EMO: Real-Time Emotion Recognition from Single-Eye Images for Resource-Constrained Eyewear Devices

Hao Wu
National Key Lab for Novel Software Technology, Nanjing University

Jinghao Feng
National Key Lab for Novel Software Technology, Nanjing University

Xuejin Tian
National Key Lab for Novel Software Technology, Nanjing University

Edward Sun
National Key Lab for Novel Software Technology, Nanjing University

Yunxin Liu
Microsoft Research

Bo Dong
Kitware

Fengyuan Xu*
National Key Lab for Novel Software Technology, Nanjing University

Sheng Zhong
National Key Lab for Novel Software Technology, Nanjing University

ABSTRACT

Real-time user emotion recognition is highly desirable for many applications on eyewear devices like smart glasses. However, it is very challenging to enable this capability on such devices due to tightly constrained image contents (only eye-area images available from the on-device eye-tracking camera) and computing resources of the embedded system. In this paper, we propose and develop a novel system called EMO that can recognize, on top of a resource-limited eyewear device, real-time emotions of the user who wears it. Unlike most existing solutions that require whole-face images to recognize emotions, EMO only utilizes the single-eye-area images captured by the eye-tracking camera of the eyewear. To achieve this, we design a customized deep-learning network to effectively extract emotional features from input single-eye images and a personalized feature classifier to accurately identify a user's emotions. EMO also exploits the temporal locality and feature similarity among consecutive video frames of the eye-tracking camera to further reduce the recognition latency and system resource usage. We implement EMO on two hardware platforms and conduct comprehensive experimental evaluations. Our results demonstrate that EMO can continuously recognize seven-type emotions at 12.8 frames per second with a mean accuracy of 72.2%, significantly outperforming the state-of-the-art approach, and consume much fewer system resources.

CCS CONCEPTS

• **Human-centered computing → Ubiquitous and mobile computing**.

*Corresponding author. Email: fengyuan.xu@nju.edu.cn

MobiSys '20, June 15–19, 2020, Toronto, ON, Canada

ACM ISBN 978-1-4503-7954-0/20/06…$15.00
https://doi.org/10.1145/3386901.3388917

KEYWORDS

Deep Learning, Emotion Recognition, Eyewear Devices, Visual Sensing, Single-eye Images

ACM Reference Format:
Hao Wu, Jinghao Feng, Xuejin Tian, Edward Sun, Yunxin Liu, Bo Dong, Fengyuan Xu, and Sheng Zhong. 2020. EMO: Real-Time Emotion Recognition from Single-Eye Images for Resource-Constrained Eyewear Devices. In *The 18th Annual International Conference on Mobile Systems, Applications, and Services (MobiSys '20), June 15–19, 2020, Toronto, ON, Canada.* ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3386901.3388917

1 INTRODUCTION

Wearable devices for eyes (eyewear) have vastly improved in recent years [47, 53, 64, 78, 88]. Many of them, such as smart glasses and Head-Mounted Displays (HMDs) of Virtual Reality (VR) and Augmented Reality (AR), have become popular in the consumer market. For those devices, eye-based interactions [13, 18, 29] are natural to users, e.g., virtually underlining text contents of a manuscript by saccadic eye movements, and recording day-to-day activities with eye-focus tracking for lifelogging. To support eye-based interactions, eye-tracking cameras are already available on commercial eyewear devices such as HTC VIVE Pro Eye [4], FOVE [2] VR headsets, and HoloLens 2 [8]. Furthermore, it is reported that more and more eyewear devices will be equipped with eye-tracking cameras to provide intelligent services [9].

Emotion recognition is highly desirable on eyewear devices. Previous research in the human-computer interaction (HCI) field has shown that emotion recognition can significantly improve the services of mobile devices [12, 21, 22, 46, 60, 89]. Particularly, the emotion expression could be a critical interaction method complementary to existing ones on eyewear devices. For instance, it is more appropriate to issue a device command via emotion expression than voice speaking in quiet public places like the library. Besides, the emotion change also plays a non-replaceable role in many social activities, which are important application scenarios of eyewear devices [50].

However, emotion recognition is a challenging task on eyewear devices for two main reasons. First, the eye-tracking cameras only

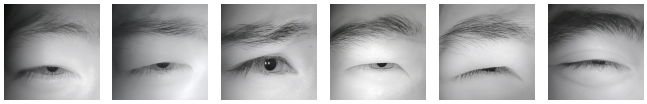

Figure 1: Eye-area images captured by the eye-tracking camera of different people expressing disgust.

capture small eye-area images rather than whole-face images. Although emotion recognition has been studied for many years, most proposed solutions are based on whole-face images [16, 24, 40, 48, 57, 62, 71, 84], and thus are ill-suited for eyewear devices. Eye-area images hold less emotional related facial changes and muscle movements compared to whole-face images, making emotion recognition more challenging. Furthermore, it is also very challenging to distinguish the emotions of different users from eye-area images. As shown in Figure 1, the difference in the same emotion between different users may be greater than the difference in the different emotions of the same user. Thus, a one-size-fits-all strategy is not suitable for the task of eye emotion recognition, unlike in the task of whole-face emotion recognition.

The second challenge is real-time emotion recognition on resource-limited eyewear devices. Emotions are transient psychology states that may last for only a few seconds or less [69]. The proposed solution should be able to quickly recognize the emotion with low latency to avoid missing any emotional changes. Also, eyewear devices usually have very limited resources in terms of computation, memory, storage, and energy. The proposed solution must be lightweight enough to run on typical eyewear devices without specialized hardware. Although there are a few existing solutions that enable effective features on eyewear devices [30, 34, 57, 67], they are of low performance or require extra hardware, impeding their application in real-world settings.

In this paper, we propose and develop EMO, a system to enable real-time emotion recognition on eyewear devices using single-eye images. EMO employs a set of novel techniques to address the challenges above. We take a Deep Learning (DL) based approach with a novel Convolutional Neural Network (CNN) architecture. To effectively extract emotion features from single-eye images, we design an effective emotion *feature extractor* tailored for eye-area images captured by the on-device eye-tracking camera. To deal with the emotion diversity among different users, we design a *personalized classifier* to quickly construct a specialized emotion classifier for each user. Despite lightweight algorithm designs, we also employ two system techniques, the *frame sampler* and the *fast forwarder*, to exploit respectively the temporal locality and the feature similarity among video frames for the sake of accelerating computation and saving resources.

We have built a prototype of EMO on two hardware platforms and implemented the proposed techniques to evaluate the performance of EMO. Our experimental results demonstrate that EMO can effectively (72.2% accuracy) and efficiently (12.8fps) recognize **seven basic emotions**, i.e., anger, disgust, fear, happiness, sadness, surprise, and neutrality, from the video streaming of the eye-tracking camera. Compared with the state-of-the-art approach [34], EMO achieves the 1.82× accuracy, 20.9× speedup, 4.3× memory reduction, and 3.3× energy saving, demonstrating our design is much more suitable for the eyewear scenario.

The main contributions of this paper are as follows:

- We propose and develop the EMO system for real-time emotion recognition from single-eye images, empowering eyewear devices with affective computing ability. (Section 3)
- We design a novel CNN architecture to effectively extract sophisticated features related to emotions from single-eye images. Together with a personalized classifier, EMO achieves high recognition accuracy for different users. The CNN architecture is also designed to be lightweight to run on resource-constrained eyewear devices efficiently. (Section 4)
- We design two system techniques that fully leverage the temporal locality and feature similarity among video frames of the eye-tracking camera to reduce the resource usage for efficient emotion recognition on eyewear devices. (Section 5)
- We build the EMO prototype on real hardware and implement the proposed techniques to evaluate the performance of EMO. Experimental results show that EMO significantly outperforms the state-of-the-art approach in terms of recognition accuracy, speed, and system resource usages like memory footprint and energy consumption. (Sections 6 & 7)

2 EMOTION RECOGNITION FROM EYE-AREA IMAGES

Theoretical study support. Psychology studies [26] show that humans have six basic emotions: fear, surprise, sadness, anger, happiness, and disgust. And the seventh one is neutrality - the lack of expression. These seven basic emotions, consisting of different facial appearance changes, are measurable, discrete, physiologically distinct, and widely accepted [11, 38, 51, 82]. We adopt this emotion categorization system in this paper.

The feasibility of recognizing emotions from eye-area images can be explained by the Emotion Facial Action Coding System (EMFACS) [27, 28]. This system was designed to associate the emotional expressions with visible facial muscle movements (e.g., upper lip raiser) labeled as action units (AUs) [58, 86]. There are seven AUs in the eye-area, i.e., inner brow raiser, cheek raiser, outer brow raiser, lid tightener, brow lowered, nose wrinkler, and upper lid raiser, that can be captured by the eye-tracking camera.

Table 1 shows the number of eye-area AUs that can be captured by the eyewear and the number of other facial AUs of the emotions we adopt in this paper. The number of the eye-area AUs is 65% of the total number of the facial AUs associated with expressing emotions, making it achievable to classify emotions solely from eye-area expressions. For example, the encoding of the sad emotion type includes two AUs in the eye area and one AU in other facial parts. We discover that four out of six emotions have in their encodings more eye-area AUs than other facial-area AUs. Furthermore, all six emotions can be distinguished from each other just by the eye-area AUs of their encodings. Therefore, recognizing emotions from single-eye images is theoretically doable if eye-area AUs can be extracted and recognized well.

Opportunities on eyewear devices. Emotion recognition on eyewear devices also has the following unique advantages.

Table 1: The number of eye-area AUs (N_{eye}) and other facial-area AUs (N_{face}) in the encodings of six emotions.

Emotion	N_{eye}	N_{face}	Emotion	N_{eye}	N_{face}
Sadness	2	1	Fear	5	2
Surprise	3	1	Anger	3	1
Happiness	1	1	Disgust	1	2

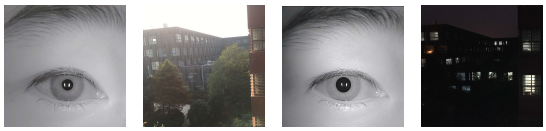

Figure 2: The single-eye images captured by the eye-tracking camera of EMO in two ambient lighting conditions. The left two figures are the eye image captured and its corresponding daylight environment, whereas the right two are for the case of a dark room at night.

Close view of fixed area. Unlike the case of surveillance/CCTV cameras, eye-tracking cameras [2, 6, 7] provide a fixed and close view of single-eye images. Likewise, the objects in the images are predictable: an eye, partial eyebrow, and facial appearance around the eye (please refer to Figure 2). This removes the influence of disturbing factors in recognition, such as face position, obstacles, sharpness in images, and others.

Advantages of infrared (IR) cameras. The eye-tracking cameras on eyewear devices are usually IR cameras, removing the concerns on lighting disruption. We also use an IR camera in our prototype. As is shown in Figure 2, the IR camera can capture clear images under different lighting conditions, and the differences in the images have a negligible impact on recognition.

Temporal locality in video streams. Temporal locality is exploited in deep learning video research to improve process efficiency and save computation resources [17, 80, 81]. We derive two observations from our empirical study in our scenario: 1) two consecutive frames in a real-time video stream of 30 frames per second (fps) are classified with the same emotion in the majority of cases; 2) there is a short interval of stagnation immediately following a change in emotion. We, therefore, exploit these two observations to optimize the system performance of emotion recognition on eyewear devices.

3 SYSTEM DESIGN OVERVIEW

EMO captures the emotional and temporal features in eye-tracking videos and applies deep learning to perform emotion recognition. Captured features are then utilized to improve the accuracy and efficiency of emotion-sensing tasks. EMO has four main system components - a CNN based feature extractor, a feature classifier personalized to each user, a recognition accelerator on the extraction network, and an opportunistic frame sampler to conserve device resources.

Feature extractor. A CNN fine-tuned on eye-tracking images. This CNN is designed to effectively and automatically extract useful emotion-sensing features from single-eye-area images with low computational overhead. It is also generalized to be able to train on imperfect datasets. (Section 4.1)

Personalized classifier. An emotion classifier based on the unique personalized features of each individual. EMO quickly builds this classifier for each user after first use. The classifier leverages personalized emotional features more accurately than a unified classifier for all wearers. (Section 4.2)

Fast forwarder. A CNN-based decision-maker, embedded in the middle of the feature extraction procedure, to assess whether two consecutive inputs are similar enough in terms of emotion distinctive features. If so, this component allows the current input to bypass the rest of the classification processing and assigns it the same label as the previous input. (Section 5.1)

Frame sampler. This frame sampler applied before the feature extractor selectively sends frames to the feature extractor according to network feedback from either the classifier or forwarder component. Its objective is to save computation resources without missing emotion transitions between frames. (Section 5.2)

The system architecture and the main workflow of EMO are illustrated in Figure 3. ❶ The video stream from the eye-tracking camera is sent to the frame sampler. The frame sampler opportunistically decides whether the current input frame shall be processed or not based upon the feedback information from ❼ the fast forwarder or ❽ the personalized classifier that labels the previous frame. If the current frame is selected for the classification pipeline, ❷ it is sent to the feature extractor to extract deep features. As shown in Figure 3, there are two stages of this extractor. After the first stage, ❸ the intermediate results of this frame are sent to the fast forwarder component for assessment. If fast forwarding is triggered after the assessment, ❹ the current frame bypasses the rest of the recognition procedure and is assigned the same emotion label as the most recent fully-processed frame. If fast-forwarding is not triggered for the current frame, ❺ the processing in the feature extractor, i.e., the second stage, is resumed. Finally,❻ the output of extractor, i.e., the deep features of the current frame, is sent to the personalized classifier to label the emotion captured in the current input frame accurately.

Two more procedures are not included in this workflow. The first procedure is EMO's deep-learning training procedure (Section 6.2), which is done before uploading the trained CNN models onto the eyewear device. The second one is the initialization procedure done in the wearer's first use (Section 4.2). This procedure is used to build a local emotion feature map as a reference to the wearer for the personalized classifier. The feature map is unique for each user and useful for improving recognition accuracy.

Next, we describe how the key components of EMO work in detail.

4 EFFECTIVE RECOGNITION

The two key components used by EMO for effective emotion recognition from single-eye images are the feature extractor and the personalized classifier.

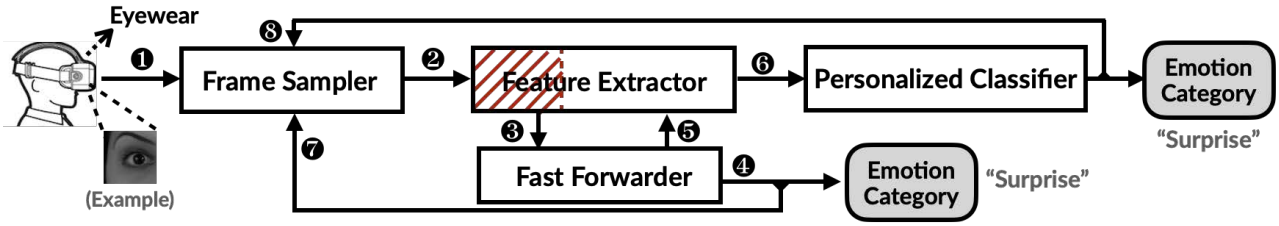

Figure 3: The system architecture and emotion-recognition workflow of EMO.

4.1 Feature Extractor

CNNs have demonstrated a remarkable ability to extract features from images. We take advantages of deep learning and design the EMO feature extractor based on a CNN tailored for single-eye images. To achieve this, we comprehensively evaluate the state-of-the-art CNN models including InceptionV3 [74], VGG16/19 [70], GoogLeNet [73], ResNet18/50 [33], SqueezeNet [37], ShuffleNet [87], and MobileNet [36], in their learning ability and resource usage with regard to our constraints. Our task does not prefer large network models (e.g., VGG16/19 and ResNet50) because of their high resource usage. For the balance between accuracy and resource usage, we choose ResNet18 as our base model, which is better suited for our task than other models according to our experiments [1].

We further improve the base model of ResNet18 according to our observations and derive the CNN model employed in the feature extractor of EMO. The network architecture is shown in Table 1. Several changes are made to tailor ResNet18 for our scenario. First, we observe that eye-area images are usually clear, objects like eyes and eyebrows occupy large areas of the image, and their positions are fixed. Therefore, we shrink the input size of video frames from the conventional 224×224 pixels to 64×64 pixels to reduce computation overhead. Second, we apply a 3×3 kernel and eliminate the pooling layer at the beginning of the network to prevent the feature map from dropping fast for the smaller input size. These modifications may degrade the performance of the feature extraction. To compensate for this issue, we extend the network to 26 layers, which maintains a similar extracting ability with significantly lower usage of the resource. As shown in Table 3, our tailored model achieves 4× faster speeds with an accuracy [2] drop of only 5.3%, compared to ResNet18.

Furthermore, to enable fast-forwarding, we split our CNN model into two stages. The first seven layers represent the first stage (the region with red striped lines in Figure 3), while the rest of the model forms the second stage. The checkpoint for fast forwarding is inserted between these two stages, as described in Section 3.

4.2 Personalized Classifier

Previous studies have shown that personalization is necessary for achieving high accuracy of emotion recognition [34, 50], and it is particularly critical for our task of recognition from single-eye images as described in Section 1. Existing solutions for personalization usually build a dedicated model for every user. These solutions suffer from not only very high model training costs but also insufficient training data for each individual user. Furthermore, they do not work for a new user without any pre-collected data, leading to big deployment issues.

To address those issues, we take a different approach by separating feature extraction from emotion classification. The rationale behind this design is from the observation that features extraction, particularly using CNNs to extract visual features from images, is generic for different inputs. Thus, we train a generic feature extractor (Section 4.1) for all users using all pre-collected training data of different users. Such a feature extractor may be powerful and

Table 2: Network architecture of our CNN model. The kernel size and filter numbers of convolutional layers are shown in the last column. The conv1_x, conv2_x, conv3_x, and conv4_x are built with 3 ResNet blocks [33].

Layer name	Output size	Layers=26
conv1	64×64	$3 \times 3, 16$
conv1_x	64×64	$3 \times \begin{Bmatrix} 3 \times 3, 16 \\ 3 \times 3, 16 \end{Bmatrix}$
conv2_x	32×32	$3 \times \begin{Bmatrix} 3 \times 3, 32 \\ 3 \times 3, 32 \end{Bmatrix}$
conv3_x	16×16	$3 \times \begin{Bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{Bmatrix}$
conv4_x	8×8	$3 \times \begin{Bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{Bmatrix}$
pooling	1×1	average pool, 128-d

Table 3: Accuracy (Acc.) and inferring time (I.T.) of our CNN model and the base model ResNet18. (The inferring time is evaluated on the Qualcomm Open-Q820 [5].)

Network	Layers	Input size	Acc.*	I.T.
ResNet18	18	$224 \times 224 \times 3$	74.4%	751ms
Our CNN	26	$64 \times 64 \times 3$	69.1%	156ms

*Please note that this is not the final accuracy which EMO achieves. See Section 7 for the comprehensive evaluation.

[1] Comparison details are omitted due to space limit.
[2] A fully-connected layer with SoftMax is appended at the end of both networks to measure accuracy.

Algorithm 1: Pseudo code for building the personalized classifier. (Only runs once in the first use.)

Input: Seven sets of frames labeled with different emotions, $\text{frames}_{(i)}$ is the frame set for emotion i; The feature extractor FE.

Output: Reference feature vector for each emotion type i, center_i; The distance between the boundary feature vector and center_i, radius_i.

for *emotion i in 7 emotions* **do**
  $\text{features}_{(i)} = FE(\text{frames}_{(i)})$;
$\text{features}^{\text{clear}}_{(i)} = IsolationForest(\text{features}_{(i)})$;
$\text{center}_{(i)} = KMeans(\text{features}^{\text{clear}}_{(i)})$;
for *emotion i in 7 emotions* **do**
  $\text{radius}_i = \max_{f \in \text{features}^{\text{clear}}_{(i)}} (Similarity(center_i, f))$;
return $\text{center}_{(i)}, \text{radius}_{(i)}$;

effective as it is trained from a large set of data. However, emotion classification may be very different for different users. Therefore, EMO personalizes the classifier, i.e., each user has her classifier. Instead of being pre-trained, the personalized classifier is built in an initialization procedure when the first time the wearer uses EMO. This building procedure only takes about 70 seconds, asking the user to express seven types of emotions for a few seconds each, while the video stream is recorded to build the personalized feature map in the background. This creates a reference feature vector of each emotion for the user. For personalized emotion recognition, we then employ a feature-matching based classifier rather than using a fully-connected layer with SoftMax for classification.

The above procedure is detailed in Algorithm 1. The user is asked by EMO to express all emotions one by one. Each expression is held for five to ten seconds, while the camera records the eye area. Then, EMO applies Isolation Forest [52], an outlier removing algorithm, onto the cluster of video frames recorded for each emotion type. For each emotion feature vector cluster, EMO uses the K-Means method [56] to calculate the reference feature vector, denoted as center_i, for each emotion type i. Next, for each emotion i, EMO calculates the similarity, using cosine distance, between the boundary feature and center_i, denoted as radius_i, which characterizes the range of corresponding emotion feature vectors. Finally, EMO stores the center_i and the radius_i on the device.

Please note that the above initialization procedure of personalization only occurs once when the eyewear device is used for the first time, and its duration is quite short. Thus, it will not disrupt the normal usage of this device. However, the benefit brought by personalization is substantial as it makes the recognition optimized for the specific user of this device. The following describes the workflow of EMO in use (after the one-time initialization).

During the emotion recognition of EMO, the incoming emotion feature vector, extracted by the feature extractor, will be measured with all seven reference vectors with the similarity metric *sim*. For the frame to be recognized, denote as frame^{u}, S_i is defined as:

$$S_i = \frac{sim(FE(\text{frame}^{\text{u}}), \text{center}_i)}{\text{radius}_i}, i \in \{7 \text{ emotions}\} \quad (1)$$

where FE represents the feature extractor, and *sim* is the cosine distance between two feature vectors. In the end, a frame is labeled as the emotion type i giving the maximum S_i.

Our model design, which explicitly combines the generality of feature extraction and the personalization of feature classification, address two unique challenges in EMO. The first one is that eye-area images contain less emotion-related facial changes and muscle movements compared to whole-face images. The second one is that the difference in the same emotion of different users may be greater than the difference in the different emotions of the same user. As a result, EMO achieves a 1.82× improvement in accuracy compared with the state-of-the-art approach (Section 7).

5 EFFICIENT RECOGNITION

EMO achieves efficient recognition through two key system components, the fast forwarder and the frame sampler, both of which leverage the temporal characteristics of live eye-tracking videos.

5.1 Fast Forwarder

The fast forwarder decides if the final classification result of a frame can be predicted at an earlier stage of the feature extraction, allowing it to bypass the rest of the operations to improve the speed significantly.

Temporal characteristics of live eye-tracking videos are the core of the fast forwarder. As mentioned in Section 2, consecutive frames may be similar, especially in the case of eye emotional expressions, thus allowing frames to bypass the bulk of the computations and re-use the emotion label of the previous frame.

The simple pixel-level comparison methods do not work to measure the similarity between consecutive frames, because the similarity targets at the semantic level. We need a method that can extract and compare high-level emotional features with a small cost. To do it, we chose to use a Siamese network [15]. The Siamese network is to learn a similarity metric from data, and it can automatically select the appropriate features with emotion recognition semantics to achieve fast forwarding in our scenario. The Siamese network is first proposed to measure the distance between two feature vectors extracted from handwriting signatures. Since then, the same network has been applied to measure the similarity between feature maps in many computer vision tasks such as face recognition [19], human re-identification [20, 59, 76, 77] and tracking [45], object tracking [14, 32, 75], etc. EMO leverages the idea of Siamese network and designs a customized network to measure the similarity between current and previous frame labeled by the personalized classifier to decide whether or not to make an early prediction.

Siamese network design. As shown in Table 4, we design our Siamese network with 10 convolution layers, 1 pooling layer, and a 128-dimension feature vector as output. To further reduce the computation, we design the first seven layers of the Siamese network structure to be the same as the CNN of the feature extractor. These shared network parameters are critical for the training procedure later described in this section.

The complete design of the fast forwarder is shown in Figure 4. For an input frame, it is first sampled by the frame sampler before

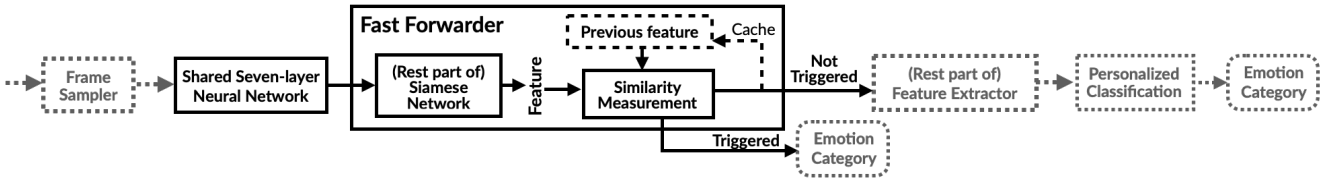

Figure 4: Design of the fast forwarder. The Siamese Network and Feature Extractor share a seven-layer neural network. Once triggered, the fast forwarder will output the emotion category immediately and bypass the rest of the Feature Extractor. Table 4 shows the architecture of the Siamese Network.

Table 4: The Siamese network architecture. The kernel size and filter numbers of convolutional layers are shown in the last column. The first seven layers, i.e., conv1 and conv1_x, are the same as the feature extractor.

Layer name	Output size	Layers=11
conv1	64×64	$3 \times 3, 16$
conv1_x	64×64	$3 \times \begin{Bmatrix} 3 \times 3, 16 \\ 3 \times 3, 16 \end{Bmatrix}$
conv2_x	32×32	$3 \times 3, 32$
conv3_x	16×16	$3 \times 3, 64$
conv4_x	8×8	$3 \times 3, 128$
pooling	1×1	average pool, 128-d

going through the seven-layer CNN feature extraction in the fast-forwarder. The fast forwarder is responsible for testing the emotion-feature similarity of this frame with the most recent feature that went through the full recognition workflow, for deciding whether this frame should resume the paused processing in the extractor's CNN or be directly assigned the previous label. The inputs of the fast forwarder are the intermediate outputs from the 7th layer of the CNN in the feature extractor. To measure the similarity, the output of the Siamese network for each frame was cached.

A threshold θ_{FF} is used to decide whether the fast forwarding will be triggered. θ_{FF} is calculated in Equation 2, when EMO collects the user emotions as described in section 4.2:

$$\theta_{\text{FF}} = \alpha \times \mathbb{E}_{i \in \{0,1,\ldots,L-1\}} sim(SN(\text{frame}_i),\ SN(\text{frame}_{i+1})) \quad (2)$$

where SN is the Siamese network. L is the total number of video frames collected for personalization. frame_i represents the i-th frame in the video. sim is the cosine distance. α is a hyperparameter that weights accuracy and efficiency. A higher α leads the fast forwarder to be triggered more easily, resulting in performance gains, but can also cause expressions to be missed when the changes are very gradual. Possible values of α are explored in Section 7.3.1, and an α of 0.75 is used in our implementation.

Siamese network training. The loss function of our Siamese network is defined as a contrastive loss:

$$\text{Loss} = \frac{1}{2} \times (y(1-s)^2 + (1-y)max(\text{margin} - (1-s), 0)^2) \quad (3)$$

where y is 1 if it is a positive sample and 0 otherwise. s is the cosine similarity of the feature vectors extracted by the Siamese network.

Algorithm 2: Pseudo code of frame sampling algorithm.

Input: Upper bound of sampling interval, Λ; Lower bound of sampling interval, λ; The similarity computed by the fast forwarder, S_{FF};

Output: The number of frames to skip, N_{skip};

$N_{\text{skip}} = 0$;
if *the last emotion is labeled by the fast forwarder* **then**
  $N_{\text{skip}} = \lceil (\Lambda - \lambda) \times \frac{S_{\text{FF}} - \theta_{\text{FF}}}{1 - \theta_{\text{FF}}} + \lambda \rceil$;
else
  $N_{\text{skip}} = (the\ emotion\ changed)?\ \Lambda\ :\ \lambda$;
return N_{skip};

margin is the hyperparameter of similarity. The feature distances of positive pairs will be the loss terms, while the negative pairs try to maximize feature distance until larger than the *margin*. We set *margin* to 5.0 in our training.

Recall that our Siamese network shares the first seven layers, including network structure and parameters, with the CNN of the feature extractor. Therefore, we perform joint training of the Siamese network and CNN to coordinate the trained parameters of their first seven layers, as illustrated in Figure 6. More training details are described in Section 6.2.

5.2 Frame Sampler

The frame sampler predicts which video frames can be skipped for recognition with little influence on the recognition functionality by using the feedback collected from both the fast forwarder and personalized classifier. Two observations inspire the design of the frame-sampling algorithm. First, the similarity measured by the fast forwarder reflects short-lasting emotional stability. Second, expressions typically last at least for 750 milliseconds [69], which means more than 22 consecutive frames for a frame rate of 30fps. In other words, in most common use cases and practical settings, there is a short time window in which emotions remain constant. For some emotions like neutrality, the time window may be much longer. Only when the similarity of two adjacent features is low, the expression is likely to be changing or has changed. Also, when the expression just changed, it can be assumed that the expression will stay consistent. These are the basis for Algorithm 2.

The frame sampler takes the last recognition feedback as input, including whether the last recognition is completed by the personalized classifier or the fast forwarder; the similarity calculated

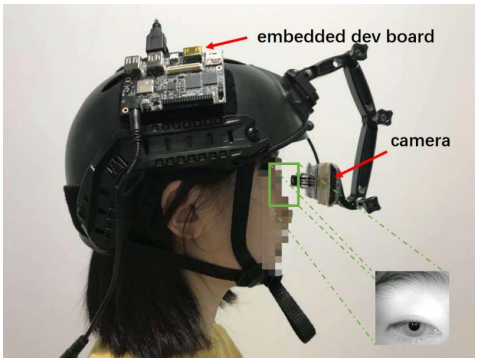

Figure 5: System prototype of EMO.

by the fast forwarder (S_{FF}); and whether the emotion has changed compared with the last recognized frame. θ_{FF} is the threshold for triggering the forwarder defined in Equation 2. Users can set the upper bound (Λ) and the lower bound (λ) of the sampling interval. The upper bound of the sampling interval is to avoid skipping too many frames and missing some ephemeral emotion. Since emotions usually last at least 750ms, we can set the upper bound to any values larger than 750ms. Therefore, any user emotion can hardly be missed by EMO, and recognition accuracy and recall are not affected. The smaller the value is, the more sensitive EMO is to emotion changes, i.e., the new emotions can be detected more quickly. With a 30fps frame rate on our EMO prototype, Λ is set to 10 (frames). The lower bound λ weighs the recognition sensitivity and performance, which can be configured by users according to their needs. A larger λ can save more CPU usage time, while a smaller λ can be used to capture the expression changes in greater detail. More details are discussed in Section 7.3.2.

The output of the sampling algorithm is the number of frames to skip (N_{skip}). If a frame is similar to the previous one and triggers the fast forwarder, the frame sampler will determine N_{skip} according to the similarity between the two frames (reflected by A and B). The larger S_{FF} is, the more stable the emotion will be, the less likely it will be to change instantly, and the larger the sampling interval will be, and vice versa. The sampler divides when the fast forwarder fails to trigger in two cases. If the emotion has not changed since the last recognition, it believes that the emotion is in a rapid change, and a lower N_{skip} will be set to capture the immediate emotion on change. If the emotion changes, the sampler assumes the emotion will remain constant for a brief window of time and set a larger N_{skip}.

6 IMPLEMENTATION

Next, we describe our prototype implementation of EMO and the training details of the two deep learning models: the CNN in the feature extractor and the Siamese network in the fast forwarder.

6.1 System Prototype

Off-the-shelf wearable devices of eye-tracking provide limited high-level APIs, and their software is hard to re-program. To facilitate the development, evaluation, and demonstration of the EMO system, we built a custom eye-tracking platform, as shown in Figure 5. Although the size of the prototype is large, the hardware capabilities and software functionalities are similar to commercial wearable eye-tracking systems.

The hardware components of the EMO prototype include a power supply, an SoC board, an infrared camera, and some cables. We built two versions of the EMO prototype, each with a different SoC board. One is the Qualcomm Open-Q820 [5], whereas the other is the HiKey 620 (LeMaker version) [3]. For each prototype, the SoC board is installed onto a helmet and connected to an infrared camera for eye tracking. This camera has a 3.6mm focal length and is held by a front mount in an eye-facing position. The camera is shown in Figure 5. Both prototypes are powered by a portable 185Wh battery.

As to software, EMO runs as an Android App on the SoC boards with Android 6.0.1 installed. The infrared camera driver we used is libuvccamera 1.1 [1]. The eye-tracking videos captured by the infrared camera are fed into the emotion-recognition workflow, as shown in Figure 3. The feature extractor and the personalized classifier are implemented in C++ through TensorFlow 1.12 [10]. The faster forwarder and the frame sampler are developed in Java.

6.2 Model Training

The model training in EMO is different in two aspects compared to the standard deep learning training procedure. Firstly, we introduce a branch network, the Siamese, to CNN, requiring joint training to achieve optimal efficiency and effectiveness. Joint training preserves the consistency of feature extraction in the shared first segment of the CNN and fast forwarder.

Secondly, we perform a two-stage fine-tuning training technique to accommodate for the lack of training data. The training dataset is generated by deriving eye-area images from preexisting facial expression datasets. Unfortunately, few images in the labeled whole-face datasets can be translated to generate proper single-eye-area images due to issues with facial orientation and distance. Although the targeted dataset of eye-area images is small, the two-stage fine-tuning can help address this problem. The training steps are highlighted below.

Phase I: pre-training. In this stage, only the top portion (the blue dashed box) of the network in Figure 6 is trained. The Siamese network is not involved. The whole-face emotion recognition dataset we use is FER2013 [31]. FER2013 is a facial expression recognition dataset containing 35,887 images of facial expressions and is labeled with the same six emotions as we consider in this work.

In training, we use the joint network architecture shown in Figure 6. The CNN is trained with a classifier that exploits back-propagation to guide the network to learn feature representations that optimize end-to-end emotion recognition. This generic classifier, as shown in Figure 6, is a simple fully-connected layer with a SoftMax layer that is appended to the end of CNN during training. The cross entropy loss function is defined as:

$$L_1 = -\sum_{k=1}^{K} log(p(k))q(k), \quad for\ label\ k \in \{1...K\}, \tag{4}$$

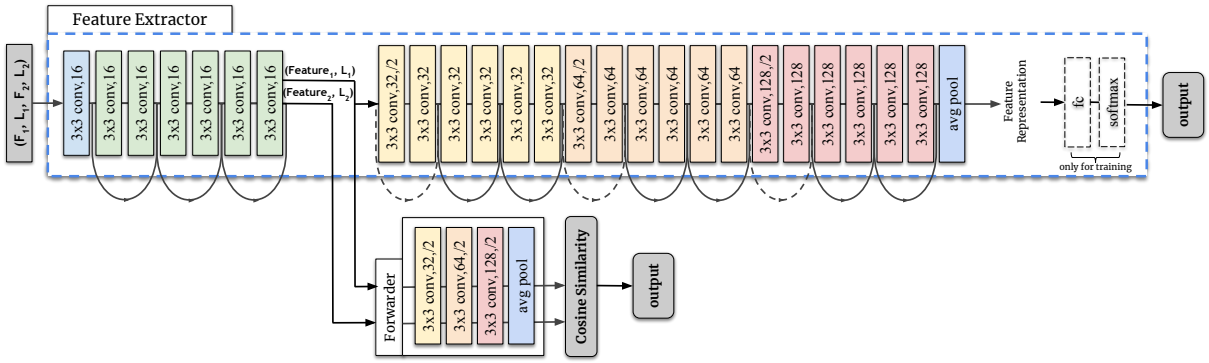

Figure 6: The training architecture contains the recognition part and the fast forwarder. The training procedure is divided into two stages: pre-training and fine-tuning. Only the feature extractor (the blue dashed box) is involved in pre-training. See Section 6.2 for more details.

Figure 7: Seven emotional expressions of the same subject of the original MUG Facial Expression Database (top row), our fine-tuning data cropped from MUG (second row), and the eye-area images captured by EMO during use (third row).

where $p(k)$ is the output of SoftMax for class k, and $q(k)$ stands for the ground-truth of this input. The error measured by the cross entropy after classification is backpropagated to CNN to adjust network parameters. Note that this generic classifier used for feature training is later replaced with EMO's personalized classifier in deployment.

Phase II: fine-tuning. In this stage, the Siamese network is sent to train with the CNN portion is in the fine-tuning process for single-eye-area emotion recognition. The first step to the joint training process is the preparation of the fine-tuning dataset.

Figure 7 depicts example training data; the first row is the MUG Facial Expression dataset [11]; the second row is the modified eye-area images cropped from the original dataset. We use the modified MUG dataset (the cropped single-eye images) consisting of 5,164 images, to fine-tune the pre-trained CNN and train the linked Siamese network from scratch in parallel. The Siamese network is untrained because it is used for similarity mapping and does not require the higher-level facial expression feature understanding. Specifically, we generate a tetrad (F_1, L_1, F_2, L_2), where F_1 and F_2 are two eye-area images in the modified MUG dataset with the labels L_1 and L_2, respectively. For each input, we randomly select two images from one subject; these can be the same emotion or two distinct ones. For each round of training, the shared layers produce the immediate results $Feature_1$ and $Feature_2$ for F_1 and F_2, respectively. The rest of the feature extractor takes the $Feature_1$ and L_1 to update the whole extractor (containing the first shared 7 layers). The Siamese model takes $Feature_1$, $Feature_2$, L_1 and L_2 as input and updates itself (including the shared 7 layers). The re-selection of the tetrad and the update of the two networks are conducted alternately for multiple iterations until the two models converge.

7 EVALUATION

We evaluate the performance of EMO in terms of the *recognition performance* and the *system performance* and measure the *resource usage* on both hardware prototypes of EMO. We also demonstrate the *advantages of continuous recognition.*

7.1 Experimental Setup

All data used in the evaluation are collected from experiment volunteers, which is separate from our training dataset (See Figure 7, for example, images). In this way, we show the robustness of EMO under strict practical situations, as well as the effectiveness of personalization. 20 volunteers, 6 females and 14 males with diverse facial characteristics, participated in our evaluation. Each of the first wear the EMO prototype and experience a 70-second initialization stage (10 seconds for each emotion). Then she or he is asked to watch video clips in FilmStim [66] and let EMO perform real-time emotion recognition. FilmStim is one of the most widely-used emotional movie databases to induce emotions. The videos in FilmStim are labeled by recording how the participants felt at the specific time they were watching the video other than the content itself. We collected our dataset following the same method, and thus we could collect the real emotions of the participants. After watching, the volunteer is asked to label ground truth emotions at frame-level granularity.

We also extract 39,780 frames from the video clips above for measuring the accuracy of the feature extractor and the personalization classifier. These frames are randomly selected from 20 volunteers, each with approximately 285 images per emotion. We randomly selected the frames of 15 people (denote as $\text{Image}^{15}_{\text{EMO}}$) for fine-tuning

Table 5: 4-emotion recognition results of EMO and Eyemotion (Eye. for short). (Support stands for the number of images. Accuracy is the ratio of correct predictions to total predictions made. F1-Score is the harmonic mean of precision and recall.)

Emotion	Accuracy		F1-Score		Support
	EMO	Eye.	EMO	Eye.	
Anger	82.8%	63.5%	0.864	0.594	1,180
Happiness	79.5%	46.6%	0.792	0.496	1,379
Surprise	67.1%	72.7%	0.732	0.769	1,317
Neutral	97.8%	68.3%	0.845	0.649	1,446
Avg/total	81.8%	62.6%	0.808	0.627	5,322

(Section 6.2) and the frames of the remaining 5 people (denote as $\text{Image}^5_{\text{EMO}}$) for testing.

Note that the way we collect the dataset is completely different from Eyemotion [34]. Our dataset is generated by volunteers induced by video clips in FilmStim. However, the dataset of Eyemotion is generated by asking volunteers to perform according to an exemplar video. Since we observe that each user expresses her/his emotions differently (Figure 1), we believe that our method is more accurate to collect the real emotions of the participants.

7.2 Recognition Performance

In this section, we evaluate the recognition effectiveness of EMO proposed in Section 4, i.e., the combined performance of the feature extractor and personalized classifier. We first evaluate the overall recognition accuracy and then demonstrate the importance of personalization in the field of eyewear.

7.2.1 Recognition Accuracy. We compare EMO's recognition accuracy and F1-score [3] with that of Eyemotion [34], a state-of-the-art method for eye-area emotion recognition. We reproduced the model of Eyemotion according to descriptions in the paper. Since the training data of Eyemotion is not available, we fine-tuned the reproduced model using the same dataset of EMO[4]. Eyemotion classifies 5 facial expressions: happiness, anger, surprise, closed eyes, and neutral. EMO performs expression recognition on 7 emotions, happiness, anger, surprise sadness, disgust, fear, and neutral, which are widely used in facial expression recognition [26]. As *closed eyes* is not widely treated as one of the 7 basic emotion types, we excluded it from our evaluation and reported only the results on the 4 shared emotion types and all 7 emotion types.

Eyemotion also uses a two-stage training method with pre-training done on the ImageNet dataset. [23]. EMO is pre-trained on FER2013, which we believe is more applicable to the select field of emotion recognition.

For 4-emotion recognition, all images labeled with the corresponding emotions from $\text{Image}^{15}_{\text{EMO}}$ are selected and applied to fine-tune both models for a fair comparison. The two models are then evaluated on $\text{Image}^5_{\text{EMO}}$. All images used by Eyemotion for fine-tuning and evaluation are personalized, as claimed in the paper. As shown in Table 5, EMO significantly outperforms Eyemotion with an accuracy of 81.8% and an F1-score of 0.808, while the corresponding numbers in Eyemotion are 62.1% and 0.627 [5], respectively.

For 7-emotion recognition, EMO is fine-tuned on two datasets: besides the modified MUG (denoted as EMO^1), we also fine-tuned it using $\text{Image}^{15}_{\text{EMO}}$ (denoted as EMO^2) to study its performance on a small dataset for fine-tuning. Eyemotion is fine-tuned on $\text{Image}^{15}_{\text{EMO}}$. Then, both models are evaluated on $\text{Image}^5_{\text{EMO}}$. Results are shown in Table 6. The accuracy of EMO slightly decreases due to the increased number of emotion types, with a mean accuracy of 75.1% for EMO^2 and 76.6% for EMO^1. However, the performance of Eyemotion becomes significantly worse, with an accuracy of only 42.1%,

These results show that EMO can achieve a high recognition performance. In comparison to Eyemotion, the accuracy of the 4-emotion recognition is 1.32× higher, and the accuracy of the 7-emotion recognition is 1.78× to 1.82× better. Keep in mind that the recognition accuracy of EMO^2 has room for improvement because of the lack of a dedicated large eye-area emotional expression dataset. Even in this case, our models can achieve a high testing accuracy on live eye-tracking videos of users not included in the training dataset.

We believe that, compared with Eyemotion, the EMO's better performance is due to our more effective personalization method. In short, Eyemotion assumes that people express their emotions in similar ways, so it takes the input image and subtracts the average picture of the neutral emotion of the same user for personalization, which does not consider the difference of emotion expressions across different users. On the contrary, we observed that people might express the same expression very differently, and sometimes the difference between the same emotion for two people maybe even larger than the difference between two emotions of the same person. Next, we further evaluate the advantages of our personalization method.

7.2.2 Personalization in Classifier. There are two main advantages brought by our personalized classifier, one is the higher accuracy (see Table 6), and the other is the better generalization, which is useful for training emotion recognition networks without large dedicated eye-area emotion datasets.

To demonstrate it, we compare the accuracy of EMO with and without personalization. Both are fine-tuned using the $\text{Image}^{15}_{\text{EMO}}$ dataset and the modified MUG dataset, and are then evaluated on the 7-emotion $\text{Image}^5_{\text{EMO}}$ dataset. The results can be found in Table 7. From Row 1 and Row 2, the accuracy of EMO with the personalized classifier is 1.61× higher than the EMO with the generic classifier. The difference between Row 3 and Row 4 is even more significant: the accuracy of EMO with the personalized classifier fine-tuned on the modified MUG is 2.09× higher than that with the

[3] We follow the most widely used definitions of accuracy and F1-score as described in [79].

[4] Please note that the training procedure of Eyemotion also consists of two phases, i.e., pre-training and fine-tuning. The pre-training phase uses ImageNet, and the fine-tuning phase uses a self-collected dataset.

[5] The numbers reported in the Eyemotion paper are higher: 74% for accuracy and 0.73 for F1-score. The difference may be caused by 1) Eyemotion reports a high precision (up to 90%) for closed-eyes expression, which is not widely considered as an emotion type and thus not included in this paper; 2) the evaluation dataset is different. However, it is hard to reproduce the results reported in Eyemotion as its model and dataset are not released.

Table 6: Recognition results of EMO and Eyemotion (Eye. for short) on 7 emotions. EMO^1 represents EMO fine tuned on the collected images, and the EMO^2 represents EMO fine tuned on the modified MUG. (Support stands for the number of images.)

Emotion	Accuracy			F1-Score			Support
	EMO^1	EMO^2	Eye.	EMO^1	EMO^2	Eye.	
Anger	84.4%	84.6%	24.8%	0.837	0.870	0.308	1,180
Disgust	69.5%	81.1%	26.6%	0.745	0.863	0.357	1,351
Fear	60.9%	58.4%	17.6%	0.695	0.650	0.221	1,320
Happiness	79.5%	70.0%	32.5%	0.719	0.658	0.327	1,379
Sadness	72.5%	59.2%	37.0%	0.794	0.674	0.486	1,407
Surprise	74.6%	73.6%	88.2%	0.803	0.716	0.741	1,317
Neutral	94.8%	96.9%	65.4%	0.736	0.702	0.396	1,446
Avg/total	76.6%	75.1%	42.1%	0.761	0.627	0.407	9,400

Table 7: 7-emotion recognition accuracy of EMO fine-tuned on different datasets with different classifiers. The general classifier consists of a fully connected layer and a Softmax layer used for training.

ID	Classifier	Fine-tune Dataset	Test Dataset	Acc.
1	General	$\text{Image}^{15}_{\text{EMO}}$	$\text{Image}^{5}_{\text{EMO}}$	47.4%
2	Personalized	$\text{Image}^{15}_{\text{EMO}}$	$\text{Image}^{5}_{\text{EMO}}$	76.6%
3	General	Modified MUG	$\text{Image}^{5}_{\text{EMO}}$	35.9%
4	Personalized	Modified MUG	$\text{Image}^{5}_{\text{EMO}}$	75.1%

generic classifier. These results show that personalized classifier can significantly improve recognition accuracy.

From Row 2 and Row 4 of Table 7, we can see the effective generalization brought by our personalized classifier. As depicted in the second and third rows of Figure 7, there are significant differences between MUG and $\text{Image}^{5}_{\text{EMO}}$ datasets, which explain the poor performance of the MUG fine-tuned general classifier on $\text{Image}^{5}_{\text{EMO}}$. After the introduction of the personalized classifier, our personalized model fine-tuned on modified MUG can almost achieve the same accuracy as the one fine-tuned on $\text{Image}^{15}_{\text{EMO}}$ (75.1% vs. 76.6%). The effective generalization ability of our personalized classifier allows for training a high-performance model without requiring sizeable eye-area emotional expression datasets.

7.3 System Performance

In this part, we evaluate the performance of the fast forwarder and the frame sampler, which relate to the recognition efficiency of EMO. Additionally, we show how to determine the values of hyper-parameters used in the fast forwarder and the sampler based on experimental results.

7.3.1 Fast Forwarding. To study the tradeoff between system resources saved by the fast forwarder and the achievable recognition accuracy, we first measured the time cost of each part of EMO on Open-Q820 and Hikey platforms. The inference time for each part of the model is shown in Table 8. When the forwarder is triggered, the current recognition exits early. Through this mechanism, the forwarder can save 57.1% of the time on Open-Q820, and 58.1% on Hikey. When not triggered, the forwarder spends an extra 10.7% of the time on Open-Q820, and 19.1% on Hikey. The more frequently the forwarder is triggered, the more computational resources will be saved. But it comes at the cost of immediate emotional change detection, which results in a decrease of overall recognition accuracy. We explore the relationship among resources, accuracy, and trigger frequency to find the optimal performance.

Table 8: Inference time of the different parts of the network. Layers_{shared} is the first 7 convolutional layers shared by the extractor and the Siamese network, the Extractor_{rest} is the rest of the extractor combined with the personalized classifier.

Board	Layers_{shared}	Forwarder	Extractor_{rest}
Open-Q820	45ms	15ms	95ms
Hikey	68ms	57ms	230ms

Recall that the trigger frequency of the fast forwarder depends on θ_{FF} (Equation 2). The larger the value, the less likely it is to trigger, and vice versa. θ_{FF} varies from person to person but is regulated by the hyper-parameter α. We test different α values, from 0.5 to 2 with a step-size of 0.25, in processing the eye-tracking video streams (described in Section 7.1). The labels of all frames are acquired and compared with the labels outputs from EMO without using the fast forwarder. These original labels are used to measure the number of incorrectly classified frames caused by the fast forwarder. The percentage of incorrectly recognized frames is referenced as accuracy degradation.

The relationship among the accuracy degradation, the trigger frequency, and α is shown in Figure 8. When α increases, it saves more computational resources but also leads to higher accuracy degradation. To determine the value of $alpha$, we propose a metric, *Performance Price*, which is defined as the percent accuracy loss per 1% performance increase.

The lowest Performance Price is reached when α is 0.75. With this value, the accuracy degradation is less than 5%, maintaining an overall accuracy of 72.2% and a forwarding trigger frequency up to

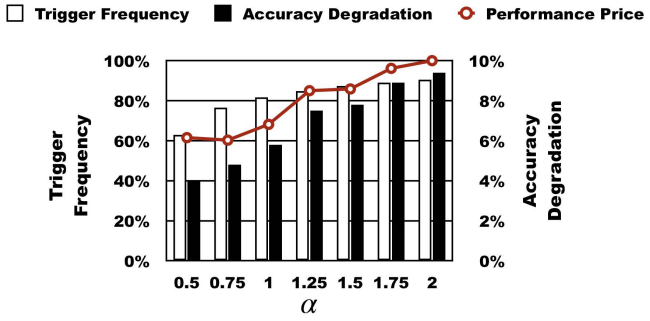

Figure 8: As α increases, trigger frequency, accuracy degradation, and performance price increases. Performance price is the tradeoff in accuracy per 1% performance improvement.

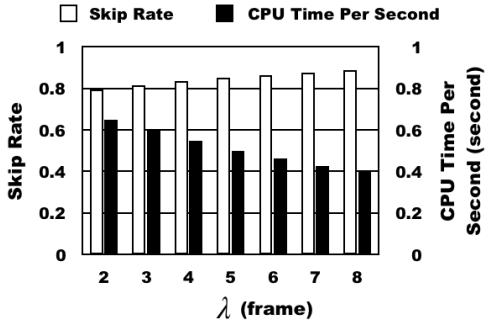

Figure 9: Skip rate has a positive correlation with λ. The higher the skip rate, the shorter the CPU time needed.

77%. The EMO prototype on Open-Q820 can reach 12.8fps, 1.8× the frame rate without fast forwarding. To compare, Eyemotion runs at only 0.58fps on the same platform, which means EMO achieves a 20.9× speedup.

7.3.2 Frame Sampling. The frame sampler is introduced to improve system performance by skipping the image analysis workflow on selected frames. Here we explore the lower bound of the sampling algorithm, λ, which weighs the sensitivity of detecting the emotion change and computational resource usage. The skip rate determines the proportion of frames that the sampler skipped, and also influences the amount of CPU time that can be saved.

Figure 9 shows the results on the Open-Q820 platform with a frame rate of 30fps. We can see that for λ values from 2 to 8, the skip rate is around 80%, and the CPU run time ranges from 600ms to 400ms per second. As λ increases, CPU time drastically decreases. λ can be configured by users according to their needs. If the user does not need to detect emotion changes immediately, it is safe to set a larger α for larger computational resource savings. It is important to note that we also limit the upper bound of Λ to ensure the sampler not miss any emotions. We set the upper bound to 10 frames in our implementation, considering the shortest time duration of emotions.

Table 9: System resources usage of the EMO and Eyemotion (Eye.).

Model	Board	CPU	Memory	Power	Storage
Eye.	Q820	53.7%	313MB	5.9W	83.3MB
Eye.	Hikey	42.06%	445MB	2.4W	83.3MB
EMO	Q820	16.20%	73MB	1.8W	4.9MB
EMO	Hikey	23.69%	57MB	1.6W	4.9MB

7.4 Resources Usage

Given that the eye-tracking cameras are always on for eye-tracking and gazing detection, EMO just shares the eye-area video with existing applications. Since the resource overhead of collecting eye-area video are not brought by EMO, we only measure the resources usage during the emotion recognition phase. We compare the system resource usage of Eyemotion and EMO on both Open-Q820 and Hikey platforms. Results are shown in Table 9. Compared to Eyemotion, EMO uses significantly fewer resources. On Open-Q820, EMO reduces the CPU, memory, power, and storage costs by 3.3×, 4.3×, 3.3×, and 17×, respectively. On Hikey, the corresponding numbers are 1.8×, 7.8×, 1.5×, and 17×, respectively. These results confirm the advantages of the lightweight model design and the power of the fast forwarder and the frame sampler in EMO.

7.5 Advantages of Continuous Recognition

Continuous recognition helps recognizing emotions effectively and quickly. We evaluate how sensitive the EMO is to the emotion changes, by measuring the time interval from when an emotion change happens to when EMO detects the emotion change. Surprise, the most ephemeral expression, lasts significantly shorter than other emotions. Thus, we have designed an extreme experiment by inviting a volunteer to perform surprise rapidly as follows.

We conducted a case study to evaluate the precision of EMO on surprise, with α set to 0.75 and λ set to 5. A volunteer was asked to put on the EMO, initialize the personalized features, and express a surprised expression once every 5 to 10 seconds for 100 times. The start and end time of surprised expression is labeled as ground truth, and the time recorded when the surprise is recognized by EMO. Only 4% of the frames labeled as surprise are missed, which shows EMO is effective in capturing ephemeral expressions. An example can be found in Figure 10.

8 DISCUSSION

Hardware Support. Although we built our own hardware prototype for easy deployment and evaluation, EMO does not depend on any customized hardware. What EMO needs is just an eye-tracking camera that is available on many existing eyewear devices and more future devices. Thus, we expect that EMO can run on many eyewear devices, including commercial off-the-shelf ones.

Technology Generality. While we focus on emotion recognition from single-eye images in this paper, we believe that the techniques we develop for EMO are also useful to build other mobile systems and applications involving deep learning. First, our model

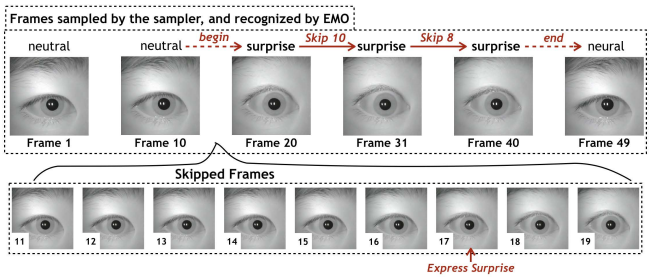

Figure 10: A continuous recognition example of EMO capturing an ephemeral emotion of surprise. The input frame rate is 30fps, and the video sampler is enabled. The frames in the first row are ones processed for recognition, and the frames in the second row are ones skipped in the inputs. It can be seen that the expression of surprise can be recognized in time with low error.

design, the combination of both generalized feature extractor and personalized classifier, could be applied to effectively accomplish other complex deep-learning tasks of personal devices. Second, our system optimizations, which leverage the temporal characteristics of videos, could also be applied to improve the processing efficiency of other video analytics tasks.

9 RELATED WORK

Whole-face images based emotion recognition has been a traditional yet active research field in computer vision. Many solutions have been proposed, including the Boosted Deep Belief Network (BDBN) [55], AU-aware Deep Networks (AUDN) [54], boosted LBP descriptors [68], RNNs [25], and many others. Among all of them, the CNN based solutions [41, 49, 63, 85] stand out due to their impressive performance and accuracy. Improving the recognition efficiency of CNNs, however, has not been the main focus in the research.

Live eye tracking applications. Eye movement has long been understood and used in many applications. Many studies focus on mobile eye tracking [39, 42, 43, 61, 65], while others explore how to utilize eye movement in various tasks. For example, Lander et al. [44] use corneal imaging to extract information for lifelogging; Steil et al. [72] use eye movement to discover users' everyday activities from their long-term visual behavior. Closely related to our work, Hoppe et al [35] and Yang et al. [83] leverage eye movement and head pose analysis for automatic recognition of different levels of curiosity and emotion. However, none of them is able to directly infer the users' emotions from the micro facial movements around the eye area.

Emotion recognition on wearable devices. Automatic emotion recognition is critical to wearable devices because these devices are highly personalized and have limited human-machine interaction interfaces. Many solutions rely on special hardware, e.g., electroencephalogram (EEG) sensors [16, 71] or photo reflective sensors [57], and thus are not readily available on existing devices. Only recently, Eyemotion [34] explored how to leverage eye-tracking cameras available on wearable devices to perform emotion recognition. EMO shares a similar high-level idea with Eyemotion but achieves significant improvements in terms of accuracy, latency, and resource usage, through CNN-based lightweight and powerful feature extracting, effective personalization, and efficient frame sampling and fast forwarding.

10 CONCLUSION

In this paper, we propose and develop EMO, an effective and efficient system to recognize user emotions from single-eye images captured by an eye-tracking camera on eyewear devices. EMO employs a generic feature extractor and a personalized classifier for accurate recognition of seven emotion types among different users. To optimize the efficiency of continuous recognition, it uses a frame sampler and fast forwarder to exploit the temporal locality of eye-tracking videos. EMO is implemented on two hardware platforms. Comprehensive evaluation results demonstrate that EMO achieves continuous emotion recognition in real-time with an average frame rate of 12.8fps, and a mean accuracy of 72.2%, significantly outperforms the previous state-of-the-art approach and consumes much fewer system resources.

ACKNOWLEDGMENTS

We sincerely thank our shepherd Prof. Mahadev Satyanarayanan and the anonymous reviewers for their valuable feedback. This work is supported in part by NSFC-61872180, Jiangsu "Shuang-Chuang" Program, Jiangsu "Six-Talent-Peaks" Program, Ant Financial through the Ant Financial Science Funds for Security Research. Sheng Zhong is supported in part by NSFC-61872176.

REFERENCES

[1] 2018. About UVCCamera. https://github.com/saki4510t/UVCCamera. [Online; accessed 11-December-2018].
[2] 2019. FOVE 0 Eye-tracking VR Devkit. https://www.getfove.com/. [Online; accessed 10-April-2019].
[3] 2019. HiKey. https://www.96boards.org/product/hikey/. [Online; accessed 11-December-2019].
[4] 2019. HTC VIVE Pro Eye Head Mounted Display. https://enterprise.vive.com/us/product/vive-pro-eye/. [Online; accessed 05-December-2019].
[5] 2019. Open-Q 820. https://www.intrinsyc.com/computing-platforms/open-q-820-usom/. [Online; accessed 11-December-2019].
[6] 2019. Pupil. https://pupil-labs.com/pupil/. [Online; accessed 10-December-2019].
[7] 2019. Tobii. https://www.tobii.com/tech/products/vr/. [Online; accessed 10-December-2019].
[8] 2020. HoloLens 2. https://www.microsoft.com/en-us/hololens. [Online; accessed 01-Apr-2020].
[9] 2020. How Eye Tracking is Driving the Next Generation of AR and VR. https://vrscout.com/news/eye-tracking-driving-next-generation-ar-vr/. [Online; accessed 01-Apr-2020].
[10] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. 2016. Tensorflow: a system for large-scale machine learning.. In *USENIX Symposium on Operating Systems Design and Implementation (OSDI)*. USENIX Association, 265–283.
[11] Niki Aifanti, Christos Papachristou, and Anastasios Delopoulos. 2010. The MUG facial expression database. In *11th International Workshop on Image Analysis for Multimedia Interactive Services WIAMIS 10*. IEEE, 1–4.
[12] Lisa Aziz-Zadeh and Antonio Damasio. 2008. Embodied semantics for actions: Findings from functional brain imaging. *Journal of Physiology-Paris* 102, 1-3 (2008), 35–39.
[13] Mihai Bâce, Sander Staal, and Gábor Sörös. 2018. Wearable eye tracker calibration at your fingertips. In *Proceedings of the 2018 ACM Symposium on Eye Tracking Research & Applications*. ACM, 22.
[14] Luca Bertinetto, Jack Valmadre, Joao F Henriques, Andrea Vedaldi, and Philip HS Torr. 2016. Fully-convolutional siamese networks for object tracking. In *European conference on computer vision*. 850–865.

[15] Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. 1993. Signature Verification Using a "Siamese" Time Delay Neural Network. In *Advances in neural information processing systems*. Morgan Kaufmann Publishers Inc., 737–744.

[16] Guillaume Chanel, Julien Kronegg, Didier Grandjean, and Thierry Pun. 2006. Emotion assessment: Arousal evaluation using EEG's and peripheral physiological signals. In *Multimedia Content Representation, Classification and Security*. Springer Berlin Heidelberg, 530–537.

[17] Jason Chang, Donglai Wei, and John W. Fisher, III. 2013. A Video Representation Using Temporal Superpixels. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, 2051–2058.

[18] Warapon Chinsatit and Takeshi Saitoh. 2017. CNN-based pupil center detection for wearable gaze estimation system. *Applied Computational Intelligence and Soft Computing* 2017 (2017).

[19] Sumit Chopra, Raia Hadsell, and Yann LeCun. 2005. Learning a similarity metric discriminatively, with application to face verification. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, 539–546.

[20] Dahjung Chung, Khalid Tahboub, and Edward J Delp. 2017. A Two Stream Siamese Convolutional Neural Network for Person Re-identification. In *IEEE International Conference on Computer Vision (ICCV)*. IEEE Computer Society, 1992–2000.

[21] Jean Costa, Alexander T. Adams, Malte F. Jung, François Guimbretière, and Tanzeem Choudhury. 2016. EmotionCheck: Leveraging Bodily Signals and False Feedback to Regulate Our Emotions. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM, 758–769.

[22] Roddy Cowie, Ellen Douglas-Cowie, Nicolas Tsapatsoulis, George Votsis, Stefanos Kollias, Winfried Fellenz, and John G Taylor. 2001. Emotion recognition in human-computer interaction. *IEEE Signal processing magazine* 18, 1 (2001), 32–80.

[23] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, 248–255.

[24] Samira Ebrahimi Kahou, Vincent Michalski, Kishore Konda, Roland Memisevic, and Christopher Pal. 2015. Recurrent neural networks for emotion recognition in video. In *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction*. ACM, 467–474.

[25] Samira Ebrahimi Kahou, Vincent Michalski, Kishore Konda, Roland Memisevic, and Christopher Pal. 2015. Recurrent Neural Networks for Emotion Recognition in Video. In *ACM on International Conference on Multimodal Interaction*. ACM, 467–474.

[26] Paul Ekman. 1992. An argument for basic emotions. *Cognition and Emotion* 6, 3-4 (1992), 169–200.

[27] Paul Ekman and Wallace V. Friesen. 1976. Measuring facial movement. *Environmental psychology and nonverbal behavior* 1, 1 (1976), 56–75.

[28] Wallace V Friesen and Paul Ekman. 1983. EMFACS-7: Emotional facial action coding system. *Unpublished manuscript, University of California at San Francisco* (1983), 1.

[29] Wolfgang Fuhl, Thiago C Santini, Thomas Kübler, and Enkelejda Kasneci. 2016. Else: Ellipse selection for robust pupil detection in real-world environments. In *Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research & Applications*. ACM, 123–130.

[30] Kurara Fukumoto, Tsutomu Terada, and Masahiko Tsukamoto. 2013. A smile/laughter recognition mechanism for smile-based life logging. In *Proceedings of the 4th Augmented Human International Conference*. ACM, 213–220.

[31] Ian J Goodfellow, Dumitru Erhan, Pierre Luc Carrier, Aaron Courville, Mehdi Mirza, Ben Hamner, Will Cukierski, Yichuan Tang, David Thaler, Dong-Hyun Lee, et al. 2013. Challenges in representation learning: A report on three machine learning contests. In *International Conference on Neural Information Processing*. Springer, 117–124.

[32] Qing Guo, Wei Feng, Ce Zhou, Rui Huang, Liang Wan, and Song Wang. 2017. Learning dynamic siamese network for visual object tracking. In *IEEE International Conference on Computer Vision (ICCV)*. IEEE Computer Society, 1781–1789.

[33] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, 770–778.

[34] Steven Hickson, Nick Dufour, Avneesh Sud, Vivek Kwatra, and Irfan Essa. 2019. Eyemotion: Classifying facial expressions in VR using eye-tracking cameras. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 1626–1635.

[35] Sabrina Hoppe, Tobias Loetscher, Stephanie Morey, and Andreas Bulling. 2015. Recognition of curiosity using eye movement analysis. In *Adjunct Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2015 ACM International Symposium on Wearable Computers*. ACM, 185–188.

[36] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. 2017. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861* (2017). http://arxiv.org/abs/1704.04861

[37] Forrest N Iandola, Song Han, Matthew W Moskewicz, Khalid Ashraf, William J Dally, and Kurt Keutzer. 2016. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and< 0.5 MB model size. *arXiv preprint arXiv:1602.07360* (2016). https://arxiv.org/abs/1602.07360

[38] Spiros V. Ioannou, Amaryllis T. Raouzaiou, Vasilis A. Tzouvaras, Theofilos P. Mailis, Kostas C. Karpouzis, and Stefanos D. Kollias. 2005. Emotion recognition through facial expression analysis based on a neurofuzzy network. *Neural Networks* 18, 4 (2005), 423–435.

[39] Moritz Kassner, William Patera, and Andreas Bulling. 2014. Pupil: an open source platform for pervasive eye tracking and mobile gaze-based interaction. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication*. ACM, 1151–1160.

[40] Bo-Kyeong Kim, Hwaran Lee, Jihyeon Roh, and Soo-Young Lee. 2015. Hierarchical committee of deep cnns with exponentially-weighted decision fusion for static facial expression recognition. In *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction*. ACM, 427–434.

[41] Bo-Kyeong Kim, Hwaran Lee, Jihyeon Roh, and Soo-Young Lee. 2015. Hierarchical Committee of Deep CNNs with Exponentially-Weighted Decision Fusion for Static Facial Expression Recognition. In *ACM on International Conference on Multimodal Interaction*. ACM, 427–434.

[42] Elizabeth S Kim, Adam Naples, Giuliana Vaccarino Gearty, Quan Wang, Seth Wallace, Carla Wall, Michael Perlmutter, Jennifer Kowitt, Linda Friedlaender, Brian Reichow, et al. 2014. Development of an untethered, mobile, low-cost head-mounted eye tracker. In *Proceedings of the Symposium on Eye Tracking Research and Applications*. ACM, 247–250.

[43] Christian Lander, Sven Gehring, Antonio Krüger, Sebastian Boring, and Andreas Bulling. 2015. Gazeprojector: Accurate gaze estimation and seamless gaze interaction across multiple displays. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*. ACM, 395–404.

[44] Christian Lander, Antonio Krüger, and Markus Löchtefeld. 2016. The story of life is quicker than the blink of an eye: using corneal imaging for life logging. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct*. ACM, 1686–1695.

[45] Laura Leal-Taixé, Cristian Canton-Ferrer, and Konrad Schindler. 2016. Learning by tracking: Siamese CNN for robust target association. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. IEEE Computer Society, 418–425.

[46] Joseph LeDoux and Jules R Bemporad. 1997. The emotional brain. *Journal of the American Academy of Psychoanalysis* 25, 3 (1997), 525–528.

[47] Uichin Lee, Kyungsik Han, Hyunsung Cho, Kyong-Mee Chung, Hwajung Hong, Sung-Ju Lee, Youngtae Noh, Sooyoung Park, and John M. Carroll. 2019. Intelligent positive computing with mobile, wearable, and IoT devices: Literature review and research directions. *Ad Hoc Networks* 83 (2019), 8 – 24.

[48] Gil Levi and Tal Hassner. 2015. Emotion recognition in the wild via convolutional neural networks and mapped binary patterns. In *Proceedings of the 2015 ACM on international conference on multimodal interaction*. ACM, 503–510.

[49] Gil Levi and Tal Hassner. 2015. Emotion Recognition in the Wild via Convolutional Neural Networks and Mapped Binary Patterns. In *ACM on International Conference on Multimodal Interaction*. ACM, 503–510.

[50] Robert LiKamWa, Yunxin Liu, Nicholas D. Lane, and Lin Zhong. 2013. MoodScope: building a mood sensor from smartphone usage patterns. In *Proceedings of the 11th annual international conference on Mobile systems, applications, and services*. ACM, 389–402.

[51] G. Littlewort, J. Whitehill, T. Wu, I. Fasel, M. Frank, J. Movellan, and M. Bartlett. 2011. The computer expression recognition toolbox (CERT). In *Face and Gesture 2011*. IEEE, 298–305.

[52] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. 2008. Isolation forest. In *2008 Eighth IEEE International Conference on Data Mining*. IEEE, 413–422.

[53] Luyang Liu, Hongyu Li, and Marco Gruteser. 2019. Edge Assisted Real-time Object Detection for Mobile Augmented Reality. In *In Proceedings of The 25th Annual International Conference on Mobile Computing and Networking*. ACM.

[54] Mengyi Liu, Shaoxin Li, Shiguang Shan, and Xilin Chen. 2013. Au-aware deep networks for facial expression recognition. In *International Conference and Workshops on Automatic Face and Gesture Recognition*. IEEE Computer Society, 1–6.

[55] Ping Liu, Shizhong Han, Zibo Meng, and Yan Tong. 2014. Facial Expression Recognition via a Boosted Deep Belief Network. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, 1805–1812.

[56] J. MacQueen. 1967. Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*. University of California Press, 281–297.

[57] Katsutoshi Masai, Yuta Sugiura, Katsuhiro Suzuki, Sho Shimamura, Kai Kunze, Masa Ogata, Masahiko Inami, and Maki Sugimoto. 2015. Affective wear: towards recognizing affect in real life. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing and the Proceedings of the 2015 ACM International Symposium on Wearable Computers*. ACM, 357–360.

[58] Daniel McDuff, Abdelrahman Mahmoud, Mohammad Mavadati, May Amr, Jay Turcot, and Rana el Kaliouby. 2016. AFFDEX SDK: A Cross-Platform Real-Time

Multi-Face Expression Recognition Toolkit. In *CHI Extended Abstracts*. 3723–3726.

[59] Niall McLaughlin, Jesús Martínez del Rincón, and Paul C. Miller. 2016. Recurrent Convolutional Network for Video-Based Person Re-identification. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, 1325–1334.

[60] Daniel N. McIntosh RB Zajonc Peter S. Vig Stephen W. Emerick. 1997. Facial movement, breathing, temperature, and affect: Implications of the vascular theory of emotional efference. *Cognition & Emotion* 11, 2 (1997), 171–196.

[61] Basilio Noris, Jean-Baptiste Keller, and Aude Billard. 2011. A wearable gaze tracking system for children in unconstrained environments. *Computer Vision and Image Understanding* (2011), 476–486.

[62] Sébastien Ouellet. 2014. Real-time emotion recognition for gaming using deep convolutional network features. *arXiv preprint arXiv:1408.3750* (2014). https://arxiv.org/abs/1408.3750

[63] Sébastien Ouellet. 2014. Real-time emotion recognition for gaming using deep convolutional network features. *arXiv preprint arXiv:1408.3750* (2014). https://arxiv.org/abs/1408.3750

[64] Fazlay Rabbi, Taiwoo Park, Biyi Fang, Mi Zhang, and Youngki Lee. 2018. When Virtual Reality Meets Internet of Things in the Gym: Enabling Immersive Interactive Machine Exercises. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 2, 2 (2018), 78:1–78:21.

[65] Javier San Agustin, Henrik Skovsgaard, Emilie Mollenbach, Maria Barret, Martin Tall, Dan Witzner Hansen, and John Paulin Hansen. 2010. Evaluation of a low-cost open-source gaze tracker. In *Proceedings of the 2010 Symposium on Eye-Tracking Research & Applications*. ACM, 77–80.

[66] Alexandre Schaefer, Frédéric Nils, Xavier Sanchez, and Pierre Philippot. 2010. Assessing the effectiveness of a large database of emotion-eliciting films: A new tool for emotion researchers. *Cognition and Emotion* (2010), 1153–1172.

[67] Jocelyn Scheirer, Raul Fernandez, and Rosalind W Picard. 1999. Expression glasses: a wearable device for facial expression recognition. In *CHI'99 Extended Abstracts on Human Factors in Computing Systems*. ACM, 262–263.

[68] Caifeng Shan, Shaogang Gong, and Peter W. McOwan. 2009. Facial Expression Recognition Based on Local Binary Patterns: A Comprehensive Study. *Image and vision Computing* (2009), 803–816.

[69] Matthew Shreve, Sridhar Godavarthy, Dmitry Goldgof, and Sudeep Sarkar. 2011. Macro-and micro-expression spotting in long videos using spatio-temporal strain. In *Face and Gesture*. IEEE, 51–56.

[70] Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014). http://arxiv.org/abs/1409.1556

[71] Mohammad Soleymani, Maja Pantic, and Thierry Pun. 2012. Multimodal emotion recognition in response to videos. *IEEE transactions on affective computing* 3, 2 (2012), 211–223.

[72] Julian Steil and Andreas Bulling. 2015. Discovery of everyday human activities from long-term visual behaviour using topic models. In *Proceedings of the 2015 acm international joint conference on pervasive and ubiquitous computing*. ACM, 75–85.

[73] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, Andrew Rabinovich, et al. 2015. Going deeper with convolutions. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, 1–9.

[74] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, 2818–2826.

[75] Ran Tao, Efstratios Gavves, and Arnold WM Smeulders. 2016. Siamese instance search for tracking. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, 1420–1429.

[76] Rahul Rama Varior, Mrinal Haloi, and Gang Wang. 2016. Gated siamese convolutional neural network architecture for human re-identification. In *European Conference on Computer Vision (ECCV)*. Springer, 791–808.

[77] Rahul Rama Varior, Bing Shuai, Jiwen Lu, Dong Xu, and Gang Wang. 2016. A siamese long short-term memory architecture for human re-identification. In *ECCV*. Springer, 135–153.

[78] Xi Wang, Xi Zhao, Varun Prakash, Zhimin Gao, Tao Feng, Omprakash Gnawali, and Weidong Shi. 2013. Person-of-interest detection system using cloud-supported computerized-eyewear. In *2013 IEEE International Conference on Technologies for Homeland Security (HST)*. IEEE, 658–663.

[79] Wikipedia contributors. 2019. Confusion matrix. https://en.wikipedia.org/w/index.php?title=Confusion_matrix&oldid=870941727. [Online; accessed 6-January-2019].

[80] Mengwei Xu, Mengze Zhu, Yunxin Liu, Felix Xiaozhu Lin, and Xuanzhe Liu. 2018. DeepCache: Principled Cache for Mobile Deep Vision. In *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking*. ACM, 129–144.

[81] Ran Xu, Jinkyu Koo, Rakesh Kumar, Peter Bai, Subrata Mitra, Sasa Misailovic, and Saurabh Bagchi. 2018. VideoChef: Efficient Approximation for Streaming Video Processing Pipelines. In *2018 USENIX Annual Technical Conference (USENIX ATC 18)*. USENIX Association, 43–56.

[82] Ekin Yağış and Mustafa Unel. 2018. Facial Expression Based Emotion Recognition Using Neural Networks. In *Image Analysis and Recognition*. Springer International Publishing, 210–217.

[83] Xiaochao Yang, Chuang-Wen You, Hong Lu, Mu Lin, Nicholas D Lane, and Andrew T Campbell. 2012. Visage: A face interpretation engine for smartphone applications. In *International Conference on Mobile Computing, Applications, and Services*. Springer, 149–168.

[84] Zhiding Yu and Cha Zhang. 2015. Image based static facial expression recognition with multiple deep network learning. In *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction*. ACM, 435–442.

[85] Zhiding Yu and Cha Zhang. 2015. Image Based Static Facial Expression Recognition with Multiple Deep Network Learning. In *ACM on International Conference on Multimodal Interaction*. ACM, 435–442.

[86] Ligang Zhang, Brijesh Verma, Dian Tjondronegoro, and Vinod Chandran. 2018. Facial Expression Analysis Under Partial Occlusion: A Survey. *ACM Comput. Surv.* 51, 2 (2018), 25:1–25:49.

[87] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. 2018. ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, 6848–6856.

[88] Yongtuo Zhang, Wen Hu, Weitao Xu, Chun Tung Chou, and Jiankun Hu. 2018. Continuous Authentication Using Eye Movement Response of Implicit Visual Stimuli. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 1, 4 (2018), 177:1–177:22.

[89] Mingmin Zhao, Fadel Adib, and Dina Katabi. 2016. Emotion recognition using wireless signals. In *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking*. ACM, 95–108.