CS518 Project Report: Image Saliency Prediction

Amritpal Singh
2017csb1068@iitrpr.ac.in
Shivam Prasad
2017csb1108@iitrpr.ac.in

Indian Institute of Technology
Ropar, India

Abstract

Saliency describe the spatial locations in an image that attract human eye gaze. There are various approaches to generate saliency map, here we try to explore the use of one such method which uses convolutional neural networks, MLNet [2].

The model combines features extracted at different levels of the CNN and the network tries to minimize the loss function through stochastic gradient descent. The model is trained and tested on the SALICON dataset [1]. This approach differs from other CNN based approaches in that it does not employ fully convolutional networks and extracts feature maps from different levels of the network.

1 Introduction

Humans are capable of quickly focus the eyes on distinctive visual regions. There are two types of vision attention, one is bottom-up, and other is top-down. The former helps us in making quick observations and decisions while latter is driven by factors such as prior knowledge of person. In this paper we focus on bottom-up saliency driven attention for machines.

Saliency prediction is important because it has application in various fields like computer vision, robotics and graphics, for example automatic cropping, automatic thumbnailing and content-based image retrieval, etc. Also that, every minute a humongous amount of data is being generated, these images contain lot of redundant data, if we focus only on the salient portion of the image the data intensive tasks can be made more efficient.

In this report we present a deep learning approach to predict the saliency prediction. Deep learning is a class of machine learning used to extract higher level features from a raw image. Deep learning approach show impressive accuracy scores in the vision related problems like classification, segmentation, saliency. We have implemented the Multi Level Network for Saliency Prediction [2] in our project using pytorch library.

The pretrained VGG-16 model is used and we add extra layers to train. Along with this, the network learns its own prior and uses its own loss function to train.

We used SALICON [1] and MIT 300 dataset. SALICON dataset contain 10,000 training, 5,000 validation and 5000 test images along with saliency annotation maps of train and validation. MIT 300 is the dataset containing 300 natural images with eye tracking data from 39 observers.

2 Literature Review

Saliency prediction has been widely studied in the previous decade and many novel methods have been proposed such as Hou [3], which use fully convolutional neural networks and holistically nested edge detectors to provide multi scale feature maps at each layer.

Some very recent methodologies such as SalGAN [4], adopt an adversarial training approach with binary cross entropy (BCE) loss function for training the deep convolutional network. Some heuristic based approaches which use features of images such as prominent corners as the more salient features of the image have also achieved appreciable scores on the MIT300 Saliency Benchmark.

Most of the CNN based approaches use Fully Convolutional Networks which predict the saliency maps based on high level features obtained from the last convolutional layer, but MLNet [2] uses features extracted at different levels of the network and shows the efficiency of using medium level features.

3 Method

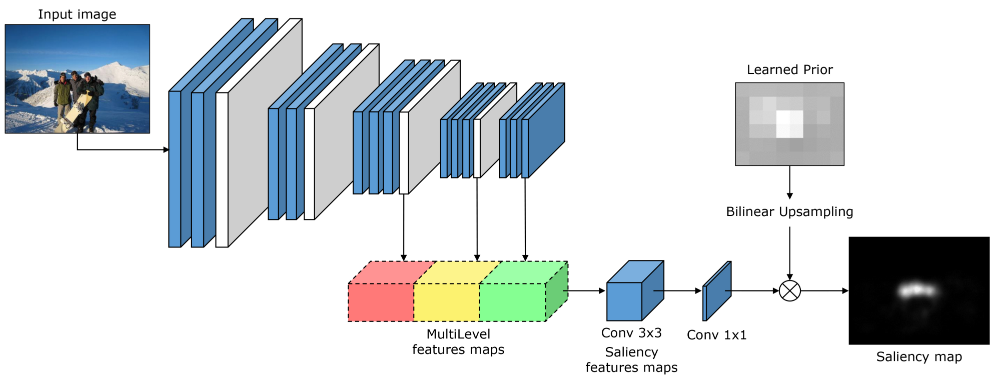


Figure 1: Overview of the model

We take the transfer learning approach to build our model architecture. Transfer learning is a machine learning method where a model developed for a task is reused as the starting point for a model on a second task. Transfer learning speeds up the training and improve the performance of the deep learning model, even when we have less data. We use VGG-16 as our starting point. VGG16 is a convolutional neural network model proposed by K. Simonyan and A. Zisserman. Along with being simple and elegant, it yields results comparable to state of the art results, but there is a drawback of this model too. It reduces the size of feature maps in higher levels. To face this drawback we removed the last max pool layer and changed the stride of the second last max pool layer to 1. This results in a final saliency map of size [W/8, H/8] instead of [W/32, H/32] where W is width and H is the height of the input image.

Feature maps of 3 different locations are taken which are, output of third max pool layer(containing 256 feature maps), last max pool layer(containing 512 feature maps) and the output of the last convolution layer(containing 512 feature maps). The spacial size of their feature maps is the same, so we concatenate these to form a tensor with 1280 channels which is fed to a dropout layer with a retain probability parameter of 0.5 for better generalization. This finally learns 64 saliency-specific feature maps with a 3 x 3 kernel. Its convolution with 1 x 1 kernel produces a saliency map.

The saliency map produced is then again convolved with the prior. Prior is a mask that is initialized to ones. It is then upsampled and convolved with the saliency map to produce the final predicted saliency map. Prior is also learned like all other parameters in the backpropagation step.

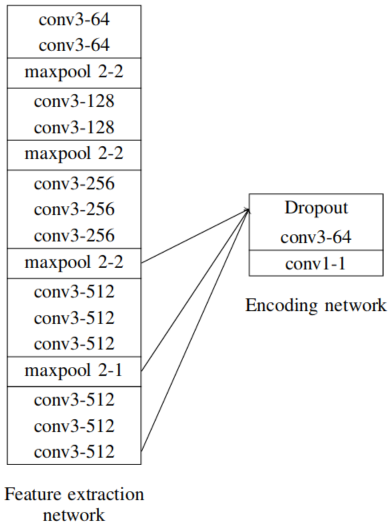


Figure 2: Architecture of the model

3.1 Loss Function

During training, the predictions are made on the random minibatch containing the training images and their saliency maps. Our main motive is to minimize the loss calculated by our loss function using Stochastic Gradient Descent.

$$L(\mathbf{w}) = \frac{1}{N}\sum_{i=1}^{N}\left\| \frac{\frac{\phi(\mathbf{x}_i)}{\max \phi(\mathbf{x}_i)} - \mathbf{y}_i}{\alpha - \mathbf{y}_i} \right\|^2 + \lambda \|\mathbf{1} - U\|^2$$

Figure 3: Loss function without prior

x_i is our predicted saliency map, y_i is the ground truth, and λ is the L2 regularization coefficient.Loss function has 3 main objectives, one is that the predictions should be pixel-wise similar to ground-truth maps, therefore a square error term is used. Second is that predicted maps should be invariant to the maximum value so the predicted results should be normalized. The third is that the loss should give the same importance to high and low ground truth values, so divided by α - y_i. The L2 regularization term is added to encourage the network to adapt to ground-truth maps by changing convolutional weights rather than modifying the prior.

3.2 Experiments

We used pytorch for the implementation of this project because of the availability of the pretrained VGG16 and ease of use while creating the layers of the network. Google Colab was used to train the network over the Salicon dataset. While the original paper trained the images on their original sizes and does not freeze layers while training, we resized the images to a 320*240 size and froze the first 23 layers of the pretrained network to be able to train the network at a faster rate and without running out of memory. The batch size was taken as 16.

Based on the observation and intuition that the prior was always being initialised by ones and was producing only minor changes in the obtained mask after convolution, we decided to forego the prior learning and changed the loss function. This produced interesting results as the time needed to train the network reduced significantly (by upto 1/3) and the obtained model still produced very competitive saliency maps. In this experiment, the L2 regularisation term is omitted from the loss function and the weighted mean square error is minimised using stochastic gradient descent.

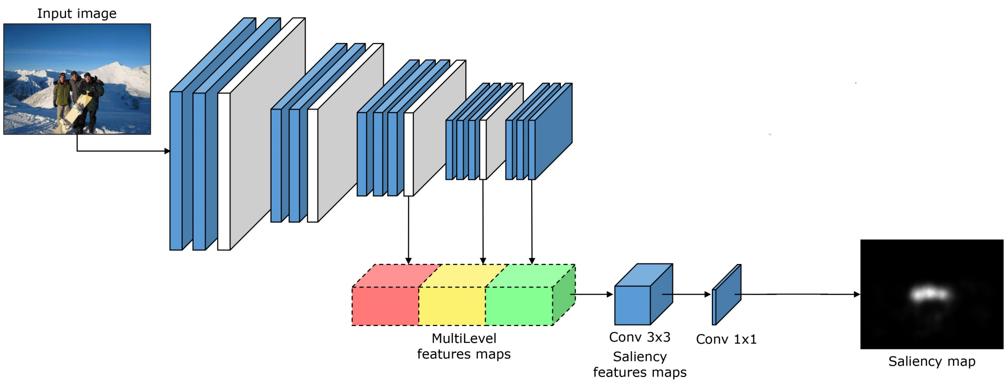


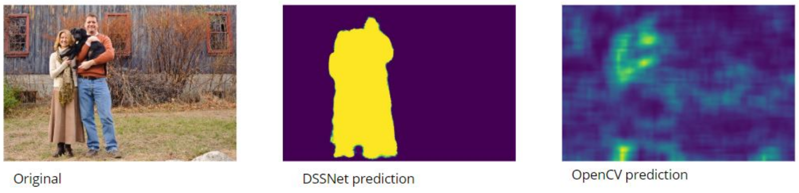
Figure 4: Experiment model without prior

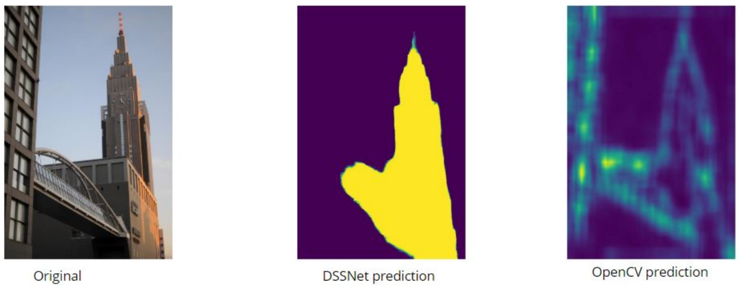
$$L(\mathbf{w}) = \frac{1}{N}\sum_{i=1}^{N}\left\|\frac{\frac{\phi(\mathbf{x}_i)}{\max\phi(\mathbf{x}_i)} - \mathbf{y}_i}{\alpha - \mathbf{y}_i}\right\|^2$$

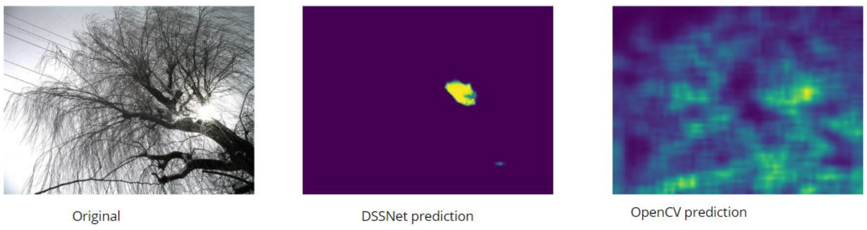
Figure 5: Experiment Loss function without prior

3.3 Experiments on other models

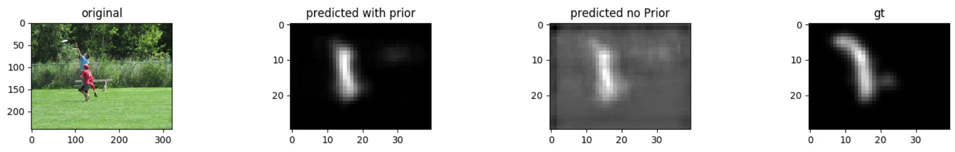
We tried to run and get output of some other methods of image saliency detection such as DSSNet [3] and OpenCV's built in saliency detection, but the achieved results were highly variable as DSSNet failed to identify salient features in images with inanimate subjects and OpenCV produced very hazy outputs. Both showed heavy bias towards human subjects. DSSNet was unable to identify salient features in sceneries whereas OpenCV produced ouputs of mostle edges in images having buildings or complex scenery.

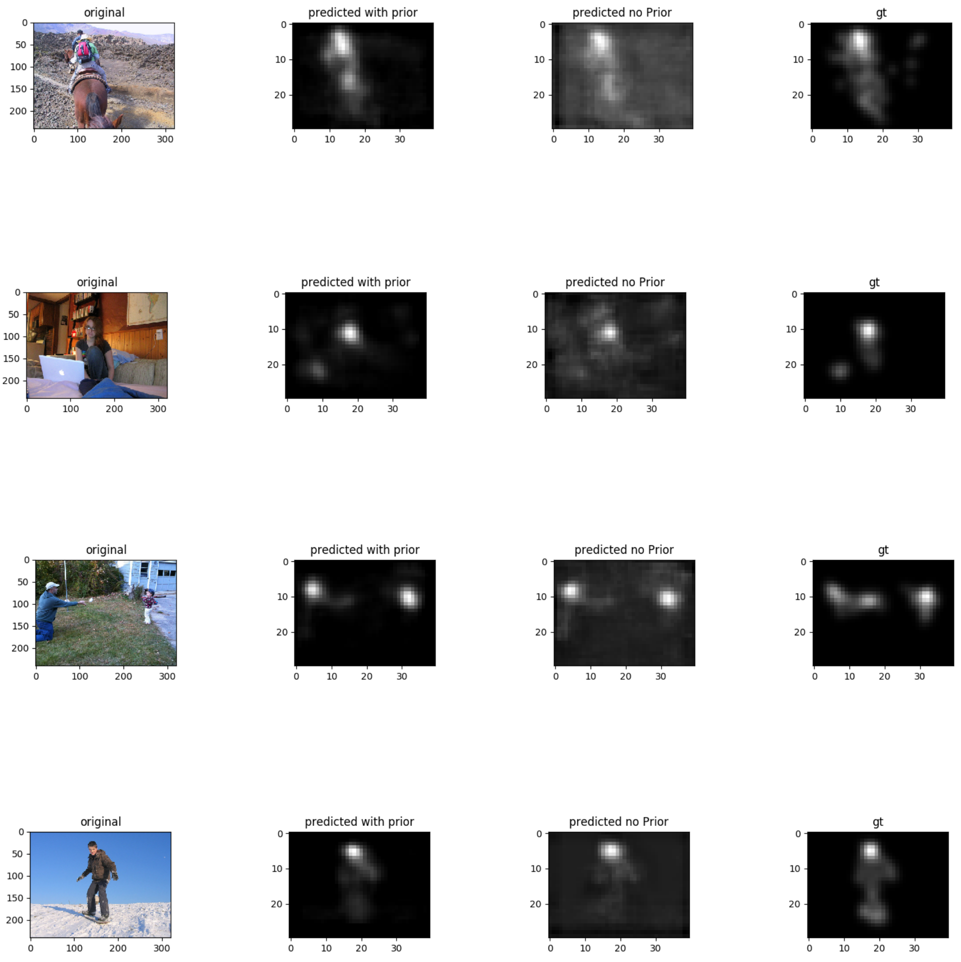






4 Outputs





Here gt denotes ground truth
We can observe that the experimental model performs well in some cases but produces hazy output compared to original.

5 Conclusion

We were able to obtain highly competitive saliency map outputs from the implemented model as well as from the experiment. The model performs well in conditions where the background is well separated from the salient regions but it is slightly biased towards the brightest regions of the image. The datasets are also slightly biased towards the central regions of the images as the data was collected by tracking the eyegaze of the human subjects while viewing the images on a screen. The model is also fast and can even be used on videos or realtime camera videos for tracking the salient regions. We tested the model on some videos and it was able to produce the video with salient maps with a slight drop in the framerate. The transfer learning on VVG16 dataset after freezing 23 layers also enables us to train the model within 3 hours on Google Colab. The performance of this model can be compared with state of the art models as the combination of medium and high level features extracted from the CNN make it robust.

The performance could possibly be further improved by freezing less number of layers in the network and taking the training set images in their original sizes, but the training time and processing power increases. By implementing this model we observe that good results can be obtained by using a concatenation of low level and high level features from different levels of the CNN instead of depending on only the high level features.

References

[1] SALICON dataset , 2017.

[2] Marcella Cornia, Lorenzo Baraldi, Giuseppe Serra, and Rita Cucchiara. A Deep Multi-Level Network for Saliency Prediction. In *International Conference on Pattern Recognition (ICPR)*, 2016.

[3] Qibin Hou, Ming-Ming Cheng, Xiaowei Hu, Ali Borji, Zhuowen Tu, and Philip H. S. Torr. Deeply supervised salient object detection with short connections. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(4):815âĂŞ828, Apr 2019. ISSN 1939-3539. doi: 10.1109/tpami.2018.2815688. URL `http://dx.doi.org/10.1109/TPAMI.2018.2815688`.

[4] Junting Pan, Cristian Canton Ferrer, Kevin McGuinness, Noel E. O'Connor, Jordi Torres, Elisa Sayrol, and Xavier Giro i Nieto. Salgan: Visual saliency prediction with generative adversarial networks, 2017.