"Blinks in the Dark": Blink Estimation With Domain Adversarial Training (BEAT) Network

Seonghun Hong, Yonggyu Kim, and Taejung Park

***Abstract*—Blink detection plays an important role in many human-computer interaction applications for consumers. Unfortunately, deep neural network-based blink detection methods are not only susceptible to poor lighting conditions, but also the deep learning model is prone to bias due to the imbalance in the dataset distribution. To solve these problems, we propose Blink Estimation with Domain Adversarial Training (BEAT) network, which robustly detects blinks in unseen out-of-sample images captured even under poor lighting conditions by extracting domain-invariant features. BEAT network is inspired by the domain-adversarial neural network (DANN) but improved with several improvements including a lambda scheduler to stabilize adversarial training and a gradient decay layer to prevent the discriminative loss from overwhelming the classification loss. As a result, BEAT achieves faster and more accurate blink detection performances than other domain generalization methods for unseen target domains. In particular, BEAT's feature extractor model achieves state-of-the-art performance in terms of AUPR on popular benchmark datasets. Also, we suggest a practical optimal threshold for blink detection based on our insights gained from our experiments for consumer applications.**

***Index Terms*—Blink detection, domain generalization, gaze estimation.**

I. Introduction

BLINK detection is an essential task in various human computer interaction (HCI) scenarios such as gaze estimation, deception detection [2], driver fatigue detection [3], face anti-spoofing [4], and dry eye syndrome recovery [5]. For these applications, researchers have been working to improve the performance of eye blink detection [6], [7], [8], [9]. Some of these methods, for example, [10], [11] have been reported to work well in real-world environments.

Nevertheless, some limitations due to racial differences, lighting, data imbalance, etc. in publicly available training datasets are not sufficiently considered in the previous studies and impede the practical application of blink detection. Those limitations become more severe when test or target datasets have different distributions - for example, different races and lighting conditions. In real-world applications, it is very likely that practitioners will have different racial distributions in the images between the training dataset and the test (target) dataset. This problem becomes more evident in target countries or regions with homogeneous ethnic groups. Also, lighting conditions can be stringent for some specific applications where lighting needs to be reduced in order not to disturb the user. Target images captured in these environments tend to be darker than images from publicly available training datasets.

Another issue that has not been fully addressed is data imbalance in the blink datasets. This means that most blink data sets have significantly more open-eye images than closed-eye images. As a consequence, deep learning models can be skewed to one side because the samples corresponding to the two classes do not exist evenly.

To address the mentioned issues, we propose and discuss several strategies in this paper. For the racial bias and poor lighting conditions in datasets, we apply domain generalization based on domain adversarial training scheme. "Domain" in this context indicates a group of similar image distribution (i.e., races, bright and dark lighting conditions, various backgrounds). Thus, each set of image data with similar racial distribution or lighting condition forms a "domain". The main goal of domain generalization is to detect blinks correctly and not be confused with unnecessary domain information (racial bias or different lighting conditions). Also, we suggest a practical guideline to determine threshold from our experiments to mitigate the data imbalance issue.

To implement the mentioned strategies, we present a baseline network (i.e., a feature extractor) that outperforms the latest results on public eye blink datasets. Based on the baseline network, we design three versions of Blink Estimation with Domain Adversarial Training (BEAT) networks, which can generalize unseen domains using adversarial training and the KL divergence loss to implement the mentioned strategies. To stabilize adversarial training, we design and apply a gradient regularization method to BEAT. Figure 1 visualizes the generalization result of BEAT using the t-SNE method [1]. Without the domain generalization, the features from an unseen target domain (highlighted in blue) are distributed in a separative cluster from source domains on the left. Since the BEAT network extracts domain-agnostic features, the feature

Manuscript received 2 January 2023; revised 8 February 2023 and 27 March 2023; accepted 9 May 2023. Date of publication 11 May 2023; date of current version 18 August 2023. This work was supported by the National Research Foundation of Korea (NRF) Grant funded by the Korea Government (MSIT) under Grant 2021R1A4A5028907. *(Seonghun Hong and Yonggyu Kim contributed equally to this work.) (Corresponding author: Taejung Park.)*

Seonghun Hong was with the Research and Development Team, VisualCamp, Seoul 06770, South Korea. He is now with the Department of Electrical and Computer Engineering, Seoul National University, Seoul 08732, South Korea (e-mail: h2o0318@snu.ac.kr).

Yonggyu Kim is with the Research and Development Team, VisualCamp, Seoul 06770, South Korea (e-mail: aiden@visual.camp).

Taejung Park is with the Department of Cybersecurity, Duksung Women's University, Seoul 01369, South Korea (e-mail: tjpark@duksung.ac.kr).

Digital Object Identifier 10.1109/TCE.2023.3275540

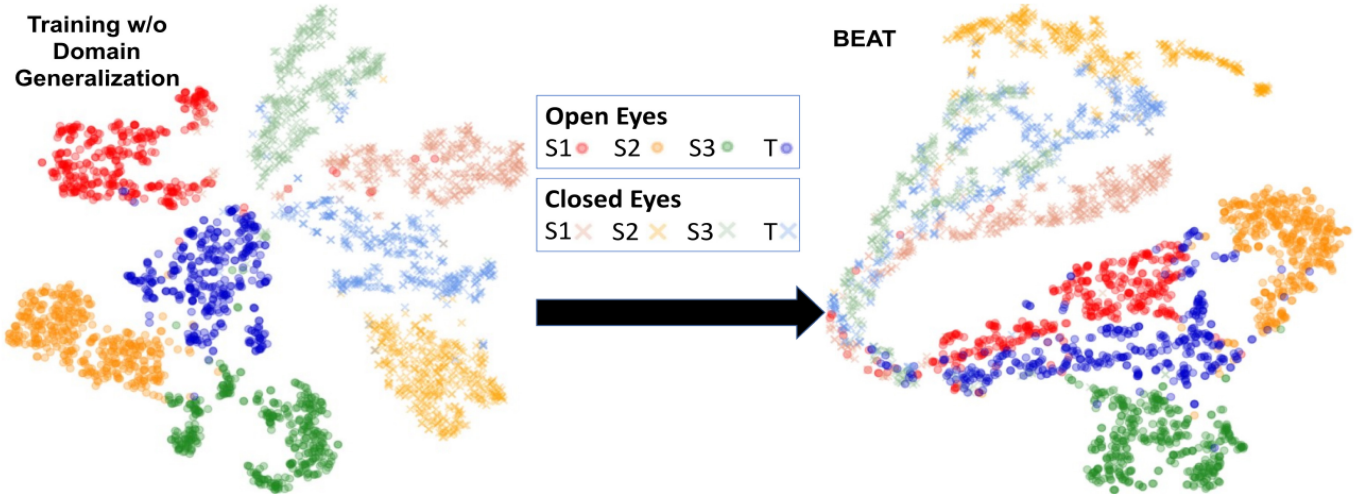

Fig. 1. Feature maps of the trained distributions without domain generalization (left) and with BEAT network (right), visualized by the t-SNE method [1]. S1, S2, and S3 represent source domains for training, and T indicates the target domain for classification. After applying BEAT, we can see that the different colors (domains) come together to be less separable and the other symbols (◯ for open and × for closed) move away from each other to be more separable (for example, using a hyperplane). This means that BEAT can better classify open/closed states without being confused by various images from different domains (i.e., different ethnic distributions, lighting conditions, backgrounds, etc.).

map on the right shows that the features of the target domain appear indistinguishable with those of source domains in an aggregated cluster.

Our contributions can be summarized as follows.

1) Performance: Our baseline network (i.e., the feature extractor) outperforms the latest results on the RT-BENE [10] dataset, which is the de facto standard for eye blink detection. For instance, our baseline network performs 1.57% higher in the AUPR (area under precision-recall curve) and is 2.86 times faster than the latest method [12] (Table V).

2) Optimal Thresholds on Data-Imbalaced Blink Datasets: As blink datasets are highly imbalanced, we discuss appropriate evaluation metrics. Based on the discussion, we propose optimal thresholds that maximize the F1-score for binary classification for imbalanced datasets. Furthermore, we also propose and discuss how to find the optimal sampling rate according to the optimal threshold for various cases.

3) Domain Generalization for Real Applications: We propose a domain adversarial training method for domain generalization using a gradient decay layer which enables stable adversarial training. The results show that our domain generalization method improves binary classification performances in the AUROC (area under receiver operating characteristic curve) and the AUPR by 2.99% and 50.24% in the Eyeblink8 target domain, 7.21% and 4.47% in the BID target domain, and 2.14% and 23.76% in the RT-BENE target domain, respectively.[1]

II. Background

Blink detection is usually implemented independently prior to gaze estimation as a natural design choice. This is important in the appearance-based gaze estimation methods [13], [14], [15] which are based on the deep neural networks (DNNs) because they predict gaze positions even when the user closes the eyes or they fail to recognize eyes correctly on the face images. Therefore, one of the important roles of the blink detection stage is to avoid unreliable outputs in the gaze estimation stage as a fail-safe. For example, Fig. 2 shows an electric wheelchair controlled by gaze estimation developed for people with disabilities who can only move their eyes and cannot operate control sticks with their arms and hands. If blink detection does not work correctly, users run a serious risk from unreliable gaze predictions when they close their eyes (e.g., sudden random changes in gaze position). In this wheelchair application, we have found that previous blink detection methods easily fail in backlit or very dim lighting conditions. Therefore, we need a robust way to predict blinks in backlit or very dark environments.[2]

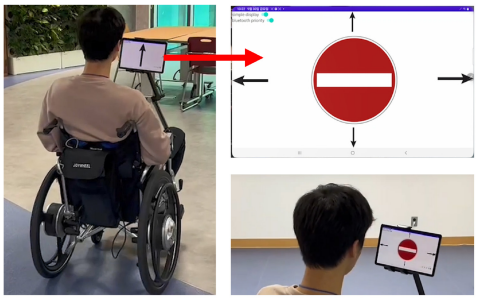

Fig. 2. Left: gaze direction control electric wheelchair for the disabled. Gaze estimation is based on an appearance-based method. Right: control screen of the wheelchair. The user determines the direction and movement of the powered wheelchair by staring at the arrow. Courtesy of Jaehyun Kim.

Another interesting consumer application for blink detection is indoor golf driving ranges (Fig. 3). Some novice golfers tend to involuntarily blink during the swing motion or at the moment of impact, resulting in unsatisfactory results. As a

[1]You can watch the video of the blink detection test results: https://youtu.be/m7b1Fsu8m4w.

[2]Beyond this, we need a reliable method to estimate gaze position in these situations for the powered wheelchair, which is beyond the scope of this paper. For focused discussion, this paper only describes how to reliably detect blinks in backlit or very dim lighting conditions.

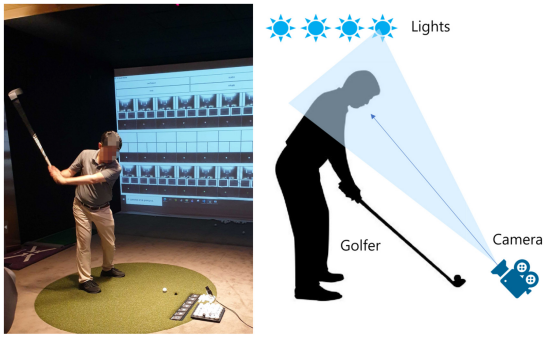

Fig. 3. Left: our lab environment simulating a typical indoor golf driving range where our dataset (BID) has been prepared (Section IV). Right: schematic diagram of the lab environment. Lighting and camera positioning lead to dark, backlit face images as shown in Fig. 4.

Fig. 4. Samples from our BID dataset, captured from the lab environment shown in Fig. 3.

natural consequence, indoor golf application developers want their systems to be able to check the condition of the golfer's eyes during the swing motion and provide useful feedback. The problem is that lights are usually installed on the ceiling to illuminate downward, and a camera is installed on the floor to capture the golfer's face upward as shown in Fig. 3, so the face images captured by the camera are very dark (Fig. 4). Additional upward lighting is generally prohibited so as not to obstruct the golfer's vision. As we will present later, previous methods do not effectively detect blinks in these backlit face images, so we need a practical way to address this issue.

Some consumer applications may rely on blink detection to prevent catastrophic accidents. For example, car or truck drivers who drive long distances are often exposed to fatal risks due to fatigue, and reliable blink detection is required while driving in low light conditions. In general, in prior studies, infrared (IR) light sources are often adopted to capture the driver's faces or eyes in dark lighting conditions, since IR illumination does not obstruct the driver's vision when driving at night [3], [16]. Typically, the IR light source is installed near the driver's face along with the camera. However, applying IR illumination to the human eye can introduce some complications. First, some studies [17], [18], [19], [20] report that IR illumination near the eye can harm the eyes. Second, we cannot directly estimate eye states from IR face images based on deep neural network models trained with regular RGB images. Since most public and private datasets are in RGB format, the ability to detect blinks in IR images can be severely limited. Finally, IR cameras and lights are less accessible and less common to the average consumers than RGB cameras, limiting their application.

Taken together, those discussed applications have common points. First, eye blink detection in backlit or very dark environments has various useful consumer applications, and failing to detect blinks could lead to catastrophic hazards for some cases. Second, additional directional lights directed toward the user's face should be avoided so as not to obstruct the user's view. Finally, consumer-level RGB cameras may be preferred over IR devices for some economic and medical reasons. On the other hand, it is not ideal if the system is only good at detecting blinks in dark conditions and not accurately in brighter environments. Therefore, a reliable system that detects blinks regardless of lighting conditions is needed.

As mentioned, the appearance-based gaze estimation technique adopts a deep neural network approach, and the eye blink estimation step is usually applied before the gaze estimation step. Considering that both steps have many possibilities to share useful information between neural networks, it would be natural to choose deep neural networks (DNNs) to implement effective blink estimation that meets all the requirements discussed so far.

The recent great success of artificial intelligence (AI) comes from the fact that rich public data sets are easily accessible on which DNNs can be trained. However, we need to overcome the following issues with datasets for training and testing for blink estimation.

Different Brightness Levels in Datasets: Most publicly available datasets have human faces in normal lighting conditions. As a result, when generic DNNs are trained on those public datasets with normal brightness levels, they do not readily detect blinks in test images captured in dim lighting conditions.

Racial Bias in Datasets: In real-world consumer applications, it is very likely that practitioners will have different racial distributions in the images between the training dataset and the test (target) dataset. This problem becomes more evident in target countries or regions with homogeneous ethnic groups.

Data Imbalance for Classification: Most blink datasets have significantly more open-eye images than closed-eye images. As a consequence, deep learning models can be skewed to one side because the samples corresponding to the two classes do not exist evenly.

To address the mentioned issues, we propose and discuss several strategies in this paper. For the racial bias and poor lighting conditions in datasets, we apply domain generalization based on the domain adversarial training scheme. Another possible approach would be to adjust the brightness level of dark images first and then to address racial bias by applying domain generalization separately. This means additional procedures for image processing are required (i.e., "two-step" approach). Also, simply increasing the brightness of a dark image tends to introduce image noise that degrades the classification performance. We avoid this two-step approach to solve both problems in the single-domain generalization scheme and make our method suitable for mobile applications. Also, we suggest a practical guideline to determine threshold from our experiments to mitigate the data imbalance issue.

III. Related Works

A. Blink Detection

Eye blink images can be captured by either a near-infrared (IR) camera or a regular RGB camera for blink detection. When capturing IR images, the IR camera and IR light source are typically placed close to the human eye to capture high-resolution images and provide better performance even in poor external lighting conditions. However, as discussed earlier, not only are IR cameras expensive, but IR sources are also claimed to cause eye damage due to the close distance between the light source and the eye in [17], [18], [19], [20]. Therefore, consumer-grade RGB cameras have a higher potential for safe blink detection.

Blink detection methods can also be divided into two categories, one exploits multiple image sequences from a video stream, and the other analyzes only a single image. In general, single-frame-based methods are reported to have faster processing times and lower computational costs in [21]. Some blink detection studies based on multiple-frame-based methods employ LSTM and RNN models to exploit features across time series [11], [22], [23]. On the other hand, some studies focus on classical image processing techniques without relying on deep learning approaches to blink detection. Some methods extract and classify eye regions using classical image processing techniques including SIFT and HOG to detect blinks [24], [25]. Others determine the degree of eye closure based on eye aspect ratio (EAR) using eye landmarks [26], [27]. Unfortunately, those approaches tend to be vulnerable to changes in face angle or skin color. Also, since the extracted eye landmarks are different for each person, different thresholds are required for each person. Recently, interest in CNN-based blink detection techniques that can utilize rich eye image datasets is increasing [10], [21], [28], [29], [30]. Researchers have proposed various CNN-based methods, such as a two-way approach that splits the input images into two streams to extract feature for eye detection [29], and a curriculum-learning-based approach [12].

In this paper, we design a CNN-based model that can detect eye blinks in a single image, inspired by [10] and other recent studies. Based on the single-frame approach, our model provides a fast interference rate that can be suitable for mobile devices and high blink detection performance in various environments.

B. Data Imbalance

Data imbalance occurs when there are large differences in the amount of data between classes. One of the common issues with blink detection is that there are often far less images with eyes closed than with eyes open. Such imbalanced datasets cause neural networks to be biased towards the prevalent (majority) class [31], [32]. Thabtah et al. [33] also have shown that evaluation measures such as precision and recall change with data imbalance. One approach to addressing data imbalance is preprocessing, usually with undersampling or oversampling applied [34]. The undersampling task removes some majority class samples to balance, but runs the risk of losing information about the majority class. On the contrary, the oversampling task increases the number of minority class samples to balance by synthesizing new samples or augmenting existing ones. Some literature including [35], [36] argues that oversampling provides a more accurate classification than undersampling based on experiments. However, we have found that oversampling does not always work in our cases, as we will discuss in a later section. Researchers also have worked on how to determine practical thresholds for binary classification problems with data imbalance. Provost [37] discuss the intricacies involved in classifying imbalanced datasets and suggests adjusting the output threshold. To analyze the intricacies associated with classifiers, data imbalances, and thresholds, various approaches have been proposed, including the ROC convex hull method [38] and cost curves [39].

In this paper, we present our approach to find an optimal threshold and test results in both undersampling and oversampling using multiple imbalance ratios for blink detection as a binary classification problem.

C. Domain Generalization

In machine learning, researchers and practitioners frequently encounter domain shifts, defined as the difference in distributions between training and test datasets. To address the domain shift problems, two methods are usually applied: domain adaptation and domain generalization. The domain adaptation method trains models to reduce domain shifts by learning the distribution difference between source and target domains. While the domain adaptation method uses both the source and target domains, the domain generalization method uses only the source domain to generalize the target domain, which may be out-of-distribution. Therefore, when obtaining information on the target domain is impossible or too expensive, the domain generalization approach is preferred. Common strategies for the domain generalization problem include data augmentation, domain alignment, meta-learning, and ensemble learning method [40]. Among them, we apply data augmentation and domain alignment method to our blink detection for domain generalization. The data augmentation method is commonly used as a way to avoid overfitting and improve generalization performance. For image data, datasets are usually augmented using image transformation methods, including random flips, rotations, and brightness and contrast modifications. However, while image transformations help to enrich datasets with different brightness or skin tones, they cannot create reasonable variations for some meaningful features, including individual eye shapes and skin textures. Recent approaches for domain alignment aim to align domains by reducing means and variances of distributions of transformed features among domains [41]; considering KL divergence [42], [43]; or applying adversarial learning [44], [45], [46]. In particular, domain adversarial training is a min-max game that the discriminator is optimized to distinguish between domains while feature extractor model is trained to extract domain-agnostic features which interferes the discriminator from differentiating domains. Some researches expand domain adversarial training for multi-source domain [47], [48]. Other studies report that if domain labels and class

TABLE I
OVERVIEW OF DATASET SPLITS

Dataset	Blink	Train	Valid	Test	Total	Ratio
RT-BENE	Open	103,684	41,208	60,694	218,548	20.95
	Closed	4,440	1,286	3,388	10,432	1
UnityEyes	Open	10,000	10,000	10,000	30,000	1
	Closed	10,000	10,000	10,000	30,000	1
BID	Open	5,089	2,329	2,321	9,739	1.51
	Closed	3,447	1,630	1,353	6,430	1
Eyeblink8	Open	24,485	8,454	10,286	43,225	39.51
	Closed	426	291	377	1,094	1

labels are not independent of each other, the classification performance may decrease in domain adversarial training [49]. AFLAC network can learn domain invariance features without interfering classification task [48].

Inspired by the AFLAC network, we use an adversarial network to train our model, but add regularization terms to reduce the effect of discriminator loss on classification performance.

IV. EYE BLINK DATASETS

Deep neural networks usually need as many datasets as possible to guarantee performance. For a fair comparison, we select RT-BENE [10], UnityEyes [50], and Eyeblink8 [51] as experimental datasets. We also prepare and use a dataset (*"Blinks in the Dark"* or BID), extracted from video clips of golfers' eye blink moments in a golf driving range under poor lighting conditions (Figure 3 and 4). We chose the golf driving range as the data collection site because it allows us to simulate real-world situations more safely than other candidate situations (e.g., blinking while driving a car at night or maneuvering an electric wheelchair based on gaze estimation). Although BID has been collected in the context of an indoor golf driving range, we believe our dataset can be applied to other contexts and applications for blink detection.

A. Dataset Details

1) RT-BENE: Cortacero et al. [10] have created RT-BENE dataset by extracting and labelling the eye blink regions from RT-GENE dataset [52] which was originally prepared for gaze estimation task. RT-BENE collected 17 subjects without glasses, excluding subjects who wore glasses in RT-GENE. The collected images have been categorized into *open*, *closed*, and *uncertain* for the 17 subjects. We apply same data split criteria as [10] in our experiments. As in [10], we ignore the data from subject 6 and images tagged with *uncertain* for same comparison. We present the details of dataset splits (i.e., *train set*, *validation set*, and *test set*) in Table I.

2) UnityEyes: Wood et al. [50] have proposed a synthetic method to create training data for appearance-based gaze estimation using a game engine (a.k.a UnityEyes). Since this approach is designed for gaze estimation, it cannot directly generate images with closed eyes. To overcome this limitation, we reverse engineered UnityEyes execution file to generate images with closed eyes. In our configuration, we set the camera angles to have random values from 0 to 30 degrees. We apply random eyeball pitch angle from 5 to 20 degree to generate eye images with *open* and from 40 to 45 degree to generate eye images with *closed*, respectively. Also, we synthesize eye images with *uncertain* tag by applying random eyeball pitch angle from 30 to 40 degree. Since UnityEyes generates images that the eyeball is located in center, we crop 60×36 pixel size box area around center including eye and eyebrow.

3) Eyeblink8: Eyeblink8 [51] is a dataset with 70,992 frames and 640×480 resolution which capture people sitting and behaving naturally in front of the camera. Because the images have been captured under natural condition, number of the eyes closed images is very small compared to number of frames with the eyes open, as shown in Table I. Also, Fogelton and Benesova [53] have pointed out that Eyeblink8 dataset may include labeling mistakes. Therefore, We preprocess the eye images in Eyeblink8 dataset again, using same method as our BID dataset preprocessing. We selected videos of 1, 3, 8, 10 based on the folder name. Each data was taken with each different subject. And we prepare them into 60x36 pixel size eye images using MediaPipe and the face normalization method in [54]. A total of 44,319 cut out images are labeled and classified into 43,225 open images and 1,094 closed images.

4) Our Dataset [Blinks in the Dark (BID)]: The gaze estimation performance for blurred or dark images tends to be poor [55]. From this it is reasonable to assume that blink detection performance also tends to deteriorate in blurry or dark images in general. As discussed, the ability to detect blinks in dark images is directly related to user safety for some applications (e.g., drowsy driving detection at night). Therefore, it is necessary to guarantee the performance of blink estimation even in a dark environment or backlight.

In order to measure the performance change in dark images, we have collected our dataset, *Blinks in the Dark (BID)*, with a relatively long distance between the camera and the face in an indoor golf driving range (Figure 3). Our target subjects consist of 11 males and 2 females in total, with minimum of 21 and maximum of 39 years old, and the average age of the subjects is 29. There are 4 people who wear glasses. All target subjects are Asian except one female Caucasian. This means that the BID dataset has a significant racial bias (12:1), which commonly occurs in the distribution of users in some East Asian countries. We have collected eye images according to various face angles and actions based on three scenarios in which the subject turns their head and blinks differently during a golf swing. First, the subjects turned their heads with their eyes open. Second, the same subjects turned their heads with their eyes closed. Finally, the same subjects blinked three times without turning their heads. Among the subjects 1 to 13, we excluded subject 7 from the training set and test set due to poor image quality (see Table VI). We recorded video with 1440×1080 resolution and 50 frames per second (FPS). The distance between the face and the camera is larger than 1 meter. After the recording, we extracted face landmarks and 6 landmarks corresponding to each eye using MediaPipe [56]. We used the normalization method proposed

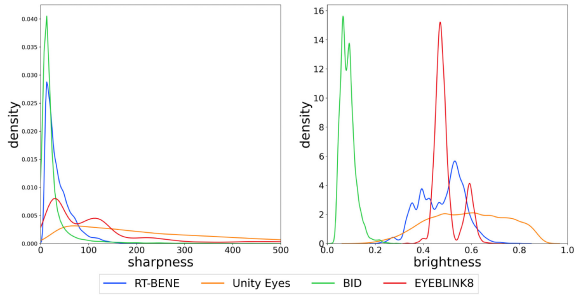

Fig. 5. Density functions for sharpness and brightness of the four datasets.

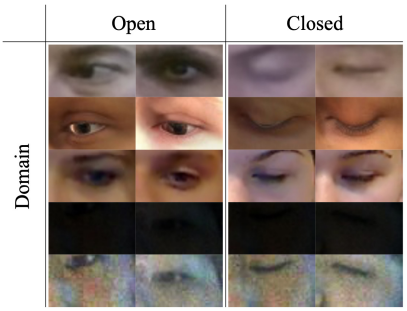

Fig. 6. Images from the datasets. Top to bottom: RT-BENE, UnityEyes, Eyeblink8, BID, BID with improved brightness. The last row is shown here for reference only and is not used for training and testing.

TABLE II
MEANS (μ) AND STANDARD DEVIATIONS (σ) OF BRIGHTNESS AND SHARPNESS FOR THE FOUR DATASETS

Dataset	Brightness		Sharpness	
	μ_B	σ_B	μ_S	σ_S
BID	0.08602	0.03117	25.45	42.69
UnityEyes	0.5859	0.1596	242.7	180.8
RT-BENE	0.4787	0.09020	36.90	28.28
Eyeblink8	0.4740	0.03062	100.6	64.19

by Zhang et al. [54], because the eye images have various features according to face yaw and pitch. This normalization method provides canonical eye images by removing various parameters, including head rotation and eye-to-camera distance.[3] We made a box containing 6 landmarks that represent the eyes, and crop the face image with vertically and horizontally $\times 1.5$ scaled region. Then we resized cropped image into 60×36 pixel size. We used both eyes by flipping right eye image to left.

B. Statistics of Datasets

The four datasets have very diverse characteristics, including skin color, brightness, and sharpness (Figure 6). In order to measure the brightness of the image, we convert RGB image into HSV image and get brightness value of image by averaging brightness value of all pixels in image. In order to get sharpness value, we convert RGB image into gray image and pass through the Laplacian filter which makes lines in the image to be emphasized. After passing through the filter, we get sharpness value by calculating variance of all pixel values. Table II summarizes the mean and standard deviation of brightness and sharpness for each dataset. Figure 5 shows the density functions for brightness and sharpness of each dataset. Mean brightness and mean sharpness are highest in the UnityEyes dataset and lowest in the BID dataset. This is because the BID dataset has been captured with backlighting in an indoor environment, while the UnityEyes dataset has been synthesized and rendered under ideal light conditions using a game engine. The RT-BENE and Eyeblink8 datasets have higher mean brightness than the BID dataset because they have been collected in natural real-world environments. However, mean sharpness of the RT-BENE dataset is lower than that of Eyeblink8 and UnityEyes datasets. This is because the RT-BENE dataset has been captured at a longer camera and subject distance as described in [10].

Table I lists the number and proportion of open and closed eye images for each dataset. As shown in Table I, Eyeblink8 is the most imbalanced (39.51:1) dataset and UnityEyes is the most balanced (1:1) dataset.

[3]Please check the normalized eye images from 0:11 in https://youtu.be/ABjrD6sFB_U.

V. METHOD

Our goal is to improve the overall performance of blink estimation in the target (test) domain, which has a different distribution than the source (training) domains. To perform a reliable classification operation on an unseen target domain, a feature extractor need to extract domain-invariant features that do not contribute to distinguishing individual domains. To this end, we propose *Blink Estimation with domain Adversarial Training network* (BEAT), a model that can improve performance on unseen target blink datasets. BEAT network is inspired from DANN [57] and AFLAC [48] network. Although Akuzawa et al. [48] argue that domain adversarial training can affect classifier performance where each domain is not independent, we find that domain adversarial training can help actually learn domain invariant features. From our observations, we combine the ideas from DANN and AFLAC in BEAT. As depicted in Figure 7, BEAT consists of a feature extractor, a blink classifier, and a domain discriminator which can be mathematically formulated as

$$f = F\big(I;\theta_f\big) \tag{1}$$

$$\hat{c} = C(f;\theta c) \tag{2}$$

$$\hat{d} = D(f;\theta d) \tag{3}$$

where $I \in \mathbb{R}^{60\times36\times3}$ is an eye image from source domain $\mathcal{S}$; F is the feature extractor with parameter θ_f; C is the blink classifier with θ_c; and D is the domain discriminator with θ_d. The feature extractor model extracts a feature vector $f \in \mathbb{R}^{36}$ that is applied as input to the classifier (C) and the discriminator (D). c represents the ground truth values for the eye condition (0 for open, 1 for closed). $\hat{c} \in [0, 1]$ is the output of the classifier representing the probabilities of blink states for the input eye images. $d = [d_1, \ldots, d_N]$ denotes the domain labels as a one-hot vector and N is the number of domains. $\hat{d} = [\hat{d_1}, \ldots, \hat{d_N}]$ represents the probabilities for each domain to which the image belongs.

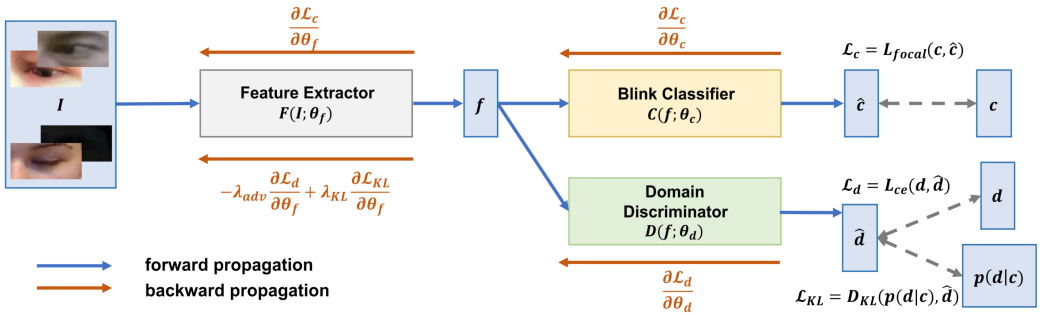

Fig. 7. BEAT network is designed to learn necessary information (i.e., blinks) and to forget unnecessary domain-specific information (i.e., lighting conditions, racial bias, etc.). For this purpose, BEAT network has a main blink classifier branch and a domain discriminator branch. The main blink classifier branch learns how to correctly classify blinks. The domain discriminator branch, on the other hand, first learns how to correctly classify domains, then passes the negative values of the learned gradients and some other values to the feature extractor in backpropagation. As a result, the entire BEAT network can learn domain-independent eye features to robustly classify blinks regardless of the various domains.

A. Optimization

1) Classification Loss: Since the blink datasets are highly imbalanced, we adopt the focal loss

$$\mathcal{L}_c(\theta;\mathcal{S}) = -\mathbb{E}_{(I,c,d)\sim\mathcal{S}}\big[\alpha\big(1-\hat{c}\big)^{\gamma} c\log\hat{c} + (1-\alpha)\big(\hat{c}\big)^{\gamma}(1-c)\log\big(1-\hat{c}\big)\big] \quad (4)$$

where $\hat{c}$ is the classifier output and c is the ground truth classification label. The classifier and the feature extractor minimize this loss to distinguish open eye and closed eye images. We use hyperparmeters $\alpha = 0.5$ and $\gamma = 2$ for all experiments.

2) Adversarial Loss: In order to extract domain invariant features, we apply the domain adversarial training method proposed by [57]. The key idea of domain adversarial training is that the discriminator and the feature extractor play a zero-sum game. The role of the discriminator is to separate features according to different domains. On the contrary, the feature extractor is optimized not to extract domain discriminating features. For the discriminator loss, we adopt the cross entropy loss as

$$\mathcal{L}_d(\theta;\mathcal{S}) = -\mathbb{E}_{(I,c,d)\sim\mathcal{S}}\sum_{i=1}^{N} d_i \log\hat{d}_i \quad (5)$$

where $\hat{d}_i$ represents the probability of each domain, d_i can be 0 or 1, indicating whether it is an image from domain i. The discriminator is trained to distinguish domains well by minimizing Equation (5), and the feature extractor is trained to extract domain invariant features by maximizing the same equation. This allows the model to ignore information not relevant to the main classification task and improve generalization performance.

3) KL-Divergence Loss: Akuzawa et al. [48] have proposed a domain generalization method using the KL-divergence. Inspired by this, we adopt the KL-divergence loss as

$$\mathcal{L}_{KL}(\theta;\mathcal{S}) = \mathbb{E}_{(I,c,d)\sim\mathcal{S}} D_{KL}\Big(p(d|c),\hat{d}\Big) \quad (6)$$

where $p(d|c)$ denotes the conditional probability of domain label d at a given classification label c. Akuzawa et al. [48] have proved that when entropy $H(\hat{d})$ and entropy $H(p(d|c))$ are equal, it is the worst case where the discriminator does not distinguish well between domains. Because the feature extractor should not discriminate domains, it is trained to minimize the KL-divergence loss, reducing the distribution difference between $\hat{d}$ and $p(d|c)$. AFLAC in [48] uses only the KL-divergence loss, but BEAT network linearly combines the adversarial loss and the KL-divergence loss.

4) Objective Functions: The objective functions are

$$\hat{\theta}_c = \arg\min_{\theta_c}(\mathcal{L}_c) \quad (7)$$

$$\hat{\theta}_d = \arg\min_{\theta_d}(\mathcal{L}_d) \quad (8)$$

$$\hat{\theta}_f = \arg\min_{\theta_f}(\mathcal{L}_c - \lambda_{adv}\mathcal{L}_d + \lambda_{KL}\mathcal{L}_{KL}) \quad (9)$$

where λ_{adv} is a hyperparameter for adversarial training and λ_{KL} is for the KL-divergence loss. In Equation (8), the discriminator is optimized to minimize the discriminator loss, while the feature extractor is optimized to maximize the discriminator loss in Equation (9).

B. Feature Extractor Details

The dilated convolution operation [58] has the advantage of extending the receptive field without lowering the resolution of the input image [58], [59]. Chen and Shi [59] have constructed a network (DilatedNet) based on the dilated convolution to extract features robust to eye shape changes in gaze estimation tasks. Inspired by this, we set the DilatedNet as a baseline model for the feature extractor.

The vanilla DilatedNet consists of a convolution stage and a dilated convolution stage. The original convolution stage has four convolution layers. To reduce computation, we modify the first and last convolution layers to depth-wise convolution layers. In the dilated convolution stage, we use four dilated convolution layers identical to the configurations in [59]. Also, we change dilated rate to (2,3), (2,3), (2,4), (2,4) due to the size difference of the input image. See Table III for details.

C. Blink Classifier Details

The blink classifier predicts the blink probability from the extracted features. The architecture details of blink classifier is described in Table IV. Batch normalization, leakyReLU, and dropout layers are added between layers. The last layer is the sigmoid function that calculates the blink probability.

TABLE III
FEATURE EXTRACTOR ARCHITECTURE

	Type	Filter Shape	Stride (dilation rate)	Input Shape $H \times W \times C$
Pre-stage	Conv	$3 \times 3 \times 64$	1	$36 \times 60 \times 3$
Convolution Stage	Depthwise Conv	3×3	1	$36 \times 60 \times 64$
	Conv	$3 \times 3 \times 64$	1	$36 \times 60 \times 64$
	Maxpool	2×2	2	$36 \times 60 \times 64$
	Conv	$3 \times 3 \times 128$	1	$18 \times 30 \times 64$
	Depthwise Conv	3×3	1	$18 \times 30 \times 128$
Pre-stage	Conv	$3 \times 3 \times 64$	1	$18 \times 30 \times 128$
Dilated Convolution Stage	Dilated Conv	$3 \times 3 \times 64$	1 (2,3)	$18 \times 30 \times 64$
	Dilated Conv	$3 \times 3 \times 64$	1 (2,3)	$14 \times 24 \times 64$
	Dilated Conv	$3 \times 3 \times 128$	1 (2,4)	$10 \times 18 \times 64$
	Dilated Conv	$3 \times 3 \times 128$	1 (2,4)	$6 \times 10 \times 128$
	FC layer	512×36	-	$2 \times 2 \times 128$

TABLE IV
CLASSIFIER ARCHITECTURE

Type	Filter Shape	Input Shape
FC layer 1	36×36	$1 \times 1 \times 36$
Batch Normalization	-	-
LeakyReLU	-	-
Dropout	-	-
FC layer 2	36×12	$1 \times 1 \times 36$
Batch Normalization	-	-
LeakyReLU	-	-
Dropout	-	-
FC layer 3	12×1	$1 \times 1 \times 1$
Sigmoid	Activation Function	$1 \times 1 \times 1$

D. Domain Discriminator Details

The domain discriminator predicts which domain the input image is from. A domain's ground truth is labeled in a one-hot vector. The last layer of the discriminator is the softmax function that predicts which domain the image will most likely belong to when there are multiple domains. We have found the optimal values of λ_{adv} and λ_{KL} as 0.01 and 1, respectively, through hyperparameter tuning experiments (see Table XI and XII).

1) Lambda Scheduler (Sc): We use a scheduler for λ_{adv}, based on Ganin et al.'s findings [57] that the scheduler makes the feature extractor less sensitive to noisy datasets during the early training epochs. λ_{adv} is defined as

$$\lambda_{adv}(k) = \lambda_o \left(\frac{2}{1 + \exp(-\sigma k)} - 1 \right) \tag{10}$$

where k denotes the number of epochs. The scheduler changes the λ_{adv} value from 0 to λ_o. As shown in Table XI, the best AUPR are achieved when λ_{adv} is 0.01. We have applied $\sigma = 0.5$ and $\lambda_0 = 0.01$ for the experiments.

2) Gradient Decay (GD) Layer: As we have experimented with different values for λ_{adv} listed in Table XI, we have found that classification performance degrades as λ_{adv} increases. We discover that if the model is trained with excessive discriminative loss, the classification loss plays little role in training, which degrades classification performance. To prevent the discriminative loss from overwhelming the classification loss, we adopt a gradient decay layer that regularizes the gradient values transferred from the discriminator to the feature extractor as

$$\delta_f = t \frac{\exp\left(\frac{\delta_d}{2t}\right) - 1}{\exp\left(\frac{\delta_d}{2t}\right) + 1} \tag{11}$$

where t is a scale factor, δ_f is the gradient of the last layer of the feature extractor, and δ_d is the gradient of the first layer of the discriminator connected to the feature extractor. The gradient decay layer (Equation (11)) converges the gradient (δ_f) to the scale factor $\pm t$ at infinity ($\delta_d \to \pm\infty$). Note that the gradient decay layer is similar to the *tanh* function and the gradient decay layer prevents the gradient values from divergence. Given that the adversarial training tends to be unstable due to unbounded gradients, we believe that our gradient regularization improves the stability of the adversarial training. We use the scale factor $t = 4$ for the experiments.

VI. EXPERIMENTS

We test the ideas discussed so far to choose the best configuration with the smallest classification loss. For our experiments, we use one RTX 2070Ti GPU, taking an average of 5 hours per experiment. During training, the batch size is set to 256 and the Adam optimizer ($\beta_1 = 0.9$, $\beta_2 = 0.999$) is used. We adopt a warm-up cosine annealing scheduler with a learning rate value between 10^{-6} and 10^{-4}. For loss functions, we apply the cross entropy loss for the domain discriminator and the focal loss for the blink classifier with a label smoothing value of 0.1.

A. Interpretation of Experiment Results

The tables in this section summarize the main experiment results measured based on the test dataset shown in Table VI. Since the test dataset is statistically independent from the training and validation datasets, the results presented in this section can be considered as the results of a statistically rigorous case study.

Table X shows the performance comparison of our blink detection method. Each column of Table X lists the results with three training datasets (source domains) and one test dataset (target domain). For example, the title in the second column "$\mathcal{D}_R, \mathcal{D}_B, \mathcal{D}_U \to \mathcal{D}_E$" means that we have trained on the source datasets, $\mathcal{D}_R$, $\mathcal{D}_B$, and $\mathcal{D}_U$, and tested our method on the target dataset, $\mathcal{D}_E$. As you can see in Table X, the three versions of the BEAT network are superior in most cases, except where the UnityEyes dataset D_U is the target domain. However, since the UnityEyes dataset is a synthetic dataset created by a game engine for training purposes only, it is not practical to use this dataset as target or test data.

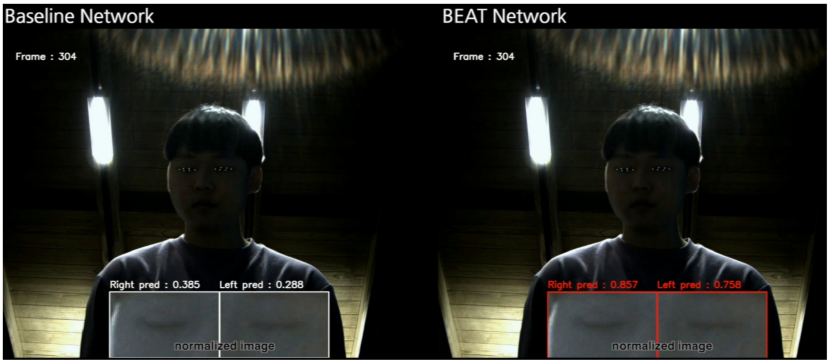

Fig. 8. Screenshot of test results. The baseline network (left) fails to detect blinks, but the BEAT network (right) detects blinks successfully based on dark original images. Normalized eye images shown below are brightened for comparison only and not applied as test or train information. Red image frames around the normalized images indicate successful predictions and white frames represent failures. The numbers above the boxes represent the expected blink probabilities. Check out the supplemental video clip (https://youtu.be/ABjrD6sFB_U) for more details.

TABLE V
BASELINE COMPARISON RESULTS. FPS REFERS TO THE NUMBER OF INFERENCED IMAGE FRAMES PER SECOND

Model	Precision	Recall	AUROC	AUPR	FPS
DenseNet121 + Ensemble Cortacero et al. (2019)	0.664	0.791	NA	0.774	20.5
ResNet18 Al-Hindawi et al. (2022)	NA	NA	**0.993**	0.921	102.3
ResNet50	0.7511	0.7615	0.9789	0.8311	34.02
MobileNetv2	0.8786	0.7928	0.9912	0.9001	38.22
DenseNet121	0.7880	0.8359	0.9883	0.8915	15.16
DenseNet121 + Ensemble	0.8702	**0.8492**	0.9913	0.9277	5.1
DilatedNet [58] [our approach]	**0.9577**	0.8022	0.9890	**0.9355**	**292.9**

TABLE VI
DETAILS OF SUBJECT ALLOCATIONS FOR REAL DATASETS

Domain	Train	Valid	Test
BID	2, 5, 6, 9, 10, 12	1, 4, 8	3, 11, 13
RT-BENE	1, 2, 3, 4, 7, 8, 9, 10	5, 12, 13, 14	0, 11, 15, 16
Eyeblink8	1, 3	10	8

The third column ($\mathcal{D}_R, \mathcal{D}_U, \mathcal{D}_E \rightarrow \mathcal{D}_B$) in Table X shows the performance of our method when our network was trained by those images from good lighting conditions ($\mathcal{D}_R, \mathcal{D}_U, \mathcal{D}_E$) and independently tested for classification of blinks in images from dark lighting condition ($\mathcal{D}_B$). As shown in the third column, our method successfully detects blinks with the highest AUROC and AUPR under dark lighting conditions (Fig. 4 and 8).

B. Baseline Models for Feature Extractor

In order to evaluate and determine the most suitable baseline model of the feature extractor in BEAT, it is necessary to compare different structures based on one common data set. For this purpose, we use the RT-BENE dataset for training and evaluation. Table V lists the comparison results for our DilatedNet-based model [58], ResNet50 [60], MobileNetv2 [61], DenseNet121 [62] and DenseNet121 with ensemble models [10]. For unbiased comparison with other reported methods, pretrained weights from the ImageNet dataset are applied to ResNet, MobileNetv2, and DenseNet121. Table VI shows the RT-BENE subjects that we have assigned for fair comparison with [10].

To measure the precision and recall in the experiments, we set the threshold to 0.5. However, the threshold of 0.5 may not be a proper choice because the datasets are imbalanced. Therefore, we choose the AUROC and the AUPR for comparison, which do not depend on threshold values. As shown in Table V, our design (DilatedNet) scores the highest in the AUPR and achieves the fastest inference speed, which is highly demanded in mobile applications. For more details, our DilatedNet-based design is approximately 28.4% better in precision, 6.38% in recall, and 18.6% in the AUPR and 12.8 times faster than [10]. The model proposed by [12] is also compared equally with ours using only the augmentation method without the curriculum learning for unbiased comparison. As a result, our DilatedNet-based design scores 1.57% higher in the AUPR than [12]. Even though they have used a better GPU (NVIDIA Titan V), our inference speed is 2.86 times faster. Based on the results, we claim that our feature extractor achieves state-of-the-art performance in terms of AUPR and higher inference speed on the RT-BENE dataset.

1) Undersampling Test: To deal with the class imbalance, we have tried the undersampling technique [63] on our feature extractor using a random subset of the majority class at the ratios of 1:1, 1:5, 1:10, and 1:15 for the RT-BENE dataset. Table VII summarizes the results according to the undersampling ratios. Note that the unsampled ratio of closed-eye to open-eye images in the original RT-BENE datasets is 1:23.

As shown in Table VII, recall increases as the sampling ratio approaches to 1:1, and precision increases as the sampling ratio (open to closed) increases. Note that the unsampled case scores highest on both the AUROC and the AUPR. We guess that this is because the model loses some important information that would contribute to classification performance due to the major class samples removed during undersampling.

2) Oversampling Test: We have tried two oversampling methods to increase the samples of the minor class. One is

TABLE VII
UNDERSAMPLING TEST RESULTS IN VARIOUS MEASURES ACCORDING TO SAMPLING RATIOS, 1:1, 1:5, 1:10, 1:15, AND 1:23

Sampling Ratio	Precision	Recall	F1-Score	AUROC	AUPR
1:1	0.3840	**0.9348**	0.5444	0.9791	0.8316
1:5	0.7449	0.8731	0.8039	0.9830	0.9113
1:10	0.6927	0.8881	0.7783	0.9872	0.9157
1:15	0.7988	0.8530	0.8250	0.9853	0.9122
1:23 (unsampled)	**0.9577**	0.8022	**0.8731**	**0.9890**	**0.9355**

TABLE VIII
OVERSAMPLING TEST RESULTS IN VARIOUS MEASURES ACCORDING TO OVERSAMPLING METHODS

Method	Precision	Recall	AUROC	AUPR
No oversampling	0.9577	0.8022	0.9890	0.9355
Data synthesis	0.9108	0.7653	0.9911	0.9152
Transformation	0.7334	0.8158	0.9833	0.8608

to synthesize new samples, and the other is to transform the existing samples in the minor class. The first method utilizes the customized UnityEyes software [50] to generate closed-eye images. Since the original software cannot create closed-eye images, we have reverse-engineered the software. The second method transforms the images in the minor class to produce other images with varying brightness, contrast, translation, scale, rotation, iso-noise, and motion blur.

Table VIII shows the results of the oversampling test using our feature extractor. All oversampled (i.e., augmented) datasets have lower AUPR values than the raw dataset.

3) Optimal Threshold: Through the undersampling and oversampling tests, we have learned that neither approach always helps to improve binary classification performance. One of the important assumptions we have to consider for the tests is that the precision and recall in Table VII and VIII are based on a threshold of 0.5, which may not be suitable for imbalanced classes. Therefore, we define an optimal threshold $\hat{T}$ based on the F1-score for varying sampling ratios in practical blink estimation applications as follows:

$$\begin{aligned}\hat{T} &= \underset{T}{}[F1_{Score}(T)] \\ &= \underset{T}{}\left[\frac{2 \times Precision(T) \times Recall(T)}{Precision(T) + Recall(T)}\right] \end{aligned} \tag{12}$$

where T is a threshold.

We find that the optimal threshold value for the raw RT-BENE dataset is 0.4598 from Equation (12) (see Table IX). The recalculated evaluation metrics based on the optimal threshold for precision, recall, and F1-score are 0.9307, 0.8409, and 0.8835, respectively. As shown in Table IX, the optimal threshold increases the F1-score and reduces the difference between precision and recall.

TABLE IX
OPTIMAL THRESHOLDS, PRECISIONS, RECALLS, AND F1-SCORES ACCORDING TO SAMPLING RATIOS

Sampling ratio	Optimal threshold	Precision	Recall	F1-Score
1:1	0.7852	0.7828	0.7255	0.7531
1:5	0.6884	0.9299	0.7952	0.8573
1:10	0.6848	0.9355	0.7966	0.8605
1:15	0.6052	0.9168	0.7869	0.8469
1:23 (unsampled)	0.4589	0.9307	0.8409	**0.8835**

C. Domain Generalization Performance of BEAT

We evaluate the domain generalization performance of BEAT by selecting one domain dataset as a target domain and other datasets as source domains. We train the BEAT network with adversarial lambda scheduler (BEAT+Sc) and the BEAT network with scheduler and gradient decay layer (BEAT+Sc+GD). For comparison, the baseline network without adversarial training, DANN [57] and AFLAC [48] are also evaluated. Table X shows that the BEAT+Sc+GD network achieves better AUROC and AUPR values than the baseline network for all target and source domain combinations, except when the UnityEyes dataset is the target domain. To be more specific, the BEAT+Sc+GD network improves the AUROC and the AUPR by 2.99% and 50.24% on the Eyeblink8, 7.21% and 4.47% on the BID, and 2.14% and 23.76% on the RT-BENE, respectively.

It is interesting to find that the baseline network scores higher in the AUROC and AUPR than other networks including DANN and AFLAC when the UnityEyes dataset is set as the target domain. The results demonstrate that decision boundary created by RT-BENE, BID, and Eyeblink8 datasets can discriminate between open and closed eye images well, even though there is no domain adversarial training which makes domain invariant features.

From the results, we speculate that, since the UnityEyes images have little noise (and therefore less domain-specific information), the model trained on other (source) datasets can easily detect eye shape features from the (target) UnityEyes dataset without adversarial training. However, the last column (i.e., $\mathcal{D}_R, \mathcal{D}_B, \mathcal{D}_E \rightarrow \mathcal{D}_U$) describes a scenario that seldom happens because $\mathcal{D}_U$ (UnityEyes) is for training, not testing, as discussed.

In summary, as shown in Table X, our BEAT+Sc+GD performs better than other methods, except in rare cases where synthetic datasets are tested as targets.

D. Hyperparameter Tuning

1) Adversarial Parameter: In order to find an optimal value for the adversarial parameter, λ_{adv} in Equation (9), we have trained the BEAT network without applying the KL-divergence loss. In this configuration, the feature extractor considers only classification and adversarial losses and is equivalent to the DANN network. We have tried with $\lambda_{adv} = 0.001, 0.01, 0.1, 1, 10$ under the same conditions depicted in

TABLE X
DOMAIN GENERALIZATION RESULTS FOR COMBINATIONS OF FOUR DATASETS (RT-BENE, UNITYEYES, BID, EYEBLINK8). SC: LAMBDA SCHEDULER, GD: GRADIENT DECAY LAYER, BID: $\mathcal{D}_B$, RT-BENE: $\mathcal{D}_R$, UNITYEYES: $\mathcal{D}_U$, EYEBLINK8: $\mathcal{D}_E$

$\mathcal{S} \rightarrow \mathcal{T}$	$\mathcal{D}_R, \mathcal{D}_B, \mathcal{D}_U \rightarrow \mathcal{D}_E$		$\mathcal{D}_R, \mathcal{D}_U, \mathcal{D}_E \rightarrow \mathcal{D}_B$		$\mathcal{D}_B, \mathcal{D}_U, \mathcal{D}_E \rightarrow \mathcal{D}_R$		$\mathcal{D}_R, \mathcal{D}_B, \mathcal{D}_E \rightarrow \mathcal{D}_U$	
Metrics	AUROC	AUPR	AUROC	AUPR	AUROC	AUPR	AUROC	AUPR
Baseline	0.9669	0.5983	0.8261	0.7587	0.9563	0.6254	0.9901	0.9912
DANN	0.9955	0.8686	0.8490	0.7130	0.9604	0.6885	0.9741	0.9748
AFLAC	0.9955	0.8527	0.8762	0.7793	0.9695	0.7307	**0.9958**	0.9961
BEAT [ours]	0.9953	0.8266	0.8298	0.7190	0.9759	0.7703	0.9957	**0.9966**
BEAT + Sc [ours]	**0.9960**	0.8780	**0.9060**	0.7889	0.9615	0.7508	0.9849	0.9879
BEAT + Sc + GD [ours]	0.9958	**0.8989**	0.8857	**0.7926**	**0.9768**	**0.7740**	0.9743	0.9802

TABLE XI
AUPR VALUES ACHIEVED ACCORDING TO λ_{adv} FOR EACH TARGET DOMAIN: $\mathcal{D}_E$, $\mathcal{D}_B$, $\mathcal{D}_R$, AND $\mathcal{D}_U$

λ_{adv}	$\mathcal{D}_E$	$\mathcal{D}_B$	$\mathcal{D}_R$	$\mathcal{D}_U$	Avg.
10	0.3771	0.7621	0.6086	0.9835	0.6828
1	0.6539	0.7119	0.5906	0.9749	0.7328
0.1	0.7773	0.7023	0.3766	0.9427	0.6997
0.01	0.8686	0.7130	0.6885	0.9748	**0.8112**
0.001	0.8132	0.6897	0.5038	0.9732	0.7450

TABLE XII
AUPR VALUES ACHIEVED ACCORDING TO λ_{KL} FOR EACH TARGET DOMAIN: $\mathcal{D}_E$, $\mathcal{D}_B$, $\mathcal{D}_R$, AND $\mathcal{D}_U$

λ_{KL}	$\mathcal{D}_E$	$\mathcal{D}_B$	$\mathcal{D}_R$	$\mathcal{D}_U$	Avg.
10	0.7842	0.7561	0.7365	0.9893	0.8165
1	0.8527	0.7793	0.7307	0.9961	**0.8397**
0.1	0.8270	0.7567	0.7254	0.9881	0.8243
0.01	0.7957	0.7662	0.7701	0.9947	0.8317

Table I. As a result, we have found that $\lambda_{adv} = 0.01$ performs the best (see Table XI).

2) KL-Divergence Parameter: We also have tested to find an optimal λ_{KL} in Equation (9). To evaluate the effect of λ_{KL} more accurately, we have trained the BEAT network by optimizing the feature extractor without the adversarial loss, using the RT-BENE, BID, and UnityEyes datasets as source domains and the Eyeblink8 dataset as target domain. We have evaluated the performances by changing $\lambda_{KL} = 0.01, 0.1, 1, 10$ and achieved the highest average AUPR with $\lambda_{KL} = 1$ as shown in Table XII.

E. Ablation Study

1) Lambda Scheduler: We have conducted an ablation study for the lambda scheduler (Sc). Note that the results in the sixth row (**BEAT**) of Table X are based on a constant $\lambda_{adv} = 0.01$ without the lambda scheduler. We have evaluated the performances of the lambda scheduler by changing λ_{adv} from 0 to 0.01, which are shown in the seventh row (**BEAT+Sc**). Although the intended purpose of the scheduler is to stabilize gradients, the results show that the lambda scheduler even improves the AUROC and the AUPR in some target domain datasets - by 0.07% and 6.22% on the Eyeblink8, and 9.18% and 9.72% on the BID, respectively. However, the scheduler degenerates the AUROC and the AUPR by 1.48% and 2.53% on the RT-BENE dataset, and 1.08% and 0.87% on the UnityEyes dataset, respectively.

2) Gradient Decay Layer: We also have conducted another ablation test to find the effect of the gradient decay (GD) layer described in Equation (11). As shown in the eighth row (**BEAT+Sc+GD**) of Table X, the BEAT+Sc+GD combination performs better in AUPR than the BEAT+Sc combination by 2.38% in the Eyeblink8, 0.47% in the BID, and 3.09% in the RT-BENE target dataset, except the UnityEyes dataset. The results prove that the gradient decay layer can also help to improve the generalization performance.

VII. CONCLUSION

Our network for Blink Estimation with domain Adversarial Training (BEAT) robustly detects eye blinks on unseen out-of-sample images captured even under poor lighting conditions in a variety of consumer applications. BEAT can generalize various domains by extracting domain-invariant features through adversarial training and the KL divergence loss. We also add a gradient decay layer which regularizes gradients for stable domain adversarial training. Based on the experiments, We conclude that our approach achieves better performances than DANN [57] and AFLAC [48] for unseen target domains.

The proposed feature extractor based on DilatedNet applied to BEAT achieves state-of-the-art performance in terms of AUPR and high inference speed on the RT-BENE dataset. We also experimentally determine the optimal threshold applicable to the RT-BENE [10] dataset.

Based on the improved classification performance and inference efficiency, we believe BEAT is suitable for a wide variety of consumer applications where robust blink detection is required to ensure critical safety even on mobile devices.

REFERENCES

[1] G. E. Hinton and S. Roweis, "Stochastic neighbor embedding," in *Advances in Neural Information Processing Systems*, vol. 15. Cambridge, MA, USA: MIT Press, 2002. [Online]. Available: https://papers.nips.cc/paper/2002/hash/6150ccc6069bea6b5716254057a194ef-Abstract.html

[2] B. S. Perelman, "Detecting deception via eyeblink frequency modulation," *PeerJ*, vol. 2, p. e260, Feb. 2014. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3932793/

[3] L. M. Bergasa, J. Nuevo, M. A. Sotelo, R. Barea, and M. Lopez, "Real-time system for monitoring driver vigilance," *IEEE Trans. Intell. Transp. Syst.*, vol. 7, no. 1, pp. 63–77, Mar. 2006.

[4] G. Pan, L. Sun, Z. Wu, and S. Lao, "Eyeblink-based anti-spoofing in face recognition from a generic Webcamera," in *Proc. IEEE 11th Int. Conf. Comput. Vis.*, Oct. 2007, pp. 1–8.

[5] M. Rosenfield, "Computer vision syndrome: A review of ocular causes and potential treatments," *Ophthalmic Physiol. Opt. J. Brit. College Ophthalmic Opticians*, vol. 31, no. 5, pp. 502–515, Sep. 2011.

[6] Q. Ji and X. Yang, "Real-time eye, gaze, and face pose tracking for monitoring driver vigilance," *Real-Time Imag.*, vol. 8, no. 5, pp. 357–377, 2002. [Online]. Available: https://journals.scholarsportal.info/details/10772014/v08i0005/357_regafptfmdv.xml

[7] J.-D. Wu and T.-R. Chen, "Development of a drowsiness warning system based on the fuzzy logic images analysis," *Expert Syst. Appl.*, vol. 34, no. 2, pp. 1556–1561, Feb. 2008. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417407000401

[8] W. Dong, P. Qu, and J. Han, "Driver fatigue detection based on fuzzy fusion," in *Proc. Chin. Control Decis. Conf.*, Jul. 2008, pp. 2640–2643.

[9] A. Królak and P. Strumillo, "Eye-blink detection system for human–computer interaction," *Universal Access Inf. Soc.*, vol. 11, no. 4, pp. 409–419, Nov. 2012. [Online]. Available: https://doi.org/10.1007/s10209-011-0256-6

[10] K. Cortacero, T. Fischer, and Y. Demiris, "RT-BENE: A dataset and baselines for real-time blink estimation in natural environments," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshop (ICCVW)*, Oct. 2019, pp. 1159–1168.

[11] G. Hu et al., "Towards real-time eyeblink detection in the wild: Dataset, theory and practices," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 2194–2208, 2020.

[12] A. Al-Hindawi, M. Vizcaychipi, and Y. Demiris, "Faster, better blink detection through curriculum learning by augmentation," in *Proc. Symp. Eye Track. Res. Appl.*, 2022, pp. 1–7.

[13] E. Wood and A. Bulling, "EyeTab: Model-based gaze estimation on unmodified tablet computers," in *Proc. Symp. Eye Tracking Res. Appl.*, New York, NY, USA, 2014, pp. 207–210. [Online]. Available: https://doi.org/10.1145/2578153.2578185

[14] Z. Zhu and Q. Ji, "Eye gaze tracking under natural head movements," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 1, 2005, pp. 918–923.

[15] D. H. Yoo and M. J. Chung, "A novel non-intrusive eye gaze estimation using cross-ratio under large head motion," *Comput. Vis. Image Understand.*, vol. 98, no. 1, pp. 25–51, 2005.

[16] J. Kang, D. V. Anderson, and M. H. Hayes, "Face recognition for vehicle personalization with near infrared frame differencing," *IEEE Trans. Consum. Electron.*, vol. 62, no. 3, pp. 316–324, Aug. 2016.

[17] E. M. Aly and E. S. Mohamed, "Effect of infrared radiation on the lens," *Indian J. Ophthalmol.*, vol. 59, no. 2, pp. 97–101, 2011. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3116568/

[18] A. Köksal, C. Dikmetaş, B. Bildik, S. Doğan, D. Atik, and B. Cander, "Exposure to infrared light: Case series," *Eurasian J. Crit. Care*, vol. 1, no. 3, pp. 103–104, 2019.

[19] "Radiation effects on the eye part 1—Optometry today." Yumpu.com. 2014. [Online]. Available: https://www.yumpu.com/en/document/view/25026347/radiation-effects-on-the-eye-part-1-optometry-today

[20] B.-C. Chen, P.-C. Wu, and S.-Y. Chien, "Real-time eye localization, blink detection, and gaze estimation system without infrared illumination," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2015, pp. 715–719.

[21] K. W. Kim, H. G. Hong, G. P. Nam, and K. R. Park, "A study of deep CNN-based classification of open and closed eyes using a visible light camera sensor," *Sensors*, vol. 17, no. 7, p. 1534, Jul. 2017. [Online]. Available: https://www.mdpi.com/1424-8220/17/7/1534

[22] Y. Li, M.-C. Chang, and S. Lyu, "In Ictu Oculi: Exposing AI created fake videos by detecting eye blinking," in *Proc. IEEE Int. Workshop Inf. Forensics Security (WIFS)*, Dec. 2018, pp. 1–7.

[23] A. Fogelton and W. Benesova, "Eye blink completeness detection," *Comput. Vis. Image Understand.*, vols. 176–177, pp. 78–85, Nov. 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S107731421830287X

[24] M. Lalonde, D. Byrns, L. Gagnon, N. Teasdale, and D. Laurendeau, "Real-time eye blink detection with GPU-based SIFT tracking," in *Proc. 4th Can. Conf. Comput. Robot Vis. (CRV)*, May 2007, pp. 481–487.

[25] L. Schillingmann and Y. Nagai, "Yet another gaze detector: An embodied calibration free system for the iCub robot," in *Proc. IEEE-RAS 15th Int. Conf. Humanoid Robots (Humanoids)*, Nov. 2015, pp. 8–13.

[26] C. Dewi, R.-C. Chen, X. Jiang, and H. Yu, "Adjusting eye aspect ratio for strong eye blink detection based on facial landmarks," *PeerJ. Comput. Sci.*, vol. 8, p. e943, Apr. 2022.

[27] T. Soukupová and J. Cech, "Real-time eye blink detection using facial landmarks," in *Proc. 21st Comput. Vis. Winter Workshop*, 2016, pp. 1–8. [Online]. Available: https://www.semanticscholar.org/paper/Real-Time-Eye-Blink-Detection-using-Facial-Soukupov%C3%A1-Cech/4fa1ba3531219ca8c39d8749160faf1a877f2ced

[28] E. R. Anas, P. Henríquez, and B. Matuszewski, "Online eye status detection in the wild with convolutional neural networks," in *Proc. VISIGRAPP*, 2017, pp. 1–8.

[29] R. Sanyal and K. Chakrabarty, "Two stream deep convolutional neural network for eye state recognition and blink detection," in *Proc. 3rd Int. Conf. Electron., Materials Eng. Nano-Technol. (IEMENTech)*, Aug. 2019, pp. 1–8.

[30] V. R. R. Chirra, S. R. Uyyala, and V. K. K. Kolli, "Deep CNN: A machine learning approach for driver drowsiness detection based on eye state," *Rev. d'Intell. Artif.*, vol. 33, no. 6, pp. 461–466, 2019.

[31] V. García, J. S. Sánchez, and R. A. Mollineda, "On the effectiveness of preprocessing methods when dealing with different levels of class imbalance," *Knowl.-Based Syst.*, vol. 25, no. 1, pp. 13–21, Feb. 2012. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0950705111001286

[32] G. Menardi and N. Torelli, "Training and assessing classification rules with imbalanced data," *Data Min. Knowl. Disc.*, vol. 28, no. 1, pp. 92–122, Jan. 2014. [Online]. Available: https://doi.org/10.1007/s10618-012-0295-5

[33] F. Thabtah, S. Hammoud, F. Kamalov, and A. Gonsalves, "Data imbalance in classification: Experimental evaluation," *Inf. Sci.*, vol. 513, pp. 429–441, Mar. 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0020025519310497

[34] C. Seiffert, T. M. Khoshgoftaar, J. Van Hulse, and A. Napolitano, "RUSBoost: A hybrid approach to alleviating class imbalance," *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 40, no. 1, pp. 185–197, Jan. 2010.

[35] G. E. A. P. A. Batista, R. C. Prati, and M. C. Monard, "A study of the behavior of several methods for balancing machine learning training data," *ACM SIGKDD Explorations Newslett.*, vol. 6, no. 1, pp. 20–29, 2004. [Online]. Available: https://doi.org/10.1145/1007730.1007735

[36] M. Buda, A. Maki, and M. Mazurowski, "A systematic study of the class imbalance problem in convolutional neural networks," *Neural Netw. Official J. Int. Neural Netw. Soc.*, vol. 106, pp. 249–259, Oct. 2018.

[37] F. J. Provost, "Machine learning from imbalanced data sets 101," in *Proc. AAAI Workshop Imbalanced Data Sets*, 2008, pp. 1–3.

[38] F. Provost and T. Fawcett, "Analysis and visualization of classifier performance: Comparison under imprecise class and cost distributions," in *Proc. 3rd Int. Conf. Knowl. Disc. Data Min.*, 1997, pp. 43–48.

[39] C. Drummond and R. C. Holte, "C4.5, class imbalance, and cost sensitivity: Why under-sampling beats over-sampling," in *Proc. ICML Workshop Learn. Imbalanced Datasets*, Jan. 2003, pp. 1–8.

[40] K. Zhou, Z. Liu, Y. Qiao, T. Xiang, and C. C. Loy, "Domain generalization: A survey," 2021, *arXiv:2103.02503*.

[41] Y. Li, M. Gong, X. Tian, T. Liu, and D. Tao, "Domain generalization via conditional invariant representations," in *Proc. AAAI Conf. Artif. Intell.*, vol. 32, 2018, pp. 1–9.

[42] Z. Wang, M. Loog, and J. Van Gemert, "Respecting domain relations: Hypothesis invariance for domain generalization," in *Proc. 25th Int. Conf. Pattern Recognit. (ICPR)*, 2021, pp. 9756–9763.

[43] H. Li, Y. Wang, R. Wan, S. Wang, T.-Q. Li, and A. Kot, "Domain generalization for medical imaging classification with linear-dependency regularization," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 3118–3129.

[44] M. M. Rahman, C. Fookes, M. Baktashmotlagh, and S. Sridharan, "Correlation-aware adversarial domain adaptation and generalization," *Pattern Recognit.*, vol. 100, Apr. 2020, Art. no. 107124.

[45] I. Albuquerque, J. Monteiro, M. Darvishi, T. H. Falk, and I. Mitliagkas, "Generalizing to unseen domains via distribution matching," 2019, *arXiv:1911.00804*.

[46] Z. Deng et al., "Representation via representations: Domain generalization via adversarially learned invariant representations," 2020, *arXiv:2006.11478*.

[47] T. Matsuura and T. Harada, "Domain generalization using a mixture of multiple latent domains," in *Proc. AAAI Conf. Artif. Intell.*, vol. 34, 2020, pp. 11749–11756.

[48] K. Akuzawa, Y. Iwasawa, and Y. Matsuo, "Adversarial invariant feature learning with accuracy constraint for domain generalization," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Disc. Databases*, 2019, pp. 315–331.
[49] Q. Xie, Z. Dai, Y. Du, E. Hovy, and G. Neubig, "Controllable invariance through adversarial feature learning," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1–12.
[50] E. Wood, T. Baltrušaitis, L.-P. Morency, P. Robinson, and A. Bulling, "Learning an appearance-based gaze estimator from one million synthesised images," in *Proc. 9th Biennial ACM Symp. Eye Tracking Res. Appl.*, 2016, pp. 131–138. [Online]. Available: https://doi.org/10.1145/2857491.2857492
[51] T. Drutarovsky and A. Fogelton, "Eye blink detection using variance of motion vectors," in *Proc. Comput. Vis. Workshops*, 2015, pp. 436–448.
[52] T. Fischer, H. J. Chang, and Y. Demiris, "RT-GENE: Real-time eye gaze estimation in natural environments," in *Computer Vision* (Lecture Notes in Computer Science), V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds. Cham, Switzerland: Springer, 2018, pp. 339–357.
[53] A. Fogelton and W. Benesova, "Eye blink detection based on motion vectors analysis," *Comput. Vis. Image Understand.*, vol. 148, pp. 23–33, Jul. 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1077314216300054
[54] X. Zhang, Y. Sugano, and A. Bulling, "Revisiting data normalization for appearance-based gaze estimation," in *Proc. ACM Symp. Eye Track. Res. Appl.*, 2018, pp. 1–9. [Online]. Available: https://doi.org/10.1145/3204493.3204548
[55] Q. Wang et al., "Style transformed synthetic images for real world gaze estimation by using residual neural network with embedded personal identities," *Appl. Intell.*, vol. 53, pp. 2026–2041, May 2023. [Online]. Available: https://doi.org/10.1007/s10489-022-03481-9
[56] Y. Kartynnik, A. Ablavatski, I. Grishchenko, and M. Grundmann, "Real-time facial surface geometry from monocular video on mobile GPUs," Jul. 2019, *arXiv:1907.06724*.
[57] Y. Ganin et al., "Domain-adversarial training of neural networks," *J. Mach. Learn. Res.*, vol. 17, no. 1, pp. 1–35, 2016.
[58] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," in *Proc. 4th Int. Conf. Learn. Represent.*, 2016, pp. 1–13.
[59] Z. Chen and B. E. Shi, "Appearance-based gaze estimation using dilated-convolutions," in *Computer Vision* (Lecture Notes in Computer Science), C. Jawahar, H. Li, G. Mori, and K. Schindler, Eds. Cham, Switzerland: Springer, 2019, pp. 309–324.
[60] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 770–778. [Online]. Available: https://www.computer.org/csdl/proceedings-article/cvpr/2016/8851a770/12OmNxvwoXv
[61] A. G. Howard et al., "MobileNets: Efficient convolutional neural networks for mobile vision applications," 2017, *arXiv: 1704.04861*.
[62] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2261–2269.
[63] X.-Y. Liu, J. Wu, and Z.-H. Zhou, "Exploratory undersampling for class-imbalance learning," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 39, no. 2, pp. 539–550, Apr. 2009.

Seonghun Hong is currently pursuing the undergraduate degree with the Department of Electrical and Computer Engineering, Seoul National University. He worked as a Machine Learning Engineer with VisualCamp in 2022. He conducted research with VisualCamp on domain generalization and image processing. His current interests include multimodal learning and building large-scale machine learning systems.

Yonggyu Kim received the B.A. and M.S. degrees in computer science engineering from Koreatech in 2013 and 2019, respectively. He has been a Researcher with the Research and Development Team, VisualCamp since 2021. He designed the progressive occupancy network architecture for 3-D reconstruction for his master's degrees. His current research interest is gaze estimation in computer vision.

Taejung Park received the B.A. and M.S. degrees in electrical and computer engineering from Seoul National University in 1997 and 1999, respectively. After working with two small startup technology businesses in South Korea, he received the Ph.D. degree from the Department of Electrical and Computer Engineering, Seoul National University for 3-D mesh compression in 2006. He has been an Associate Professor with the Department of Cybersecurity/IT Media, Duksung Women's University since 2013. During his sabbatical, he worked as a Technical Advisor with VisualCamp. His current research interests include gaze estimation based on AI and parallel numerical simulation techniques.