Machine Perceptron Project: Eye Gaze Estimation

Dusan Svilarkovic
ETH Zürich
dusan.svilarkovic@inf.ethz.ch

Nikola Popovic
Computer Vision Lab, D-ITET
ETH Zürich
nipopovic@vision.ee.ethz.ch

Nikhil Prakash
Engineering Geology, D-ERDW
ETH Zürich
nikhil.prakash@erdw.ethz.ch

ABSTRACT

Estimating the gaze from real-world images is a difficult task that has many potential applications. We leverage the labeled images from large publicly available datasets like GazeCapture and MPIIFaceGaze to train an end-to-end convolutional neural network for gaze estimation tasks. We achieved a 4.517 mean angular gaze direction error in a cross-dataset testing set, which has ranked second in the public leaderboard.

1 INTRODUCTION

Methods that can track the gaze direction of a person directly from images are important for the advances in human-computer interaction technologies [4]. Eye-tracking has also been used for medical diagnosis [8], supporting disabled individuals [3], and understanding the cognitive processes, such as attention or perception [2, 10, 14]. Methods used to estimate gaze direction from images can be categorized as model-based or appearance-based [7, 19]. Model-based methods have been actively used in the early works and are based on the detection of key features in images to generate a geometric model for the eye. On the other hand, appearance-based methods track eyes directly from images to extract gaze direction. Recent works on appearance-based methods have adopted a data-driven approach to learn the mapping from eye/face images to the gaze angles using convolutional neural networks (CNN) [5, 11, 12].

2 DATA SET

Publicly available datasets, such as GazeCapture [11] and MPIIGaze [17], have been used to train CNN for gaze estimation tasks. These datasets tend to include a large number of examples from multiple persons and varying illumination conditions to facilitate a cross-person gaze estimation in real-world situations. The GazeCapture dataset had almost $2.5M$ frames captured from 1474 people using iPad and iPhones. On the other hand, MPIIGaze tried to step away from the laboratory conditions and collected $213K$ images from 15 persons during natural everyday laptop use over more than three months. In both the dataset, the images were also accompanied by additional information as head pose and gaze direction.

In this work, images from GazeCapture have been used to train, validate, and test the CNN. Images from MPIIFaceGaze [18] were also used to do cross-dataset testing. The detailed distribution of the training, validation, and testing split is described in Table 1. The dataset was already pre-processed, and we used the available full face image, the left and right eye cropped images, head pose vector, and facial landmarks as an input to our CNN (Figure 1).

3 MODEL

The model that we used was inspired by the Asymmetric Regression-Evaluation Network (ARE-Net) from [5]. The intuition behind the

Data source	Training	Validation	Testing
GazeCapture [11]	100000 (200)	4000 (40)	5000 (50)
MPIIFaceGaze [18]	–	–	7500 (15)

Table 1: Number of images used in this work for training, validation, and testing the CNN. Also reported in brackets are the number of people who contributed equally in the dataset.

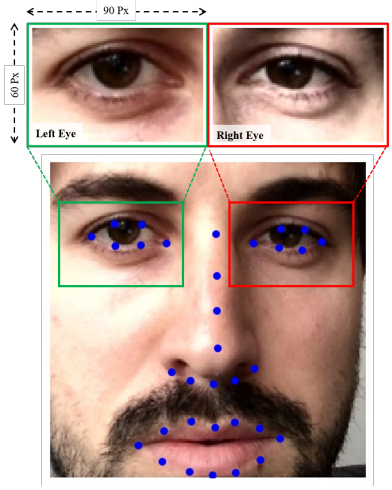

Figure 1: An example from the dataset used in this work. The left-eye image (in green) and right-eye image (in red) have been cropped from the full face image (bottom). The blue dot overlay visualizes the 2D coordinates of 33 facial landmarks points, which are also available in the dataset.

ARE-Net is that the gaze directions of two eyes should be consistent physically, but even if we apply the same regression method, the gaze estimation performance on different eyes can be very different. Based on that intuition it makes different predictions for different eyes (while having different gaze labels for each eye). In our problem we only have one gaze label for a pair of eyes. Thus, we decided to use a similar approach and high-level architecture as in ARE-Net, but instead of predicting different gaze vectors for different eyes, we predict two gaze vectors and combine them into the final prediction.

We will first describe the AR++Net, which is an extension of the AR-Net from [5]. The eye gaze direction is contained both in the eyes, but also in the position and the orientation of the persons head. Because of that, apart from just using the cropped eye images, we use different input modalities like the head pose

vector, facial landmarks and the masked position of the head in the given image. First, we use both the left and the right eye image. They are processed like in Figure 2. The first two eye streams process the features of the eyes separately, while last two process them jointly. This follows the idea that both the separate and combined eye features should be useful [5]. Unlike in the original work, we used a more powerful backbone CNN architecture to process the eye images. We used the DenseNet [9] architecture as the backbone, which has shown to be able to express complex non-linearities with a modest amount of network weights, compared to other modern architectures. It does so by taking inputs from all previous layers inside one block, instead of just from the previous one. The eye processing backbone has shared network weights for all four streams. The second input modality is the head orientation. It is concatenated directly onto the feature map, just before the last fully connected layer which predicts the gaze direction, like in the ARE-Net. Next, we have the vector containing facial landmarks. It is first processed through a fully connected network, to extract useful features from the landmarks, before it is concatenated to the final feature vector. Finally, we have the mask of the head's position inside the whole image. It is processed through a small CNN with 6 convolutional layers, which use ReLU activations. At the end of every CNN used here, a global average pooling was applied to reduce the spatial size of the feature map to $1x1$, because we want global information from the image while estimating the gaze direction. The features extracted from all input modalities are concatenated together into a feature vector and fed to the a fully connected layer. Unlike in the ARE-Net, where the last layer estimates the gaze direction for the left and the right eye, here we make two different estimates of the gaze direction: $\hat{g}_1$ and $\hat{g}_2$.

The E-net from Figure 2, is the same as the E-net in ARE-net [5], but it just uses a more powerful MobileNet v2 backbone architecture [15]. The purpose of this network is toprocess both eye images as the main sources of useful features, and predict which of the two gaze predictions is going to be better. This is done by standard classification using the binary cross-entropy loss. The labels are one-hot vectors that tell which gaze prediction received a lower error after a forward pass in the AR++Net. The output of this network are probabilities $\hat{p_1}$ and $\hat{p_2}$ that $\hat{g}_1$ or $\hat{g}_1$ will be a better gaze estimate.

Using the AR++Net we measure the angular error of the predicted 3D gaze directions $e_i = arccos(\frac{g \cdot \hat{g}_i}{\|g\| \|\hat{g}_i\|})$. We then calculate the symmetric gaze loss $L_{avg} = \frac{e_1+e_2}{2}$ as well as the asymmetric loss $L_{asym} = \frac{e_1 e_2}{e_1+e_2}$. The asymmetric loss takes the better prediction into account with a larger factor, meaning that during gradient backpropagation it gives more learning signal. The final loss is a combination of the two: $L_{final} = wL_{asym} + (1-w)\beta L_{avg}$, where $\beta = 0.1$ as in [5], $w = \frac{1+(2\gamma-1)p_1+(1-2\gamma)p_2}{2}$ and $\gamma = 1$ if $e_1 \leq e_2$ and $\gamma = 0$ if $e_1 > e_2$. The component weight w is close to 1 when both the AR++Net and E-Net have a strong agreement about the better gaze estimate ($\hat{g}_1$ or $\hat{g}_2$), meaning that the asymmetric part of the loss should be emphasized, while w is close to 0 when they strongly disagree.

The final prediction of the model is $\hat{p}_1\hat{g}_1 + \hat{p}_2\hat{g}_2$.

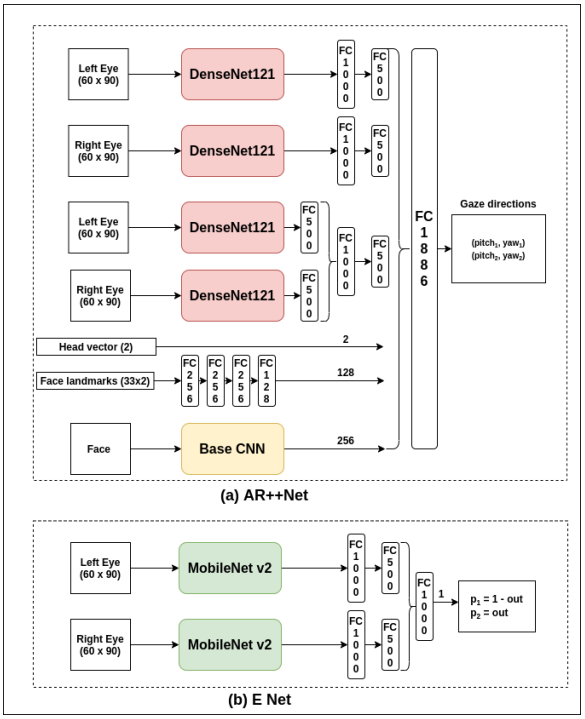

Figure 2: Architecture of our best model. (a) represents Asymmetric regression network used in the original paper with our addition of Dense Net architecture as the eye feature extractor, as well as face landmarks and face features as additional input modalities. (b) represents Evaluation network from the original paper with MobileNet v2 used for determining the strength of each intermediate gaze prediction in the final prediction. Base CNN is the same model used in the original paper by Cheng et al [5]. DenseNet121 is the model proposed in the work by Huang and Liu [9] on Densely Convoluted Neural Networks. MobileNet v2 is the work by Sandler et al [15].

4 EXPERIMENTS

The data set has been already divided into a training, validation and testing partition. The test labels are not included in the data handout. They are only used to evaluate solutions submitted to the project server. The training set has been used to learn the models weights, while the validation set has been used to learn hyper-parameters such as learning rate, weight decay strength, which type of model to use and with which network backbone. The validation set has also been used to determine the optimal epoch when to stop training, by doing early stopping on the validation loss.

The network has been trained using the ADAM optimization algorithm . The initial learning rate was 0.0001, and it is divided by 10 every time the validation loss doesn't improve after 4 whole epochs. The strength of the weight decay term was 0.005. The backbone networks were initialized with weights pre-trained on the ImageNet dataset. During training, data was augmented by slight random translation and scaling, slight color jittering and by

erasing a rectangle of size 10% of the image with probability $p = 0.5$. Both the training and validation data was normalized by reducing the mean and dividing with the standard deviation (the mean and standard deviation computed on the ImageNet data set)

5 FRAMEWORK

As a substantial contribution to this project we are also mentioning our team usage of the PyTorch [13] framework for our models and experiments. Whole skeleton was developed from scratch and includes codebase for training models, logging training results, hyperparameter setup from configuration files for convenient batch submission, saving the losses on Tensorboard [1] logs and most importantly, reusable framework for creating new experiments for additional models.

6 RESULTS

Regarding results, we have explored various models in order to achieve our best results. Recommended papers were explored, and various state of the art models from the ImageNet competition [6] were tried out, and the mixture between Asymmetric network [5] and Eyes Tracking For Everyone [11] with the usage of [9] gave us the soft spot on the leaderboard. On figure 3 how the process of training went on. On Table 2, we included both DenseNet161 and Resnet18 (same as for figure 2, but replacing DenseNet161 module) implementation of our model and their best validation and public test results. We have also tested our best model with the Evaluation Network excluded, which as concluded makes a small difference as opposed to our best model, but at this point on test score, a really significant one.

	Validation score	Public test score
AR++ DenseNet161	4.0503	4.5127
AR++ Resnet18	4.3732	4.6548
AR++ No-ENet DenseNet161	4.0367	4.5248

Table 2: Validation versus Public test score results.

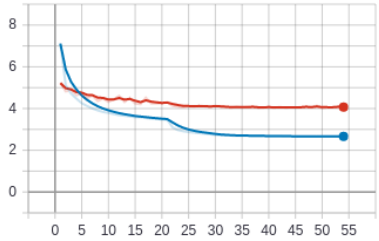

Figure 3: Learning curve of our best model, it is depicted as mean squared angular error [16] on the y-axis versus the exact epoch on the x-axis. For finding the best model, we used early stopping of 14 epochs.

7 WORK DISTRIBUTION

All team members contributed equally and made improvements on our project in different aspects. Down below in Table 3 we have different parts of the work that took most of the time on developing this project.

Task	Dusan	Nikhil	Nikola
Model code	X	X	X
Submission code		X	
Experiments code	X	X	X
Hyperparameter tuning	X		
Importing SOTA models			X
Report writing	X	X	X

Table 3: Work distribution of each team member. "X" marker shows which member contributed on which part of the project.

8 CONCLUSION AND FUTURE WORK

Our final model was inspired by the ARE-Net model. We make two gaze predictions based on the same feature map and combine them, unlike in the ARE-Net where they make gaze predictions for each eye using separate labels that they have. Also, we used additional input modalities and network backbones compared to ARE-Net, to improve the predictive power. Based on the public leaderboard, we conclude that we arrived to a good solution. In our experiments we noticed that E-Net brings only a small improvement compared to running the model without it, and we chose that solution because it was the best one just before the submission deadline.

REFERENCES

[1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2015. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. http://tensorflow.org/ Software available from tensorflow.org.

[2] Sylwester Białowąs and Adrianna Szyszka. 2019. Eye-tracking in Marketing Research. In *Managing Economic Innovations – Methods and Instruments*. Bogucki Wydawnictwo Naukowe, 91–104. https://doi.org/10.12657/9788379862771-6

[3] Maria Borgestig, Jan Sandqvist, Richard Parsons, Torbjörn Falkmer, and Helena Hemmingsson. 2015. Eye gaze performance for children with severe physical impairments using gaze-based assistive technology—A longitudinal study. *Assistive Technology* 28, 2 (Oct. 2015), 93–102. https://doi.org/10.1080/10400435.2015.1092182

[4] S. Chandra, G. Sharma, S. Malhotra, D. Jha, and A. P. Mittal. 2015. Eye tracking based human computer interaction: Applications and their uses. In *2015 International Conference on Man and Machine Interfacing (MAMI)*. 1–5.

[5] Yihua Cheng, Feng Lu, and Xucong Zhang. 2018. Appearance-Based Gaze Estimation via Evaluation-Guided Asymmetric Regression. In *Computer Vision – ECCV 2018*, Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss (Eds.). Springer International Publishing, Cham, 105–121.

[6] J. Deng, W. Dong, R. Socher, L. Li, Kai Li, and Li Fei-Fei. 2009. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*. 248–255.

[7] D. W. Hansen and Q. Ji. 2010. In the Eye of the Beholder: A Survey of Models for Eyes and Gaze. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32, 3 (2010), 478–500.

[8] Philip S. Holzman. 1974. Eye-Tracking Dysfunctions in Schizophrenic Patients and Their Relatives. *Archives of General Psychiatry* 31, 2 (Aug. 1974), 143. https://doi.org/10.1001/archpsyc.1974.01760140005001

[9] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. 2016. Densely Connected Convolutional Networks. arXiv:cs.CV/1608.06993

[10] Q Ji. 2002. Real-Time Eye, Gaze, and Face Pose Tracking for Monitoring Driver Vigilance. *Real-Time Imaging* 8, 5 (Oct. 2002), 357–377. https://doi.org/10.1006/rtim.2002.0279

[11] Kyle Krafka, Aditya Khosla, Petr Kellnhofer, Harini Kannan, Suchendra M. Bhandarkar, Wojciech Matusik, and Antonio Torralba. 2016. Eye Tracking for Everyone. *CoRR* abs/1606.05814 (2016). arXiv:1606.05814 http://arxiv.org/abs/1606.05814

[12] Seonwook Park, Adrian Spurr, and Otmar Hilliges. 2018. Deep Pictorial Gaze Estimation. *CoRR* abs/1807.10002 (2018). arXiv:1807.10002 http://arxiv.org/abs/1807.10002

[13] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in PyTorch. In *NIPS-W*.

[14] Keith Rayner. 1998. Eye movements in reading and information processing: 20 years of research. *Psychological Bulletin* 124, 3 (1998), 372–422. https://doi.org/10.1037/0033-2909.124.3.372

[15] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. 2018. MobileNetV2: Inverted Residuals and Linear Bottlenecks. arXiv:cs.CV/1801.04381

[16] Xucong Zhang, Yusuke Sugano, and Andreas Bulling. 2018. Revisiting Data Normalization for Appearance-Based Gaze Estimation. In *Proceedings of the 2018 ACM Symposium on Eye Tracking Research Applications (ETRA '18)*. Association for Computing Machinery, New York, NY, USA, Article 12, 9 pages. https://doi.org/10.1145/3204493.3204548

[17] Xucong Zhang, Yusuke Sugano, Mario Fritz, and Andreas Bulling. 2015. Appearance-Based Gaze Estimation in the Wild. *CoRR* abs/1504.02863 (2015). arXiv:1504.02863 http://arxiv.org/abs/1504.02863

[18] Xucong Zhang, Yusuke Sugano, Mario Fritz, and Andreas Bulling. 2017. It's written all over your face: Full-face appearance-based gaze estimation. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2017 IEEE Conference on*. IEEE, 2299–2308.

[19] Xucong Zhang, Yusuke Sugano, Mario Fritz, and Andreas Bulling. 2017. MPIIGaze: Real-World Dataset and Deep Appearance-Based Gaze Estimation. *CoRR* abs/1711.09017 (2017). arXiv:1711.09017 http://arxiv.org/abs/1711.09017