Semi-Supervised Image Segmentation

CS 726 Course Project

Sanjeev Mk, 14305R002

Eeshan Malhotra, 14305R001

Mihir Kulkarni, 12D020007

Ashish Goyal, 12D070051

1. Introduction

The project aims to segment a given image into the 'foreground' – a recognised object and 'background' – everything else, without being trained explicitly on an image segmentation dataset. The segmentation of a given image, then, is a map the same size as the original image, with each pixel containing a value of 1 (object) or 0 (everything else). This is done in three steps or phases:

1. A supervised image classification model that classifies the image into one of a set of 20 pre-defined categories (such as 'airplane', 'car', 'horse', 'duck', etc.)
2. An unsupervised segmentation, that tries to extract the key areas of the image that led to it being classified in a particular class in phase 1.
3. Improving object boundaries using image properties, such as colour gradients.

The key idea is that lets us perform step 2 in an unsupervised fashion is that if a model is able to classify images correctly, it must contain latent information about which pixels in the image are actually influential in making this decision. We just need to develop a method to extract this information.

Multiple approaches were tried for each phase, and are described in the relevant sections.

We used the PASCAL Visual Objects Classes (VOC2012) dataset for evaluation, which contains a large number of segmented images.

2. Existing Work

Lots of work has been done on image classification using deep neural networks Notably, AlexNet [1] surpassed human accuracy and is one of the state-of-the-art models in image classification. Work also exists on supervised image segmentation, but was not extensively surveyed, because the approaches are considerably different.

The idea of extracting segmentation from classifier networks was first described in [2]. The key concept is that the gradient of the correct output label class with respect to each of the input layer nodes (each corresponding to a pixel in the image), will be high (in magnitude) for pixels that

contribute to the image being classified correctly, and low for others. We use this idea, and ideas from [3], and improve on these for our project.

3. Methodology and Approaches Tried

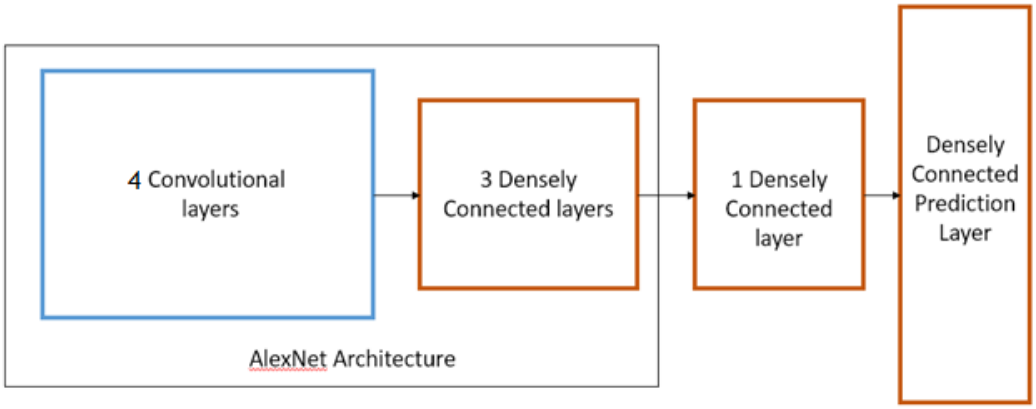
Phase 1: Supervised Image Classification

AlexNet [1] is known to have great classification accuracy. However, this network is trained for specific object classes, which do not match with the object classes in our dataset.

AlexNet's architecture consists of 4 convolution layers, followed by 3 densely connected layers, and one prediction layer. For a first baseline score, we used the same model as AlexNet, replacing the prediction layer with a fully connected layer corresponding to the number of classes in our dataset. Since the original network has hopefully already learnt some important image classification features, such as edges, curves, corners, we utilized these values as the initialization for our network. However, the complete network was trained again, on the VOC2012 dataset.

Our hypothesis was that features specific to the classes in the dataset need to be learnt. To do this, we added another fully connected layer before the prediction layer. The final architecture of this network is: 4 convolution layers, 4 densely connected layers, 1 prediction layer.

Classification Network Architecture



Once again, the initial values for the first 7 layers were used from AlexNet, but all layers were updated during the training phase. This approach gave a significantly higher classification accuracy.

Adding more densely connected layers to the network did not result in any improvement in the accuracy.

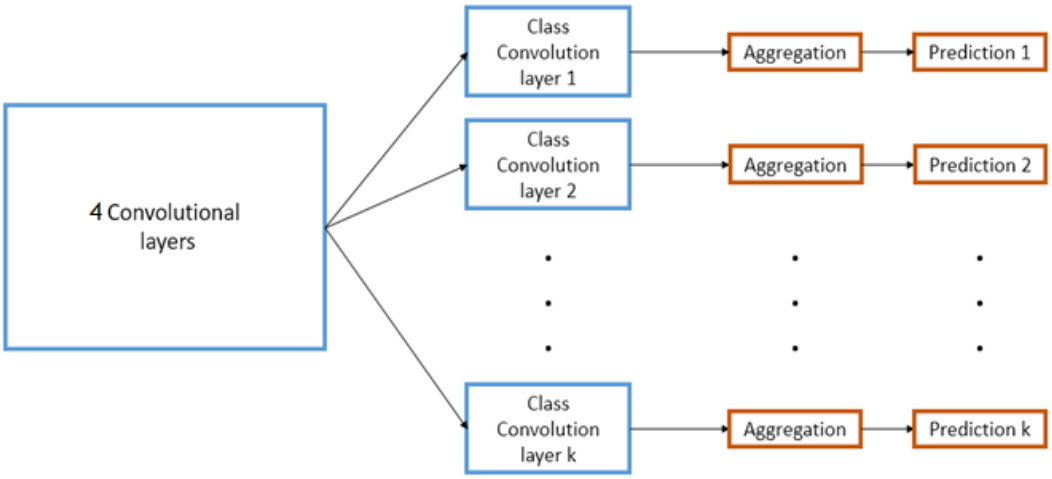
Phase 2: Unsupervised Segmentation Extraction

In the first approach, we used an idea described in [2]. The absolute value of the gradient of the correct output label class with respect to each of the input layer will be high for pixels that contribute to the image being classified correctly. These pixels can be labelled as belonging to the foreground object, while other pixels with low gradient can be labelled as background. An array of the same size as the original image, mapping each pixel to the gradient of the output class with

respect to that pixel is called a saliency map, and can be obtained directly from the trained classification network from Phase 1

A completely different approach (described in [3]) adds one convolution layer for each class directly after the initial 4 convolutional layers. Since all the layers so far are convolutional layers, every element of the output can be mapped to a pixel-set in the input image. The aggregation layer simply takes an average, to go from pixel level classification scores to image level classification scores. For each class, the output of the aggregation layer is transformed to the classification probability for that class.

In essence, here, we are training the Class convolution layer to learn the image segmentation, *while* training the classification network.



Phase 3: Object boundary improvement

The output saliency map is not perfect, because often pixels in the near vicinity of the relevant object are also responsible for correct classification. Three approaches were tried to improve upon the segmentation boundaries:

i. Denoising and thresholding
ii. CRF for bilateral filtering
iii. Denoising + thresholding + CRF for bilateral filtering

For approaches (ii) and (iii), the hyperparameters for CRF were trained using a grid-search approach.

4. Experiments and Results

Phase 1: Image Classification

The two networks as described in section 3 were tried

Architecture	Train Accuracy	Validation Accuracy
Baseline network (AlexNet retrained)	78.4%	70.2%
Final Network (one additional densely connected layer)	99.2%	82.3%

Adding more layers to the network did not improve accuracy

Phase 2 & 3: Unsupervised Segmentation Extraction and Object Boundary Improvement

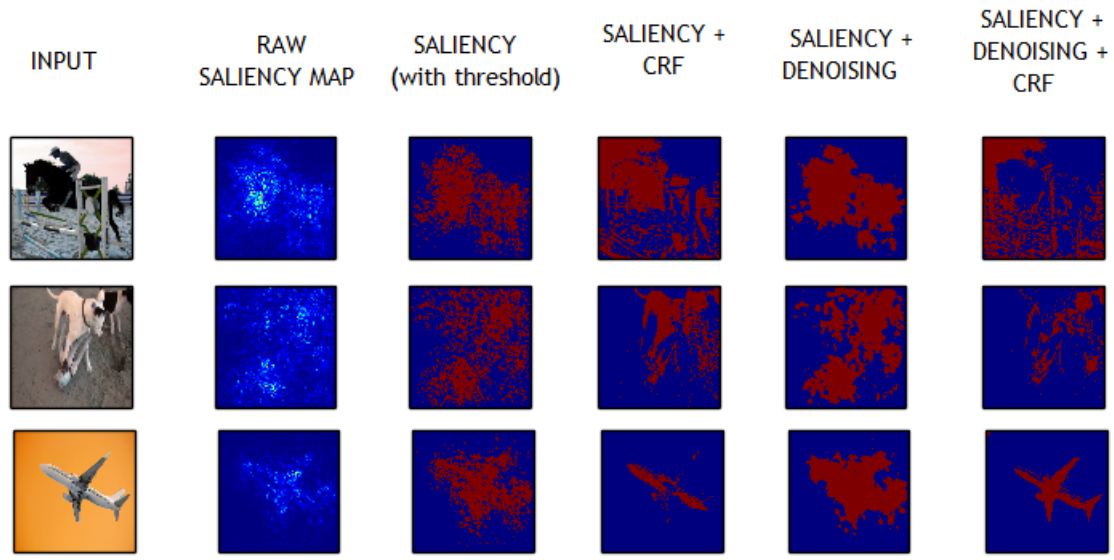
Saliency based methods

Technique	Precision (Train)	Precision (Test)	Recall (Train)	Recall (Test)	F1 (Train)	F1 (Test)
Raw Saliency Map	**0.44**	**0.53**	**0.62**	**0.57**	**0.51**	**0.55**
Saliency Map + Gaussian Denoising	**0.48**	**0.53**	**0.75**	**0.69**	**0.59**	**0.60**
Saliency Map + Tuned CRF	**0.32**	**0.27**	**0.51**	**0.51**	**0.39**	**0.35**
Saliency Map + Gaussian Denoising + Tuned CRF	**0.37**	**0.28**	**0.48**	**0.48**	**0.41**	**0.35**

Joint Segmentation and classification network-based methods

Technique	Precision (Train)	Precision (Test)	Recall (Train)	Recall (Test)	F1 (Train)	F1 (Test)
Joint Training Network	**0.55**	**0.61**	**0.53**	**0.54**	**0.54**	**0.57**
Joint Training Network + Gaussian Denoising	**0.58**	**0.61**	**0.54**	**0.55**	**0.56**	**0.58**

Some sample images help understand the performance of each method a little better



Gaussian denoising almost always improves the predicted segmentation map. However, the uncertain nature of the CRF yields great results at times (case 3), where the object boundaries are well-defined, but completely obliterates the information value of the saliency map when the background is noisy (case 1). The Saliency+CRF result for case 2 is quite good, but the Denoising operation catches a lot more background pixels as well, which confuse the CRF.

The entire codebase and set of trained models is included at
https://drive.google.com/open?id=0B5lIX_IG0rxJQ3pHaDJyMlZlRTA

5. Training Setup and Time

Deep neural network models were trained on a machine with 8 GB Quadro M4000 GPU, 32 GB RAM

Machine Time consumed

Classification network training (5700 images): 25 minutes

Using stochastic gradient descent, with learning rate = 0.001

Joint classification+segmentation network training (5700 images): 12 minutes

Using stochastic gradient descent, with learning rate = 0.001

CRF training (per image): ~3 seconds

6. Effort

Fraction of time spent on different parts of the project

- Augmenting AlexNet for classification: 25%
- Segmentation recovery using Saliency maps: 30%
- Post-processing (blurring, CRF tuning): 15%
- Developing Joint classification+segmentation network: 30%

Developing the joint network architecture proved to be the most complex part of the whole process, although the rewards were worth it. The work was more-or-less equally split among all team members.

7. References

1. Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*
2. Simonyan, K., Vedaldi, A., & Zisserman, A. (2013). Deep inside convolutional networks: Visualising image classification models and saliency maps. arXiv preprint arXiv:1312.6034.
3. Pinheiro, P. O., & Collobert, R. (2015). From Image-level to Pixel-level Labeling with Convolutional Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 1713-1721)
4. Koltun, V. (2011). Efficient Inference in Fully-Connected CRFs with Gaussian Edge Potentials. *Advances in Neural Information Processing Systems*.