Shallow Convolutional Networks for Saliency Predictions

Sanyam Vyas
Faculty of Engineering
University of Bristol
Bristol, United Kingdom
sv17657@bristol.ac.uk

***Abstract*— The standard estimate of salient points in images is predominantly addressed in the field of neuroscience. However, recent success from deep learning architectures have been used as a regression problem to generate saliency maps that are very similar to the ground truth. With the recent increase in the datasets online available for saliency prediction over the last decade, several researchers have contributed to the area of saliency prediction. This paper addresses the saliency problem (inspired by Pan et al's [5] work with the implementation of a shallow convolutional neural network. The implemented model utilises the SALICON [6] dataset for training and validation. The implementation replicates Pan et al's shallow architecture through the PyTorch library as opposed to Theano.**

Keywords—deep-learning, neural-networks, saliency

I. INTRODUCTION

The human eyes prowess the capability of visually processing information at very high rate. A very important mechanism of the human visual system is the capability to specifically focus on the parts of image/video's salient areas. While this research area has conventionally been tackled in neuroscience, machine learning techniques have been able to produce a genuine attempt to replicate the saliency in a regular visual system of a human. Recently, a surge of data-driven, deep learning models have been used to detect image saliency through deep neural networks. The deep models have produced superior performance to the traditional machine learning models used in this field. Specifically, Convolutional Neural Networks (CNN) have been widely used within image classification research areas. Structural CNN models endeavour to replicate the behaviour of regular visual models of humans. With training of datasets for a particular purpose, the CNN models have been able to produce very high performance in recent times.

Within the saliency prediction research area, two challenges exist compared to the regular image classification problems (as Junting et al mentioned, reference!). The first requires collecting a large amount of data to capture fixation points of human observers instead of a numerical/ text label used in image classification problems. The second problem to report is that a saliency measure must be produced for each image pixel, along with a saliency maps at the output. The maps require spatial coherence and small transition between its nearby pixel values.

The paper is organised as follows. Section 2 focuses on the related work covered in recent saliency prediction, while section 3 focuses on the dataset and how it has been utilised. Section 4 provides information on the architecture implemented, input and focus on the steps undertaken to generate the results. Section 5 provides results from the implemented architecture along with its comparison to implementations covered in the area by other researchers. Lastly, section 6 outlines conclusion and future directions in the model.

II. RELATED WORK

This section discusses the recent works in the area of visual saliency using deep learning strategies. A brief description of the training of these regression models is included along with the final respective saliency predictions.

In 2018, Pan et al's SalGAN [1], implemented a novel saliency prediction technique on MIT300 and SALICON dataset using generative adversarial network. The training procedure involved utilising the adversarial loss function within its training network. Two networks (generator and discriminator) were used in the procedure; the first one predicted saliency maps from raw pixels of an input image. Transfer learning was implemented since the network was initialised with the weights of a VGG-16 model trained on ImageNet data set for object classification [2], a decoder architecture is then used to produce a saliency map using a total of 10 convolutional neural layers in the generator network. The discriminator network is used consisting of six kernel convolutions interspersed with three pooling layers. While, excellent training results have been produced, the model implements a deep architecture which has a high run-time.

Kummerer et al [3] also used VGG features for saliency predictions along with readout layers on the MIT300 dataset. Along with VGG layers, four 1x1 convolution layers are used outputting very high performance. However, along with SalGAN, deep architectures are used in the implementation, leading to high training run-time.

Cornia et al [4] implement an architecture that combines features extracted from different levels of their CNN from the SALICON dataset. The model is composed of three main blocks; feature extraction CNN, feature encoding network and a prior learning network. The feature extraction CNN comprises of 13 layers taking an input image and forming feature maps of the encoded network, the model also builds on the 16 layer model of VGG. The feature encoding network takes feature maps at three different locations, a final 1x1 convolution layer is added in order to weight the importance of each feature map and produce a final predicted map. The prior learning network implemented at the end of the overall architecture lets the implemented network learn its own custom prior, allowing up sampling to the predicted saliency map with pixel-wise multiplication.

III. Dataset Used

SALICON dataset (Cite) has been used for training model. This is the largest dataset available for saliency predictions. The dataset has been built from images of Microsoft CoCo dataset, while crowdsourcing strategies on the internet have formed the saliency maps to act as ground truth for the model implemented. Mouse clicks were captured in the crowdsourcing campaign, while this is not ideal for pinpoint saliency, it acts good source for the model since there is a high quantity of images for the model to learn from.

For the implementation, training and validation datasets are extracted from pickle files. The training dataset comprises of 20000 data points featuring three keys of a dictionary, X, Y and filename. X for each data point is a 3x96x96 image, while Y consists of 48x48 image saliency maps for the respective X data point. The validation dataset consists of 500 data points where each data point is represented as a dictionary. The dictionary comprises of five keys: X, Y, y_original, X_original, and filename. X and Y are of the same image dimensions as those in the training dataset, while y_original saliency maps of dimensions 480x640 and X_original is the original training data before of dimensions 3x480x640.

IV. Architecture and Input

A. Architecture

Neural Network	Kernel Size	Stride and Padding	Dimension by Layer
Input Image	96x96x3	-and-	96x96x3
Conv. Layer 1	5x5x32	1 and (2,2)	96x96x32
Max Pooling 1	2x2	2 and -	48x48x32
Conv. Layer 2	3x3x64	1 and (1,1)	48x48x64
Max Pooling 2	3x3	2 and -	23x23x64
Conv. Layer 3	3x3x128	1 and (1,1)	21x21x128
Max Pooling 3	3x3	2 and -	10x10x128
FC Layer 1	4608	-and-	4608
Slice 1 and Slice 2	2304 and 2304	-and-	2304 and 2304
MaxOut Layer	2304	-and-	2304
FC Layer 2	2304	-and-	2304
Output Vector	2304	-and-	2304

The network comprises of three convolutional layers (with max pooling layers after each of these layers) along with two fully-connected neural network layers. The network comprises a total of 64.4 million free parameters. With the use of required padding and strides on each of the convolutional layers, the resized images of dimension [96 x 96] feature maps down to [11 x 11] before the data is inputted into the initial fully-connected layer. As opposed to Pan et al's implementation where the dimensions are [10,10] before it is inputted into the initial fully-connected layer (as mentioned in Table 1.

After observing several run-throughs, the loss performance of the neural networks were above 2.0 at the end of 20 epochs, which is considered high. As a solution, ReLU were added to the convolution layers to act as an activation function that prevents linearity of tensor computations in order to allow for more optimised and relevant outputs to be formed.

In order to reduce the overfitting of data, the output dimensional vector from the first fully-connected layer is sliced to form two 2304-dimensional vector slices, the slice with the maximum overall activation is selected to be inputted into the second fully-connected layer. The 2304-dimensional vector at the output is mapped into a 2D array of [48 x 48], corresponding to a saliency map. The image is resized to match the input image and posterior filtering using a gaussian kernel with a standard deviation of 2.0.

The architecture was implemented through predominantly utilising the PyTorch deep learning library. Nvidia P100 GPUs have been used for the processing. 1000 epochs of the architecture required a run-time of 3 hours.

B. Input

The implementation initially loads the SALICON dataset, reshapes the input image to [96x96] coloured images and trains (and validates) them to produce output saliency maps of dimension [48x48] as shown in Figure.

C. Further Implementation Details

The training process uses the MSELoss function (Mean Squared Loss) function, which, in essence, measures the Euclidean distance of the predicted saliency map with the ground truth. The network was trained using the SGD (Stochastic Gradient Descent) optimiser along with the Nesterov momentum method, while the network was validated using the validation dataset in order to monitor convergence and overfitting. The learning rate was kept varied between 0.03 to 0.0001 and the network was trained for 1,000 epochs. For data augmentation, the training saliency maps and images was mirrored horizontally before it was trained in order to increase the volume of data being trained, while this leads to a higher training time, it also reduces overfitting and thus, increases performance. The saliency maps were normalised to [0,1].

V. Results and Comparison

The results of the experiment aimed to replicate that of Pan et al's shallow architecture. The numerical results produced are within 0.15 range to Pan et al's work and two metrics within 0.02 range of the implementation of lecturers at Bristol University. Figure 1 below compares the ground truth along with the saliency output corresponding to the image in the left.

The loss curves shown in Figure 2 suggests show that no overfitting has occurred since the train and validation loss plateau at around the same level. The value of the point of stability for both, validation and training curves is ~0.017 which is considered low.

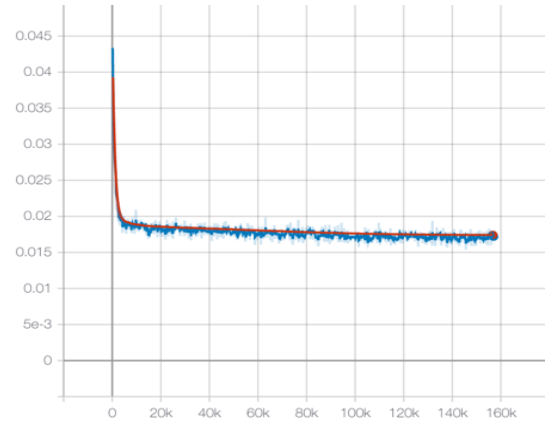

Figure 1: Loss Curves

The visualisation filters shown below are those from the first convolutional layer. These present patterns showing edge detectors which can be identified through them.

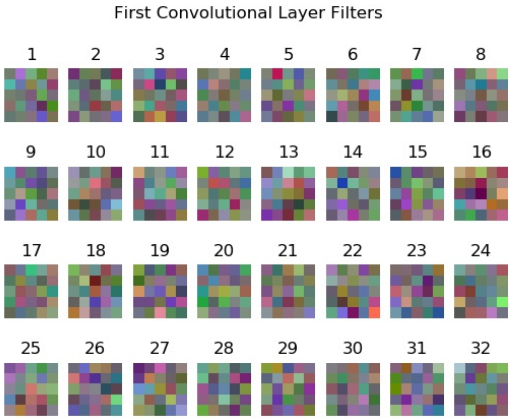

Figure 2: Visualisation Filters from First Convolution Layer

As shown in Figure 3, the second column, ground truth shows the ground truth of the image on the first column. The third column corresponds to the final saliency output maps of the respective images.

Due to the strong center-bias in the saliency output, the salient features of most images have been blocked. The 1st and 3rd row prediction seem to have a center-bias and the most of the output is covered with uniform noise, producing low saliency detection performance for the images. Specifically, 1st row saliency output has a very minor focus on the salient parts of the image i.e. a relatively low intensity is shown at points where saliency is high. Additionally, the prediction has a very high center bias that does not reflect on a salient area in the image. The 3rd row saliency prediction produces an overall dim intensity within the whole image, meaning that the salient areas are not recognised visually.

In contrast, the 2nd row saliency prediction detects the salient features being in the middle (horizontal strip) of the image.

The saliency level below and above the center-bias are significantly lower compared to the level in the middle. While there are noticeable anomaly 'spots' in the edges of the third row saliency output, it is still clear that the saliency output is significantly better than the other two saliency predictions.

SALICON (val)	CC	AUC Shuffled	AUC Borji
Pan et al [2016]	0.58	0.67	0.83
Shallow Convnet (Replicated results)	0.51	0.55	0.69

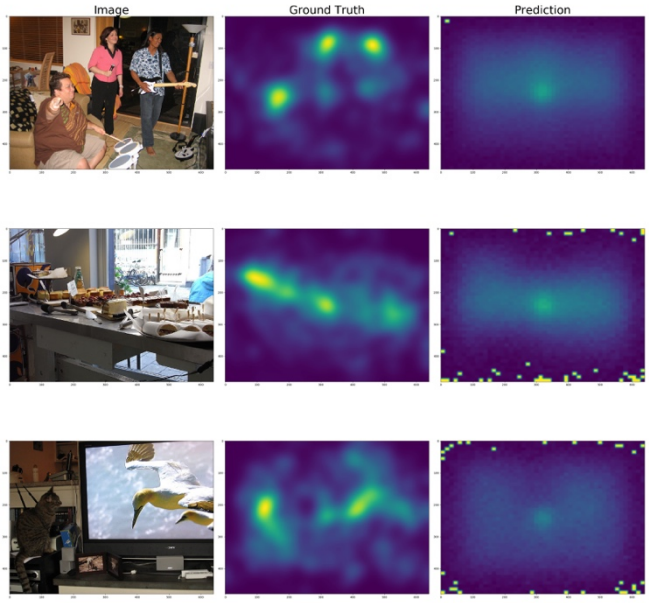

Figure 3: Image, Ground Truth and Saliency Prediction

VI. Conclusion

In conclusion, an attempt to replicate the results of Pan et al's implementation was produced and achieved within 0.14 of the performance metrics of their implementation. The loss curves further highlight the fact that no overfitting occurred in the implementation since the validation and training curves stabilize at approximately the same level. Additionally, many of the visual saliency output predictions indicate similarity to ground truth images. However, with the architecture being shallow.

References

[1] J.Pan, C.Canton-Ferrer, K.McGuinness, "SalGAN: A visual saliency prediction with adversarial networks," Computer Vision and Image Understanding

[2] Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L., 2009. ImageNet: A large-scale hierarchical image database, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR)

[3] Kummerer, M., Wallis, T.S., Gatys, L.A., Bethge, M., 2017. Understanding ¨ low-and high-level contributions to fixation prediction, in: 2017 IEEE International Conference on Computer Vision, pp. 4799–4808.

[4] M.Cornia, L.Baraldi, G.Serra, R. Cucchiara, 2016, "A Deep Multi-Level Network for Saliency Prediction" in : IEEE International Converence on Pattern Recognistion (ICPR), Cancun, pp. 3488-3493.

[5] J. Pan, K. McGuinness, S. E., N. O'Connor, and X. Giro-i Nieto, ´ "Shallow and Deep Convolutional Networks for Saliency Prediction," in CVPR, 2016.

[6] M. Jiang, S. Huang, J. Duan, and Q. Zhao, "Salicon: Saliency in context," in CVPR, 2015