GaGaNet: Deep Learning Model for Gaze Estimation

Jan Leutwyler
janleu@student.ethz.ch

Zsombor Kalotay
zsombor.kalotay@hest.ethz.ch

ABSTRACT

In this report, we will talk about Gaze-Gaze-Net (abrv. GaGaNet) which is a novel combination of two GazeNets [11].

1 INTRODUCTION

Human Eye-Gaze-Estimation is a research topic in the field of Human Computer Interaction and Computer Vision. As its name says, the goal of this field is to estimate the users focus point (eye gaze) by analising images of his eyes. Applications vary from assistive technologies for paralyzed users [1], human-robot interaction [8], affective computing [2] and social signal processing [10].
The goal of this project is to develop a model which is being trained on the Gazecapture dataset [4] and then evaluated on both Gazecapture and the MPII dataset [11]. Difficulty is to create a model which generalises from the Gazecapture dataset to inputs from any other dataset, e.g., MPII dataset.

2 RELATED WORK

Our work is mainly based on Zhang et al.'s [11] MPII model which is a modification of the famous VGG-Net16 [7] introduced by Liu et al.

2.1 Datasets

Our project was conducted on the provided datasets gazecapture [4] and MPII [11]. Our train and validation dataset contains face images, eye-region images, left-eye images, right-eye images, head orientation, face-landmarks, and the gaze direction.

All our models were trained on the complete GazeCapture train set, which contains 500 images each from 200 people, and were validated on the provided GazeCapture test set, which contains 100 images each from 40 people.

3 REFERENCE MODELS

3.1 AlexNet

As a simple exercise we implemented AlexNet [5] and tested it with left and right eye patches and the whole eye region. As expected, AlexNet was not able to outperform any of the following models and therefore will be left out from the evaluation part.

3.2 VGG16

As a base for the GazeNet model [11] we implemented a simple VGG16-Net [7]. We also tested this network with either the left, right eye, the whole eye region, or the complete face image. And, as expected VGG16 was not able to outperform the GazeNet model.

3.3 GazeNet

Our GazeNet [11] implementation followed strictly the original description of the paper with concatenating the headpose to the dense layer input. The input image was the left eye patch of a given input user. For our GazeNet implementation we used the original proposed reduction gain of 0.1 and applied each after conducting 20'000 steps.

4 MODELS AND METHODS

We produced multiple models over the course of our work. In this subsection we will quickly explain the implementation of each model.

4.1 GoogLeNet

We implemented our version of GoogLeNet [9] and modified it for our purpose. As input image we used the full face of the user and let it run through the GoogLeNet. We modified the fully connected part into two subparts. The first sub-part concatenated output 1 and 2 of the GoogLeNet which then was connected to a dense layer with 1000 neurons in combination with an L2 regularizer (lambda = 0.0002) which was followed by a batch normalization layer and a dropout layer (rate = 0.6).
The second sub-part followed a similar manner, but concatenated the output of sub-part 1, output 3 of GoogLeNet and the head pose which then was connected again to a dense layer with 1000 neurons (incl. L2 regularizer) and a batch normalization layer followed by a dropout layer (rate = 0.2).
Finally the output of sub-part 2 was reduced by a dense layer to the desired eye gaze output of the network. Figure 1 visualizes our architecture for our modified GoogLeNet.

4.2 ResNet

Out of curiosity, we also implemented a scalable ResNet [3]. As in GoogLeNet we also used the full face image as input. And we modified the output layer, to get the estimated gaze as output.

4.3 GaGaNet

In this section we will explain our novel model Gaze-Gaze-Net (GaGaNet).

4.3.1 GaGaNet Architecture. Our GaGaNet is an extension of GazeNet. The most significant difference to GazeNet is that it takes the left and right eye image as inputs. Both input images get processed in two individual convolutional neural networks with no shared weights. To account for the larger input images of size 60x90x1, we changed the stride of the second pooling layer to 2. After the first 13 convolutional layers, we flattened the output of each eye image and added dropout layers (r=0.25), then we concatenated the flattened outputs and the 2D head pose orientation.

Finally, we added three dense layers (8192, 4096, and 4096 units) and applied the ReLU activation function to each output. And the final 2D gaze is produced by another dense layer with 2 units and no activation function applied.

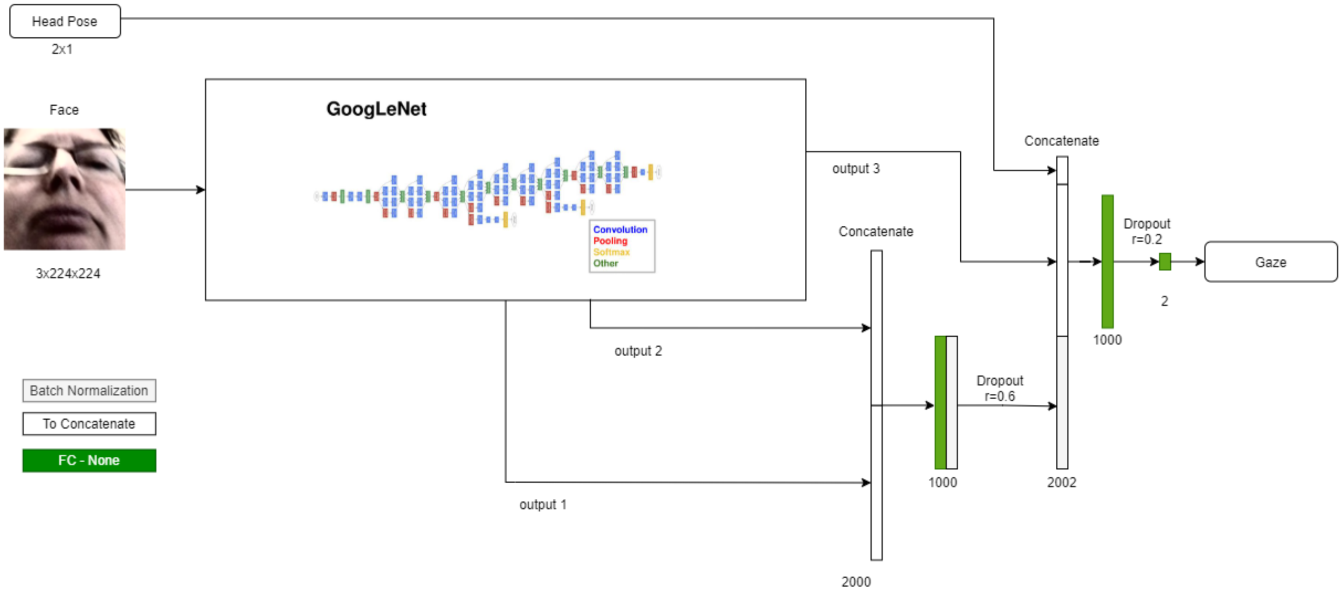

Figure 1: Schematic of our modified GoogLeNet

See figure 2 for a more detailed view of the overall architecture.

4.3.2 Learning-Rate Reduction. Thanks to GazeNet, we had to implement LR-Reduction. The implementation tried to be as general as possible, therefore we enabled for the programmer to define an array of tuples of (step number, reduction coefficient). This enabled us to control the reduction points and the reduction gain with a fine granularity.

4.3.3 Data Augmentation. We added methods to randomly modify the input data at run-time. Properties that can be changed are image brightness and/or image saturation. These methods should help to prevent the model from overfitting.

4.4 Training of our GagaNet

We trained our model for 20 epochs with batches of size 16. We used an initial learning rate of $1e^{-4}$, which we multiplied after 16600 steps by 0.1, at step 40250 by 0.01, at step 90000 by 0.1, and at step 110000 by 0.1. For the optimization we used ADAM with $\beta_1 = 0.9$ and $\beta_2 = 0.95$.

4.4.1 Loss Function. We used the mean squared error as our model's loss function.

$$L(y, \tilde{y}) = \frac{1}{n} \sum_{i=1}^{n} (y_i - \tilde{y}_i)^2 \tag{1}$$

5 EVALUATION

Our evaluation metric is the mean angular gaze direction error (in degrees). Which is the angular distance between the predicted gaze directions and the ground truth gaze directions, averaged over all samples.
We were able to pass the easy baseline and get close to the hard one with our GaGaNet. However, our GoogLe and ResNet implementations were not able to pass the easy baseline.

Model	Local Score	Public Score
GaGaNet	4.43	5.01
GazeNet (left eye)	5.47	5.99
GoogLeNet	5.10	5.70
ResNet34	5.28	5.78

As we can see in the table above GaGaNet outperformed SOTA computer vision models and also SOTA GazeNet.

6 DISCUSSION & OUTLOOK

GaGaNet outperformed the SOTA computer vision models and also the GazeNet. However, there would have been newer papers which we did not try to implement due to the Leonhard incidence. A good example is Liu et al.'s DiffNet [6] which is promising very good generalizability.
Furthermore there were several tricks we applied to GaGaNet but did not test on the other models since they were not getting close to the scores of the initial implementation of GaGaNet (e.g. Data Augmentation).
Another implementation/variation of the GoogLeNet, which was inspired by the GaGaNet, would have been putting two GoogLeNets parallel (GaGooGagLeNet), where one is getting the eye region as input (let's call that eye-net) and the other the whole face (analogously called face-net). These two networks would have been run in parallel with no shared weights and the outputs of the eye-net would have been concatenated with the head pose and the face-nets outputs with the face landmarks. As a last step we would have concatenated the two huge output vectors and run it through a fully connected layer (1000 neurons) after which we would have predicted the eye gaze.

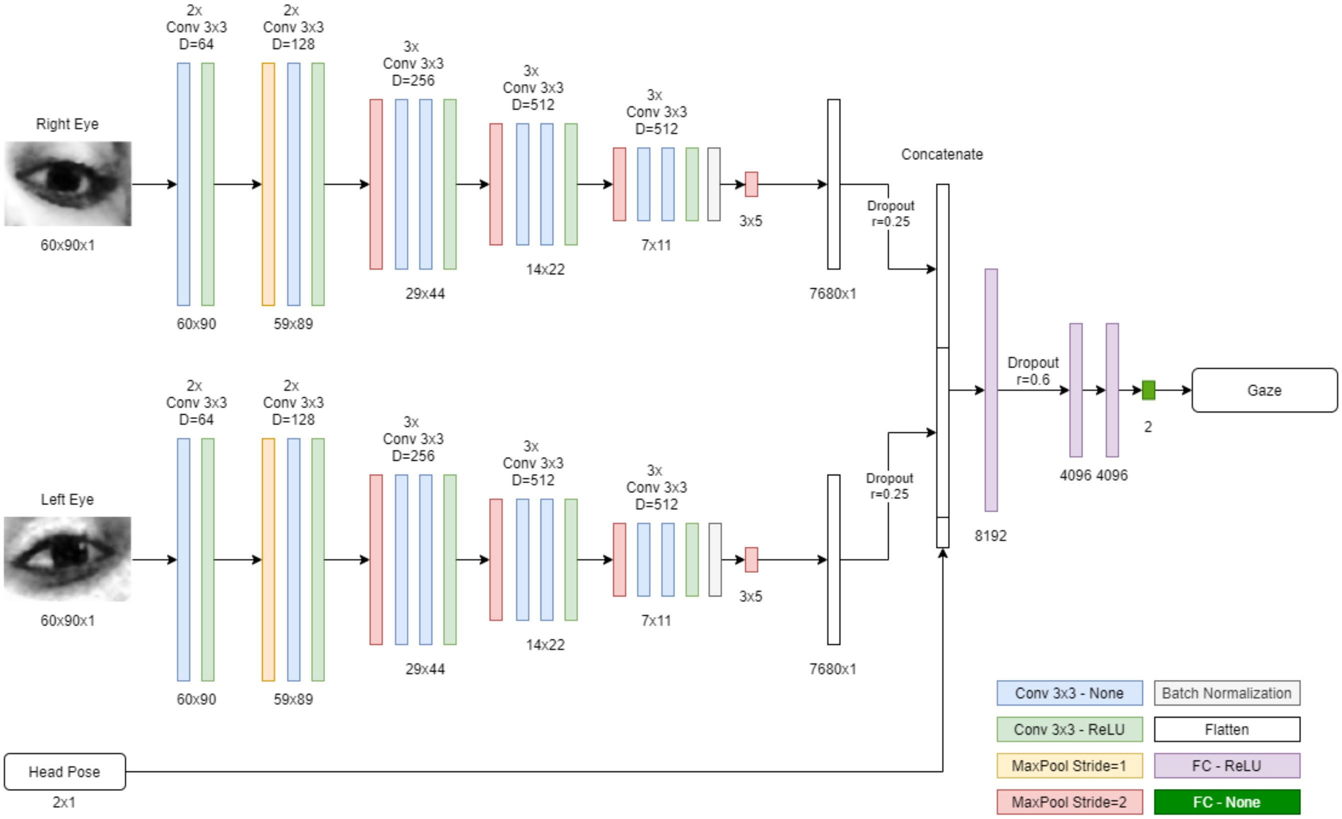

Figure 2: Schematic of our GaGaNet

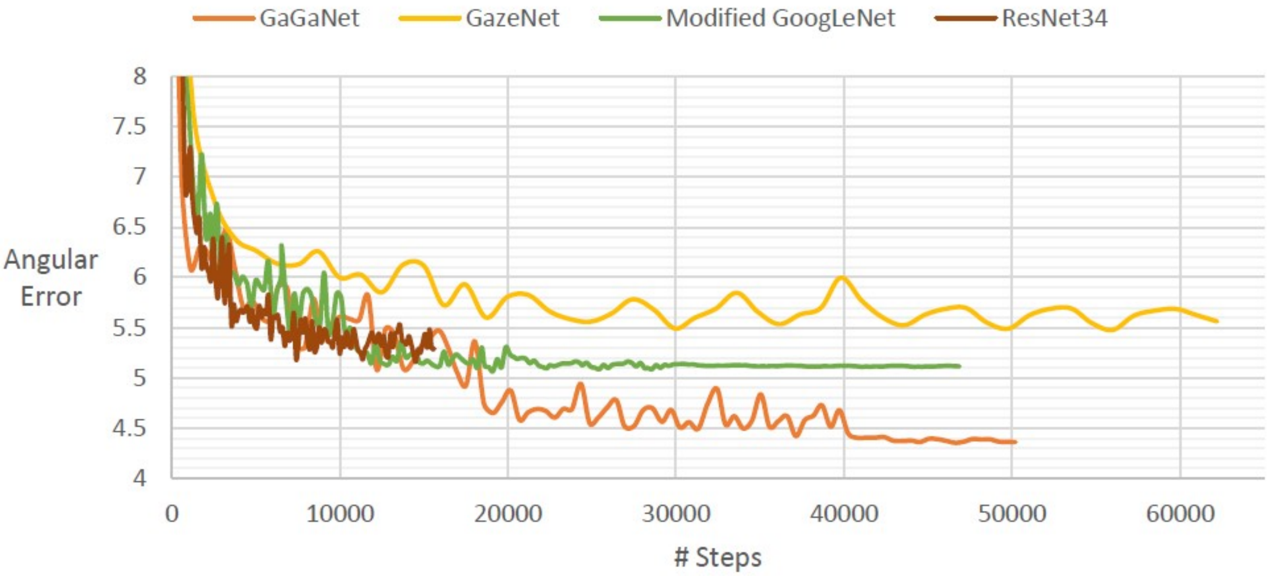

Figure 3: Comparison Graph which shows our best performing models. Evaluated on the provided validation set.

Unfortunately neither our resources on Leonhard nor our private computational resources were enough for running this huge network.

REFERENCES

[1] Guruprasad Bhat. 2017. Eye Gaze Recognition System to Assist Paralyzed Patients. *Perspectives in Communication, Embedded-Systems and Signal-Processing* 1 (05 2017), 4–5.

[2] Artem Dementyev and Christian Holz. 2017. DualBlink: A Wearable Device to Continuously Detect, Track, and Actuate Blinking For Alleviating Dry Eyes and Computer Vision Syndrome. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 1, 1, Article 1 (March 2017), 19 pages. https://doi.org/10.1145/3053330

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Deep Residual Learning for Image Recognition. http://arxiv.org/abs/1512.03385 cite arxiv:1512.03385Comment: Tech report.

[4] K. Krafka, A. Khosla, P. Kellnhofer, H. Kannan, S. Bhandarkar, W. Matusik, and A. Torralba. 2016. Eye Tracking for Everyone. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2176–2184.

[5] Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton. 2012. ImageNet Classification with Deep Convolutional Neural Networks. *Neural Information Processing Systems* 25 (01 2012). https://doi.org/10.1145/3065386

[6] Gang Liu, Yu Yu, Kenneth Funes Mora, and Jean-Marc Odobez. 2019. A Differential Approach for Gaze Estimation.

[7] S. Liu and W. Deng. 2015. Very deep convolutional neural network based image classification using small training sample size. In *2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR)*. 730–734.

[8] B. Mutlu, T. Shiwa, T. Kanda, H. Ishiguro, and N. Hagita. 2009. Footing in human-robot conversations: How robots might shape participant roles using gaze cues. In *2009 4th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. 61–68.

[9] C. Szegedy, Wei Liu, Yangqing Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. 2015. Going deeper with convolutions. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 1–9.

[10] Alessandro Vinciarelli, Maja Pantic, Hervé Bourlard, and Alex Pentland. 2008. Social Signal Processing: State-of-the-Art and Future Perspectives of an Emerging Domain. In *Proceedings of the 16th ACM International Conference on Multimedia (MM '08)*. Association for Computing Machinery, New York, NY, USA, 1061–1070. https://doi.org/10.1145/1459359.1459573

[11] Xucong Zhang, Yusuke Sugano, Mario Fritz, and Andreas Bulling. 2017. MPIIGaze: Real-World Dataset and Deep Appearance-Based Gaze Estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* PP (11 2017). https://doi.org/10.1109/TPAMI.2017.2778103