

# Gaze-Tracking with Convolutional Neural Networks

Alberto Castañeda Arana

## CONTENTS

<b>I.</b>	<b>State-of-the-art</b>	1
I-A.	Gaze-Tracking . . . . .	1
I-B.	Model-based methods . . . . .	1
I-C.	Appearance-based methods . . . . .	1
I-D.	Related Work . . . . .	2
I-E.	Datasets . . . . .	3
<b>II.</b>	<b>Evaluation</b>	3

## I. STATE-OF-THE-ART

### I-A. Gaze-Tracking

Applications of estimating the human eye gaze direction, or Gaze-Tracking, can be found in multiple study areas such as health care [4], virtual-reality [10], and human-computer interaction [9]. Devices that utilize gaze-tracking methods for human-computer interaction are often referred to as Eye Trackers. While there is commercial use for Eye Trackers, they have not reached the accuracy needed to allow users with fine-motor disabilities to interact with a computer by using their own eyes as a replacement for a traditional mouse device.

Conventional methods for gaze-tracking require additional devices such as infrared light in order to make the gaze estimations. However, these methods are prone to error in poor light conditions.

Research in computer vision and Deep Learning have allowed improvement in Gaze-Tracking systems by allowing better predictions while also being less intrusive for users with no additional devices required.

Gaze-Tracking with computer vision can be achieved with two different type of methods: Model-based methods and Appearance-based methods. These methods will be described in the following sections.

### I-B. Model-based methods

Model-based methods, or sometimes referred to as feature-based methods, use the geometric form of the eyes to make the eye gaze prediction. As seen on Fig. 1, these can be interpreted as 3D models. Model-based 3D Gaze-Estimation methods use 3D eyeball models and estimate the gaze direction using geometric eye features, such as the iris center and the eye corners [5].

Furthermore, most these methods require an infrared light source (IR) in order to create a refraction in the human eye which is then used to calculate the eye gaze direction by comparing the refraction position with the pupil position. The main drawback of model-based methods is that they can

have a lower accuracy with low-resolution images and poor-condition lighting environments. Additionally, all of them have limited working distance.

Some state-of-the-art commercial products that have adapted these techniques are Tobii Eye Tracker (<https://www.tobii.com/>), Eye Tribe (<https://imotions.com/hardware/the-eye-tribe-tracker/>) and EyeSee (<https://eyesee-research.com/>). While these products allow very accurate predictions for the Eye Tracking, they suffer from the previously mentioned model based methods disadvantages while also requiring users to purchase additional expensive equipment. Most Eye-Tracking products also require wearable glasses peripherals that can be intrusive for daily usage.

With the increase of interest and usage of Gaze-Tracking, there is also a growing need for new Eye-Trackers that are cheap and easy to use. Gaze-Tracking methods that utilize built-in or external cameras included in almost every computer can be used as a way of making eye tracking more accessible. Appearance-based approaches are beginning to be researched to achieve the previous goal.

### I-C. Appearance-based methods

Recently, investigations for appearance based methods with Deep Learning has surfaced which have demonstrated high accuracy while only using on-the-shelf web cameras to capture the human eye appearance hence the method name.

Conventional appearance-based methods used regression in order to learn the mappings from appearance to human gaze. However, since these methods use the appearance of the eye, the main challenges are head motion and subject differences that alter the eye appearance on use.

To overcome the previous challenges, appearance-based Gaze-Estimation with Deep Learning has become a trend in research. Deep Learning based methods can extract high-level abstract gaze features from high-dimensional images, making them more robust and accurate than conventional methods. The mapping between eye-images and gaze direction can be achieved with various regression techniques, such as neural networks, local interpolation, or Gaussian process regression. Convolutional Neural Networks (CNNs) models are commonly used in computer vision tasks for its high accuracy and performance. Current state-of-the-art solutions have begun to adopt CNN based architectures, adopting the advantages of appearance-based methods and achieving high accuracy. CNN models used for Gaze-Tracking learn from RGB images which are preprocessed to segment the human eyes in training.

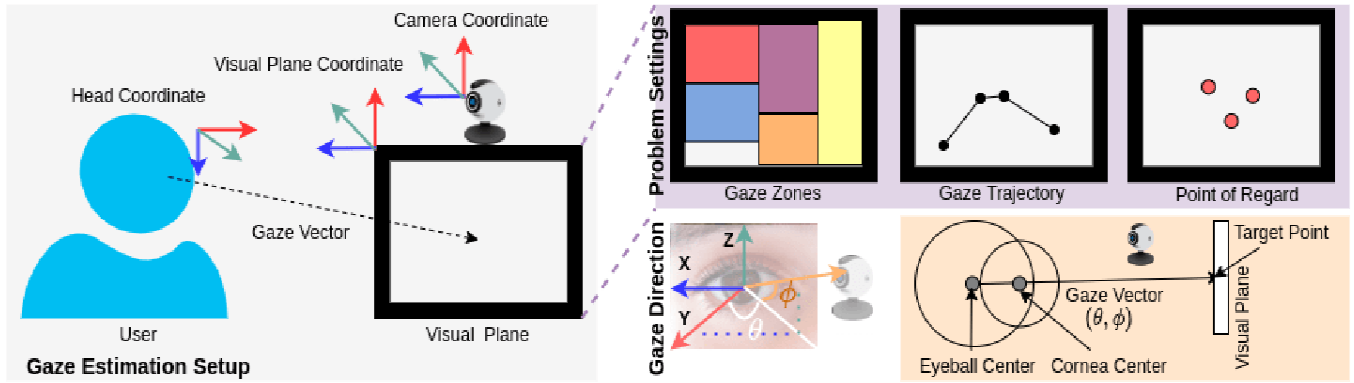


Fig. 1. Overview of Gaze-Estimation setups. Recovered from [3]

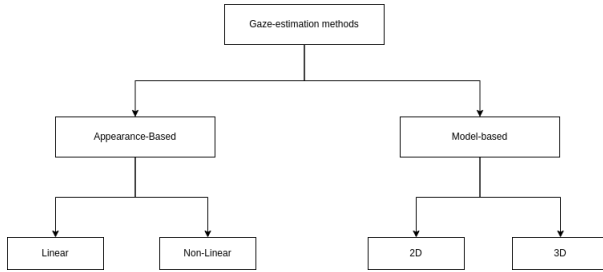


Fig. 2. Overview classification of Gaze-Tracking methods

Some of the drawbacks that appearance-based methods have been that they require more images for training than model-based methods, which are not easily retrieved. Additionally, since most appearance-based methods require deep-learning architectures such as Convolutional Neural Networks, they require higher computational power for training. Several works apply CNNs for gaze estimation, as they have shown to outperform traditional model-based approaches [11].

#### I-D. Related Work

##### Multimodal Convolutional Neural Network

Zhang et al. proposed a six-layered CNN based on LeNet that takes the eye image as input and combines the head pose in the last fully connected layer of the network [11]. The results demonstrated a large performance gap between person-specific training results (RF model) with the proposed CNN model and revealed the potential of appearance-based gaze estimation.

##### Spatial Weight CNN

After the previous proposal, Zhang et al. proposed a spatial weights CNN based on the AlexNet architecture [8] that takes the full face image as input [12]. The spatial weights classify the importance of different facial areas in order to perform the Gaze-Estimation. This mechanism includes three additional  $1 \times 1$  convolutional layer followed by a ReLU activation. Compared to other approaches only use eye regions, this method was demonstrated to be more robust against appearance variation such as head pose as well as illumination. The results achieved an error of  $4.8-6.0^\circ$ .

##### iTracker

Kannan et al. proposed a CNN architecture for an iOS based Gaze-Tracking system called iTracker [6]. The model's large neural network was constructed with multiple smaller neural networks, with four total inputs: a right eye image, a left eye image, a face image, and a face-grid. The face-grid is a  $25 \times 25$  binary grid that indicates where the user face is placed relatively to the camera view.

This model achieved 1.66 centimeter error with no calibration needed. The GazeCapture dataset which is described in Section 1-E was used to train this model.

While achieving good results, this model is biased with mobile usage which makes it unusable for laptop or computer desktop usage. Additionally, the authors describe there being room for improvement for better segmentation of eyes. However, by adjusting the model's face-grid dimensions to simulate webcam dimensions and with a different dataset, the model could potentially be used for desktop/laptop use.

##### Geometric & Texture-based Networks

Jyoti et al. proposed an ensemble of networks that uses the full face, left and right eye, similarly to iTracker [1]. A Deep Neural Network (DNN) is trained using the face geometric features. Secondly, three CNNs are trained for the texture based features i.e. the segment of the left eye, right eye, and the combined eyes. The model was trained with the publicly available Columbia Eye Gaze and TabletGaze datasets for experiments. When trained with the Columbia Eye Gaze dataset, experiments resulted in an error of  $2.22^\circ$ . Additionally, the authors emphasize on the importance of choosing different activation functions for different results by comparing the model outcome with ReLU vs Swish, where the former achieved better performance for geometric network while Swish showed better performance on the rest.

##### RT-GENE

Fischer et al. proposed a two-stream VGG network that takes the left and right as inputs [2]. This approach considers large camera-to-user distances and high variations in head pose found in natural environments. This approach allows automatic annotation of subject's ground truth gaze and head pose regardless of large camera-user distances, achieving an

error of  $7.7^\circ$ . However, the approach setup requires additional devices such as mobile eye-tracking glasses (Kinect v2 RGB-D camera).

#### Hybrid Two-step Training

Deng and Zhu proposed a two-step training architecture, where a head CNN and eye CNN are independently trained and then together transformed with a "gaze transform layer"[1]. The proposed model does not suffer from head-gaze correlation overfitting which makes it possible to use existing datasets for training. The solution achieves an error of  $5.6^\circ$  when trained with a large dataset, achieving state-of-the-art gaze tracking accuracy.

#### I-E. Datasets

Until recently, datasets for Gaze-Estimation models training were built manually by taking pictures of different users' faces by placing their head firmly in a stand. Training models require multiple high-quality images, which makes the previous methods slow and not ideal. To improve the previous methods' problem, multiple data sets containing images for eye tracking have been compiled with crowd founding to train new models.

#### GazeCapture

Krafka et al. built and released the GazeCapture dataset, which crowdfunded a mobile-based data set of approximately 1500 subjects from a wide variety of backgrounds captured by an iOS application [7]. However, given that this dataset is built with mobile users, training with this dataset may lead to a huge bias towards mobile users which could lead to poor results with laptop or desktop use.

#### MMPIIGaze

Another state-of-the-art dataset that has been widely used is the MMPIIGaze dataset created by Zhang et al. [11]

#### Columbia Eye Gaze

#### TabletGaze

## II. EVALUATION

### REFERENCES

- [1] Haoping Deng y Wangjiang Zhu. «Monocular Free-Head 3D Gaze Tracking with Deep Learning and Geometry Constraints». En: oct. de 2017, págs. 3162-3171. DOI: 10.1109/ICCV.2017.341.
- [2] Tobias Fischer, Hyung Chang y Yiannis Demiris. «RT-GENE: Real-Time Eye Gaze Estimation in Natural Environments». En: sep. de 2018. DOI: 10.1007/978-3-030-01249-6\_21.
- [3] Shreya Ghosh y col. «Automatic Gaze Analysis: A Survey of Deep Learning based Approaches». En: *arXiv:2108.05479 [cs]* (ago. de 2021). arXiv: 2108.05479. URL: <http://arxiv.org/abs/2108.05479> (visitado 31-03-2022).
- [4] Alessandro Grillini y col. «Towards Using the Spatio-Temporal Properties of Eye Movements to Classify Visual Field Defects». En: *Proceedings of the 2018 ACM Symposium on Eye Tracking Research Applications*. ETRA '18. Warsaw, Poland: Association for Computing Machinery, 2018. ISBN: 9781450357067. DOI: 10.1145/3204493.3204590. URL: <https://doi.org/10.1145/3204493.3204590>.
- [5] E.D. Guestrin y M. Eizenman. «General theory of remote gaze estimation using the pupil center and corneal reflections». En: *IEEE Transactions on Biomedical Engineering* 53.6 (2006), págs. 1124-1133. DOI: 10.1109/TBME.2005.863952.
- [6] Harini D. Kannan. «Eye tracking for the iPhone using deep learning». eng. Accepted: 2018-01-12T20:59:18Z. Thesis. Massachusetts Institute of Technology, 2017. URL: <https://dspace.mit.edu/handle/1721.1/113142> (visitado 27-03-2022).
- [7] K. Krafka y col. «Eye Tracking for Everyone». En: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Los Alamitos, CA, USA: IEEE Computer Society, jun. de 2016, págs. 2176-2184. DOI: 10.1109/CVPR.2016.239. URL: <https://doi.ieeecomputersociety.org/10.1109/CVPR.2016.239>.
- [8] Alex Krizhevsky, Ilya Sutskever y Geoffrey E Hinton. «ImageNet Classification with Deep Convolutional Neural Networks». En: *Advances in Neural Information Processing Systems*. Vol. 25. Curran Associates, Inc., 2012. URL: <https://papers.nips.cc/paper/2012/hash/c399862d3b9d6b76c8436e924a68c45b-Abstract.html> (visitado 15-04-2022).
- [9] Raphael Menges y col. «GazeTheWeb: A Gaze-Controlled Web Browser». En: *Proceedings of the 14th International Web for All Conference*. W4A '17. Perth, Western Australia, Australia: Association for Computing Machinery, 2017. ISBN: 9781450349000. DOI: 10.1145/3058555.3058582. URL: <https://doi.org/10.1145/3058555.3058582>.
- [10] Benjamin Outram y col. «Anyorbit: orbital navigation in virtual environments with eye-tracking». En: jun. de 2018, págs. 1-5. DOI: 10.1145/3204493.3209579.
- [11] Xucong Zhang y col. «Appearance-based gaze estimation in the wild». en. En: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Boston, MA, USA: IEEE, jun. de 2015, págs. 4511-4520. ISBN: 978-1-4673-6964-0. DOI: 10.1109/CVPR.2015.7299081. URL: <http://ieeexplore.ieee.org/document/7299081/> (visitado 06-04-2022).
- [12] Xucong Zhang y col. «It's Written All Over Your Face: Full-Face Appearance-Based Gaze Estimation». En: *arXiv:1611.08860 [cs]* (mayo de 2017). arXiv: 1611.08860. URL: <http://arxiv.org/abs/1611.08860> (visitado 31-03-2022).