7510	0.7493	0.1624	0.7463	0.7435	0.2798	0.7532	0.7497	0.3478	0.7631	0.7587	0.4450
	34	0.7591	0.7575	0.1634	0.7618	0.7597	0.2129	0.7692	0.7666	0.2614	0.7703	0.7657	0.4561
	37	0.7936	0.7914	0.2107	0.7911	0.7887	0.2335	0.7932	0.7899	0.3301	0.7976	0.7940	0.3638
64	15	0.7401	0.7388	0.1238	0.7617	0.7598	0.1923	0.7540	0.7510	0.2999	0.7548	0.7511	0.3699
	28	0.7659	0.7648	0.1135	0.7670	0.7647	0.2264	0.7534	0.7502	0.3148	0.7626	0.7579	0.4706
	34	0.7645	0.7626	0.1861	0.7581	0.7561	0.1993	0.7699	0.7666	0.3308	0.7715	0.7673	0.4200
	37	0.7986	0.7971	0.1540	0.7971	0.7958	0.1272	0.7978	0.7954	0.2384	0.7979	0.7944	0.3478

Figure 1.1: Classification results (F1-values) obtained by applying the different labeling approaches. [2, C. Gonzalo-Martin] , Deep Learning For Superpixel-based Classification Of Remote Sensing Images

While using ML approaches for image segmentation there can be a large variations in viewing and lighting conditions. To increase illumination invariance and discriminative power, color descriptors have been proposed. [7] mentions a structured overview of color invariant descriptors in the context of image category recognition. These color descriptor have proven to be highly effective under many circumstances especially on PASCAL VOC 2007 dataset[8]. Such descriptors have proved to be light intensity and light color invariant. It was observed that SIFT based color descriptors performed better on object category recognition. Out of the many SIFT based color descriptors CSIFT and OpponentSIFT descriptors have proven to give good results in category recognition.
Locality-constrained Linear Coding for Image Classification is an effective coding com-

pared to the traditional Bag-of-Words model to perform Vector Quantization [9]. LLC utilizes the locality constraints to project each descriptor into its local-coordinate system, and the projected coordinates are integrated by max pooling to generate the final representation.

Prior to Supervised methods for Object Co-segmentation, other works using unsupervised as well as semi-supervised based co-segmentation were popular. Most of the unsupervised methods were based on energy minimization[3] Figure 1.2. These methods were implemented on a pair of images and were later extended to multiple images with multiple foregrounds[4] Figure 1.3.

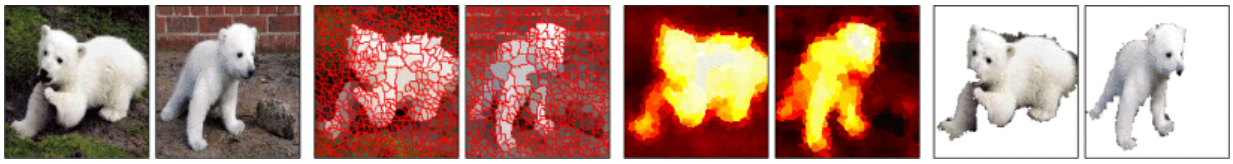


Figure 1.2: Co-segmentation on pair of images. Image courtesy:[3, Joulin et al.,] Discriminative clustering for image co-segmentation", CVPR 2010

Figure 1.3: Co-segmentation on multiple images with multiple foreground. Image courtesy: [4, Quann et al.] ,Object Co-segmentation via Graph Optimized-Flexible Manifold Ranking", CVPR 2015

Besides unsupervised methods some of the semi-supervised ways were also popular in giving good object Co-segmentation results.Semi-supervised methods were also known as Interactive co-segmentation. In this method the images were annotated using lines called scribbles. Separate scribbles were done for foreground and background. All the pixels lying on these scribbles were used to estimate the models for both foreground and background separately. These models are usually Gaussian Mixture Models.

1.2 Report Outline

The following report is a compilation of our work on Object Co-segmentation. It begins with some insights on previous works in Object Co-segmentation and how they have been

Figure 1.4: Interactive co-segmentation. Image courtesy: [5, Batra et al.], iCoseg: Interactive Co-segmentation with Intelligent Scribble Guidance", CVPR 2010

useful followed by the challenges associated with co-segmentation. Then it review topics that are needed as background for understanding the later implementations.
Our work initial work starts with simple object segmentation using both supervised and unsupervised methods. The supervised method uses CNN network to do the job (Chapter-3). The unsupervised method is based on using color descriptors as the distinguishing feature (Chapter-3).
Following this the implementation of the SegNet-Basic model to perform Semantic segmentation of road scene images from CamVid dataset (Chapter-4). This was followed by implementation of Object co-segmentation using conditional random fields (Chapter-5).

Chapter 2

Background

2.1 Superpixel Segmentation

Prepossessing technique where an image is over-segmented into superpixels. Superpixel is a group of connected pixels with similar colors or gray levels. This is an unsupervised algorithm that uses local k-means of predetermined k = (# of superpixels) to over segment the image into superpixels. The initialization is a uniform grid structure to ensure that the resulting superpixels are relatively uniform.

For a set of images

$$\omega = \{I_1, I_2, ..., I_m\}$$

that contain a common object and similar background, our goal is to segment the common object instance in each image. As a pre-processing step, each image I_i in is first over-segmented into n_i superpixels by the SLIC algorithm. Then the whole image set ω contains $\sum_{i=1}^{m} n_i$ superpixels. Our object co-segmentation process is performed on superpixel level.

2.2 SegNet[1]

SegNet is a deep fully convolutional neural network architecture for semantic pixel-wise segmentation. Its architecture consists of an encoder network and a decoder network followed by pixel-wise classification layer. The encoder network in SegNet is topologically identical to the convolutional layers in VGG16 after removing the fully connected layers. The decoder network consists of hierarchy of decoders one corresponding to each encoder.

2.3 Conditional Random Fields

CRF is a conditional modelling method used to encode some relationship between observations and deduce corresponding interpretations. CRFs are often used in object recognition and image segmentation.

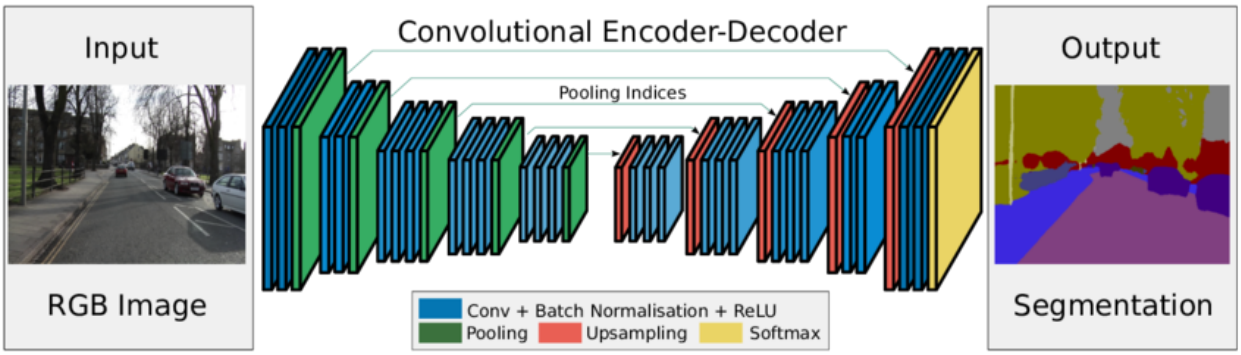


Figure 2.1: SegNet architecture, Image courtesy: [1, Vijay Badrinarayanan, Alex Kendall, Roberto]

Suppose a CRF is defined on observations X and random variables Y. Let G = (V , E) be a graph such that Y is indexed at the vertices of the graph. Then (X,Y) is a conditional random field conditioned on X and obey the Markov property w.r.t. the graph such that,

$$P(Y_v|X, Y_w, w \neq v) = P(Y_v|X, Y_u, u \sim v) \tag{2.1}$$

where $u \sim v$ means that u and v are neighbours in G.

2.3.1 Conditional Random Fields as Recurrent Neural Network

Conditional Random Fields (CRFs) based probabilistic modelling and Conditional Neural Networks are combined to perform pixel-level labelling tasks such as Semantic Segmentation where we use the mean field approximation for the Conditional Random fields with Gaussian pairwise potential obtained from RNN.

The key idea for CRF inference in semantic segmentation is to have a probablistic inference of label assignment of similar pixels in an image. CRF inference helps to refine the course pixel-level label prediction to get sharp image segmented boundaries through improved label prediction.

Let X_i be a random variable associated with the i^{th} pixel of an image representing its corresponding pixel label. Let X be the vector consisting of labels of all the pixels. Let **G(V,E)** be a graph where $V = \{X_1, X_2, ..., X_n\}$ conditioned over the image **I**. The pair **(I, X)** can be modelled as a CRF characterized by a Gibbs distribution of the form

$$P(X = x|I) = \frac{1}{Z(I)} exp(E(x|I)) \tag{2.2}$$

where $E(x)$ is the energy of the configuration with x belongs to the set of possible labels. $Z(I)$ is the partition function. Therefore in the fully connected pairwise CRF model, the energy of a label assignment x is given by:

$$E(x) = \sum_i \psi_u(x_i) + \sum_{i<j} \psi_p(x_i, x_j) \tag{2.3}$$

where the unary energy components $\psi_u(x_i)$ measure the cost of the pixel i taking the label x_i, and pairwise energy components $\psi_p(x_i, x_j)$ measure the cost of assigning labels x_i, x_j to pixels i, j simultaneously. Therefore more the cost more unlikely two pixels will have same labels.

For semantic segmenetation the above pairwise potential is formulated as [10, P. Krahenbuhl and V. Koltun]:

$$\psi_p(x_i, x_j) = \mu(x_i, x_j) \sum_{m=1}^{M} w_{(m)} k^G_{(m)}(f_i, f_j) \tag{2.4}$$

Minimizing the above CRF energy $E(x)$ yields the most probable label assignment x for the given image.

2.3.2 Mean-field inference

Since the exact minimization the above energy function is intractable, Mean Field approximation to the CRF distribution is used for approximate maximum posterior marginal inference. In this approximation the CRF distribution P(x) can be approximated to Q(x) which can be written as the product of independent marginal distributions, i.e., $Q(X) = \prod_i Q_i(X_i)$. These $Q_i's$ are approximated by an iterative algorithm:

Algorithm 1: Mean-field in dense CRFs, broken down to common CNN operations [11]

Input : $\mathrm{Q}_i(l) \leftarrow \frac{1}{Z_i} exp(U_i(l))$ for all i

Output: Marginal Probabilities

while *not converged* **do**

- $\tilde{Q}_i^{(m)}(l) \leftarrow \sum_{j \neq i} k^{(m)}(f_i, f_j) Q_j(l)$ for all m
- $\bar{Q}_i(l) \leftarrow \sum_m w^{(m)} \tilde{Q}_i^{(m)}(l)$
- $\hat{Q}_i(l) \leftarrow \sum_{l' \in L} \mu(l, l') \bar{Q}_i(l')$
- $\mathrm{Q}_i(l) \leftarrow U_i(l) - \hat{Q}_i(l)$
- $\mathrm{Q}_i \leftarrow \frac{1}{Z_i} exp(Q_i(l))$

end

Chapter 3

Image Segmentation Approaches

3.1 Image Segmentation using Convolutional Neural Nets

3.1.1 Background

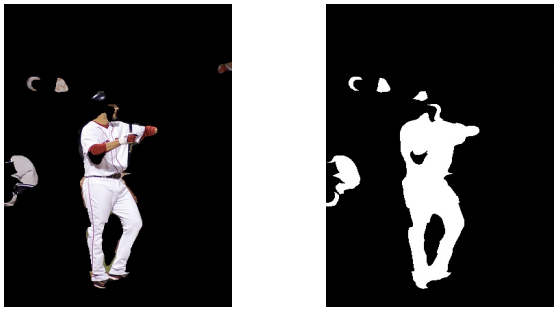
CNNs have demonstrated a good performance in computer vision tasks such as classification where a single label is assigned to the entire image. However, to perform a pixel-based classification it is necessary to break the image down into overlapping patches. Each patch is centered on a pixel which provide the class for the entire patch. CNN is trained using a large set of patches randomly selected trying to maintain a good distribution of the classes of interest. Since all pixels must be processed, the sliding window approach is a time consuming task.

To avoid this we have implemented the approach based on the use of superpixels during labeling process. The Superpixels are used as minimum processing units during labeling process. The superpixels obtained from over-segmentation tend to be similar in size and color, as well as belonging to only one object. These properties allowed to generate superpixels completely contained by a window of determined size, maintaining all the characteristics of a reduced environment inside of the area to analyze with a CNN. The proposed method reduces the number of windows to those centered on the pixels corresponding to the centroids of the superpixels.

Figure 3.1 and Figure 3.2 shows the predicted output of the shown input images. The Segmented output is the collection of superpixels predicted as the foreground superpixel with its generated masks.

3.1.2 Output Segmented test images

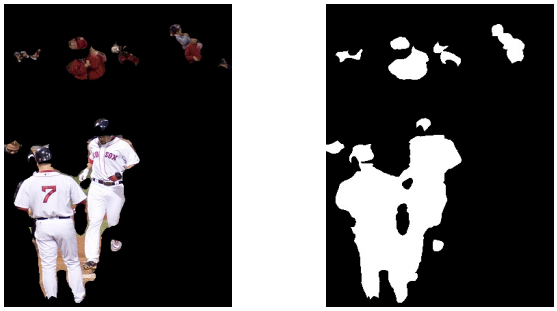
(a) Original image (b) Ground truth



(c) Segmented output (d) Predicted ground truth

Figure 3.1: Red Sox Players image from iCoseg Dataset. (a) is used as the test image

(a) Original image (b) Ground truth



(c) Segmented output (d) Predicted ground truth

Figure 3.2: Red Sox Players from iCoseg Dataset. (a) is used as the test image

3.2 Image Segmentation involving Color Descriptors

3.2.1 Background

In this Image segmentation approach color descriptors are used instead of the intensity based descriptors to extract illumination-invariant features from the image for better segmentation results. These descriptors are further encoded using LLC encoding followed by max-pooling to obtain a single feature vector for each suprepixel of an image.

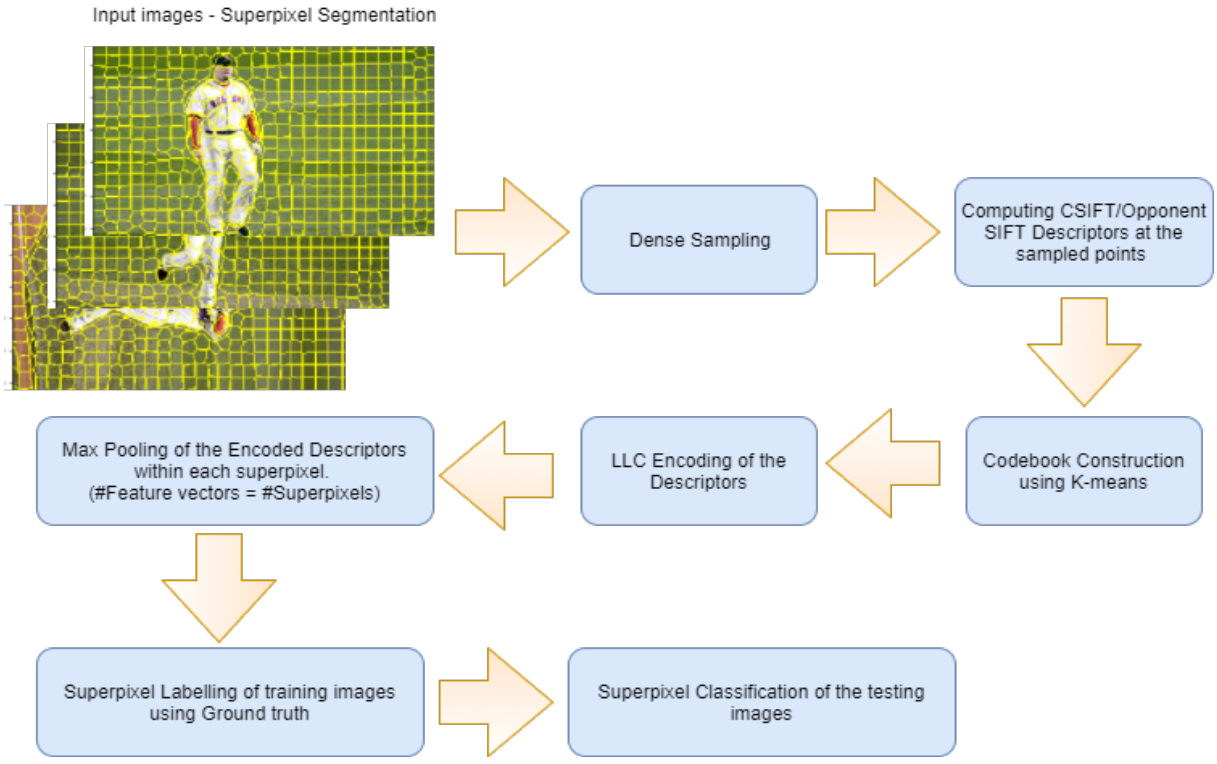
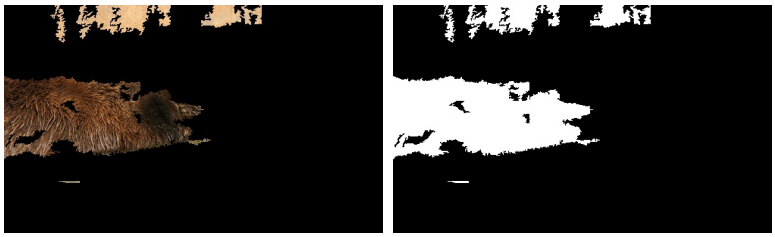


Figure 3.3: Flowsteps for proposed unsupervised image segmentation approach

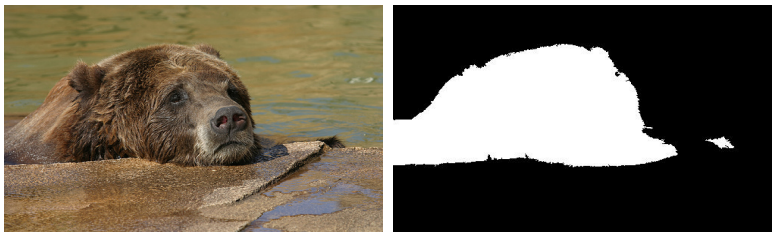
3.2.2 Output Segmented test images

(a) Original image (b) Ground truth



(c) Segmented output (d) Predicted ground truth

Figure 3.4: Alaskan Brown Bear (iCoseg Dataset). (a) is used as the test image while the other images were used as training images from the same category



(a) Original image (b) Ground truth

(c) Segmented output (d) Predicted ground truth

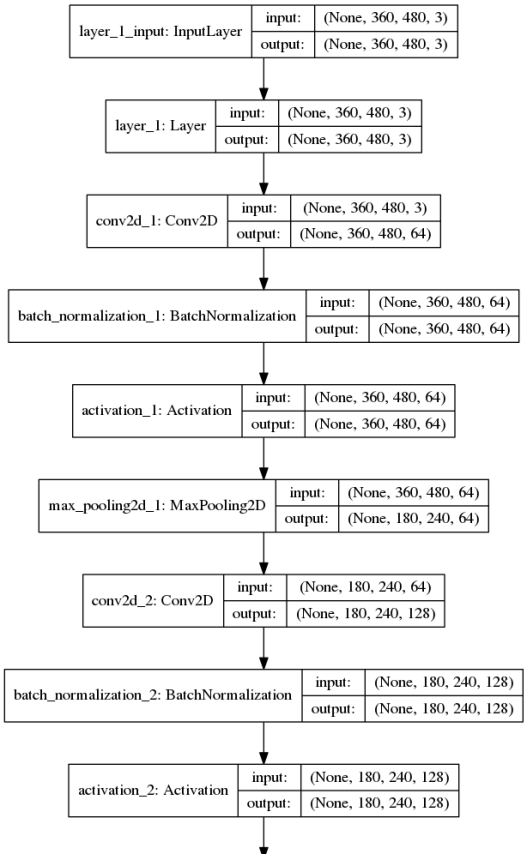
Figure 3.5: Alaskan Brown Bear (iCoseg Dataset). (a) is used as the test image while the other images were used as training images from the same category

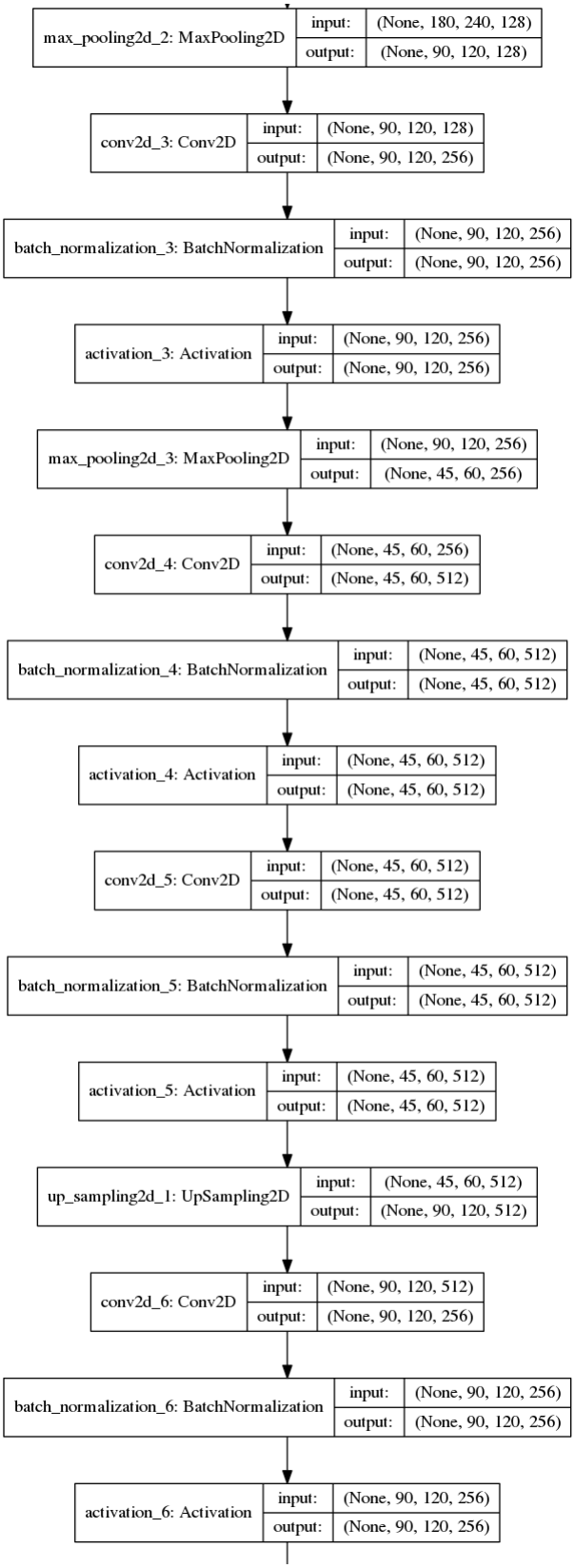
Chapter 4

Semantic Segmentation using SegNet

4.1 About the Model

In order to analyse SegNet and compare its performance we use a smaller version of SegNet, termed as SegNet-Basic, which has 4 encoders and 4 decoders. All the encoders in SegNet-Basic perform max-pooling and subsampling and the corresponding decoders upsample its input using the received max-pooling indices. Basically it's a mini-segnet.




max_pooling2d_2: MaxPooling2D
input: (None, 180, 240, 128)
output: (None, 90, 120, 128)
conv2d_3: Conv2D
input: (None, 90, 120, 128)
output: (None, 90, 120, 256)
batch_normalization_3: BatchNormalization
input: (None, 90, 120, 256)
output: (None, 90, 120, 256)
activation_3: Activation
input: (None, 90, 120, 256)
output: (None, 90, 120, 256)
max_pooling2d_3: MaxPooling2D
input: (None, 90, 120, 256)
output: (None, 45, 60, 256)
conv2d_4: Conv2D
input: (None, 45, 60, 256)
output: (None, 45, 60, 512)
batch_normalization_4: BatchNormalization
input: (None, 45, 60, 512)
output: (None, 45, 60, 512)
activation_4: Activation
input: (None, 45, 60, 512)
output: (None, 45, 60, 512)
conv2d_5: Conv2D
input: (None, 45, 60, 512)
output: (None, 45, 60, 512)
batch_normalization_5: BatchNormalization
input: (None, 45, 60, 512)
output: (None, 45, 60, 512)
activation_5: Activation
input: (None, 45, 60, 512)
output: (None, 45, 60, 512)
up_sampling2d_1: UpSampling2D
input: (None, 45, 60, 512)
output: (None, 90, 120, 512)
conv2d_6: Conv2D
input: (None, 90, 120, 512)
output: (None, 90, 120, 256)
batch_normalization_6: BatchNormalization
input: (None, 90, 120, 256)
output: (None, 90, 120, 256)
activation_6: Activation
input: (None, 90, 120, 256)
output: (None, 90, 120, 256)

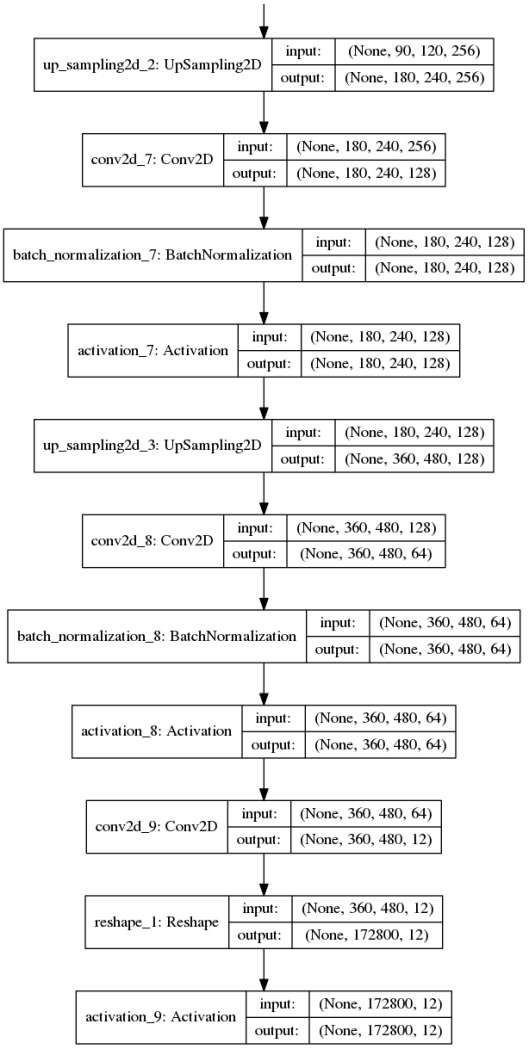


Figure 4.1: Implemented SegNet-Basic Architecture

4.2 Dataset

SegNet-Basic model is trained and tested on The Cambridge-driving Labelled Video Database (CamVid). The database is unique since it is a video sequence and consists of high resolution images. It includes the original frame sequence and the corresponding labelled frames which constitute the groundtruth. In the labeled frames, each object has been painted with a given class color by human operators.

The sequence of images depicts a moving driving scene in the city of Cambridge filmed from a moving car. It consists of 101 images of size 960X720 pixels in which each pixel was manually assigned to one of the 32 object classes that are relevant in a driving environment.

4.3 Implementation details

The described SegNet-Basic model was implemented in Keras with Tensorflow as backend.

Pre-processing of input data The image dataset is read into an array matrix after normalization. From the annotations available we perform the one hot encoding for all the 12 classes in an image and hence the label matrix comes out to be of dimension 360 X 480 X 12 where each 360 X 480 layers have element 1 corresponding to the object position of that specific class as the 3rd index.

Training Training of the above network was done with 367 training and 101 validation images for 50 epochs.

4.4 Results

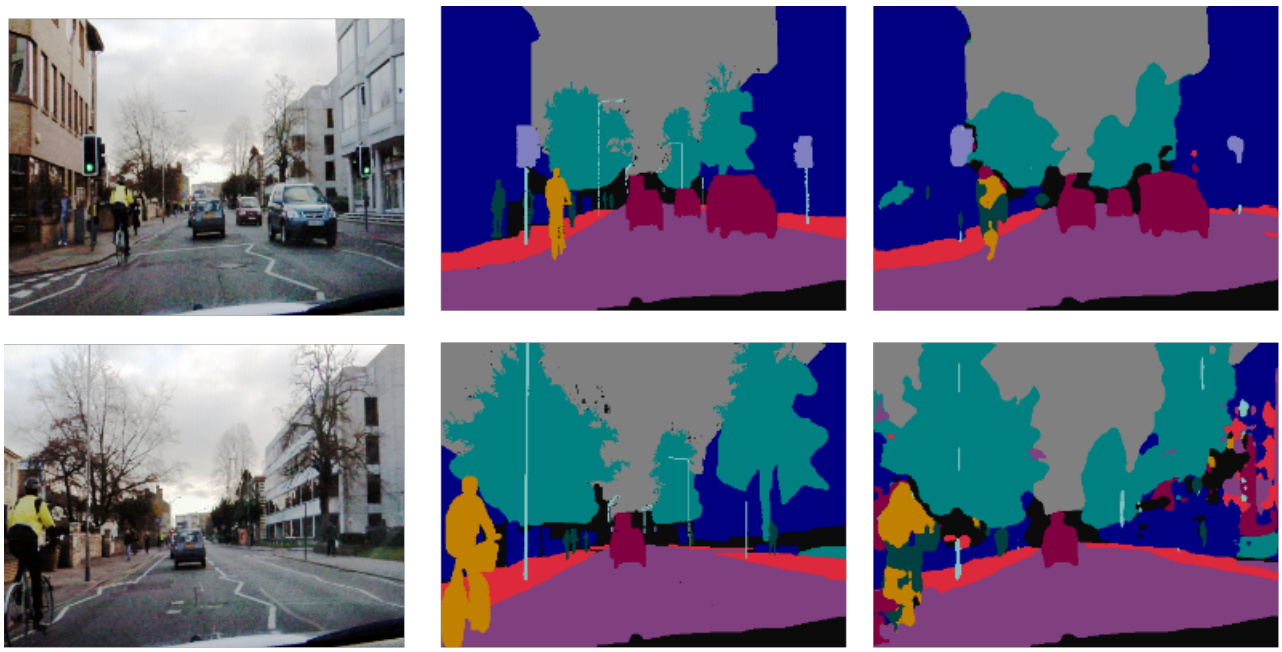


Figure 4.2: SegNet-Basic output

The above SegNet-Basic model gives a test accuracy of 82.26% which is close to 84% as claimed by[1] on images from CamVid dataset. From the above sample test image results we can observe that the predicted output of the SegNet-Basic model have nearly defined boundaries separating different classes.

Chapter 5

CRF based Co-segmentation Approach

5.1 Method

The approach is on the lines of the work by Zehuan Yuan1 , Tong Lu1 and Yirui Wu. on Deep-dense Conditional Random Fields for Object Co-segmentation [6]. In the mentioned paper they introduce a deep-dense conditional random field framework to infer co-occurrence maps which gives the objectness scores of object proposals across images. These co-occurrence maps are further used in object segmentation of the images.

5.2 Model

This model aimed at performing Image Co-segmentation is performed on a set of images having a single common object across all the images. In this method object proposals are generated for each image and the objective is to predict the common object proposal labels for the input object proposals and to further generate their masks.

5.3 Deep-dense Conditional Random Field

In a given set of n images $\{I_1, I_2, ..., I_n\}$, firstly a pool of object proposals are generated for each image. Let G be the total number of object proposals generated. Therefore we can write $O = \{o_1, o_2, ..., o_G\}$ where O represents the set of object proposals. Each object proposal is represented by two variables c_i and m_i where $c_i \in \{0,1\}$. $c_i = 1$ indicates presence of common object in the particular object proposal and 0 if background or uncommon object. m_i represents the masks of the respective object proposal to indicate the pixel locations of potential objects. Hence a deep-dense conditional random field is used to model the joint distribution $\mathbf{P(C,M \mid O)}$ of $C = \{c_1, c_2, ..., c_G\}$ and $M = \{m_1, m_2, ..., m_G\}$ given O,

$$P(C, M|O) = \frac{1}{Z(O)} exp\Big(- \sum_{i=1}^{G} \phi(c_i, m_i \mid o_i) - \sum_{i}^{G} \sum_{j>i}^{G} \psi(c_i, c_j, m_i, m_j \mid o_i, o_j) \Big)$$

(5.1)

where $\phi()$ and $\psi()$ represent the unary term and pairwise potential, respectively.

5.3.1 Unary Potential

The first term in the above equation is the unary potential which represents how likely an object proposal o_i contains a common object with the segmentation mask m_i.

$$\phi(c_i, m_i|o_i) = \phi(c_i|o_i) + \phi(m_i|c_i, o_i) \tag{5.2}$$

5.3.2 Pairwise Potential

Since segmentation masks does not rely on the predictions from other object proposals, the pairwise potential can be written as

$$\psi(c_i, c_j, m_i, m_j|o_i, o_j) = \psi(c_i, c_j|o_i, o_j) = \mu(c_i, c_j)\kappa(f_i, f_j) \tag{5.3}$$

where μ represents the compatibility between the labels and $\kappa(.,.)$ represents the similarity matrix. A typical Potts model is used to represent $\mu(c_i, c_j) = [[c_i \neq c_j]]$.

$$\kappa(f_i, f_j) = exp\Big(- \frac{1}{2}\|f_i - f_j\|^2 \Big) \tag{5.4}$$

5.4 Experiments

5.4.1 CNN architecture

Vgg16 architecture [12, Very Deep Convolutional Networks for Large Scale Image Recognition.] is used as the backbone of this architecture with the fully connected layers removed. This is further divided into two branches where the first branch consists of two fully connected dense layers followed by another branching into two. One models the unary potential $\phi(c_i)$ while the other models the high level features of the object proposal f_i respectively. The unary potentials and the features are then inputed into the CRF inference to get the final objectness score of the object proposals.
The other branch consists of a convolution layer followed by a dense layer. This models the segmentation masks of the object proposals $\phi(m_i)$. In order to make the final output

to have fixed dimensions a 40x40 mask is used. The detailed structure is shown in Figure 4.1

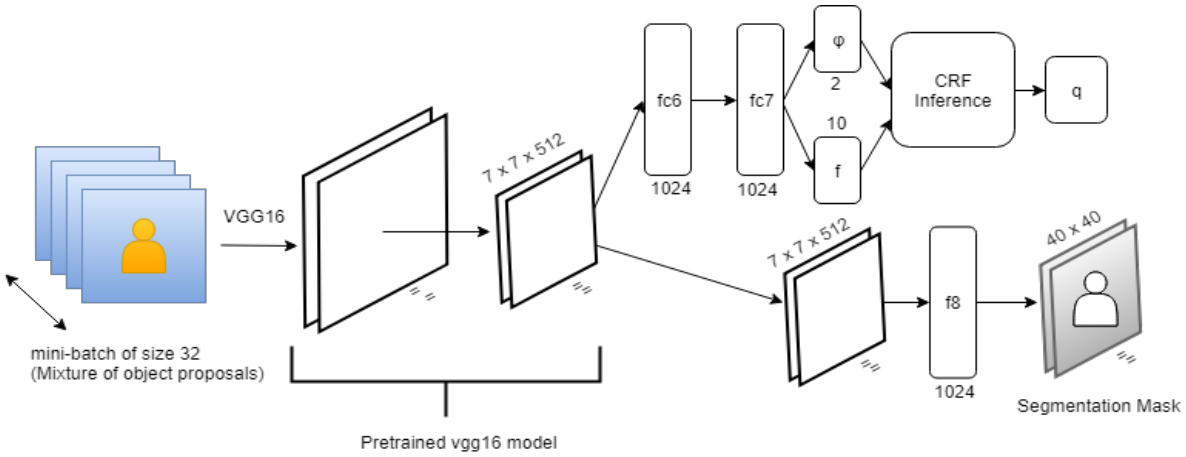


Figure 5.1: Deep Dense CRF architecture used for Co-segmentation inspired from [6]

5.4.2 MAP Estimates

In order to generate the co-occurrence maps we first maximize **P(C,M|O)** and get MAP estimates for object proposals. Using the fact that in pairwise potential the mask generation does not depend upon other proposals, **P(C,M|O)** can be further written as,

$$P(C, M \mid O) \propto exp\Big(\sum_{i}(-\phi(c_i|o_i) - \sum_{j>i}\psi(c_i, c_j|o_i, o_j))\Big)\prod_{i} exp(-\phi(m_i|o_i)) \tag{5.5}$$

Respective masks can be predicted by just doing a forward pass. For C the mean field variational inference [13, Krahenbuhl and Koltun, 2013] is adopted to get MAP estimate. Therefore **P(C|O)** can be approximated to $Q(C) = \prod_i q_i(c_i)$ where the marginal distribution $q_i(c_i)$ is calculated in an iterative way as described in [13, Krahenbuhl and Koltun, 2013].

$$q_i(c_i) \propto exp(-\phi(c_i) - \sum_{j\neq i}\sum_{c_i}\psi(c_i, c_j)q_j(c_j)) \tag{5.6}$$

The iteration continues until the $q_i(c_i)$ converge or reach a threshold. In case of semantic segmentation the pairwise potential can be calculated using convolution. The obtained get $q_i(c_i)$ for any object proposal encode its probability to appear in other image.

Finally the co-occurance map is obatined by max-pooling over the object proposal masks using the estimated marginal probabilities,

$$R(x, y) = \max_{i,(x,y)\in i} q(c_i = 1)m_i(\overline{x}, \overline{y}) \tag{5.7}$$

where $(\overline{x}, \overline{y})$ corresponds to the relative coordinate of (x, y) in o_i.

5.4.3 Dataset

The Deep-dense CRF network is trained on PASCAL VOC 2007 dataset and is tested on the iCoseg dataset which a highly used dataset for Co-segmentation. The PASCAL VOC 2007 dataset consists of 9,963 images in total, containing 24,640 annotated objects. There are total of 20 classes in the dataset.
Since the model is built only for single object co-segmentation, we select only those images from the dataset that contains only one object belonging to any one of the 20 classes. Therefore only 1,865 images are considered for training the classification branch of the network.

5.4.4 Pre-processing of images

For training classification branch For every image in the dataset conataining single object, we first generate object proposals. Selective search algorithm is used to perform this task. These object proposals are then labelled depending on their overlap with the groundtruth bounding box. If the overlap is greater than 0.7 the proposal is labelled as 1 (positives) i.e. containing an object where as the proposals with overlap lying between 0.1 to 0.4 are labelled as 0 (negative) i.e does not contain an object. Other than these some of the proposals without any overlap with the groundtruth bounding box are also considered but are less in number (approx total (positive samples)/2).

For training CRF inference layer and Segmentation branch In the PASCAL VOC 2007 dataset we only have 422 images with segmenatation masks out of which only 131 images contain single object hence these images are used for training the segmentation branch. While training the CRF branch we input object proposals of images belonging to the same object class hence the labelling of the object proposals can now be defined as whether a proposal belongs to the common object class having overlap with the groundtruth bounding box as greater than 0.7. And those having overlap with the groundtruth bounding box between 0.1 and 0.3 as 0. Some of the proposals belonging to uncommon classes are also considered but are less in number.

5.4.5 Training Methodology

A pretrained VGG16 model is used as the backbone of the architecture. The pre-trained weights without batch normalization is used for the VGG16 architecture. The weights of the remaining layers of this pre-trained network are intialized from a zero-mean Gaussian distribution with the standard deviation 0.01. Two stage training strategy is being used in the implementation.

Training stage-I In the first stage the network is trained to perform object proposal classification without involving the pairwise term as described in [14, Pinheiro et al.,

Table 5.1: Training stage-II Class weights

w_0 / Common Object class	w_1 / Uncommon/Background class
0.0025	0.001

2015]. In this case the training reduces to classifying object proposals.
Even though we have only 131 images with given segmentation masks in the dataset we still can use images and their proposals without their segmentation masks since we require only ground truth bounding boxes of each images in this stage. This leaves a total of 1865 images with single object out of the entire dataset.
Initially the training dataset is taken to be small and is allowed to overfit so as to confirm that the fine tuning of the model is done correctly. After this result comes out to be desirable, futher finetuning with bigger dataset is done as specified earlier.

Training stage-II In the second stage of training we introduce the pairwise term modelled using CRF inference. In this case two images are used at a time containing common object. Object proposals are generated for each image such that those with overlapping threshold more than 0.7 with the groundtruth bounding box are labelled as 1 (positive samples) whereas the object proposals with the overlapping threshold lying between 0.1-0.3 are labelled as 0 (negative samples). We take approximately equal number of positive and negative samples to avoid any kind of bias while training.

5.5 Results

Figure 6.2 contains two screenshots of part of the matrix containing the L2 norm of the feature vector distance between 2 proposals in a mini-batch. Both the image belong to consecutive epochs. As per the equation 6.3 as we train the model its task is to maximize the similarity between very similar proposals that is those having different labelling after the classification output. Hence theoretically the L2 normalization of the distance feature vector should increase at every iteration until the loss is minimized.
It can be observed that most of the L2 norm values shows increment whereas some doesn't. The reason behind some of the values showing undesirable behavior is due to the non-convergence of the previous layer classification loss which keeps on fluctuating.

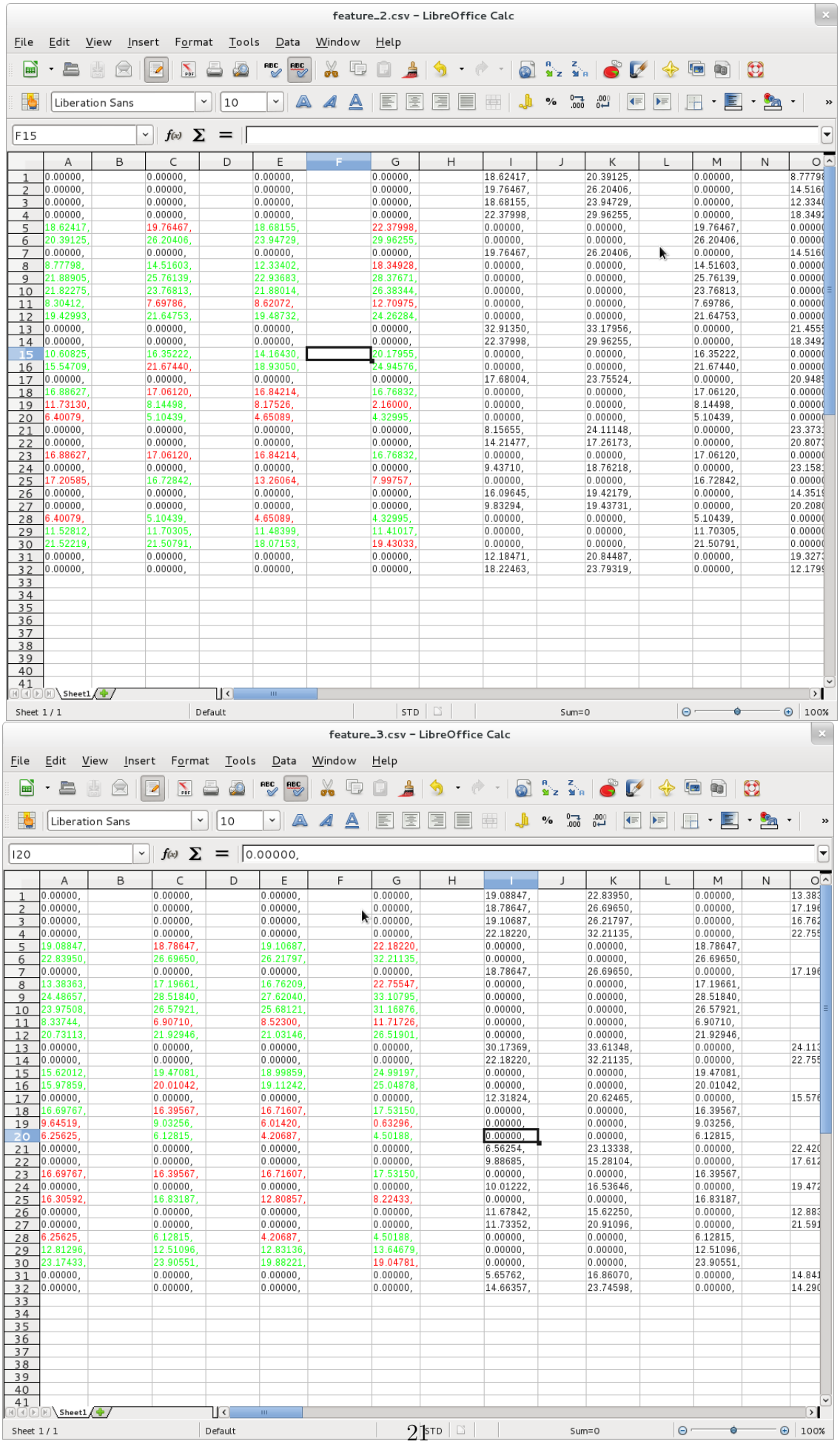


Figure 5.2: Feature distance matrix

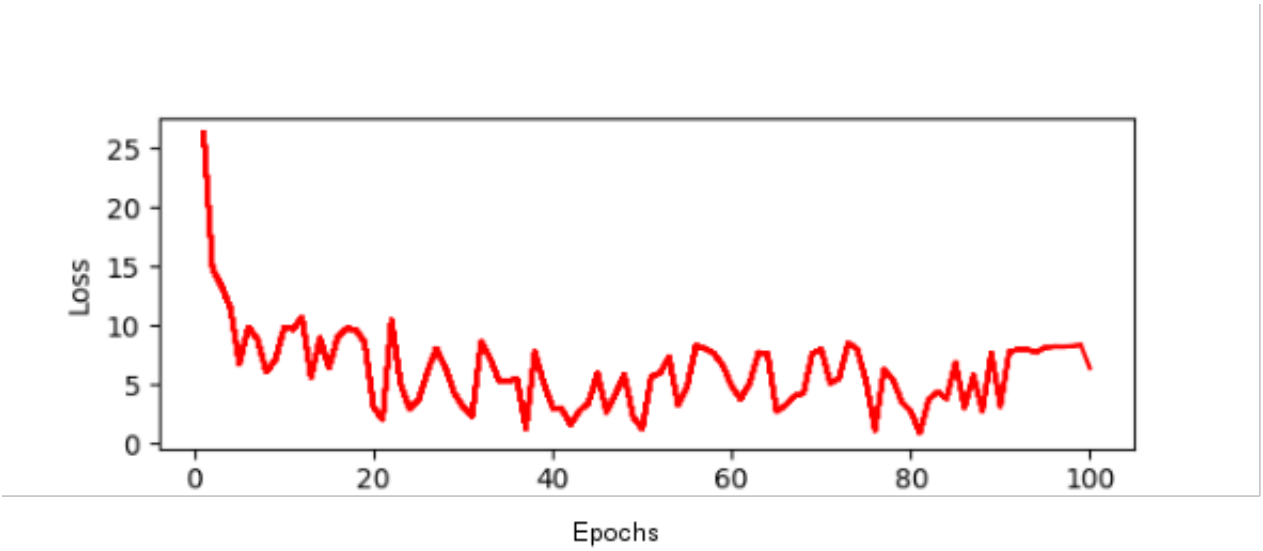


Figure 5.3: Training CrossEntropy Loss for CRF layer output

Figure 6.3 shows that the Cross entropy loss is unable to converge to a final stable value and keeps on fluctuating with increasing epochs. The reason for this lies in the criteria of labelling the positives and the negatives proposals while training.

5.5.1 Testing sample images

a.) Alaskan Brown Bear image

b.) Goose-Riverside image

Figure 5.4: a.) Alaskan Brown Bear b.) Goose-Riverside (iCoseg Dataset)

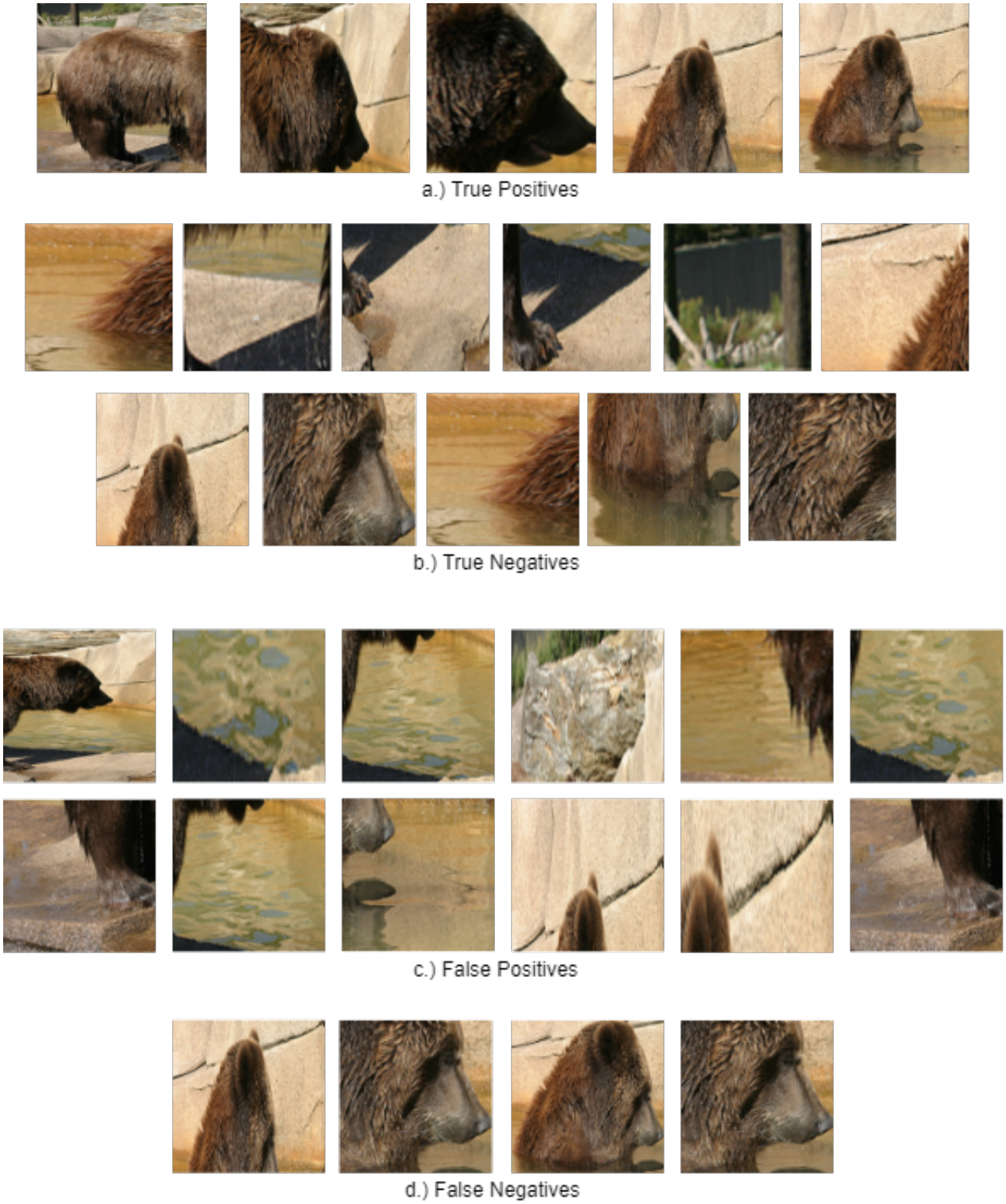


Figure 5.5: Predicted Common Object Proposals - Alaskan Brown Bear(iCoseg Dataset)

Figure 5.6: Predicted Common Object Proposals - Goose Riverside (iCoseg Dataset)

5.6 Conclusions

It is observed that the CrossEntropy loss calculated during the training of the classification output ϕ couldn't converge to a stable value and kept fluctuating. One of the reasons for such a happening lies in the way the training dataset is created. The positive object proposals that is the ones with more than 0.7 overlap with the groundtruth bounding box tended to be very similar to the negative object proposals with overlap lying between 0.3 to 0.1 with the groundtruth bounding box. This makes the network difficult to different the positive and negative proposals.
While generating the object proposals for the training dataset we get only a few positive object proposals as compared to negative object proposals which also affects the stability of the network.

Training the segmentation branch for generating masks of the predicted common class labelled object proposals. Some of the possible advances that can be done to this algorithm is to extend this for multiple image co-segmentation since the current method only works for a pair of images given as input. This can further be extended to perform multiple foreground co-segmentation.
Right now the network is trained on only a part of the entire PASCAL VOC 2007 dataset. Hence training it for the entire dataset.

Chapter 6

Conclusions and Future Work

With the aim of developing an end-to-end supervised co-segmentation technique we first looked into various segmentation techniques using both supervised and unsupervised methods. The main aim for doing this was to analyze both supervised as well as unsupervised ways of retaining visual input data and how is one better than other. We find that the unsupervised method using the color descriptor features did not give good segmentation results whereas the supervised method using CNN network gave better segmentation accuracy but requires a very large dataset for training the model.
Further looking into segmentation we see that SegNet has given really good results in road scene image semantic segmentation with 82.26% as the test accuracy. Such high accuracy signifies the uniqueness of the feature extracted for each class leading to better segmentation. CRF based image co-segmentation has given good results. The exact accuracy of the predicted mask can obtained after we generate masks for each of the predicted common object proposal hence this remains as the future work.

References

[1] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, pp. 2481–2495, Dec 2017.

[2] C. Gonzalo-Martin, A. Garcia-Pedrero, M. Lillo, and E. Menasalvas, "Deep learning for superpixel-based classification of remote sensing images," 09 2016.

[3] A. Joulin, F. Bach, and J. Ponce, "Discriminative clustering for image co-segmentation," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 1943–1950, June 2010.

[4] R. Quan, J. Han, D. Zhang, and F. Nie, "Object co-segmentation via graph optimized-flexible manifold ranking," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 687–695, June 2016.

[5] D. Batra, A. Kowdle, D. Parikh, J. Luo, and T. Chen, "icoseg: Interactive co-segmentation with intelligent scribble guidance," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 3169–3176, June 2010.

[6] Z. Yuan, T. Lu, and Y. Wu, "Deep-dense conditional random fields for object co-segmentation," in *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pp. 3371–3377, AAAI Press, 2017.

[7] K. E. A. van de Sande, T. Gevers, and C. G. M. Snoek, "Evaluating color descriptors for object and scene recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1582–1596, 2010.

[8] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International Journal of Computer Vision*, vol. 88, pp. 303–338, June 2010.

[9] R. Gray, "Vector quantization," *IEEE ASSP Magazine*, vol. 1, pp. 4–29, April 1984.

[10] P. Krähenbühl and V. Koltun, "Efficient inference in fully connected crfs with gaussian edge potentials," *CoRR*, vol. abs/1210.5644, 2012.

[11] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. H. S. Torr, "Conditional random fields as recurrent neural networks," *CoRR*, vol. abs/1502.03240, 2015.

[12] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014.

[13] P. Krähenbühl and V. Koltun, "Parameter learning and convergent inference for dense random fields," in *Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28*, ICML'13, pp. III–513–III–521, JMLR.org, 2013.

[14] P. H. O. Pinheiro, R. Collobert, and P. Dollár, "Learning to segment object candidates," *CoRR*, vol. abs/1506.06204, 2015.