Machine Perception: Eye Gaze Estimation

Lukáš Jendele*
ETH Zürich
jendelel@ethz.ch

Ondrej Skopek*
ETH Zürich
oskopek@ethz.ch

ABSTRACT

We study cross-person single-eye gaze estimation on a realistic dataset. We reproduce recent appearance-based results and attempt to improve on the used methods by adding regularization, pre-training, pre-trained initialization, and refining synthetic data to extend the existing training set. Our results show promise, but do not perform exceedingly well.

1 INTRODUCTION

In eye gaze estimation, we aim to predict the direction a person is looking in. Purely appearance-based methods have been growing more popular lately due to several reasons:

- monocular RGB cameras are ever present in our lives and can be used for applications ranging from accessibility to tracking ad impressions,
- the availability of both realistic [21] and synthesized [17] datasets with larger numbers of samples, and
- recent advances in using deep learning to solve increasingly complex computer vision tasks, like classification or segmentation from images or videos.

Ultimately, we would like to be able to do accurate gaze estimation, even in noisy, real-life environments. The fewer assumptions about facial appearance, camera settings, head poses, and surrounding environments we can make, the closer we get to the goal of unconstrained gaze estimation.

In this work, we study appearance-based unconstrained single eye gaze estimation. We validate all our methods using the MPIIGaze [21] dataset, which contains using eye images of 15 individuals over a time period of several weeks. Specifically, we evaluate our methods on individuals that the models were not trained on (cross-person evaluation).

The contributions of our work are the following:

1. Introduce a model that refines synthetic images in order the breach the gap between the distributions of real and synthetic dataset.
2. Achieve reasonable performance on our task (5.2 degrees error).
3. Evaluate the effects of techniques like additional modalities, image augmentation, and transfer learning on model performance.

2 RELATED WORK

2.1 Synthetic data refinement

Shrivastava et. al. [14] were the first to come with the idea of using generative models to refine synthetic datasets. Due to reproducibility issues, we decided to leverage a different approach, called CycleGan [22]. It adds a cycle-consistent term into the loss function for GANs and trains the network to translate the domain of an image to a different one, and subsequently translate it back. This unpaired image translation method has achieved a lot of successes lately.

2.2 Estimation methods

A lot of work has been done on model-based eye gaze estimation [2, 5], appearance-based eye-gaze estimation [11, 19–21], and combinations thereof [12].

Specifically relevant to our approach is GazeNet [21], because their method is purely appearance-based and uses single-eye images. They also use head pose angles as inputs to their model, the effects which we will discuss later (they reported it does not have significant effect). GazeNet achieves an error degree of 5.4 on cross-person evaluation, recent state-of-the-art is 4.8 degrees [12].

3 METHODS

For eye gaze estimation task, the data, its preprocessing, and augmentation play a big factor in the resulting model performance. We chose to solve the single eye gaze estimation problem using two joint approaches. Firstly, we utilized the CycleGan model to refine the synthetic UnityEyes dataset. Secondly, we tried to use a convolution neural network architecture that estimates the eye gaze angles best.

3.1 Datasets and preprocessing

Out of all the work done on gaze estimation datasets, like EYEDIAP [6], GazeCapture [11], MPIIGaze [21], and synthetic datasets UnityEyes [17, 18], we focus on MPIIGaze and UnityEyes.

MPIIGaze contains single-eye images of 15 people, collected over the range of days to weeks in real-life conditions, from a laptop web camera. For our purposes, we obtained a cross-person train/validation/test split of this dataset for the project.

UnityEyes is a synthetic dataset of larger eye regions and more larger gaze angles than MPIIGaze. For the purpose of transfer learning to MPIIGaze, we crop the eye images and leave out samples where the gaze angles are not in the appropriate range. Different gaze angle distributions have been shown to negatively impact model performance before [21].

For preprocessing, we augment the image by varying contrast, adding Gaussian noise and performing Gaussian blur on top of that. Finally, we equalize the histogram of the image. All of the augmentation values were chosen such that the visual appearance stays largely consistent across the MPIIGaze dataset. For raw UnityEyes, we increase the values marginally, to simulate more noise that the MPIIGaze dataset has.

Whenever we learn on UnityEyes images that were refined to look like MPIIGaze images, we apply the same preprocessing as for original MPIIGaze images.

[*] Both authors contributed equally.

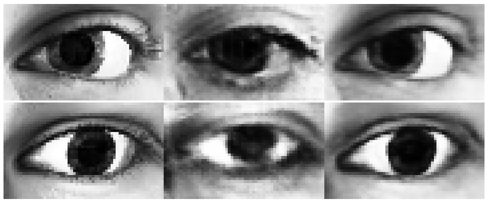

Figure 1: Images of synthetic eyes (left), refined images (middle), and synthetic images generated from the refined images for cycle consistency (right).

3.2 Synthetic data refinement

The synthetic UnityEyes images come from a different domain then MPIIGaze images. Thus, we propose to use generative adversarial models to transform these images from domain X to domain Y. More specifically, we adopted the CycleGan architecture as described Zhu et al. for the synthetic dataset refinement. Zhu et. al. shows that CycleGan is capable of appearance changes and struggling with geometric changes, which is desired behavior since we must keep the angle, in which the eye is looking. The only difference between their and our use case is that our pictures have dimension 36×60 instead of square images with dimensions 128×128. During training, we try to optimize the least square adversarial losses and the cycle consistency loss. As shown in Figure 1, we managed to acquire realistically looking images of eyes, and thus nearly double the original dataset.

3.3 Estimation models

For the actual angle estimation, we propose several Convolutional Neural Network (CNN) models based on VGG19 model of Simonyan and Zisserman [15]. It is a well-known CNN architecture originally developed for the ILSVRC competition [13] on the ImageNet dataset, and has been successfully used for learning on other datasets as well.

3.3.1 EyeVGG19. The EyeVGG19 model is based on the VGG19 model of Simonyan and Zisserman [15]. Contrary to GazeNet, we do not change strides of max-pooling operations in the first two blocks and keep them at 2. This has the positive effect of faster training times, while (to our knowledge) not impacting performance greatly. Instead of the original two 4096-sized fully connected layers, we only use 2048 neurons in each, activated by a ReLU function.

Dropout [8, 16] is applied before and after each of the two fully connected layers, with a keep probability of 0.5.

A linear output layer is added at the end, with 2 output units representing the pitch and yaw of the gaze angle. The loss we optimize is the mean squared error (MSE) of the produced output and the target pitch/yaw.

3.3.2 EyeVGG19_synth. A variant of EyeVGG19, but trained on a combination of MPIIGaze training data and refined UnityEyes data as seen in Section 3.2. The dataset is available for download temporarily[1], and can be reproduced by code in our repository.

3.3.3 EyeVGG19_noMP. A variant of EyeVGG19, that removes all max-pooling operations and changes the last convolution in each block to a strided one (with stride 2).

3.3.4 Other variants. As variant of EyeVGG19_noMP, we added a convolution with 1 output filter on top of the last convolution of the last block to squeeze the outputs of the previous layer. This network only has a linear layer on top, because the number of outputs of the squeeze layer is too small. The lack of fully connected layers made this method perform sub-par in our limited experiments and we did not explore it further.

3.4 Implementation

For ease of use, readability and comparability reasons, we base the implementations of all our models on Keras[4] Applications' sources. This also enables us to seamlessly experiment with ImageNet-pretrained weights for all mentioned models. We incorporate these models into our sources by means of a Keras wrapper[2]

As a backend to Keras we use TensorFlow[1] 1.7. In all our models, we use a batch size of 32 and use the ADAM[10] optimizer.

The source code of all our models will be made available later on our GitHub repository[3] and is also attached to this report.

4 EXPERIMENTS

We were provided with already split MPIIGaze dataset into train, validation and test parts. We trained all models on the train dataset until the MSE loss converged on validation, by decaying the learning rate by a factor of 10 every 5000 steps, with the starting learning rate being 10^{-4}. All other base learning rates performed worse (slow convergence, sub-optimal convergence).

For all our experiments, we report the best achieved MSE and error during validation steps performed at regular intervals. The reported test MSE is the public score on Kaggle of submitted predictions from that checkpoint. Most submitted models trained for 6000-8000 steps and took about 30-50 minutes to converge to that point, but training for more doesn't make the model overfit based on the validation MSE, so we can train for the full 15000 steps we used, which is around 60-70 minutes on a single Tesla K80 GPU.

The results of experiments described above can be found in Table 1.

Compared to the top performing teams, our results even for our best model (EyeVGG19) were sub-optimal. We managed to beat the hard baseline, but only marginally, and our models were very sensitive to early stopping – the difference in taking a few more steps at the time of stopping could very well mean a difference of up to 15% in MSE.

When trained on refined UnityEyes and real MPIIGaze data (EyeVGG19_synth), it performed reasonably well. It took longer to converge to a good result, but failed to get top performance. Even when successively fine-tuned on just MPIIGaze data, the model did not perform better.

The strided convolution model EyeVGG19_noMP did not perform well, the model was saturated at an early point during training and the loss did not go down later.

[1] http://people.ee.ethz.ch/~jendelel/MPIIGaze_augmented_50K.h5

[2] See file `src/util/keras_wrapper.py`

[3] https://github.com/oskopek/mp

Model	Validation		Test
	Degrees	MSE	MSE
GazeNet [21]	5.4		
Park et al. [12]	4.8		
Hard baseline			0.00665
Randomly weights in GazeNet			0.03264
The Convolution…			0.00377
Bonus Baseline			0.00503
EyeVGG19	5.2	0.00573	0.00649
EyeVGG19_synth	5.7	0.00685	0.00753
EyeVGG19_noMP	6.2	0.00810	0.01157

Table 1: Validation and test results. Values left blank are unknown. MSE means Mean Square Error. The top section describes published results (possibly on a different train/test split) and baselines. Middle section contains the two top teams on Kaggle for this project. Bottom section shows our results. Please note that numbers from the top two entries might not be directly comparable to the rest.

5 DISCUSSION

We also experimented with adding L2 loss, trying different drop-in replacement architectures (DenseNet [9], Xception [3], ResNet [7]), but they all proved to be outperformed by VGG.

We discovered that conditioning our model on head pose does not improve the results, which we believe is partially caused by the normalization procedure when collecting MPIIGaze dataset. Taking the average as the face model is inaccurate, and therefore, introduces a lot of noise.

Our image augmentation (contrast + noise) did improve the convergence speed, but the overall performance remained largely the same.

For transfer learning, we experimented with adding a pre-training phase for training on UnityEyes for a number of steps before training on MPIIGaze, or initializing weights of the convolutions that were pre-trained on ImageNet.

Unfortunately, this proved to not be of any help, as once the pre-training was finished, the model's loss went up significantly (to the point of having a random initialization) and in the end re-learned everything "from scratch" and performed the same as without any pre-training or initialization. ImageNet pre-trained models expect very different scenes; however, we are not sure, why pre-training on synthetic images does not boost the performance.

Furthermore, we found out that all our models are very sensitive to hyper-parameter changes, especially learning rate.

6 CONCLUSION AND FUTURE WORK

In conclusion, our contribution was two-fold. First, we found a way to refine the synthetic dataset and obtain better pre-training results. Second, we examined several convolutional neural network architectures, measured their performance, and attempted to explain it.

ACKNOWLEDGMENTS

We would like to thank the Advanced Interactive Technologies Lab for providing the code skeleton and computational resources for this project, and answering all our questions; we would also like to thank Microsoft for the provided computational resources on Azure.

REFERENCES

[1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, Manjunath Kudlur, Josh Levenberg, Rajat Monga, Sherry Moore, Derek G Murray, Benoit Steiner, Paul Tucker, Vijay Vasudevan, Pete Warden, Martin Wicke, Yuan Yu, Xiaoqiang Zheng, and Google Brain. 2016. TensorFlow: A System for Large-Scale Machine Learning. In *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI '16)*. 265–284. https://doi.org/10.1038/nn.3331

[2] Kenneth Alberto Funes Mora and Jean-Marc Odobez. 2014. Geometric generative gaze estimation (g3e) for remote rgb-d cameras. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1773–1780.

[3] François Chollet. 2016. Xception: Deep Learning with Depthwise Separable Convolutions. *CoRR* abs/1610.02357 (2016). arXiv:1610.02357 http://arxiv.org/abs/1610.02357

[4] François Chollet et al. 2015. Keras. https://keras.io.

[5] Stefania Cristina and Kenneth P. Camilleri. 2016. Model-based Head Pose-free Gaze Estimation for Assistive Communication. *Comput. Vis. Image Underst.* 149, C (Aug. 2016), 157–170. https://doi.org/10.1016/j.cviu.2016.02.012

[6] Kenneth Alberto Funes Mora, Florent Monay, and Jean-Marc Odobez. 2014. EYEDIAP: A Database for the Development and Evaluation of Gaze Estimation Algorithms from RGB and RGB-D Cameras. In *Proceedings of the ACM Symposium on Eye Tracking Research and Applications*. ACM. https://doi.org/10.1145/2578153.2578190

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[8] Geoffrey E. Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R. Salakhutdinov. 2012. Improving neural networks by preventing co-adaptation of feature detectors. (2012). https://doi.org/arXiv:1207.0580

[9] Gao Huang, Zhuang Liu, L v. d. Maaten, and Kilian Q Weinberger. 2017. Densely Connected Convolutional Networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2261–2269. https://doi.org/10.1109/CVPR.2017.243

[10] Diederik P. Kingma and Jimmy Lei Ba. 2015. Adam: A Method for Stochastic Optimization. *International Conference on Learning Representations (ICRL)* (dec 2015). http://arxiv.org/abs/1412.6980

[11] Kyle Krafka, Aditya Khosla, Petr Kellnhofer, and Harini Kannan. 2016. Eye Tracking for Everyone. *IEEE Conference on Computer Vision and Pattern Recognition* (2016), 2176–2184. https://doi.org/10.1109/CVPR.2016.239

[12] Seonwook Park, Xucong Zhang, Andreas Bulling, and Otmar Hilliges. 2018. Learning to Find Eye Region Landmarks for Remote Gaze Estimation in Unconstrained Settings. In *ACM Symposium on Eye Tracking Research and Applications (ETRA) (ETRA '18)*. ACM, New York, NY, USA.

[13] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C Berg, and Li Fei-Fei. 2015. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision* 115, 3 (2015), 211–252. https://doi.org/10.1007/s11263-015-0816-y

[14] Ashish Shrivastava, Tomas Pfister, Oncel Tuzel, Josh Susskind, Wenda Wang, and Russell Webb. 2016. Learning from Simulated and Unsupervised Images through Adversarial Training. *CoRR* abs/1612.07828 (2016). arXiv:1612.07828 http://arxiv.org/abs/1612.07828

[15] Karen Simonyan and Andrew Zisserman. 2015. Very Deep Convolutional Networks for Large-Scale Image Recognition. *International Conference on Learning Representations (ICRL)* (sep 2015), 1–14. https://doi.org/10.1016/j.infsof.2008.09.005

[16] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research* 15 (2014), 1929–1958. https://doi.org/10.1214/12-AOS1000

[17] Erroll Wood, Tadas Baltrušaitis, Louis-Philippe Morency, Peter Robinson, and Andreas Bulling. 2016. Learning an appearance-based gaze estimator from one million synthesised images. In *Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research & Applications - ETRA '16*. 131–138. https://doi.org/10.1145/2857491.2857492

[18] Erroll Wood, Tadas Baltrusaitis, Xucong Zhang, Yusuke Sugano, Peter Robinson, and Andreas Bulling. 2015. Rendering of Eyes for Eye-Shape Registration and

Gaze Estimation. In *Proc. of the IEEE International Conference on Computer Vision (ICCV 2015)* (2015-12-12).

[19] Xucong Zhang, Yusuke Sugano, Mario Fritz, and Andreas Bulling. 2015. Appearance-based Gaze Estimation in the Wild. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 4511–4520.

[20] Xucong Zhang, Yusuke Sugano, Mario Fritz, and Andreas Bulling. 2017. ItâĂŹs 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.

[21] 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* (2017). https://doi.org/10.1109/TPAMI.2017.2778103

[22] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. 2017. Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks. *CoRR* abs/1703.10593 (2017). arXiv:1703.10593 http://arxiv.org/abs/1703.10593