Light-weight Head Pose Invariant Gaze Tracking

Rajeev Ranjan
University of Maryland
rranjan1@umiacs.umd.edu

Shalini De Mello
NVIDIA
shalinig@nvidia.com

Jan Kautz
NVIDIA
jkautz@nvidia.com

Abstract

Unconstrained remote gaze tracking using off-the-shelf cameras is a challenging problem. Recently, promising algorithms for appearance-based gaze estimation using convolutional neural networks (CNN) have been proposed. Improving their robustness to various confounding factors including variable head pose, subject identity, illumination and image quality remain open problems. In this work, we study the effect of variable head pose on machine learning regressors trained to estimate gaze direction. We propose a novel branched CNN architecture that improves the robustness of gaze classifiers to variable head pose, without increasing computational cost. We also present various procedures to effectively train our gaze network including transfer learning from the more closely related task of object viewpoint estimation and from a large high-fidelity synthetic gaze dataset, which enable our ten times faster gaze network to achieve competitive accuracy to its current state-of-the-art direct competitor.

1. Introduction

Gaze tracking has numerous applications in the design of industrial and web interfaces: for enhancing automotive safety by monitoring the visual behavior of drivers, for increased accessibility via gaze-controlled interfaces, for correcting gaze direction during video conferencing, and for rendering foveated displays in the bandwidth-constrained environments of augmented and virtual reality headsets.

Unconstrained gaze tracking refers to calibration-free, subject-, viewpoint-, and illumination-independent gaze tracking using a remotely placed off-the-shelf camera. It allows for free head motion of the subject. Recently, various promising appearance-based techniques for unconstrained remote gaze estimation have been proposed [19, 25, 7, 8, 26, 27]. Among them, techniques based on convolutional neural networks (CNN) are the most successful [8, 27]. Researchers have also proposed computer graphics models to generate synthetic training data for gaze estimation [23, 22], along with techniques to adapt them to real-

world data [23, 15]. Nevertheless, many open problems remain to be solved in order to increase the accuracy of gaze tracking including robustness to variability in head pose, illumination, subject identity, image quality and distribution of head pose and gaze angles [27]. Furthermore, many applications of gaze tracking require operation in mobile environments on low-cost processors. Applications like foveated rendering require extremely fast gaze estimation (several hundred hertz) [13]. Hence, achieving realtime performance, without lowering quality is a practical concern for the deployment of gaze estimation technology in the real-world. To improve computational efficiency, Krafkra *et al.* [8] use dark knowledge to reduce the size of their gaze CNN, but incur some loss of accuracy.

In this work we address several challenges of unconstrained gaze tracking. First, we systematically study the effect of variable head pose on machine learning-based gaze regressors. We show that one effect of variable head pose is the change in the prior distribution of gaze angles that the regressor needs to predict. This motivates the need for higher specificity in the design of gaze networks that process eye images captured from widely varying viewpoints. To address this, first, we propose a novel and efficient head-pose-dependent branched CNN architecture to improve robustness to variable head pose, which does not increase the computational cost of inference. Second, for realtime high-quality gaze estimation, we present various effective training procedures for gaze CNNs, which help our ten times faster CNN to achieve the same accuracy as the current state-of-the-art more complex algorithm [27]. Our contributions to the training procedure include (a) transfer-learning from the closely related task of general object view-point estimation [18] instead of from object classification – the latter is the de-facto approach for gaze estimation [8, 27]; and (b) from on a CNN trained on 1M synthetic photo-realistic images generated from the SynthesEyes model [23].

2. Related Work

Research on conventional gaze trackers is summarized in [4]. Recently, various appearance-based gaze tracking

algorithms have been proposed that train machine learning regression functions using adaptive linear models [11, 12], support vector machines [2], random forests (RF) [19, 7], and more recently CNNs [25, 23, 8, 26, 27, 3, 15]. Other approaches use fitting to generative face and/or eye region models [1, 21, 20]. Among appearance-based approaches, CNNs [26, 27, 3] are the most successful.

Inputs to gaze CNNs, investigated previously, include eye images and head orientation information [25, 23, 27], or full face images along with their location information [8, 26, 3]. Full-face-based CNNs are more accurate for gaze estimation than eye image-based CNNs [8, 26], but are limited in their applicability as the entire face may not be visible to the camera in all situations [7]. Our algorithm uses information from the eye region only along with head orientation, and thus is more widely applicable.

Most existing algorithms regress 3D gaze angles, but a few directly regress the point-of-regard in 2D screen co-ordinates [8, 26]. As Zhang *et al.* [26] also note, computing 3D gaze directions is a more general solution, which, along with the 3D positions of the eye centers, can be used to compute any 3D point-of-regard. Hence, we too compute the more general 3D gaze angles.

Many open challenges in gaze estimation including robustness to variable head pose, image resolution and quality, individual eye-shape, illumination and the presence of eye glasses are noted in [27]. To deal with head pose, the early work [11] limits head pose to close to frontal. Others generate frontal images of the face by capturing depth information in addition to color and warping [12, 1]. Sugano *et al.* [19] create overlapping clusters of head pose and train one RF regressor per cluster. At inference time, for each input, the outputs of multiple neighboring pose regressors are computed and averaged together, which improves accuracy, but because it involves inferencing via multiple regressors it also increases the computational cost. Zhang *et al.* apply Sugano *et al.*'s [19] approach of head pose clustering to two different gaze CNN architectures [25, 27] and observe a reduction in accuracy with it. We also use the idea of clustering head pose for gaze estimation. However, unlike the previous approaches, we employ late branching of one CNN, where only the few final fully-connected layers are specific to individual pose clusters and the rest of the layers are shared. We show that when large head poses are present this architecture results in improved accuracy without incurring any additional computational cost for inference.

Using progressively more complex CNNs architectures (LeNet [10], AlexNet [9] and VGG [16]) and their weights trained on ImageNet as the bases for gaze networks has helped to improve the accuracy of gaze estimation [25, 26, 27]. However, this comes at the expense of increased computational cost, which is a real challenge for deploying gaze CNNs on embedded devices. Krafka *et al.* [8] explore re-ducing the size of their CNN using dark knowledge [6], but it results in reduction in accuracy of their model. Orthogonal to these works, we propose effective training procedures including transfer learning from the closely related task of object view-point estimation and from a large photo-realistic synthetic dataset to train a much (10×) faster and smaller CNN than Zhang *et al.*'s [27] state-of-the-art network and still achieve competitive accuracy to it.

Lastly, there exist multiple synthetic models to generate synthetic training images for gaze CNNs: (a) UT Multi-view [19] with limited discrete gaze angles, (b) SynthesEyes [23] with 10 real head scans and highly photo-realistic rendering, and (c) UnityEyes [22] with realtime rendering of less photo-realistic images, but with a morphable model to generate infinite shapes and textures for the head region surrounding the eyeball. Of these, large datasets (of a few millions images) generated with the UnityEyes model are used most widely [22, 15]. A few works use a smaller dataset of 12K photo-realistic images generated from the SynthesEyes model [23, 22, 27] and found it to be inferior to the 1M UnityEyes dataset in improving gaze estimation accuracy. Unlike these previous works, we generate a large dataset of 1M photo-realistic images using the SynthesEyes model and show its effectiveness in improving the accuracy of gaze estimation.

3. Method

3.1. Head Pose Dependence

The goal of unconstrained gaze tracking is to estimate the angle between the visual axis of the eye and that of a remotely placed stationary camera that observes the subject, all while allowing the subject to move their head freely. Head pose is not strictly required to compute gaze direction and it can be computed purely based on the shape of the pupil and iris in the camera's image [4]. However, for machine learning regressors trained to regress gaze direction from eye images, changing head pose introduces a number of sources of variability into the problem.

First, for the same gaze angles, head rotation causes significant change in the appearance of the observed eye images. For example, Figure 1 shows various eye images with nearly identical gaze angles close to zero, but with large variation in head pose. In all these images, while the shape of the pupil and iris region is similar (almost circular because of the zero gaze angle), differences in head pose cause (a) the pupil and iris region to be occluded differently, and (b) significant differences in the appearance of the sclera and skin regions, surrounding the pupil. Nevertheless, when presented with sufficient and variable training examples a CNN model with adequate capacity may learn to disregard these appearance changes as non-contributing factors to determining gaze direction.
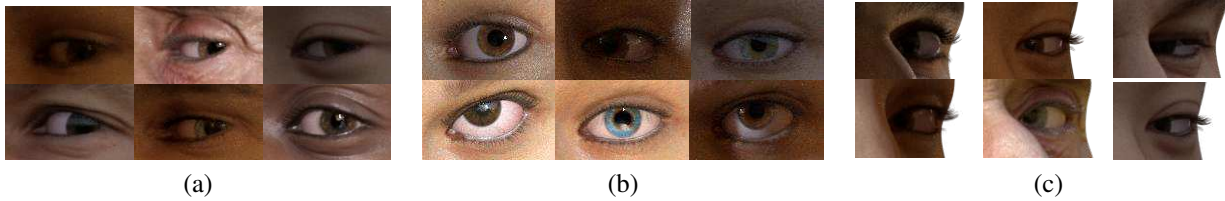
(a)   (b)   (c)

Figure 1: Example eye images from our high-quality synthetic gaze dataset with (a) large positive head pose yaw angles, (b) near-zero head poses, and (c) large negative head pose yaw angles. Note how eye images with different head poses have significantly different appearances.

The second effect of variable head pose, which is more subtle to understand and harder to solve is that with variable head pose, the prior distribution of gaze angles that a machine learning regressor needs to predict changes. To understand this effect, we create a large synthetic dataset in-house by rendering 1M eye images using the SynthesEyes [23] computer graphics model. We render eye images with uniformly distributed $[-60, 60]°$ head pitch (up/down) and yaw (left/right) angles, and uniformly distributed eye-in-head rotation pitch and yaw angles of $[-25, 25]°$ and $[-35, 35]°$, respectively, distributed about each head pose. We cluster head poses into various clusters (using cosine distance) shown in Figure 2(a). The distributions of gaze pitch and yaw angles for different head pose clusters are shown in Figure 2(b). Observe the distinctly different distributions of gaze angles for the various head pose clusters.

The reasons for this phenomenon are two-fold. First, as the head rotates, the eyeballs rotate in synchrony with it and gaze angles for a particular head orientation tend to be distributed roughly centrally about the instantaneous head pose. Hence, the mean of the distribution of gaze angles is a function of the observed head pose. Second, head rotation causes the shape of the prior distribution of the gaze angles to change (see Figure 2(b)). To understand why this happens consider a stationary camera that observes a frontal face, (*e.g.*, the cluster number 7 in Figure 2(a)). For this head pose, for the entire range of motion of the eyeball, the eye, pupil and iris regions are visible to the camera and

hence all gaze angles corresponding to these eyeball rotations can be regressed by the CNN. As the head rotates, either the eye or the pupil and iris regions, for certain eyeball rotations, become invisible to the camera. For example, for a head turned to the left, the left eye may not be visible to the camera. Additionally, even for the visible right eye, its pupil and iris may be visible in the camera image only for rotations of the eyeball to the right relative to the head. Hence for oblique head poses, eye-image-based CNNs are presented with eye images with limited gaze angles and hence have smaller distributions of gaze angles that they need to regress (see clusters 1-6 in Figure 2(b)).

We observe a similar trend for the real-world MPII Gaze dataset [25] as well on clustering it into 5 head pose clusters in Figure 3. Note that this dataset was collected while subjects viewed their laptop screens and hence contains only a narrow range of head poses angles. This explains the noticeable, but less prominent differences in the distributions of gaze angles for the various head pose clusters of this dataset (Figure 3(b)) versus the synthetic dataset (Figure 2(b)).

These observations about the effect of head pose on the problem of unconstrained appearance-based gaze estimation via machine learning methods, motivate the need to incorporate greater specificity in their design for different head poses. In the following section, we describe one such novel CNN architecture that we propose to tackle this problem.
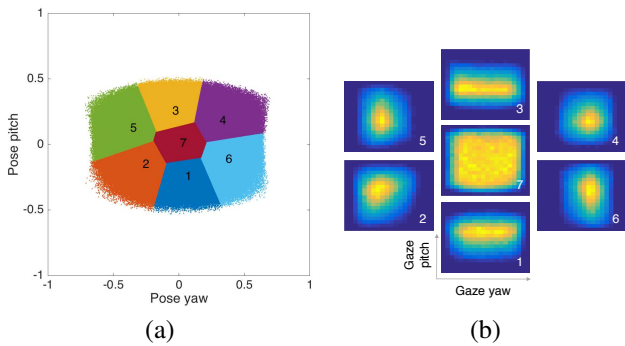

(a)   (b)

Figure 2: This figure contains (a) 7 head pose clusters for our synthetic dataset, and (b) the 2D distributions of the gaze angles for these clusters.
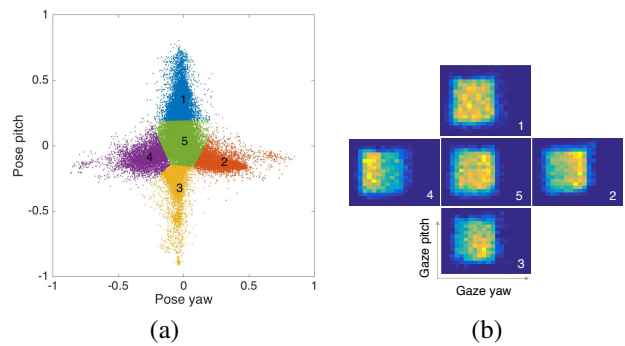

(a)   (b)

Figure 3: This figure contains (a) 5 head pose clusters for the MPIIGaze dataset, and (b) the 2D distributions of the gaze angles for these clusters.

3.2. Branched Gaze Network

A naive approach to tackling this problem of variable head pose is to train multiple gaze networks, each for a given head pose. However, this is infeasible and would result in a very large number of classifiers. An alternative solution is to train a network specific to a group of head poses with small variations. Hence, we cluster the head poses in our training data into K groups using k-means clustering and cosine distance. The value of K is chosen empirically via cross-validation.

One obvious solution is to train multiple gaze networks, one for each head pose cluster. Zhang *et al.* applied this approach in [25, 27] and observed a decrease in accuracy with multiple gaze networks. Using different CNNs for each head pose cluster has the drawback of data scarcity during training. For K clusters, the number of training samples for each network decreases to roughly $1/K^{\text{th}}$ of the original size of the training dataset.

To overcome this issue, we instead use weight-sharing among the different gaze networks for head pose clusters. We use a modified version of the Alexnet [9] CNN architecture, as shown in Figure 4. It consists of 5 convolutional layers (conv1-conv5) followed by a max pooling layer (pool5) and a fully connected layer (fc6). We also incorporate residual skip connections for the classifier to have easy access to the lower level features of the network. We share the weights of the lower layers (conv1 to fc6) across all the head pose clusters, but add multiple fully connected (fc7 and fc8) layers on top of the fc6 layer, one pair for each head pose cluster. Similar to Zhang et al. [25], we concatenate head pitch and yaw angles with the fc7 layers of all head pose clusters. The final output layers (fc8) produce the pitch and yaw gaze angles (in radians). We use the rectified Linear Unit (ReLU) as the activation function after all convolution and fully connected layers.

The inputs to our network are an eye image, head pitch and yaw angles and head pose cluster ID. During training, only the training cases that belong to a particular cluster update the weights of their corresponding fc7-fc8 layers, but all training samples from all clusters update the conv1-fc6 layers, which reduces the problem of data scarcity. Additionally, during inference, based on the known cluster ID, each input is conditionally routed through only one pair of fc7-fc8 layers corresponding to its pose cluster. This ensures no increase in computational cost per input versus an un-branched network. Thus our proposed architecture is efficient in terms of training and testing.

3.3. Training

We describe a number of effective techniques that we employ to train our gaze classifier, which help to improve its accuracy significantly.
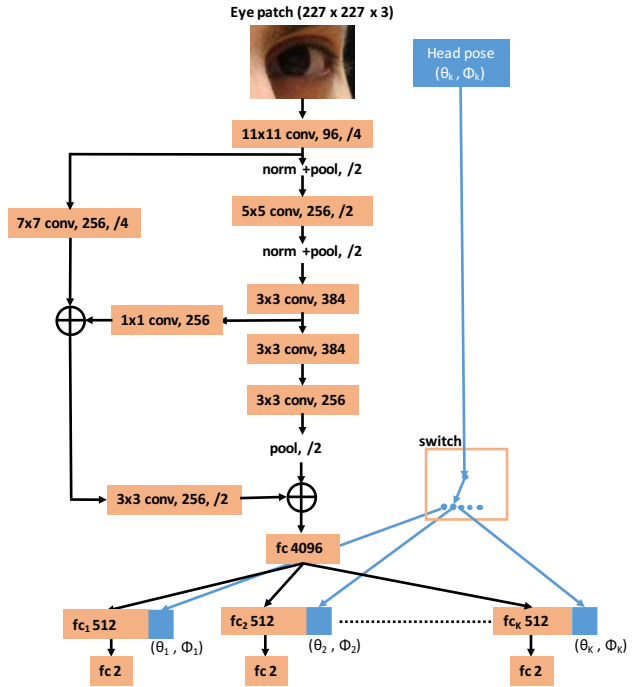

Figure 4: The proposed CNN architecture for estimating gaze angles from the image of an eye and head pose. The annotations in each box represent in order – the filter kernel sizes, the CNN layer type, the number of output feature maps, and the stride.

3.3.1 Object Viewpoint Estimation

Transfer learning from CNNs trained for the task of object recognition on the large ImageNet dataset [14] is a widely used approach, especially when the amount of training data available for the task at hand is small. This approach is also used exclusively for initializing all the successful gaze networks [25, 8, 26, 27, 3]. A careful visualization of the activations of a network trained on ImageNet [24] suggests that its earlier layers capture low level features that describe the location and orientation of an object, while the deeper layers capture the global characteristics of the objects and are invariant to their position and viewpoint. Their insensitivity to object pose makes networks trained for object recognition not ideal for initializing networks for gaze estimation. Gaze estimation fundamentally involves estimating the 3D orientation of the eyeball.

We instead transfer the learning of a network trained for general object viewpoint estimation, which is a more closely related task to gaze estimation than object recognition. We initialize the weights of our gaze network with the pre-trained weights of the "Render for CNN" [18] (RFC) network, which shares architectural elements with our gaze network. The RFC network was designed to estimate the 3D orientation of 20 categories of objects, including cars, airplanes, chairs, etc. Additionally, it was trained using a large

corpus (2.4M) of synthetically rendered images of different object categories. We show through multiple experiments that since RFC is designed to distinguish between subtle changes in the appearance of objects caused by their rotation, it is a better choice as a pre-trained network for gaze estimation.

3.3.2 High-quality Synthetic Data

We create a dataset of 1M synthetic eye images by rendering left eye images of size 134×80 pixels using advanced graphics techniques and ray tracing with the SynthesEyes model [23]. It contains high-quality 3D scans of 10 human head models (5 female and 5 male from different age groups and ethnicities) and detached eyeball models, which can be posed independently of the head models. Additionally, the subjects' eyelids can be posed in synchrony with the up/down movement of the eyeball. The images contain wide variations in illumination, which is achieved by image-based relighting with different HDR environment maps. Some example images from this synthetic dataset are shown in Figure 1. These images are of much higher quality in terms of photo-realism than the popular real-time rendered UnityEyes dataset [22]. Our work is the first to create and use such a large dataset of high-fidelity rendered eye images.

For all experiments with synthetic data, we transfer learn from the RFC network only. For the real-world datasets we perform an additional step of training on the 1M synthetic data first and then fine-tuning with the real data. Similar to previous work [23, 27], we too observe that to effectively transfer learn form synthetic data, it is important that both datasets have the same distribution of head pose and gaze angles. Hence we target the synthetic dataset to match the distribution of the real dataset.

4. Experiments

4.1. Synthetic Data

4.1.1 Network Initialization

We evaluate the performance of our gaze network with different weight initializations. For all experiments, we use 800K images of 8 subjects from the the 1M synthetic dataset for training and 200K images of two randomly selected subjects (female 2 and male 4) for testing. The inputs to all networks are normalized eye images and head pitch and yaw angles (radians), which are computed with the normalization procedure outlined in [25]. In this experiment, all networks are un-branched, meaning they contain one set of fc layers for all head pose clusters. We initialize our gaze network with (a) random weights, (b) weights from AlexNet [9] trained for object recognition on ImageNet, and

Table 1: The angular gaze errors for different network architectures (AlexNet and ResNet-50) initialized with different pre-trained networks.

Network	Weights Initialization	Error (°)
Ours	ImageNet [9]	5.03
Ours	RFC [18]	**4.40**
ResNet-50	ImageNet [5]	5.07

(c) with weights from the RFC [18] network trained for object viewpoint estimation. Additionally, we also implement a gaze network based on the more complex ResNet-50 network [5] with head pose concatenated with its pool5 layer and initialize it with weights from ImageNet.

The angular errors for gaze estimation with these networks are shown in Table 1. Our gaze network when initialized with random weights fails to converge, but all other networks converge. When initialized with weights from RFC, our network has the lowest gaze error of $4.4°$, which is considerably smaller than initializing from ImageNet. Interestingly, our AlexNet-style gaze network, despite being much less complex than the ResNet-50-based network, when initialized with RFC, also performs better than it. This shows the importance of proper initialization for training gaze networks and how transferring knowledge from the more closely related task of object viewpoint estimation versus object recognition can help it to converge to a better solution.

4.1.2 Effect of Head Pose

We investigate the effect of head pose on our gaze network. We evaluate the performance of our gaze network (a) with and without head pose as input and (b) with and without late branching of the network for different head pose clusters. We use the same experimental protocol (Experiment I) described in Section 4.1.1. In addition, we perform a leave-one-subject-out experiment (Experiment II) where we train with 900K images from 9 subjects and test on the 100K images of the one held-out subject. We repeat this experiment

Table 2: The effect of head pose on the accuracy of gaze estimation of our network. We compare the gaze errors (°) for networks (a) with and without head pose as an input and (b) with and without branching of the fully connected fc7-8 layers for the different head pose clusters.

Input	fc7-8	Experiment I	Experiment II
Eye	single	5.66	5.76
Eye	branched	5.18	4.75
Eye, pose	single	4.40	3.91
Eye, pose	branched	**4.15**	**3.68**

Table 3: The individual gaze errors (°) of each of the head pose clusters of the synthetic dataset shown in Figure 2. These errors are for our gaze network without head-pose-dependent branching of the fc7-8 layers (second row) and with branching (third row).

CNN Architecture	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5	Cluster 6	Cluster 7	Overall
single fc7-8	4.13	4.43	4.02	4.59	4.33	4.02	4.61	4.40
branched fc7-8	**4.09**	**4.19**	**3.92**	**4.08**	**4.31**	**3.92**	**4.61**	**4.15**

10 times for all subjects and average the gaze errors.

The results are shown in Table 2. We observe that providing head pose as input to our network, in addition to the eye image, has a significant positive impact on its accuracy. Additionally, branching the high-level fc7-8 layers for different head pose clusters results in improved accuracy of gaze estimation. The positive effect of branching is particularly pronounced when the head yaw and pitch angles are not provided as inputs to the gaze network. These observations are consistent for both Experiment I and Experiment II.

To better understand how our gaze network behaves with and without head pose-dependent branching, we compute the gaze errors (Table 2) for the 7 individual head pose clusters as shown in Figure 2 for Experiment I. Obsreve that with the branched architecture (row 3 of Table 3), the accuracy of all head pose clusters either remains the same or improves versus the un-branched architecture (row 2 of Table 3). Clusters 2 and 4, which contain oblique head poses, benefit the most from the branched network architecture, while the central cluster (number 7), which contains head pose angles close to zero remains unchanged. Plausible explanations for this observation are that since head pose is close to null and the gaze angles are less strongly correlated to the head pose for this cluster, head pose has minimal impact on it.

Overall, these results confirm the presence of correlations between oblique head poses and gaze angles, which, given sufficient data can be exploited to improve the accuracy of appearance-based gaze estimation via machine leaning regressors. Our branched gaze network design is one such proposed architecture that helps to improve the accuracy of gaze estimation by effectively using head pose information, while being efficient in terms of training and inference.

4.1.3 Number of subjects

On comparing the results of Experiment I and Experiment II in Table 2, one can also conclude that having greater subject variability in the training data helps to improve the accuracy of our gaze network. We perform an additional experiment where we train our network with 80% of the data from all 10 subjects in the synthetic dataset and test on the remaining 20%. This produces an an accuracy of 1.99° and

(a)   (b)

Figure 5: Examples of normalized eye images from (a) the Columbia gaze dataset, and (b) the MPIIGaze dataset's "Evaluation" set (top) and "Annotation" set (bottom). Note the erroneous image (top-center) from the MPIIGaze dataset.

represents the accuracy of our gaze network for the person-specific training scenario.

4.2. Real-world Data

We evaluate the performance of our gaze network and the efficacy of our training procedure, which involves transfer learning from RFC and the 1M synthetic dataset on the real-world benchmark Columbia [17] and MPIIGaze [25] datasets. For these datasets, we mirror the right eye images about the vertical axis and reverse the signs of their gaze and head yaw angles to process all images with one gaze network as was done in [25].

4.2.1 Columbia Dataset

The Columbia gaze dataset [17] contains 5,880 images of 56 subjects (22 with eyeglasses and 34 without), captured indoors in a laboratory from 5 viewpoints with head pose yaw angles ranging from $-30°$ and $30°$ in steps of $15°$. Additionally, for each viewpoint, images for 21 different gaze directions are present. We extract normalized color eye

Table 4: The gaze errors (°) of our gaze network and its comparison to the current state-of-the-art [21] on the Columbia dataset [17]. The second row contains errors for only 34 subjects without eyeglasses, while the third row contains the errors of all 56 subjects with/without eyeglasses.

Subjects	Wood [21]	Ours branched	
		RFC	1M Synthetic
No glasses	7.54	6.26	**5.58**
All	-	6.68	**5.65**

Table 5: A comparison of the performance of our proposed algorithm against the comparable state-of-the-art algorithm on the MPIIGaze dataset [25]. The table contains the average angular errors (°) across 15 repetitions of the leave-one-subject-out experiment on the MPIIGaze dataset.

	Zhang [25] 12K Synth [23]		Ours single			Ours Branched		
			ImageNet	RFC	1M Synth	ImageNet	RFC	1M Synth
MPIIGaze	5.5	Evaluation	5.88	5.56	**5.38**	5.77	5.49	5.48
MPIIGaze+	5.4	Annotation	5.88	5.38	**5.30**	5.63	5.46	5.42

images of size $134{\times}80$ pixels and compute the normalized head pose and gaze directions, using the procedure of [19]. We also apply histogram equalization to the Y channel of the YCbCr representation of the images and convert them back to RGB. Example normalized images are shown in Figure 5(a).

We divide the 56 subjects into three partitions and for each partition, we train with two-thirds of the subjects and test with the remaining one-third. We repeat this thee-way cross-validation-across-subjects procedure three times and average the gaze errors for all subjects. We cluster the Columbia data into 5 head pose clusters centered about the five discrete head yaw angles present in this dataset. We compute the performance of our gaze network for initialization with (a) RFC only and (b) with RFC followed by fine tuning with a targeted subset of the 1M synthetic dataset, sampled to match the gaze and head pose distribution of the Columbia dataset. Wood *et al*. [21] report the state-of-the-art error of $7.54°$ with their method of morphable eye-region matching on a randomly selected set of 680 images of 20 subjects without eyeglasses from this dataset. To compare against their results, we also repeat our three-fold cross validation experiment on the subset of 34 subjects without eyeglasses.

The results are shown in Table 4. Our gaze network produces the best known accuracy on this dataset with a gaze error of $5.58°$ for images without eyewear and $5.65°$ for all images. Thus there is only a marginal decrease in accuracy on including images with eyeware. Both versions of our gaze network, trained with and without the synthetic data, outperform Wood *et al*.'s algorithm. Furthermore, the additional step of pre-training with synthetic data and then fine tuning with the real-world data (column 3) helps to improve the accuracy of our gaze network by nearly $1°$.

4.2.2 MPIIGaze Dataset

The MPIIGaze dataset [25] contains images of 15 subjects (with/without eyeglasses) captured while they were viewing their laptop screens using the laptop's front-facing camera. The images contain widely varying illuminations. The dataset contains an "Evaluation" set of 45K normalized grayscale eye images (3K per subject) of size $60{\times}36$ pixels with their accompanying normalized head pose and gaze

angles (top row of Figure 5(b)). For normalizing the images in the "Evaluation" set, the authors use an automatic facial fiducial point detection algorithm, which is erroneous 5% of the time. This in turn produces erroneous normalized images that often do not even contain an eye (*e.g.*, top-center image of Figure 5(b)). In our analysis, we include these erroneous images as well and did not discard them. In addition to the "Evaluation" set, the authors also provide an "Annotation" Set of 10,848 images, which contain manually annotated facial fiducial points. We normalize the images in the "Annotation" set and evaluate the performance of our gaze network on it as well.

We follow the same leave-one-subject-out experimental protocol that was previously employed in [25, 27]. We train with images from 14 subjects, test on images of the left-out subject and repeat this procedure 15 times. We report the average errors across all repetitions. We evaluate the accuracy of both the un-branched and branched versions of our gaze network on the MPIIGaze dataset. For the branched network, we cluster the head poses into 5 clusters (Figure 3(a)) using k-means clustering and cosine distance. Furthermore, we evaluate initialization of the weights of our gaze network with the RFC network with and without further training on the 1M synthetic dataset.

Recently, Zhang *et al*. [27] report the state-of-the-art gaze error of $5.5°$ on the "Evaluation" set of the MPIIGaze dataset with erroneous facial fiducial point annotations included in it. They also report a gaze error of $5.4°$ for another version of the "Evaluation" subset called (MPIIGaze+), wherein, they manually annotate the facial fiducial points. Note that the MPIIGaze+ dataset is not publicly available. The backbone of their gaze network is the VGG network [16], which is $10\times$ slower than our gaze network. Additionally, similarly to us, they initialize their gaze network with weights from ImageNet, first train with a targeted subset of the 12K synthetic images generated by Wood *et al*. [23], and later fine-tune on the MPIIGaze dataset for cross-subject evaluation.

The results of these experiments are presented in Table 5. We observe that the un-branched version of our gaze network when initialized with weights from pre-trained RFC network and the synthetic data produces the lowest gaze errors for both the "Evaluation" and "Annotation" subsets of the MPIIGaze dataset of $5.4°$ and $5.3°$, respectively. These

Table 6: The cross-dataset angular errors (°) of various state-of-the-art gaze networks and ours on the "Evaluation" set of the MPIIGaze dataset.

	Dataset	Network	Initialization	Domain Adaptation	Error (°)
Shrivastava [15]	UnityEyes 1M	LetNet [25]	Random	GAN	7.80
Zhang [27]	UT Multiview	VGG	ImageNet	targeting	8.0
Zhang [27]	SynthesEyes 12K	VGG	ImageNet	targeting	**7.3**
Ours	SynthesEyes 1M	AlexNet	ImageNet	none	10.19
Ours	SynthesEyes 1M	AlexNet	RFC	none	8.72
Ours	SynthesEyes 1M	AlexNet	RFC	targeting	7.74

are the lowest reported gaze errors on this dataset. Furthermore, our gaze network outperforms the state-of-the-art network of Zhang *et al.* [27], both for automatically and manually annotated facial fiducial points, despite the fact our network is $10\times$ faster and significantly smaller than theirs. Proper initialization of the weights of our gaze network from the more closely related task of object viewpoint estimation performed by the RFC network contribute considerably towards improving its accuracy. We observe a further improvement by employing the synthetic data for pretraining the network.

Similar to what Zhang *et al.* [27] report, for the MPIIGaze dataset, we too observe a decrease in accuracy with head-pose-dependent branching. As Zhang *et al.* [27] also note, it is not surprising that the effect of head pose is minimal for this dataset as it mostly contains near-frontal head poses. However, the results of our previous simulation analyses with the larger, denser and much more variable 1M synthetic dataset confirm that in the presence of large head pose variations the branched CNN architecture is beneficial to improve the accuracy of gaze estimation.

4.2.3 Cross-dataset Experiment

Lastly, we evaluate the performance of our gaze network for the cross-dataset evaluation protocol. We train our gaze network (without branching) on the 1M synthetic dataset with and without targeting it to match the distribution of the MPIIGaze dataset. We test its performance on real-world images from the "Evaluation" set of the MPIIGaze dataset. We also evaluate the performance of various weight initializations: ImageNet and RFC.

Table 6 shows the gaze errors for the various configurations of our gaze network and for other gaze estimation algorithms. Shrivastava *et al.*'s [15] use 1M images from the low quality UnityEyes [22] dataset for training, but they adapt the appearance of the rendered images to match that of the real images using generative adversarial networks. For gaze estimation they use the original LeNet-style network proposed by Zhang *et al.* [25]. Zhang *et al.* [27], recently experiment with the UT MultiView [19] and the 12K SynthesEyes [23] datasets and subsample them to target the

distributions of the synthetic datasets to that of the real MPIIGaze dataset. They use the more complex VGG network for gaze estimation and achieve the lowest gaze errors of all others.

Our algorithm is competitive to the other competing algorithms in terms of gaze error. Furthermore, as we observe in the other experiments, here too initializing the weights with the RFC is the major contributing factor in improving the accuracy of our network. Targeting of the synthetic dataset also helps to improve its accuracy.

5. Conclusions

Unconstrained gaze estimation using off-the-shelf cameras that observe subjects from a remote location is a nontrivial problem as it requires handling many different head poses. We systematically study the affects of head pose on the problem of gaze estimation via machine learning regressors. We show that head pose changes the appearance of eye images along with the prior distribution of gaze angles that the machine learning regressor must predict. To address this, we cluster head poses into various clusters and design a branched gaze network, which contains shared lower-level layers for all head pose clusters, but multiple cluster-specific higher-level layers. This design is efficient in terms of training and inference. We show via a large simulation study containing 1M synthetic images that in the presence of large head poses, such a branched network is beneficial to improve the accuracy of gaze estimation.

Lastly, we propose new effective procedures for training gaze networks including transfer learning from the more closely related task of general object viewpoint estimation versus object recognition, and from a large synthetic dataset of high-quality rendered images. We show that these training procedures help our light-weight AlexNet-style gaze network achieve best-in-class accuracy on two real-world datasets, despite being significantly ($10\times$) faster and smaller than its directly competing state-of-the-art algorithm.

References

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

[2] M.-C. Chuang, R. Bala, E. A. Bernal, P. Paul, and A. Burry. Estimating gaze direction of vehicle drivers using a smartphone camera. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 165–170, 2014. 2

[3] H. Deng and W. Zhu. Monocular free-head 3d gaze tracking with deep learning and geometry constraints. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, pages 3162–3171. IEEE, 2017. 2, 4

[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*, 32(3):478–500, 2010. 1, 2

[5] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 5

[6] G. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 2

[7] Q. Huang, A. Veeraraghavan, and A. Sabharwal. Tabletgaze: unconstrained appearance-based gaze estimation in mobile tablets. *arXiv preprint arXiv:1508.01244*, 2015. 1, 2

[8] K. Krafka, A. Khosla, P. Kellnhofer, H. Kannan, S. Bhandarkar, W. Matusik, and A. Torralba. Eye tracking for everyone. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2176–2184, 2016. 1, 2, 4

[9] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012. 2, 4, 5

[10] Y. LeCun, L. Jackel, L. Bottou, C. Cortes, J. S. Denker, H. Drucker, I. Guyon, U. Muller, E. Sackinger, P. Simard, et al. Learning algorithms for classification: A comparison on handwritten digit recognition. *Neural networks: the statistical mechanics perspective*, 261:276, 1995. 2

[11] F. Lu, Y. Sugano, T. Okabe, and Y. Sato. Adaptive linear regression for appearance-based gaze estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(10):2033–2046, 2014. 2

[12] K. A. F. Mora and J.-M. Odobez. Gaze estimation from multimodal kinect data. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*, pages 25–30. IEEE, 2012. 2

[13] A. Patney, J. Kim, M. Salvi, A. Kaplanyan, C. Wyman, N. Benty, A. Lefohn, and D. Luebke. Perceptually-based foveated virtual reality. In *ACM SIGGRAPH 2016 Emerging Technologies*, page 17. ACM, 2016. 1

[14] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015. 4

[15] A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang, and R. Webb. Learning from simulated and unsupervised images through adversarial training. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 3, page 6, 2017. 1, 2, 8

[16] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 2, 7

[17] B. A. Smith, Q. Yin, S. K. Feiner, and S. K. Nayar. Gaze locking: passive eye contact detection for human-object interaction. In *Proceedings of the 26th annual ACM symposium on User interface software and technology*, pages 271–280. ACM, 2013. 6

[18] H. Su, C. R. Qi, Y. Li, and L. J. Guibas. Render for cnn: Viewpoint estimation in images using cnns trained with rendered 3d model views. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2686–2694, 2015. 1, 4, 5

[19] Y. Sugano, Y. Matsushita, and Y. Sato. Learning-by-synthesis for appearance-based 3d gaze estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1821–1828, 2014. 1, 2, 7, 8

[20] K. Wang and Q. Ji. Real time eye gaze tracking with 3d deformable eye-face model. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1003–1011, 2017. 2

[21] E. Wood, T. Baltrušaitis, L.-P. Morency, P. Robinson, and A. Bulling. A 3d morphable eye region model for gaze estimation. In *European Conference on Computer Vision*, pages 297–313. Springer, 2016. 2, 6, 7

[22] E. Wood, T. Baltrušaitis, L.-P. Morency, P. Robinson, and A. Bulling. Learning an appearance-based gaze estimator from one million synthesised images. In *Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research & Applications*, pages 131–138. ACM, 2016. 1, 2, 5, 8

[23] E. Wood, T. Baltrusaitis, X. Zhang, Y. Sugano, P. Robinson, and A. Bulling. Rendering of eyes for eye-shape registration and gaze estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3756–3764, 2015. 1, 2, 3, 5, 7, 8

[24] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014. 4

[25] X. Zhang, Y. Sugano, M. Fritz, and A. Bulling. Appearance-based gaze estimation in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4511–4520, 2015. 1, 2, 3, 4, 5, 6, 7, 8

[26] X. Zhang, Y. Sugano, M. Fritz, and A. Bulling. Its written all over your face: Full-face appearance-based gaze estimation. In *Proc. IEEE International Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2017. 1, 2, 4

[27] 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*, 2017. 1, 2, 4, 5, 7, 8