Face-Based Gaze Estimation Using Residual Attention Pooling Network

Chaitanya Bandi\textsuperscript{a} and Ulrike Thomas\textsuperscript{b}

Robotics and Human-Machine-Interaction Lab, Chemnitz University of Technology, Reichenhainer str. 70, Chemnitz, Germany

Keywords: Gaze, Attention, Convolution, Face.

Abstract: Gaze estimation reveals a person’s intent and willingness to interact, which is an important cue in human-robot interaction applications to gain a robot’s attention. With tremendous developments in deep learning architectures and easily accessible cameras, human eye gaze estimation has received a lot of attention. Compared to traditional model-based gaze estimation methods, appearance-based methods have shown a substantial improvement in accuracy. In this work, we present an appearance-based gaze estimation architecture that adopts convolutions, residuals, and attention blocks to increase gaze accuracy further. Face and eye images are generally adopted separately or in combination for the estimation of eye gaze. In this work, we rely entirely on facial features, since the gaze can be tracked under extreme head pose variations. With the proposed architecture, we attain better than state-of-the-art accuracy on the MPIIFaceGaze dataset and the ETH-XGaze open-source benchmark.

1 INTRODUCTION

Eye gaze is a crucial nonverbal cue that determines a person’s intent. The person’s intent is extremely useful in human-robot interaction applications (Huang and Mutlu, 2016) such as attracting a robot’s attention by glancing at it, and when combined with body motions, it is possible to strengthen communication between human and robot. Aside from robotics, the gaze can be used in human-computer interface (Zhang et al., 2019; Li et al., 2019; Wang et al., 2015), virtual reality (Patney et al., 2016; Konrad et al., 2019), and behavioral analysis (Hoppe et al., 2018). Model-based methods and appearance-based methods (Hansen and Ji, 2010) are used to estimate eye gaze. Although classic model-based eye gaze assessment approaches (Guestrin and Eizenman, 2006; Nakazawa and Nitschke, 2012; Valenti et al., 2012; Funes Mora and Odobez, 2014; Xiong et al., 2014) are accurate, the environment is extremely regulated (i.e., slight occlusions and static laboratory settings). Furthermore, the distance between the customized device or camera with RGB-D sensors and the eye is fixed (often to 60 cm) in order to estimate the gaze. The eye model is assumed to be constant across all participants, and without proper calibration, the system frequently fails to estimate the right gaze. Model-based solutions fail frequently in real-world applications that involve estimating the gaze in the wild (i.e., in an uncontrolled environment).

Because of the restrictions of the model-based techniques, recent research has switched to appearance-based models. Dedicated devices are not essential for appearance-based gaze estimating techniques because standard cameras are adequate for image processing and gaze regression. The appearance-based models are further classified into two types:

1) Feature-based methods and deep learning-based methods. The early works focus on effective feature extraction techniques like the histogram of oriented gradients (Martinez et al., 2012) to estimate gaze. The histogram of oriented gradients works well for low-level feature extraction but fails to effectively extract high-level features for gaze in images. One of the early efforts (Baluja and Pomerleau, 1994) tracks gaze using artificial neural networks using $15 \times 15$ retina input. Later appearance-based approaches (Tan et al., 2002) estimate eye gaze from images using non-linear mapping functions. Each calibrated subject has its mapping function. The work (Williams et al., 2006) uses linear interpolation to do an appearance-based closest manifold point query. Training data is frequently used in appearance-based models. The paper introduces a semi-supervised Gaussian process with an uncertainty measure that learns map-
pings from partially labeled input. For unseen images, the sparse regression model infers mappings from processed pixel data in real time. The saliency gaze (Chang et al., 2019) estimates gaze on uncalibrated users to solve the problems of classic gaze estimation methods such as calibration, lighting, and position fluctuations. Using 11 optimization, the adaptive linear regression (Lu et al., 2014) approach handles calibration problems based on a large number of training samples, image resolution, and blinking. Although there is a slight improvement in accuracy, they are not reliable enough to apply in real-world scenarios.

2) Deep learning approaches such as convolutional neural networks (CNN) have been proved to be effective in extracting high-level image characteristics and learning non-linear information for regression applications. Recent research indicates that CNN-based design regresses the direction of human attention in eye images (Zhang et al., 2015; Yu et al., 2018; Fischer et al., 2018; Cheng et al., 2018; Lorenz and Thomas., 2019a), face images (Zhang et al., 2016; Xiong et al., 2019; Zhang et al., 2020; Park et al., 2019), or from both face and eye images (Kraftka et al., 2016; Chen and Shi, 2019; Cheng et al., 2020a).

We focus on regressing the 2D gaze vector from face images in this work because CNNs can regress outputs even with eye occlusions. To directly regress the 2D gaze vector, an image of the face is sent through the proposed network. Figure 1 depicts the basic architecture flow. This paper makes the following contributions:

[1] We provide a novel network design that regresses the 2D gaze vector using a Panoptic-feature pyramid network (PFPN) (Kirillov et al., 2019), residual blocks, pooling, and self-attention modules.

[2] Using the proposed technique, the network achieves cutting-edge performance on two separate datasets: the MPIIFaceGaze (Zhang et al., 2016) dataset and the ETH-XGaze (He et al., 2015) dataset.

2 RELATED WORK

Deep learning-based appearance methods have been found to be more efficient than model-based and feature-based learning methods in cross-subject gaze estimation.

2.1 Convolutional Neural Network Architectures

CNNs have proven to be useful in a variety of computer vision applications, including eye-gaze estimation. CNN-based gaze estimate is affected by the input features. CNNs can regress eye gaze utilizing features such as eyes and face either dependently or independently.

**Eye-based Methods.** The paper (Zhang et al., 2015) provides the first CNN-based gaze estimating methodology that works in real-world situations. LeNet architecture (Lecun et al., 1998) inspired the proposed multimodal CNN architecture. The multi-model CNN receives 60 × 36 pixel eye images as input and outputs a 2D gaze vector, providing an open-source unconstrained high-resolution dataset known as the MPIIGaze dataset. (Yu et al., 2018) present a multi-task framework that uses an end-to-end model known as the constrained landmark gaze model to localize eye landmarks and eye gaze. (Yu et al., 2018) use UnityEyes (Wood et al., 2016) and supplement data for training and evaluation to build an end-to-end model. In natural settings, the distance between the camera and subject is greater, and the resolution of the eye is fairly low. The work in (Lorenz and Thomas., 2019b) present a multi-task CNN architecture that extracts the facial features at first and then eye features for gaze estimation using geometric method. (Fischer et al., 2018) present a CNN model for a new large-scale dataset known as the RT-GENE that feeds two eye regions to the VGG-16 (Simonyan and Zisserman, 2014) network individually. The characteristics are later concatenated with head posture information to regress the 2D eye gaze vector. (Cheng et al., 2018) proposes two networks for eye gaze regression that take advantage of eye asymmetry. The first network is an asymmetry regression network with four streams for 3D gaze regression and a two-stream assessment network for asymmetry correction.

**Face and Eye Combined Methods.** iTracker (Kraftka et al., 2016) is one of the first attempts to use CNNs to forecast gaze based on the face, patches of both eye regions, and face grid. The iTracker is specifically built for commodity hardware such as mobile phones and tablets, and it makes use of a novel dataset known as GazeCapture. According to the study in (Chen and Shi, 2019), most CNN architectures use multi-layer downsampling, which degrades spatial resolution. Dilated convolutions are used to extract features to avoid this. The dilated convolutions are taken into account for both eye pictures but not for the facial region. A coarse to fine strategy (Cheng et al., 2020a) estimates coarse gaze direction from a facial image, fine gaze direction from an eye image, and final output gaze is refined. (L R D and Biswas, 2021) suggests the AGE-Net...
3 METHODOLOGY

In this section, we present the Panoptic feature pyramid and Residual Attention Pooling (P-RAP) framework for eye gaze estimation. We investigate the essential building blocks of this architecture, such as panoptic-FPN, residual blocks, and attention processes, to gain a deeper understanding. Residual networks (He et al., 2015) were designed to increase the accuracy and performance of image recognition. Residual networks have been found to be more efficient in feature extraction and to optimize quicker with skip connections than networks such as VGG (Simonyan and Zisserman, 2014). The residual blocks are the fundamental backbone of the panoptic-FPN in our architecture. The panoptic-FPN (Kirillov et al., 2019) architecture is commonly used for object detection and semantic segmentation in order to acquire multi-scale characteristics for detecting smaller and larger objects. The network is made up of bottom-up and top-down layers containing lateral connections to increase object detection accuracy and image segmentation. We employ the panoptic-FPN architecture in this work to preserve the multi-scale aspects of the face and eyes. The basic architecture flow of panoptic features is shown in Figure 1.

3.1 Attention Mechanism

The attention mechanism accepts \( n \) input features and returns \( n \) output features. Attention’s core operation
Figure 2: The self-attention mechanism.

is that it learns to pay greater attention to the required elements. The attention method proposed in (Vaswani et al., 2017) works well for a wide range of applications, including natural language processing (Vaswani et al., 2017) and computer vision (Dosovitskiy et al., 2021; Heo et al., 2021). The attention mechanism also referred to as scaled dot product attention, takes as inputs queries ($Q$), keys ($K$), and values ($V$). The identical characteristics from the input are replicated and passed as the queries, keys, values, and attention is calculated as

$$Attention(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

(1)

where $\sqrt{d_k}$ is a scaling factor. The attention mechanism is applicable to $n$-dimensional (D) space. Figure 2 depicts the single-head attention mechanism. By merging several heads simultaneously, the single-head attention process is expanded to multi-head attention. We experiment with 2, 4, and 8 heads for spatial or convolutional attention in this paper.

### 3.2 Panoptic and Residual Attention Pooling Network

We feed the input face image of size $I \in \mathbb{R}^{224 \times 224 \times 3}$ to the P-RAP architecture. The panoptic-FPN consists of five bottom-up layers and four top-down layers. The last four top-down layers share the lateral feature information from the bottom-up layers. Each top-down layer from the FPN is then passed to a convolution layer to obtain a size of $256 \times 56 \times 56$. The four-layer outputs are then element-wise summed to obtain features of size $256 \times 56 \times 56$ which are known as panoptic features. The panoptic features are then forward to a simple convolutional layer with pooling to obtain features of size $128 \times 28 \times 28$. The features are then combined with position embeddings similar to transformer encoder (Vaswani et al., 2017). We pass the features to the residual-attention-pooling (RAP) network as in Figure 1 purple block. The RAP network consists of residual blocks and attention convolution. Each residual-attention block consists of: attention $\rightarrow$ convolution $\rightarrow$ batchNorm $\rightarrow$ ReLU $\rightarrow$ attention $\rightarrow$ convolution $\rightarrow$ batchNorm $\rightarrow$ ReLU $\rightarrow$ attention $\rightarrow$ convolution $\rightarrow$ batchNorm $\rightarrow$ skipconnection $\rightarrow$ ReLU. After the activation layer, the features are pooled and passed through the residual-attention block. The process is repeated 2 times and we linearize the output and pass through $1 \times N$ self-attention layer. The process of linearized output is illustrated in Figure 1. Finally, the attended features are forwarded to a linear layer to regress the pitch and yaw angles of eye gaze (i.e., 2D gaze).

### 3.3 Gaze and Loss Function

Designing the network for a 2D gaze vector is more efficient than optimizing the network for a 3D gaze vector. We use the same nomenclature as (Zhang et al., 2016; He et al., 2015) to transform the regressed 2D gaze vector to a 3D gaze and vice versa as needed. We compute 2D gaze yaw ($\theta$) and pitch ($\phi$) values from a 3D gaze vector.

$$\theta = \arcsin(y)$$

(2)

$$\phi = \arctan2(x, z)$$

(3)

Similarly, we compute 3D unit gaze vector $[x,y,z]^T$ given 2D gaze angles as

$$x = \cos(\theta) \cdot \sin(\phi)$$

(4)

$$y = \sin(\theta)$$

(5)

$$z = \cos(\theta) \cdot \cos(\phi)$$

(6)

This conversion is unique to the ETHX-Gaze dataset (He et al., 2015) and the MPIIFaceGaze (Zhang et al., 2016). We employ the $L_1$ loss function with regularization to backpropagate the weights of the proposed architecture. The loss function is

$$L_1 = |\text{predicted}_\text{gaze} - \text{actual}_\text{gaze}| + \lambda \sum_{i=1}^{N} |w_i|$$

(7)

where $\lambda$ is a regularization parameter and $w_i$ are the weights of the network.
4 EXPERIMENTS

In this section, we evaluate the proposed P-RAP architecture and experiment with two open-source gaze datasets.

4.1 Gaze Datasets

We utilize datasets with face images to train and assess our proposed network technique because we intend to regress gaze purely on face images. We use two open-source datasets to test our architecture. 1) The MPIIFaceGaze (Zhang et al., 2016) dataset contains high-resolution photos of people who are closer to the camera. 2) The ETH-XGaze (He et al., 2015) collection contains incredibly high-resolution images.

MPIIFaceGaze Dataset. The MPIIFaceGaze (Zhang et al., 2016) dataset is a subset of the MPIIGaze (Zhang et al., 2015) dataset. The MPIIGaze dataset was originally composed of eye images for experimentation, but a subset of face images was eventually provided as MPIIFaceGaze. The MPIIGaze dataset is totally captured in an uncontrolled context, such as daily laptop usage over long periods of time. When the target was displayed, the individuals were prompted to hit a key. The dataset contains 213,659 photos from 15 distinct contributors. In the MPIIFaceGaze dataset, a subset of 45,000 samples with full facial images is released from this dataset. Each participant’s dataset has 3,000 samples. A few samples are shown in Figure 3.

ETH-XGaze. The ETH-XGaze (He et al., 2015) dataset contains extremely high-resolution images acquired with 18 Canon 250D digital SLR cameras. The images captured have a resolution of 6000 × 4000 pixels. The ETH-XGaze dataset contains a large number of head position variations ranging from ±80°, ±80°. The ETH-XGaze dataset is a massive collection of over 1 million photos from 110 individuals. Figure 4 shows the facial cropped data images.

4.2 Evaluation Metric and Training Parameters

The evaluation metric for measuring the performance is the 3D angular error. The angular error between the actual $g_{\text{actual}}$ and the predicted gaze $g_{\text{predicted}}$ is computed as

$$L_{\text{angular}} = \frac{g_{\text{actual}} \cdot g_{\text{predicted}}}{\|g_{\text{actual}}\| \|g_{\text{predicted}}\|}. \quad (8)$$

The metric is utilized for both within and cross-dataset evaluation. To train the proposed architecture, we use the Adam optimizer (Kingma and Ba, 2017) with a learning rate of 0.0001 and a weight decay of 1e-6. We use an Nvidia RTX Quadro with a 48GB graphical processing unit to train the networks. Each training batch consists of 256 - 224 × 224 × 3 pixel images. For each evaluation, the architecture is trained for 50 epochs, and in most cases, the loss curves (training and validation loss) stabilize around 30th epoch.

4.3 Within Dataset Evaluation

Within dataset evaluation evaluates the effectiveness of data from a similar subset on unknown subjects. The MPIIFaceGaze dataset started cross-validation with leave-one-person-out. The dataset contains 15 persons, 14 of which are used for training and one for testing. The procedure is evaluated 15 times, with the overall accuracy calculated by averaging the results. We use a similar cross-validation technique to evaluate the performance of the suggested architectures.

4 EXPERIMENTS

In this section, we evaluate the proposed P-RAP architecture and experiment with two open-source gaze datasets.

4.1 Gaze Datasets

We utilize datasets with face images to train and assess our proposed network technique because we intend to regress gaze purely on face images. We use two open-source datasets to test our architecture. 1) The MPIIFaceGaze (Zhang et al., 2016) dataset contains high-resolution photos of people who are closer to the camera. 2) The ETH-XGaze (He et al., 2015) collection contains incredibly high-resolution images.

MPIIFaceGaze Dataset. The MPIIFaceGaze (Zhang et al., 2016) dataset is a subset of the MPIIGaze (Zhang et al., 2015) dataset. The MPIIGaze dataset was originally composed of eye images for experimentation, but a subset of face images was eventually provided as MPIIFaceGaze. The MPIIGaze dataset is totally captured in an uncontrolled context, such as daily laptop usage over long periods of time. When the target was displayed, the individuals were prompted to hit a key. The dataset contains 213,659 photos from 15 distinct contributors. In the MPIIFaceGaze dataset, a subset of 45,000 samples with full facial images is released from this dataset. Each participant’s dataset has 3,000 samples. A few samples are shown in Figure 3.

ETH-XGaze. The ETH-XGaze (He et al., 2015) dataset contains extremely high-resolution images acquired with 18 Canon 250D digital SLR cameras. The images captured have a resolution of 6000 × 4000 pixels. The ETH-XGaze dataset contains a large number of head position variations ranging from ±80°, ±80°. The ETH-XGaze dataset is a massive collection of over 1 million photos from 110 individuals. Figure 4 shows the facial cropped data images.

4.2 Evaluation Metric and Training Parameters

The evaluation metric for measuring the performance is the 3D angular error. The angular error between the actual $g_{\text{actual}}$ and the predicted gaze $g_{\text{predicted}}$ is computed as

$$L_{\text{angular}} = \frac{g_{\text{actual}} \cdot g_{\text{predicted}}}{\|g_{\text{actual}}\| \|g_{\text{predicted}}\|}. \quad (8)$$

The metric is utilized for both within and cross-dataset evaluation. To train the proposed architecture, we use the Adam optimizer (Kingma and Ba, 2017) with a learning rate of 0.0001 and a weight decay of 1e-6. We use an Nvidia RTX Quadro with a 48GB graphical processing unit to train the networks. Each training batch consists of 256 - 224 × 224 × 3 pixel images. For each evaluation, the architecture is trained for 50 epochs, and in most cases, the loss curves (training and validation loss) stabilize around 30th epoch.

4.3 Within Dataset Evaluation

Within dataset evaluation evaluates the effectiveness of data from a similar subset on unknown subjects. The MPIIFaceGaze dataset started cross-validation with leave-one-person-out. The dataset contains 15 persons, 14 of which are used for training and one for testing. The procedure is evaluated 15 times, with the overall accuracy calculated by averaging the results. We use a similar cross-validation technique to evaluate the performance of the suggested architectures.

Figure 3: MPIIFaceGaze (Zhang et al., 2015) sample images from dataset.

Figure 4: ETH-XGaze (He et al., 2015) sample images from dataset.

Figure 5: The output 2D gaze vector on the ETH-Xgaze and the MPIIFaceGaze test set. The red arrow is the predicted gaze and the green arrow is the ground truth gaze.
Table 1: Comparison of the proposed architecture results to the state-of-the-art. The values are within-dataset evaluation errors. The gaze angular errors are in degrees.

<table>
<thead>
<tr>
<th>Datasets</th>
<th>Method</th>
<th>MPIIFaceGaze</th>
<th>ETH-XGaze</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>FewShotGaze (Park et al., 2019)</td>
<td>5.2°</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>MPIIFaceGaze (Zhang et al., 2016)</td>
<td>4.8°</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>ETH-XGaze (He et al., 2015)</td>
<td>4.8°</td>
<td>4.5°</td>
</tr>
<tr>
<td></td>
<td>RT-GENE (Fischer et al., 2018)</td>
<td>4.3°</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>FARE-Net (Cheng et al., 2020b)</td>
<td>4.3°</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>CA-Net (Cheng et al., 2020a)</td>
<td>4.1°</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>AGE-Net (L RD and Biswas, 2021)</td>
<td>4.09°</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>L2CS-Net (Abdelrahman et al., 2022)</td>
<td>3.92°</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>P-RAP (Ours)</td>
<td>3.8°</td>
<td>4.09°</td>
</tr>
</tbody>
</table>

Table 2: Cross-dataset evaluation results in degrees.

<table>
<thead>
<tr>
<th>Test</th>
<th>Train</th>
<th>MPIIFaceGaze</th>
<th>ETH-XGaze</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MPIIFaceGaze (Zhang et al., 2016)</td>
<td>3.8°</td>
<td>27.95°</td>
</tr>
<tr>
<td></td>
<td>ETH-XGaze (He et al., 2015)</td>
<td>6.79°</td>
<td>4.09°</td>
</tr>
</tbody>
</table>

on the MPIIFaceGaze dataset. The architecture regresses the 2D gaze vector, and to measure 3D angular error, we transform both the actual and predicted 2D gaze vectors to the previously specified 3D unit vector. Figure 7 depicts the mean 3D angular gaze error for 15 subjects in the MPIIFaceGaze dataset. The 3D angular error resulting from the proposed network trained on the MPIIFaceGaze dataset is represented in Figure 7. The average inaccuracy for all participants is about 3.8°. The chart demonstrates that the majority of the participants have an angle error of less than 4.5°. The most deviations are 4.8° and 6.27° for participants P02 and P14, respectively. The ETH-XGaze dataset has predefined training and test samples. We obtain a 3D angular accuracy of 4.09° on test set. The ground truth gaze of the ETH-XGaze test set is not available for direct evaluation. As the dataset is very recently released not many works are available for comparison.

Finally, we compare the P-RAP network results to the state-of-the-art methods for face-based gaze estimation. Table 1 contains the comparison findings. According to the results, our proposed model delivers state-of-the-art results on the MPIIFaceGaze and ETH-XGaze datasets (as far as published work). Figure 5 and 6 depict a few output samples from both datasets with low and high precision.

4.4 Cross Dataset Evaluation

Cross dataset evaluation tests the performance of a model on a completely different dataset. For cross dataset evaluation, we retrained the architecture with complete MPIIFaceGaze dataset. We did not retrain the ETH-XGaze dataset as leave-one-out cross-validation is not performed during training. The cross dataset evaluation accuracy is mentioned in Table 2. The model trained on MPIIFaceGaze dataset is used for obtaining the gaze for the test set of ETH-XGaze dataset resulting in 27.9° angular error. Next, the model is trained on ETH-XGaze dataset and tested on MPIIFaceGaze dataset to obtain 3D angular error of 6.8°. From this, we can see that the model trained on the ETH-XGaze dataset performs better for the cross dataset evaluation.
4.5 Human Robot Interaction Application

We use the gaze estimation method in a real-time human-robot interaction setting in this section. The purpose of this environment is to capture a robot’s attention in a human-robot interaction scenario. The camera is approximately 1 to 2 meters away from the subject. According to cross-dataset examination, the ETH-XGaze dataset performs better than others from Table 2. We cascade the dlib (King, 2009) face detection and head posture estimation with the suggested FPN-AP architecture for gaze estimation for real-time applications. The suggested model is applied in a real-time situation, and the direction of gaze in the surroundings is shown in Figure 8. We can clearly discern the directions left, right, up, and down based on the experiments. In addition, we also tested the gaze in another human-robot environment for picking objects by gazing at them. From the experiments, we noticed that the distance between the camera and the human as well as the distance between objects are highly dependent. Although it worked for certain distances, it requires quite a huge improvement for real-time object-picking applications. As a further improvement, we are currently working on combining multi-modal communication information for the object-picking human-robot application.

5 CONCLUSION

We presented the P-RAP network design for eye gaze estimation with a panoptic feature pyramid network, residual blocks, and attention mechanism. We evaluated the framework using two large-scale open-source datasets. On both datasets, we conducted within-dataset and cross-dataset evaluations and obtained state-of-the-art performance. We aim to further improve the accuracy of gaze for real-time robotic applications in combination with multimodal communication.
255–258, New York, NY, USA. Association for Computing Machinery.


