Gaze-Net: Appearance-Based Gaze Estimation using Capsule Networks

Bhanuka Mahanama
Department of Computer Science
Old Dominion University
Norfolk, VA 23529
bhanuka@cs.odu.edu

Yasith Jayawardana
Department of Computer Science
Old Dominion University
Norfolk, VA 23529
yasith@cs.odu.edu

Sampath Jayarathna
Department of Computer Science
Old Dominion University
Norfolk, VA 23529
sampath@cs.odu.edu

ABSTRACT

Recent studies on appearance based gaze estimation indicate the ability of Neural Networks to decode gaze information from facial images encompassing pose information. In this paper, we propose Gaze-Net: A capsule network capable of decoding, representing, and estimating gaze information from ocular region images. We evaluate our proposed system using two publicly available datasets, MPIIGaze (200,000+ images in the wild) and Columbia Gaze (5000+ images of users with 21 gaze directions observed at 5 camera angles/positions). Our model achieves a Mean Absolute Error (MAE) of 2.84° for Combined angle error estimate within dataset for MPIIGaze dataset. Further, model achieves a MAE of 10.04° for across dataset gaze estimation error for Columbia gaze dataset. Through transfer learning, the error is reduced to 5.9°. The results show this approach is promising with implications towards using commodity webcams to develop low-cost multi-user gaze tracking systems.

CCS CONCEPTS

• **Human-centered computing** → *Interaction techniques*; • **Theory of computation** → **Models of learning**.

KEYWORDS

Gaze estimation, Gaze Tracking, Capsule Networks, Deep Learning, Transfer Learning

ACM Reference Format:
Bhanuka Mahanama, Yasith Jayawardana, and Sampath Jayarathna. 2020. Gaze-Net: Appearance-Based Gaze Estimation using Capsule Networks. In *11th Augmented Human International Conference (AH '20), May 27–29, 2020, Winnipeg, MB, Canada.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3396339.3396393

1 INTRODUCTION

Human gaze estimation has wide range of applications from human computer interaction [11, 13, 14, 18] to behavioural [1, 10], and physiological studies [2, 9]. There has also been a growing interest towards identifying the direction of gaze. Recent studies [8, 21] using convolutional neural networks (CNN) for gaze estimation have shown promising results by learning features from both ocular and facial regions. However, the extraction of facial images or the entire ocular region can be challenging in naturalistic environments, where occlusions such as hair or objects obstruct the view [6].

Intuitively, the direction of gaze is linked with the pose of the eyeballs, i.e. the location of the pupil with respect to the ocular region. Thus, an image patch of a single ocular region (i.e. a single eye) should encompass important information such as the eye type (left or right), yaw, and pitch to represent its orientation in space. Hence, a model that could learn such information from an ocular image should be able to reliably estimate the direction of the gaze.

CNNs work extremely well in detecting the presence of objects, but are intolerant to feature translations unless accompanied with pooling layers. However, pooling introduces translation-invariance as opposed to translation-equivariance, which makes it challenging for CNNs to preserve pose and orientation information of objects across convolutional layers. A possible solution is to replicate feature detectors for each orientation, or to increase the size of the training data set to accommodate varying poses of features. However, this approach becomes challenging in terms of model complexity, data acquisition and generalization.

On the other hand, capsule networks [16] present an exciting avenue with capabilities towards learning equivariant representations of objects. Capsules [4] converts pixel intensities to instantiation parameters of features, which aggregates into higher level features as the depth of the network grows. In this study, we propose Gaze-Net, a pose-aware neural architecture to estimate the gaze based on the concept of capsules [4], and dynamic routing [16].

Given a capsule network's ability to learn equivariant representations of objects, we expect it to learn the orientation of eyes and reliably estimate the gaze direction. We utilize image patches of individual eyes instead of multi-region data to train our network. For training and evaluation, we use two publicly available datasets, *MPIIGaze* [20] and *Columbia Gaze* [19] datasets. We present a network of encoding of orientation information corresponding to the ocular region, and use transfer learning to apply our pre-trained network for different tasks, and evaluate implications in terms of performance.

2 RELATED WORK

Gaze estimation methods can be classified as either model-based or appearance-based. Model-based methods estimate gaze using a geometric model of the eye [7, 13], or face [3, 5]. Appearance-based methods directly use image patches of the eye [15, 18, 20] or face [8, 21] for estimation.

AH '20, May 27–29, 2020, Winnipeg, MB, Canada
© 2020 Association for Computing Machinery.
ACM ISBN 978-1-4503-7728-7/20/05…$15.00
https://doi.org/10.1145/3396339.3396393

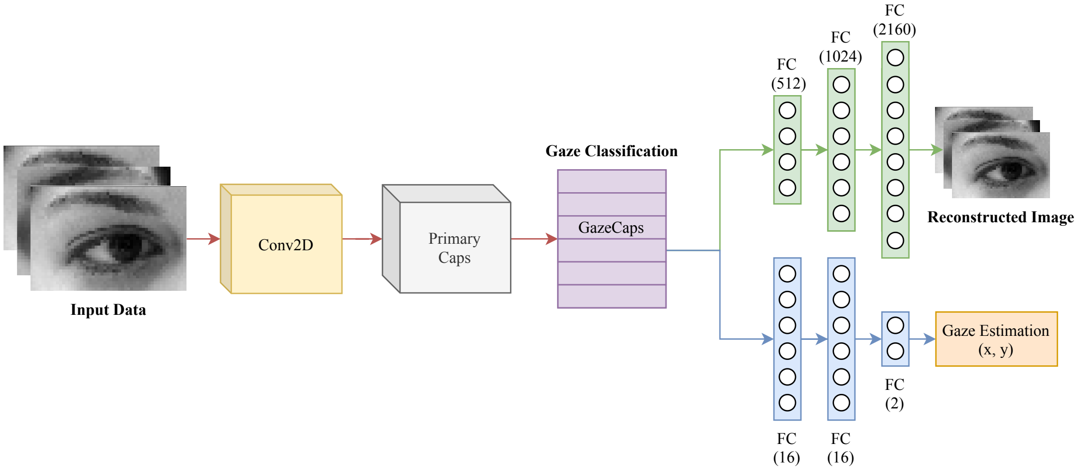

Figure 1: Gaze-Net Architecture for Gaze Estimation

Appearance-based methods can be modeled either through user-specific examples or through data-driven approaches. Due to practical limitations in collecting large amounts of user-specific examples, data driven approaches are preferred for appearance-based methods [20]. A key limitation of data-driven approaches is that the estimation models are generalized across all examples they were trained with. This may not be preferable when building personalized gaze estimation models. For such cases, in addition to gaze examples [7], user interaction events [5, 15] have been used. However studies have not been conducted on adapting generalized models trained through data driven approaches for personalized gaze estimation.

In the early work of gaze estimation, a fixed head pose was assumed for the sake of simplicity [17]. In more recent work, the orientation of the head was provided either explicitly [20] or implicitly through facial images [8, 21], which has led to improved gaze estimations. However, extracting that information from the ocular region itself has not been explored.

3 METHODOLOGY

Capsule networks are tailored for classification tasks, however, estimation of gaze is a regression task. We follow a two-step approach to build and train our network for gaze estimation.

First, we train a portion of our network to classify the gaze direction for image patches of individual eyes using 6 class labels: upper-left, upper-center, upper-right, lower-left, lower-center, and lower-right. Here, image patches of shape (36×60×1) are first passed through a (9×9) convolution layer (Conv2D) of 256 filters and a stride of 1. Its output is then passed into the primary capsule layer (PrimaryCaps) to perform a (9×9) convolution on 32 filters using a stride of 2. Its output is normalized using the *squash* function [16],

$$squash(s_j) = \frac{||s_j||^2}{1 + ||s_j||^2} \frac{s_j}{||s_j||} \tag{1}$$

Here, the *squash* function accepts a vector input s_j, and returns a normalized vector having the same direction, but the magnitude squeezed between 0 and 1. This output is passed into the GazeCaps layer, which has 6 capsules corresponding to each class label. It performs dynamic routing using 3 iterations to generate a 16-dimensional activity vector v_i from each capsule. The length of each vector represents the probability of the gaze being directed in a specific region, and the parameters of the vector represents ocular features that correspond to that direction. The class label of the gaze capsule having the highest activity $||v_i||$ is interpreted as the output. We use *margin loss* L_k for each gaze capsule k,

$$L_k = T_k max(0, m^+ - ||v_k||)^2 + \lambda(1 - T_k) max(0, ||v_k|| - m^-)^2 \tag{2}$$

Here, $T_k = 1$ iff the categorization is k, $m^+ = 0.9$, and $m^- = 0.1$.

Next, the first branch consists of three fully-connected layers of 512, 1024, and 2160 respectively. It accepts the (6×16) output from the GazeCaps layer, and provides a 2160-dimensional vector as output. This output is reshaped into (36×60×1) to calculate the pixel-wise loss of reconstructing the original input [16], i.e. *reconstruction loss (RL)*. The second branch consists of three fully-connected layers having sizes of 16, 16, and 2, respectively. It accepts the (6×16) output from the GazeCaps layer, and outputs the (x, y) gaze directions of the input image. We calculate the mean-squared error of gaze direction, i.e. *gaze loss (GL)*. Since the first portion of the network learns to encode the orientation and relative intensity of features, the combined network learns to transform these into gaze estimates, and to reconstruct the original image (see Figure 3).

3.1 Training

We define our objective function L as a combination of margin loss, reconstruction loss, and gaze loss,

$$L = \Sigma_k L_k + \lambda_1 RL + \lambda_2 GL \tag{3}$$

where L_k is the margin loss of the k^{th} capsule, RL is the reconstruction loss, GL is the gaze loss, and λ_1, λ_2 are regularization parameters. During training, we use reconstruction loss and gaze loss in isolation by tweaking λ_1 and λ_2, and evaluate its impact on the model performance.

4 RESULTS

4.1 Gaze Estimation

We traine Gaze-Net using multiple data sets, and evaluate its performance through classification accuracy (for gaze categorization) and mean absolute error (for gaze estimation). We use two publicly available datasets, the *MPIIGaze* [21] dataset for experimentation, and the *Columbia Gaze* dataset [19] for transfer learning.

For the MPIIGaze dataset, we use a 75-25 split to create separate training and test sets. We kept aside 10% of the training data as the validation set, and trained the model for 100 epochs using the remaining 90% of training data. After each epoch, the validation set was used to measure the performance of the model. We consider both left-eye and right-eye images to be the same, to test our hypothesis of a single eye image having sufficient information to reliably estimate the gaze. We train Gaze-Net using different regularization parameters for reconstruction loss and gaze loss (see Table 1, and Figure 2).

Table 1: Classification Accuracy (ACC) and Mean Absolute Error (MAE) of Gaze Estimation for each Regularization method.

Regularization Method	ACC (%)	MAE
No Regularization ($\lambda_1 = 0, \lambda_2 = 0$)	**67.15**	-
Image Reconstruction ($\lambda_1 = 0.005, \lambda_2 = 0$)	65.97	-
Gaze Error ($\lambda_1 = 0, \lambda_2 = 0.005$)	63.98	2.88
Image Reconstruction + Gaze Error ($\lambda_1 = 0.005, \lambda_2 = 0.005$)	62.67	**2.84**

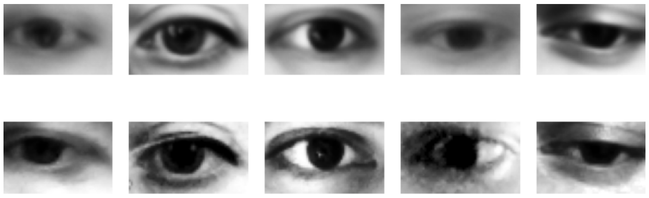

Figure 2: Comparison of MPIIGaze image reconstruction with the original images. The top row shows the reconstructed images, and the bottom row shows the original images.

4.2 Transfer Learning

We evaluated personalized gaze estimation from the Gaze-Net weights using the Columbia Gaze [19] dataset.We used PoseNet [12] to obtain the (x, y) coordinates of ocular regions in them. We extracted a $(36 \times 60 \times 1)$ image patch around each coordinate to generate data for evaluating Gaze-Net. When PoseNet predicted multiple (x, y) coordinates, we only selected the most confident ($\geq 80\%$) predictions.

Next, we evaluated Gaze-Net using the extracted image patches with 75-25 split for each participant to create 39 personalized training and test sets. Next, we re-trained Gaze-Net for each training set *while freezing all weights up to the GazeCaps layer*, such that only the gaze estimation weights (i.e. last fully connected layers) get updated. This resulted in 39 personalized Gaze-Net models, which we evaluated using the corresponding test sets to obtain 39 MAE values (see Table 2).

Table 2: Mean Absolute Error (MAE) of gaze estimation before and after training on Columbia Gaze Dataset.

Model	MAE
Transfer Learning	10.04 ± 0.470
Transfer Learning + Retraining Gaze Estimator	5.92 ± 0.457

5 DISCUSSION

An important observation from this evaluation is the lower mean absolute error (MAE) despite of the low classification accuracy (ACC). One possible reason for this observation is the *crisp* categorization that was used to map the gaze directions into 6 classes. This forces the suppression of the activity vectors not associated with the class label. Instead, we could adapt a probabilistic mapping which takes angular distance into account, to provide more relevant accuracy estimates. Alternatively, we could adapt a similarity metric with a minor modification to T_k in the margin loss function (see Eq. 2), such that the adjacency of categorical values are taken into account.

$$T_k = \bar{v}_k \cdot \bar{v}_l \tag{4}$$

Here, $\bar{v}_k$ corresponds to k^{th} capsule and $\bar{v}_l$ is a directional encoding for the class label l. It produces $T_k = 1$ if the activity vector is from the right capsule, conforming the original implementation of the capsule network. For a gaze categorization problem, we can define $\bar{v}_l$ as the centroid of the region that belongs to class l.

6 CONCLUSION

The transfer learning approach presented in this paper is capable of providing personalized gaze estimates by leveraging the generalized pre-trained eye-gaze model with capsule layers. In real-world applications, gaze estimation software can be shipped with a generalized model, which could be personalized through calibration. Since the generalized network is pre-trained to encode ocular information accurately, a personalized network could learn to estimate gaze by integrating with the generalized network and training through an interaction driven approach, such as mouse clicks. Overall, Gaze-Net combines components trained via both data-driven and interaction-driven approaches, which enables to realize the benefits of both methodologies.

This approach shows promise for conducting behavioral studies in the wild. For instance, gaze estimation can be applied in a classroom setting to monitor how students interact with the environment, and by doing so, gain insights into their attentiveness, learning preferences, cognition, and underlying medical conditions. Moreover, in the context of human computer interaction, this approach shows promise in facilitating zero-calibration, multi-user interactions such as AR/VR gaming and gesture-based device operation.

REFERENCES

[1] Stylianos Asteriadis, Paraskevi Tzouveli, Kostas Karpouzis, and Stefanos Kollias. 2009. Estimation of behavioral user state based on eye gaze and head

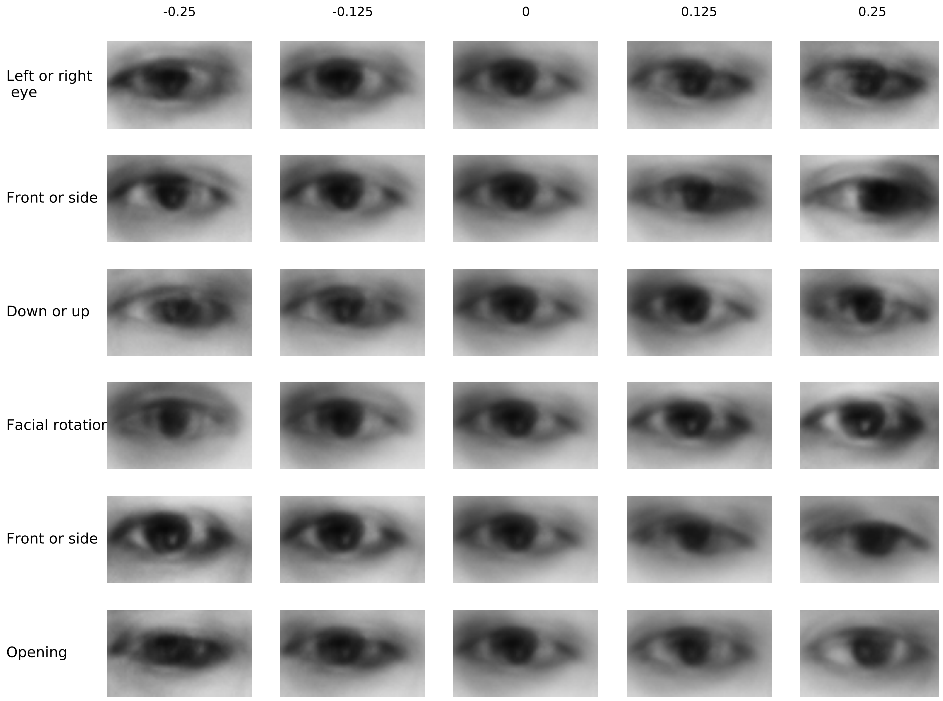

Figure 3: Dimension perturbations. Each row shows the reconstruction when one of the 16 dimensions in the GazeCaps output is tweaked by intervals of 0.125 in the range $[-0.25, 0.25]$.

pose—application in an e-learning environment. *Multimedia Tools and Applications* 41, 3 (2009), 469–493.

[2] Senuri De Silva, Sanuwani Dayarathna, Gangani Ariyarathne, Dulani Meedeniya, Sampath Jayarathna, Anne MP Michalek, and Gavindya Jayawardena. 2019. A Rule-Based System for ADHD Identification using Eye Movement Data. In *2019 Moratuwa Engineering Research Conference (MERCon)*. IEEE, 538–543.

[3] Lex Fridman, Philipp Langhans, Joonbum Lee, and Bryan Reimer. 2016. Driver gaze region estimation without use of eye movement. *IEEE Intelligent Systems* 31, 3 (2016), 49–56.

[4] Geoffrey E Hinton, Alex Krizhevsky, and Sida D Wang. 2011. Transforming auto-encoders. In *International conference on artificial neural networks*. Springer, 44–51.

[5] Michael Xuelin Huang, Tiffany CK Kwok, Grace Ngai, Hong Va Leong, and Stephen CF Chan. 2014. Building a self-learning eye gaze model from user interaction data. In *Proceedings of the 22nd ACM international conference on Multimedia*. Association for Computing Machinery, 1017–1020.

[6] Qiong Huang, Ashok Veeraraghavan, and Ashutosh Sabharwal. 2017. TabletGaze: dataset and analysis for unconstrained appearance-based gaze estimation in mobile tablets. *Machine Vision and Applications* 28, 5-6 (2017), 445–461.

[7] Moritz Kassner, William Patera, and Andreas Bulling. 2014. Pupil: an open source platform for pervasive eye tracking and mobile gaze-based interaction. In *Proceedings of the 2014 ACM international joint conference on pervasive and ubiquitous computing: Adjunct publication*. ACM, 1151–1160.

[8] Kyle Krafka, Aditya Khosla, Petr Kellnhofer, Harini Kannan, Suchendra Bhandarkar, Wojciech Matusik, and Antonio Torralba. 2016. Eye tracking for everyone. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. IEEE, 2176–2184.

[9] Anne MP Michalek, Gavindya Jayawardena, and Sampath Jayarathna. 2019. Predicting adhd using eye gaze metrics indexing working memory capacity. In *Computational Models for Biomedical Reasoning and Problem Solving*. IGI Global, 66–88.

[10] Philipp Müller, Michael Xuelin Huang, Xucong Zhang, and Andreas Bulling. 2018. Robust eye contact detection in natural multi-person interactions using gaze and speaking behaviour. In *Proceedings of the 2018 ACM Symposium on Eye Tracking Research & Applications*. ACM, 1–10.

[11] Bilge Mutlu, Toshiyuki Shiwa, Takayuki Kanda, Hiroshi Ishiguro, and Norihiro Hagita. 2009. Footing in human-robot conversations: how robots might shape participant roles using gaze cues. In *Proceedings of the 4th ACM/IEEE international conference on Human robot interaction*. 61–68.

[12] D Oved, I Alvarado, and A Gallo. 2018. Real-time human pose estimation in the browser with TensorFlow.js. *Retrieved from TensorFlow Blog: https://blog.tensorflow. org/2018/05/real-time-humanpose-estimation-in. html* (2018). https://blog.tensorflow.org/2018/05/real-time-humanpose-estimation-in.html

[13] Oskar Palinko, Francesco Rea, Giulio Sandini, and Alessandra Sciutti. 2016. Robot reading human gaze: Why eye tracking is better than head tracking for human-robot collaboration. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 5048–5054.

[14] Alexandra Papoutsaki, James Laskey, and Jeff Huang. 2017. Searchgazer: Webcam eye tracking for remote studies of web search. In *Proceedings of the 2017 Conference on Conference Human Information Interaction and Retrieval*. 17–26.

[15] Alexandra Papoutsaki, Patsorn Sangkloy, James Laskey, Nediyana Daskalova, Jeff Huang, and James Hays. 2016. Webgazer: Scalable webcam eye tracking using user interactions. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence-IJCAI 2016*.

[16] Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. 2017. Dynamic routing between capsules. In *Advances in neural information processing systems*. 3856–3866.

[17] Weston Sewell and Oleg Komogortsev. 2010. Real-time eye gaze tracking with an unmodified commodity webcam employing a neural network. In *CHI'10 Extended Abstracts on Human Factors in Computing Systems*. ACM, 3739–3744.

[18] Brian A Smith, Qi Yin, Steven K Feiner, and Shree K Nayar. 2013. Gaze locking: passive eye contact detection for human-object interaction. In *Proceedings of the 26th annual ACM symposium on User interface software and technology*. ACM, 271–280.

[19] Brian A Smith, Qi Yin, Steven K Feiner, and Shree K Nayar. 2013. Gaze locking: passive eye contact detection for human-object interaction. In *Proceedings of the 26th annual ACM symposium on User interface software and technology*. ACM, 271–280.

[20] Xucong Zhang, Yusuke Sugano, Mario Fritz, and Andreas Bulling. 2015. Appearance-based gaze estimation in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. IEEE, 4511–4520.

[21] Xucong Zhang, Yusuke Sugano, Mario Fritz, and Andreas Bulling. 2017. It's written all over your face: Full-face appearance-based gaze estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. IEEE, 51–60.