ADL Coursework Report

Gabriel Galadyk
School of Computer Science
University of Bristol
us21186@bristol.ac.uk

Maciej Braszczok
School of Computer Science
University of Bristol
tc19422@bristol.ac.uk

I. Signed Agreement

We agree that all members have contributed to this project (both code and report) in an approximately equal manner

Gabriel Galadyk

Maciej Braszczok

II. Introduction

Human visual system fixates on the informative elements of the viewed visual scenes. In order to predict such human fixation regions, numerous computational systems have been proposed in the past. Traditional approaches to computationally model human fixations involve hand-crafting feature selection mechanisms. Such approaches rely on throughout understanding of human visual attention mechanisms, which has not been yet achieved. Altough some machine learning models have been proposed in for saliency detection, these models do not learn patterns sufficiently from complex images, due to their shallow architecture. To overcome the limitations of traditional approaches, the research paper Predicting Eye Fixations using Convolutional Neural Networks[1] was proposed by Nian Liu1, Junwei Han1, Dingwen Zhang1, Shifeng Wen1 and Tianming Liu. The deep architecture of the network allows the model to learn features related to saliency, with increasing detail and complexity. The Multi-resolutional architecture captures varying level of details influencing saliency. The model proposed in that paper, introduces a novel approach to saliency detection, which proves to make accurate predictions and resolves the aforementioned issues.

III. Related Works

Since the release of the paper, multiple scientists have adopted deep neural networks for saliency detection. In the paper 'A novel fully convolutional network for visual saliency prediction' [2] the authors implemented a novel Contextual Encoder-Decoder Network, which also learns on multi resolution data, to capture local and global context of the saliency regions, similarly to what was aimed to achieve in a Predicting Eye Fixations using Convolutional Neural Networks [1], the paper we are implementing. The proposed model and a Mr-CNN[3] implemented in another paper, was tested on Toronto[4] and MIT[5] datasets, however was not evaluated using the AUC score. Some evaluation metrics used by the authors, favored MR-CNN model, while other favored the model proposed by the authors of the paper. Hence, in our work we aim to improve MR-CNN's model's performance, without changing the original architecture proposed by the authors of the paper: Predicting Eye Fixations using Convolutional Neural Networks[1].

IV. Datasets

In our work we utilised 3 out of 4 saliency datasets that the authors of the original paper used. The first dataset was the same MIT[5] dataset that the model from the original paper was trained on. This is the largest eye fixation dataset of all 3 and contains 1003 images from the Flicker and LabelMe datasets which have been viewed by 15 humans to obtain eye tracking data that showed fixation points on images of various resolutions from 405x1024 to 1024x1024. There are, in total, 779 landscape and 228 portrait images in the dataset. Second dataset, Toronto[4], consists of 120 color and fixed size photos with resolution 511x681 pixels. The images are indoor and outdoor scenes, which were free-viewed by 20 humans. The last dataset, NUSEF[6], contains 758 images, collected from Flickr, Photo.net, Google images, and the IAPS dataset, with affective context. Each image was free-viewed on average by 25 human subjects, from a pool of 75 humans. We have used 438 images from this dataset, due to not having copyrights for IAPS images, as well as not being able to match label data for 6 photos.

V. CNN Architecture

Our network is a replication of the original Multi-resolution Convolutional Neural Network that was implemented in the original paper. It contains 3 separate streams for each input where each stream contains a series of 3 convolutional & pooling layers followed by one fully connected layer; they are then concatenated into one input to another fully connected layer before being fed into the output layer which returns a binary classification of whether this image is a fixation point or not.

Each input is a 42x42 cropping of a resizing of the original image that is put is an input, the x3 represents the 3 colour channels, the first layer, a convolutional layer, applies a 7x7x3 kernel resulting in a layer size of 36x36x96; the kernel size for the remaining convolutional layers is set to 3x3x**C** where **C** is

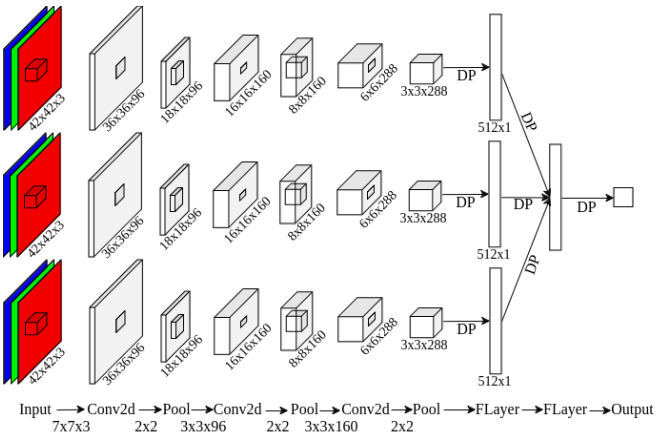

Fig. 1. Diagram of our MrCNN. Arrows show the progression of the data through the network, labels at bottom indicate the purpose of the layer. Dropout layers are included in the diagram whenever they occur. Kernel size for the layers is included under the arrows.

the number of channels in the layer. After the convolution, a pooling layer is applied with a 2x2 kernel with a stride of 2, effectivelly halving the height and width of the convolutional layer, this repeats 2 more times resulting in the last pooling layer being of size 3x3x288, we apply dropout before feeding into the first fully connected layer, this is then fed into the first fully connected layer of size 512x1 where dropout is applied again after. The streams are then concatenated together and fed into the second 512x1 fully connected layer where we apply dropout for the final time before feeding into the output layer.

VI. Implementation Details

A. Data processing

The training, validation and testing splits of the MIT data were already prepared for us, each split prepared as follow. For the train split, from each of the 1003 images 10 fixation and 20 non-fixation locations were sampled. Location being a size 42x42 pixels patch from the image having all corresponding saliency values in the eye fixation density map grater than 0.9 for fixation location or less than 0.1 for non-fixation locations. For 100 images in validation and 100 in testing data, each image was divided into 2500 samples, by evenly spliting the image into 50 coordinates across width and 50 across height of the image. Coordinates denote the centre points of 42x42 crops, such that if the coordinates of the points lie too close to the borders, the border pixels are copied. The ground truth fixation maps corresponding to each image were created by applying gaussian blur to the ground truth images with fixation points.

For NUSEF dataset, we extracted all the ground truth fixation points recorded for all of the subjects for each corresponding image and we applied Gaussian blur to create ground truth fixation maps to be used as labels for the data. After that, we randomly chose 350 images for train split, 44 for validation and 44 for test split, and than processed data for each split in thesame way as described above MIT dataset.

Toronto dataset, was processed in the same manner as validation and testing data of MIT, as it was only used to test performance of model trained on MIT. For the labels, the ground truth fixation maps prepared by the authors were used.

B. CNN parameters and settings

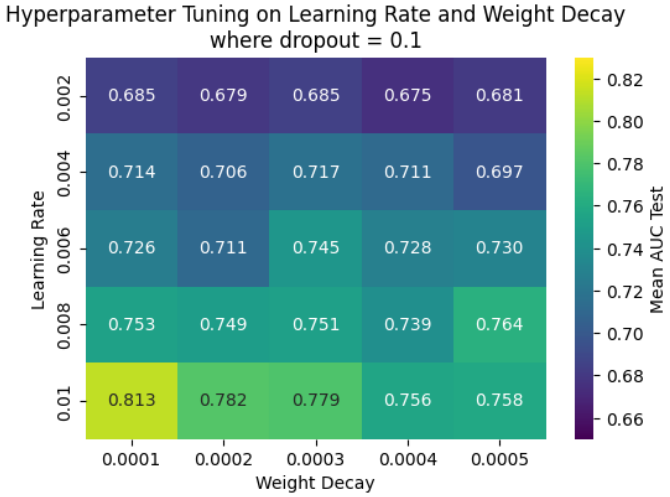

Fig. 2. Results of hyperparameter tuning performed on learning rate and weight decay with dropout probability being 0.1. This tuning gave us the best result model.

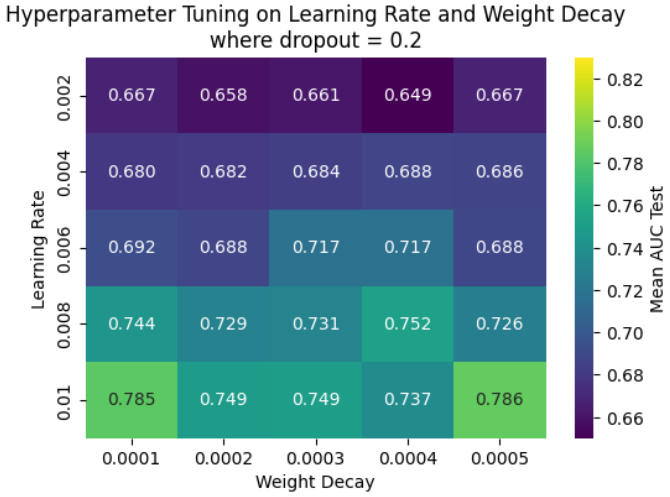

Fig. 3. Results of hyperparameter tuning performed on learning rate and weight decay with dropout probability being 0.2.

While our MrCNN architecture is made with replicating the original in mind, we did not utilise the same hyperparameters that the original paper's model has used. Instead we had performed hyperparameter tuning using a grid search strategy with predefined ranges and intervals, we have included the original paper's model hyperparameters within this range. Additionally, momentum was left untouched at 0.9; the original model includes learning rate decay which our model implements by multiplying the learning rate by 0.999 each step,

momentum on the other hand increased by 0.09 linearly during training which we have included by having the momentum update as well. Finally, as the original model did, we trained our model for 5000 steps, and performed validation every 200 steps where we saved a checkpoint of the model. At the end of the training we picked the best scoring model which usually ended up being around the time when the model began to overfit for the data.

Overall, we have tested the scores with differing learning rate, dropout probability, and weight decay. The ranges for these parameters were 0.0001-0.0005 for weight, 0.01-0.002 for learning rate, and 0.1-0.5 for dropout probability. In total, there were 125 different runs of the model, for each model, we selected the best scoring checkpoint, and our results have shown that dropout beyond 0.3 always did worse than any model with a dropout probability of 0.1 or 0.2 so for our final figures we have discarded these results.

Figure 2 and figure 3 show the results of these tunings. It can be seen from the results that training tends to perform better with a higher learning rate and there seemed to be no noticeable impact on the AUC score of the model with varying weight decay. The average mean AUC for the first figure is 0.729, while the second figure has an average mean AUC of 0.785, this would imply that a lower dropout did improve the AUC score of the model. Through hyperparameter tuning we identified the most performing hyperparamaters as 0.01 for learning rate, 0.0001 for weight decay, and 0.1 for dropout probability.

VII. Replicating Quantitative Results

Dataset	Mean AUC	Zero-Shot AUC	TL AUC	Original AUC
MIT	0.813			0.719
NUSEF		0.86846	0.880	0.670
Toronto		0.86811	N/A	0.726

TABLE I

TABLE SHOWING THE AUC SCORE OF OUR MODEL ON THE 3 DATA SETS. 4TH COLUMN CONTAINS THE SHUFFLED AUC SCORE OF THE ORIGINAL PAPER'S MODEL. TRANSFER LEARNING AUC FOR THE TORONTO DATASET WAS NOT OBTAINED DUE TO ITS SMALL SIZE.

VIII. Training Curves

Figure 4 and figure 5 show the accuracy and loss curves of our MrCNN model. The very aggressive spikes in the accuracy and loss shown by the figures can be inferred as the consequence of using a high momentum hyperparameter that only increases over time, despite that, our validation loss and accuracy steadily improved until around 3000 epochs where the validation loss and accuracy begins to stagnate; with the training set loss and accuracy improving, it is indicator that our model might be beginning to overfit for the training data. Because of the application of checkpointing, the final model we selected ended up being checkpoint with the lowest validation loss which usually ended up being before the overfitting started.

Fig. 4. Loss curves of our base model during training and testing over time

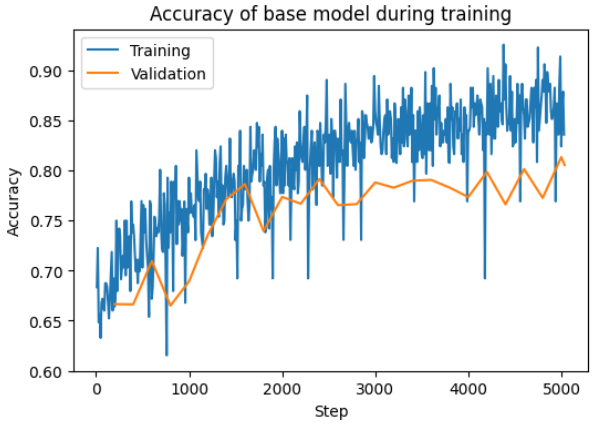

Fig. 5. Accuracy of our model on the training and validation data set during training. Accuracy during validation is considered to the mean AUC on the validation set

IX. Qualitative Results

Visualized predictions from figure 7 show that our model is better at making predictions on data with high-contrast, and little details. Comparing the model's predicitons with the ground truth for the second and third image would suggest that the accuracy of the model's performance is very low. However, on the image from the first row, our model achieved 0.8815 AUC score, while on the second and third row, 0.6073 and 0.5966 respectively. This visual verification confirms what the authors of the original paper stated, that AUC is not the best metric to evaluate model's performance.

X. Extensions

As part of our extensions we have sought to identify how our trained model would fare when we have it train and test on other datasets, the two techniques are zero-shot and transfer learning. The datasets we have used are the NUSEF and Toronto datasets which were also used in the original paper.

Fig. 6. Visual comparison of a set images from MIT dataset (row 1), our Mr-CNN model's predictions that achieved AUC scores over 0.78 against the ground truth (row 2) and the ground truth saliency maps (row 3).

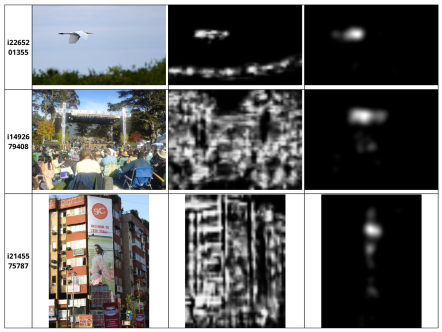

Fig. 7. Selected images (column 1), predicted fixation maps (column 2) and ground truth fixation maps (column 3) for comparison.

A. Zero-Shot

1) Introduction to Zero-Shot: Zero-shot learning refers to the practice of testing a neural network and seeing if it can accurately label data classes unseen during its training[7]. In our experiment, we trained and tuned the model on the MIT dataset and tested how well would it perform in predicting saliency regions for the two other datasets, Toronto and NUSEF.

2) Implementation and Results: Using the best model we have found using checkpointing and hyperparameter tuning, we proceeded to test this model using the images from the NUSEF and Toronto datasets, we used the same AUC metric to evaluate. The model tested on all 120 images from Toronto dataset, and achieved a 0.86846 AUC score. Second test on 44 images from NUSEF test split had achieved a 0.86811 AUC score. The results have shown us that the model was able to generalise very well to unseen data, it can be theorised that this could be due to the fact that by checkpointing the model and picking the best checkpoint with the lowest validation loss, we have obtained a model that can generalise well.

3) Critical Analysis: Zero-shot technique aims to demonstrate the model's ability to generalize well to new unseen during the training. Our model achieved a higher AUC score (as seen on I) on data it has not seen before than on the testing data it saw on the MIT dataset. This could be the result of many factors. Apart from the one mentioned in results, the nature of the two other datasets could also hold a reason, for one, the small sizes of the datasets compared to the MIT dataset could imply that not enough samples were tested in order to obtain an accurate metric, additionally it could also be the case that the datasets simply share images or the types of images included are not too dissimilar from each other to cause a considerable a considerable drop in AUC score as a result of zero shot.

B. Transfer learning

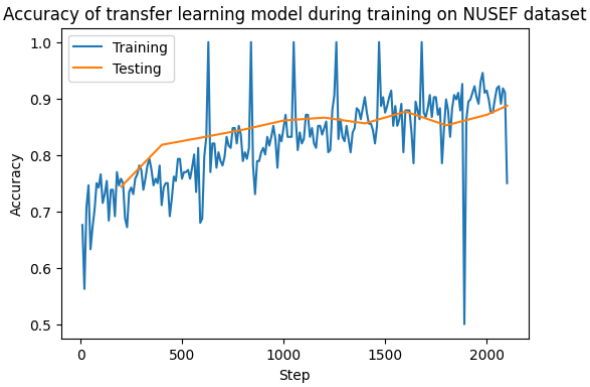

Fig. 8. Graph showing the accuracy of our model on the NUSEF dataset. Six separate times the model achieved perfect accuracy during training, including one time where the model completely blundered and achieved a 0.5 accuracy.

1) Introduction to Transfer Learning: Transfer Learning involves taking out pre-trained model, and performing additional training on a separate dataset with the intention of applying knowledge gained from the original dataset to accurately predict the new and similar dataset[8]. We performed transfer learning on the NUSEF dataset, we did not perform transfer learning on the Toronto dataset due to the very low sample size. This method was also used by the authors of the original paper to further train their model.

2) Implementation and results.: Using the same model that we trained and used in zero-shot, we continued the training of our original but this time using images from the NUSEF dataset, our hyperparameters remained the same and we have performed validation every 200 steps using the testing data. Figure 8 and 9 show the change in the models accuracy and

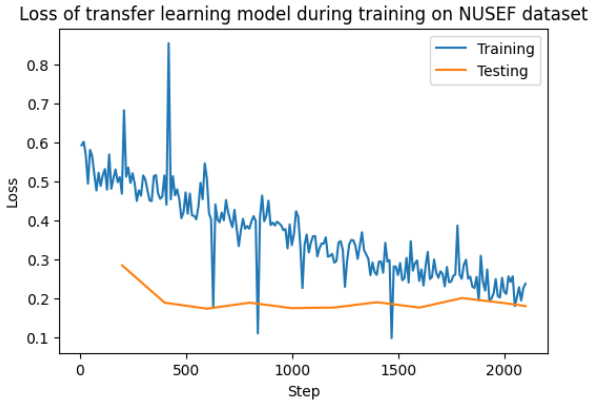

Fig. 9. Graph showing the loss of our model on the NUSEF dataset. The model consistently lowers its loss on the training data over time but fails to improve on the testing data.

loss during training, the final result of the training can be seen in I. Despite the evidence of overfitting happening, the model still performed remarkably well when evaluating its AUC score. The high accuracy and the 1.0 peaks could be indicators of the benefit of checkpointing and picking a model that did not fall victim to overfitting which let it be effective at generalising over unseen data.

3) Critical reflection.: The results from the figures do imply that there was evidence of overfitting during transfer learning, but despite that, the model still achieved a high mean AUC score of 0.880. This is considerably higher than the models AUC score on the original MIT dataset, but as was the case with zero-shot, the fact that the model was able to generalise well over unseen data it did not train on implies that either the model is capable of generalising, or the datasets have more overlaps than thought.

XI. Conclusion & Future Work

In this report we have replicated the original architecture of the Multi-resolution Convolutional Neural Network used by the original authors to train a model for predicting eye fixations. We have made minor adjustments to the overall architecture, and additionally we have performed hyperparameter tuning to find a new set of optimal hyperparameters that have produced a better overall result compared to the original paper. Finally, through the usage of checkpointing, we have potentailly avoided the issue of too much overfitting and have created a model that is capable of generalising over unseen data by testing itself on it, and further training itself on it. Summarise what your report contains in terms of content and achievements. Suggest future work that might extend, generalise, or improve the results in your report. On the other hand, based on the visual inspection of the saliency map generated by our model, it can be argued that the metric we used for evaluation may not be the most accurate for scoring the saliency maps our model has created due to the massive discrepancy between some saliency maps and fixation maps for a given image. Some works[9] do indeed argue that it is the case in general and overview the pros and cons of AUC as a metric. Our extensions to the original paper may imply a common trend within eye fixation datasets, that usually, human attention is mostly drawn towards the most distinctive location within the image and it does not matter what sort of image is being observed whether it is a synthetic image or a real image. Further work on this could explore new and more diverse datasets that could further illustrate how models trained on a eye fixation dataset generalises over other eye fixation datasets and learn more about human eye tendencies and fixations. Additionally, with our current model, one improvement that could be made is investigating the effect a change in the momentum hyperparameter would have on the overall scoring of the model.

References

[1] N. Liu, J. Han, D. Zhang, S. Wen, and T. Liu, "Predicting eye fixations using convolutional neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 362–370.

[2] B. M. Ghariba, M. S. Shehata, and P. McGuire, "A novel fully convolutional network for visual saliency prediction," *PeerJ computer science*, vol. 6, e280, 2020.

[3] N. Liu, J. Han, T. Liu, and X. Li, "Learning to predict eye fixations via multiresolution convolutional neural networks," *IEEE transactions on neural networks and learning systems*, vol. 29, no. 2, pp. 392–404, 2016.

[4] N. Bruce and J. Tsotsos, "Attention based on information maximization," *Journal of Vision*, vol. 7, no. 9, pp. 950–950, 2007.

[5] T. Judd, K. Ehinger, F. Durand, and A. Torralba, "Learning to predict where humans look," in *2009 IEEE 12th international conference on computer vision*, IEEE, 2009, pp. 2106–2113.

[6] S. Ramanathan, H. Katti, N. Sebe, M. Kankanhalli, and T.-S. Chua, "An eye fixation database for saliency detection in images," in *Computer Vision–ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part IV 11*, Springer, 2010, pp. 30–43.

[7] F. Pourpanah, M. Abdar, Y. Luo, *et al.*, "A review of generalized zero-shot learning methods," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 4, pp. 4051–4070, 2023. DOI: 10.1109/TPAMI.2022.3191696.

[8] K. Weiss, T. M. Khoshgoftaar, and D. Wang, "A survey of transfer learning," *Journal of Big data*, vol. 3, pp. 1–40, 2016.

[9] D. M. W. Powers, "The problem of area under the curve," in *2012 IEEE International Conference on Information Science and Technology*, 2012, pp. 567–573. DOI: 10.1109/ICIST.2012.6221710.