Self-Adaptive Appearance-Based Eye-Tracking with Online Transfer Learning

Bruno Klein Salvalaio
SAP Labs Latin America
São Leopoldo, RS, Brazil
bruno.klein01@sap.com

Gabriel de Oliveira Ramos
Universidade do Vale do Rio dos Sinos
São Leopoldo, RS, Brazil
gdoramos@unisinos.br

***Abstract*—Eye-tracking plays a role in human-computer interactions and has proven useful in a wide variety of domains. We consider appearance-based eye-tracking, where one tracks eye movements based solely on conventional images (rather than on sophisticated additional hardware). Recent advances made in Deep Learning and, in particular, convolutional neural networks have allowed appearance-based eye-tracking to achieve better results than ever. However, current literature still lacks methods that generalize to different combinations of user, environment and device. In this work, we introduce Online Deep Appearance-Based Eye-Tracking (ODABE), which overcomes such a limitation by considering online transfer learning, thus enabling eye-tracking models to self-adapt to different context very rapidly. Our results show that ODABE improves upon previous research when context changes, decreasing the prediction error by 50.95% on average, on tested cases.**

***Keywords*-Deep Learning, Eye-Tracking, Gaze-Tracking, Transfer Learning**

I. Introduction

Eye-tracking represents an important human-computer interaction scheme. Eye-tracking provides a mean to infer the *gaze* or the position the user is looking at, based on imaging or sensorial data. Not surprisingly, eye-tracking applications include health-care [1], human computer interface [2], neuroscience, industrial engineering and advertising [3].

Eye-tracking techniques can be classified based on their input assumptions [3]. Shape-based approaches rely on the shape of the eye and pupils and use border detection algorithms, such as image gradients. Feature-based shape approaches are similar to the prior, in that they also make of use of the eye, only to identify characteristics around it that are less sensitive to variations in illumination and viewpoint. Appearance-based methods are able to detect eyes directly, based only on image data. Moreover, appearance-based methods are also less susceptible to illumination variance, and can detect and track other objects besides eyes in a multi-tasking fashion. Overall these methods are easier to implement and more robust as they work in an end-to-end manner and do not require specialized equipment, only an image which can be provided by a consumer grade webcam [4].

Recently, Deep Learning has shown great modelling capabilities [5], achieving state-of-the-art results in many computer vision problems, most notably [6]. As a consequence, many appearance-based eye-tracking proposals have been made, reaching state-of-the-art results in different domains [7]–[9]. These solutions use Deep Learning to train models in an end-to-end fashion, which are then able to infer the coordinates a user is looking on the screen. Although these solutions have introduced extensive datasets, such as [10], they fail to ultimately generalize to any user, environment or device combination, which we will refer to simply as the problem *context*. Masko [9] shows that, by using transfer learning, one can achieve better models for specific contexts at development time only, limiting the value added by using only this technique on its own.

In this paper, we introduce Online Deep Appearance-Based Eye-Tracking (ODABE, for short). ODABE builds upon the concept of transfer learning in an online streaming setting, which allows a model learned for a specific context to be adjusted to new contexts. Specifically, our method is pre-trained using an existing dataset [8] and then, when a new context arises, it keeps improving and fine-tuning the model using the idea of online transfer learning. In order to validate our approach, we also present a novel dataset, which includes samples of different contexts to validate the online learning capability of ODABE. We have empirically evaluated our approach using these datasets. Our results indicate that ODABE is able to improve upon previous research results when new contexts arise, being able to decrease the prediction error by 50.95% on average.

This paper is organized as follows. Section II addresses and references some of the methods used in our proposal. Section III presents an overview of related work. Section IV introduces the datasets used in this work. Section V devises our main contribution, ODABE. Our method is evaluated in Section VI. Concluding remarks and future work direction are discussed in Section VII.

II. Background

In this section we present the background upon which we build our work.

A. Neural Networks

Neural Networks are universal function approximators, used to map input values x into expected output values y. This is called the Supervised Learning paradigm. By mapping these inputs to outputs, neural networks are expected to not

2643-6264/19/$31.00 ©2019 IEEE
DOI 10.1109/BRACIS.2019.00074

memorize instances of samples, but to capture patterns and behave accordingly for unseen instances. In the case of this work, the x is an image of the user gazing at the screen and the y is a tuple containing the actual screen coordinates the user was staring when x was taken.

The networks which have several layers are called Deep Neural Networks, studies have shown that the deeper a network is, the higher the order of the function it is able to model, by being able to take advantage of larger datasets [11].

B. Convolutional Neural Networks

Convolutional Neural Networks (CNNs) [12] are a type of neural network that use several steps of convolutions, which is an operation that applies a function over several overlapping spatial areas of the input, making so the networks activations are only high when certains visual patterns occur in the image. Lower layers in the network are strongly activated when simple features are observed, for instance particularly shaped lines. Deeper layers, on the other hand, when combinations of the shallow layers is observed, meaning they activate when finding, in some instances, entire objects within an image. We refer the reader to [11] for a thorough overview of CNNs.

In our work we also make use of Batch Normalization, which helps us to deal with Internal Covariate Shift [13]. ICS is a phenomenon that happens when training networks using mini-batches of data. Because each mini-batch on their own is not a good representation of the distribution of the entire dataset. At each optimization step the network makes weights adjustments that are optimal for the particular mini-batch, but not for the entire dataset. By using Batch Normalization, the network gets a chance to learn parameters to normalize the covariates being given as input at each block, making sure the distribution of the mini-batches is more similar to one another. This is particularly useful in the context of online learning where there is a single training sample, for each step, enabling the model to online converge faster.

III. Related Work

We now discuss representative literature on eye-tracking. There are several works which make use of CNNs to create models that learn to infer user gaze and on screen coordinates end-to-end without relying on hand made features. Zhang et al. [14] used a multimodal CNN to infer gaze, and also contributed with a so called real world dataset [15], which is collected from users doing daily activies, outside of the controlled environment of a lab. They crop the user face, use landmark detection algorithms to be able to normalize the position of the face within the frame, then they extract eyes and give that as input to the network. Because their proposal is based on trying to infer user gaze angles, their solution is inherently highly coupled with the specifics contexts used during trained. So, even though they make valuable contributions, namely their dataset MPIIGaze, their overall proposal is still very much susceptible to bad generalization.

Krafka et al. [7] built a 2.5M samples dataset, collected from users using mobile devices. They used mobile devices because these are more widely spread nowadays than other alternatives, such as workstations. Their data was acquired by crowd sourcing, the same as in [16]. They use a multimodal approach in which the network receives as input a picture of each eye, the face, and a grid which encodes the position of the face within the frame. Such as us, they also infer the on screen coordinates using a euclidean distance between target and actual. They achieve a error of 1.71cm and 2.53cm in phones and tablets respectively.

Algorithm 1: Data collection procedure employed to build our dataset.

```
while True do
    $x \sim U(0, 1)$;
    $y \sim U(0, 1)$;
    $plot\_circle(x, y)$;
    // at this point, the user should be looking at the
      circle
    $wait\_for\_key()$;
    $p \leftarrow fetch\_webcam\_frame()$;
    $persist(p, x, y)$;
end
```

The work of Zhang et al. [17], follows with that of Krafka et al. [7] with a more elegant approach in which they use a Spatial Weights CNN, where only the face is given as input to the model. They argue that there might be cues in the face that are relevant for gaze inference.

Transfer Learning has also been used to verify its capabilities in provided unconstrained context generalization. In [9], the author shows that Transfer Learning can be used to bring final user observed performance improvements of up to 21%.

These solutions, when put in practice do not really generalize well for unseen contexts, leaving room for further improvement. Our aim in this work is to explore what we can get from Transfer Learning in an online fashion that does not rely on pre-training for a specific user, but instead can be trained by the user's device while the application is already deployed.

IV. Dataset

In this work, we use MPIIGaze [8] to create the baseline model which gets fine-tuned during online transfer learning. MPIIGaze was originally created for gaze estimation in uncontrolled contexts. Here, MPIIGaze is used only for pre-training our model.

As for the online part of our algorithm, we introduce a new dataset. The new dataset consists of 1000 images, split evenly among two different environments. We use this dataset to feed the model, one image at a time, when doing online transfer learning. We created this dataset because the images contained in them are all related and taken in sequences of user interaction with the workstation. This contrasts with other datasets, which cannot be used to evaluate the online learning training scheme because they do not have this sequential nature.

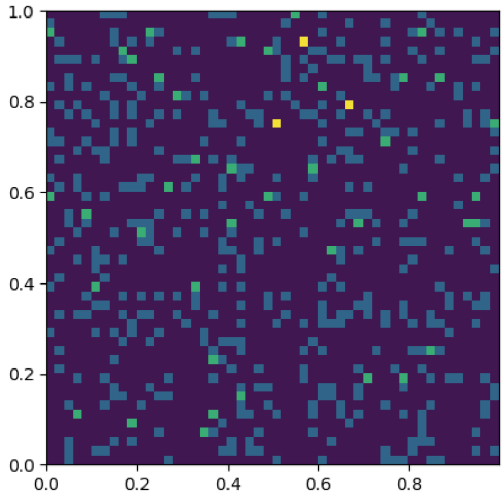

Fig. 1. Heatmap showing the distribution of user gaze on our own data collection process. The brighther the spot, the higher the number of gaze examplea at the specific coordinate.

In order to collect data for our dataset, we developed a small application that interacts with the user, collecting images (from a conventional webcam) and screen coordinates. Specifically, at each iteration, the application displays a circle on the screen. The screen coordinates of the circle, (x, y), are drawn uniformly at random. The user then keeps looking at the circle and presses a pre-defined keyboard key to confirm. Once the key is pressed, the application saves the webcam image and the circle coordinates as the input and output samples, respectively. This whole process is repeated several times, once for creating each sample on the dataset. An sketch of this procedure is described in Algorithm 1. A heatmap of the coordinates of the collected samples is shown in Figure 1.

In order to normalize data, we crop the images from both MPIIGaze and our dataset so as to keep only the faces. To this end, we used dlib [18], which yields different results from the pre-made cropping already existent in MPIIGaze. To keep both datasets similar, we also applied dlib's face cropping in MPIIGaze. Figure 3 shows as example an image used as input (left) and the resulting image after cropping (right).

We also normalized pixel values, to the range $[0, 1]$ and normalized pixel values dataset wise, image channel wise, by subtracting the mean and dividing by standard deviation.

A small sample of the datasets after pre-processing is shown in Figure 3. In the figure, images on the left are from MPIIGaze and images on the right are from our new dataset.

V. Online Deep Appearance-Based Eye-Tracking

In this section we introduce Online Deep Appearance-Based Eye-Tracking (ODABE, for short). ODABE builds upon previous works [7], [14], [15] and includes an online learning fine-tuning procedure on top of our new architecture. Our approach also performs batch normalization and it is

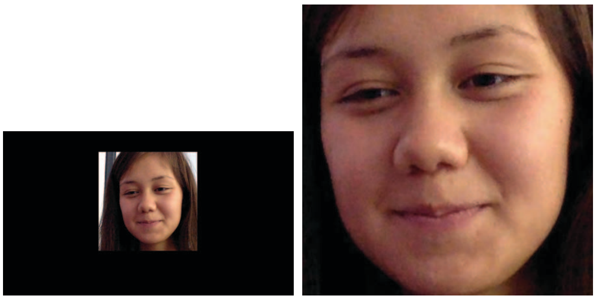

Fig. 2. Example of the cropping process. On the left, the original image from MPIIGaze. On the right, the post-processed image after cropping, which is then used by our method.

trained from scratch. In particular, ODABE extends previous approaches by introducing the online learning element, which is able to generalize without creating coupling related to user appearance, environment characteristics and device properties. ODABE is composed by two parts: the pre-training of the baseline model (Section V-A), and the online model fine-tuning within a new context (Section V-B).

A. Pre-training

During online learning, we fine-tune a pre-trained model in order to make the online learning process converge faster to an acceptable user perceived result. The resulting model is then expected to suited better the current combination of user, environment, and device. To this end, we train a baseline model at development time by using 213.650 images from MPIIGaze [15]. Before training, the network's parameters are initialized with pre-existing weights from ImageNet [6]. MPIIGaze is split in three parts, train, validation and test, resulting in 75%, 15%, 15% splits respectively. Hyperparameters were empirically chosen based on observed performance on the validation set. When an optimal model is converged, it is finally tested on the test split.

The data is also normalized channel-wise, for which the mean and standard deviation of each color channel in MPIIGaze is computed. The final images being input into the network have the respective channel mean subtracted and divided by the standard deviation for the respective channel.

The network used is a VGG11 [19], with added batch normalization, as depicted in Figure 4. Batch Normalization is added to reduce the aggravated Internal Covariate Shift for online learning inexistent batch size, since images are consumed one at a time [13]. Batch Normalization works by learning parameters which attempt to make the dataset mean and variance similar among all batches.

B. Online Learning

Given the pre-trained model, the purpose of the online learning training procedure is to fine-tune some layers of the same network used for pre-training and therefore adapt them to be more suitable in a given new context, which the user might have just transitioned to.

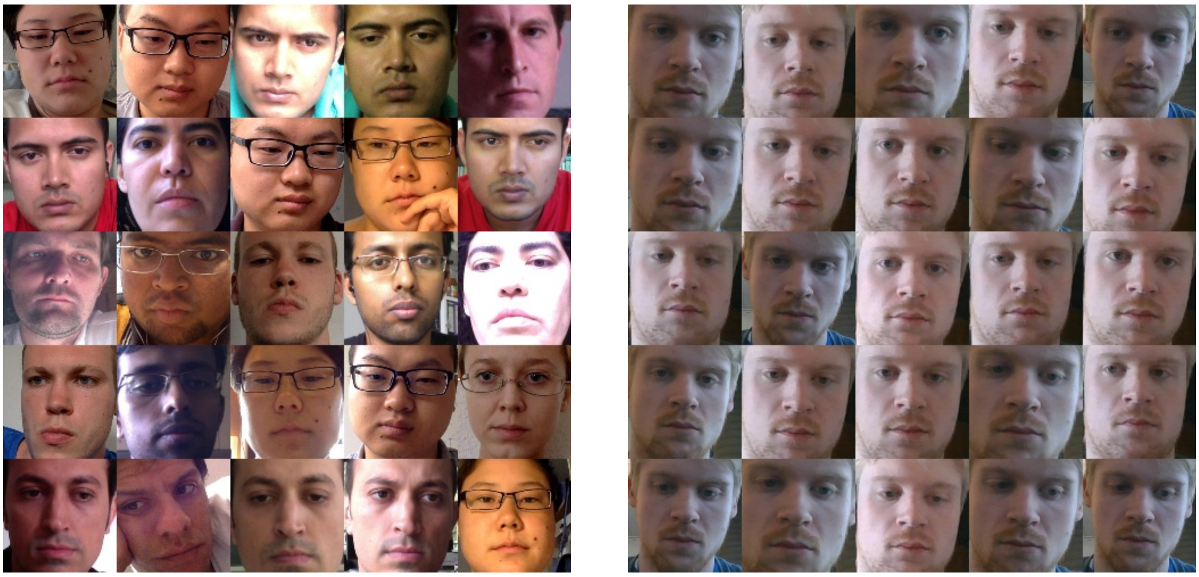

Fig. 3. Small sample of the datasets used in this work. In the left, a sample of the pre-processed images from MPIIGaze. In the right, a sample of the pre-processed images collected by us.

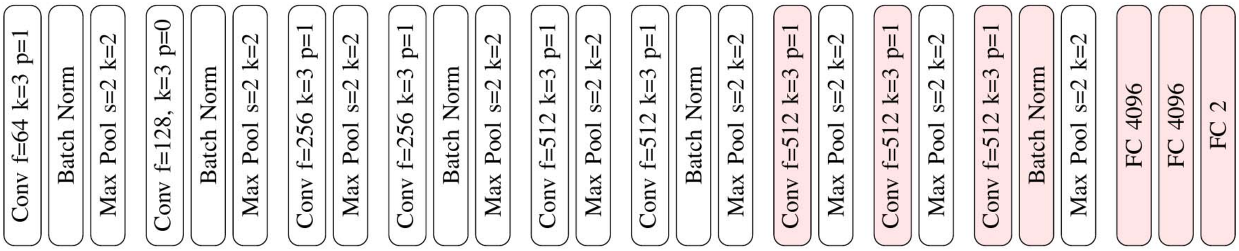

Fig. 4. The employed VGG11 network architecture, with added *Batch Normalization*. The red layers are fine-tuned during *Online Learning*.

.

The online learning process works in the same way as the pre-training. Here, images are normalized based on statistics from MPIIGaze, since the model is only being fine-tuned, and considering that images are being acquired and trained on during user usage of the model. They are captured whenever the user does a mouse click, at which point, an image of the user gaze is taken and the mouse click coordinates are used as ground truth. To make training a streamlined process we collect the online learning dataset as described in section IV and train using a single epoch and batch-size 1.

During online learning the training procedure has no feedback of overfitting such as the validation and test split during pre-training. So what is done is that each new image is forwarded through the network and its loss is computed. If the loss is above a pre-defined threshold, then we backprogate for that particular image, updating the model weights, otherwise we discard it and the model remains the same. The threshold used is the final mean loss for the validation set found during pre-training. The heuristics come from the observation that the training/validation loss for pre-training can be used as a proxy for the network modelling capabilities.

VI. Experimental Evaluation

This section presents the experimental evaluation of our method. Our aim is to test whether ODABE behaves well when different contexts arise.

A. *Methodology*

As explained in previous sections, before doing online-learning, first we pre-trained a model suited for fine-tuning during online learning. The model pre-training is done with the dataset MPIIGaze [15]. After the baseline is trained we have to test how it behaves with new images which we collected, as explained in Section IV. For MPIIGaze we did a pre-processing, also explained in Section IV, since images from MPIIGaze have a different cropping scheme.

The network used is a VGG11, depicted in Figure 4, where the red layers are the ones which have their parameters adjusted during online learning, the remaining ones have fixed

TABLE I
RESULT OF EXPERIMENTS WITH AND WITHOUT ONLINE LEARNING

Stage	Env. 1	Env. 2
without *Online Learning* (valid)	0.330 ± 0.025	0.266 ± 0.031
with *Online Learning* (train)	0.150 ± 0.027	0.188 ± 0.021
with *Online Learning* (valid)	0.150 ± 0.017	0.140 ± 0.004
%	54.54%	47.36%

parameters, during the online learning phase. Pre-training is done for four epochs with a batch-size of 128. Stochastic Gradient Descent is used with a learning-rate of 0.001 and momentum 0.9. During online learning, the same learning rate is used.

The objective function is an Euclidean distance, as shown in Equation 1. This function is used for training is the same both for pre-training and online learning.

$$L((x,y),(\hat{x},\hat{y})) = \frac{1}{N}\sum_{i=1}^{N}\sqrt{(x_i-\hat{x}_i)^2+(y_i-\hat{y}_i)^2} \tag{1}$$

We remark that the screen coordinates are normalized on the range $[0,1]$, meaning the diagonal distance of the screen is $\sqrt{2}$, which is therefore the highest error the model can make.

During online-learning, for reproducibility reasons, we train each environment for only a single epoch with batch-size, to emulate the online learning nature of the incoming data.

B. Numerical Results

Table I shows the overall results of our experiments, which were ran in two different environments, one after the other, with the online learning dataset collected, as explained in Section IV. The table shows the comparison between the scenarios with online learning and without online learning (previous works). The training procedure was ran a total of five times, the shown results are the mean and standard deviation. As seen, an improvement of 54.54% is observed in the first environment and 47.36% in the second scenario as compared to the baseline without online learning. This corresponds to an average decrease in the prediction error by 50.95% on average, as compared to the scenario where online learning is not used.

The evolution of the loss along iterations is shown in Figure 5. As seen, the loss consistently decreases while the the context is fixed. When a new context emerges, our method quickly adapts to it, keeping the loss somewhat stable in all new contexts. This situation can be seen in more detail in Figure 6, where the same curves are shown but for a smaller interval. Therefore, our results evidence that our method is able to properly, and effectively, handle new contexts, which gives an important advantage to our approach as compared to previous ones.

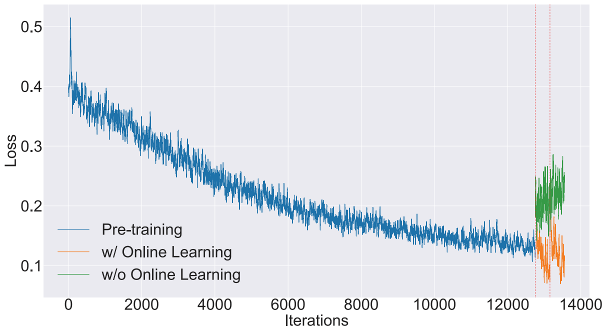

Fig. 5. Loss along iterations. In blue, the train loss during pre-training, in orange, the loss during online learning in the first environment, and green during online learning in the second environment.

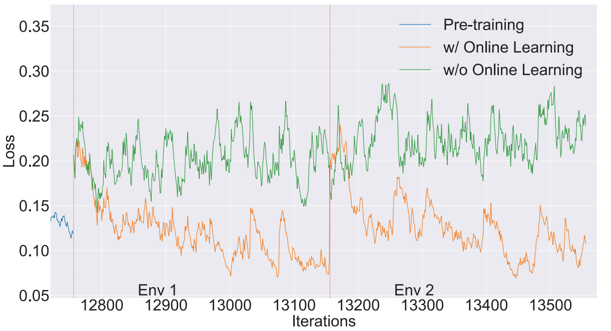

Fig. 6. Loss along iterations for a small window of Figure 5, from iterations 12700 and 13600. In blue, the train loss during pre-training, in orange, the loss during online learning in the first environment, and green during online learning in the second environment.

VII. CONCLUSION

In this paper, we introduced ODABE, an online deep appearance-based eye-tracking algorithm that effectively tackles different eye-tracking contexts. To this end, ODABE pre-trains a model using an existing dataset and then, as new contexts arise, uses online transfer learning to fine-tune the learned model. Additionally, we also devised a new dataset in an attempt to foster research on online, appearance-based eye-tracking. We performed experiments, which empirically shown that ODABE outperforms state-of-the-art eye-tracking methods when new contexts arise, being able to decrease the prediction error by 50.95% on average.

As future work, we would like to expand our dataset, which is publicly available at (double blind-review), with more contexts. Furthermore, we plan to further reduce the prediction error by experimenting variation on the network architecture.

ACKNOWLEDGMENT

We thank the anonymous reviewers for the valuable feedback. This research was partially supported by FAPERGS and SAP Labs Latin America.

REFERENCES

[1] R. S. Keefe, J. M. Silverman, R. C. Mohs, L. J. Siever, P. D. Harvey, L. Friedman, S. E. L. Roitman, R. L. DuPre, C. J. Smith, J. Schmeidler *et al.*, “Eye tracking, attention, and schizotypal symptoms in nonpsychotic relatives of patients with schizophrenia,” *Archives of general psychiatry*, vol. 54, no. 2, pp. 169–176, 1997.

[2] R. J. Jacob and K. S. Karn, “Eye tracking in human-computer interaction and usability research: Ready to deliver the promises,” in *The mind’s eye*. Elsevier, 2003, pp. 573–605.

[3] A. T. Duchowski, *Eye Tracking Methodology: Theory and Practice*. Berlin, Heidelberg: Springer-Verlag, 2007.

[4] D. W. Hansen and Q. Ji, "In the eye of the beholder: A survey of models for eyes and gaze," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 3, pp. 478–500, March 2010.

[5] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, "Understanding deep learning requires rethinking generalization," *CoRR*, vol. abs/1611.03530, 2017.

[6] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," in *CVPR09*, 2009.

[7] K. Krafka, A. Khosla, P. Kellnhofer, H. Kannan, S. Bhandarkar, W. Matusik, and A. Torralba, "Eye tracking for everyone," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[8] X. Zhang, Y. Sugano, M. Fritz, and A. Bulling, "Appearance-based gaze estimation in the wild," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015, pp. 4511–4520.

[9] D. Masko, "Calibration in eye tracking using transfer learning," Master's thesis, KTH, School of Computer Science and Communication (CSC), 2017.

[10] X. Zhang, Y. Sugano, M. Fritz, and A. Bulling, "Mpiigaze: Real-world dataset and deep appearance-based gaze estimation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 1, pp. 162–175, Jan 2019.

[11] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[12] Y. LeCun *et al.*, "Generalization and network design strategies," in *Connectionism in perspective*, 1989, vol. 19.

[13] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37*, ser. ICML'15. JMLR.org, 2015, pp. 448–456. [Online]. Available: http://dl.acm.org/citation.cfm?id=3045118.3045167

[14] X. Zhang, Y. Sugano, M. Fritz, and A. Bulling, "Appearance-based gaze estimation in the wild," *CoRR*, vol. abs/1504.02863, 2015. [Online]. Available: http://arxiv.org/abs/1504.02863

[15] ——, "Mpiigaze: Real-world dataset and deep appearance-based gaze estimation," *CoRR*, vol. abs/1711.09017, 2017. [Online]. Available: http://arxiv.org/abs/1711.09017

[16] P. Xu, K. A. Ehinger, Y. Zhang, A. Finkelstein, S. R. Kulkarni, and J. Xiao, "Turkergaze: Crowdsourcing saliency with webcam based eye tracking," *CoRR*, vol. abs/1504.06755, 2015. [Online]. Available: http://arxiv.org/abs/1504.06755

[17] X. Zhang, Y. Sugano, M. Fritz, and A. Bulling, "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, 2017, pp. 2299–2308.

[18] D. E. King, "Dlib-ml: A machine learning toolkit," *Journal of Machine Learning Research*, vol. 10, pp. 1755–1758, 2009.

[19] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.