applied sciences

MDPI

Article

Real-Time Facial Emotion Recognition Framework for Employees of Organizations Using Raspberry-Pi

Navjot Rathour [1], Zeba Khanam [2], Anita Gehlot [1], Rajesh Singh [1], Mamoon Rashid [3,*], Ahmed Saeed AlGhamdi [4] and Sultan S. Alshamrani [5]

[1] School of Electronics and Electrical Engineering, Lovely Professional University, Jalandhar 144001, India; navjot.16885@lpu.co.in (N.R.); anita.23401@lpu.co.in (A.G.); rajesh.23402@lpu.co.in (R.S.)
[2] College of Computing and Informatics, Saudi Electronic University, Dammam 15515, Saudi Arabia; z.khanam@seu.edu.sa
[3] Department of Computer Engineering, Faculty of Science and Technology, Vishwakarma University, Pune 411048, India
[4] Department of Computer Engineering, College of Computer and Information Technology, Taif University, PO Box 11099, Taif 21994, Saudi Arabia; asjannah@tu.edu.sa
[5] Department of Information Technology, College of Computer and Information Technology, Taif University, PO Box 11099, Taif 21944, Saudi Arabia; susamash@tu.edu.sa
* Correspondence: mamoon.rashid@vupune.ac.in; Tel.: +91-7814346505

Abstract: There is a significant interest in facial emotion recognition in the fields of human–computer interaction and social sciences. With the advancements in artificial intelligence (AI), the field of human behavioral prediction and analysis, especially human emotion, has evolved significantly. The most standard methods of emotion recognition are currently being used in models deployed in remote servers. We believe the reduction in the distance between the input device and the server model can lead us to better efficiency and effectiveness in real life applications. For the same purpose, computational methodologies such as edge computing can be beneficial. It can also encourage time-critical applications that can be implemented in sensitive fields. In this study, we propose a Raspberry-Pi based standalone edge device that can detect real-time facial emotions. Although this edge device can be used in variety of applications where human facial emotions play an important role, this article is mainly crafted using a dataset of employees working in organizations. A Raspberry-Pi-based standalone edge device has been implemented using the Mini-Xception Deep Network because of its computational efficiency in a shorter time compared to other networks. This device has achieved 100% accuracy for detecting faces in real time with 68% accuracy, i.e., higher than the accuracy mentioned in the state-of-the-art with the FER 2013 dataset. Future work will implement a deep network on Raspberry-Pi with an Intel Movidious neural compute stick to reduce the processing time and achieve quick real time implementation of the facial emotion recognition system.

Keywords: emotion recognition; face detection; face recognition; machine learning (ML); real-time systems; Raspberry-Pi; support vector machine (SVM)

Citation: Rathour, N.; Khanam, Z.; Gehlot, A.; Singh, R.; Rashid, M.; AlGhamdi, A.S.; Alshamrani, S.S. Real-Time Facial Emotion Recognition Framework for Employees of Organizations Using Raspberry-Pi. *Appl. Sci.* **2021**, *11*, 10540. https://doi.org/10.3390/app112210540

Academic Editor: Monica Perusquia Hernandez

Received: 19 September 2021
Accepted: 3 November 2021
Published: 9 November 2021

1. Introduction

In today's context, video cameras can be easily accessed by everyone. These video cameras can be mobile-based cameras or other static cameras like surveillance cameras, smartphone cameras, Raspberry-Pi cameras, or laptops, etc. With the help of these cameras, it is easy to capture human faces from any location at any place. This kind of freedom has enabled the research community to implement these smart systems for understanding the behavior of humans in real-time. To understand the behavior of a human being, expression plays the most important role. Various surveys have already been done to understand different components that play major roles in understanding human emotions. The outcome of those surveys concluded that non-verbal components, facial expressions, play the most

important role during interpersonal communication [1]. Research in the field of emotive facial recognition has gathered attention in the last couple of decades, as the applications are not only limited to computer science but can be implemented in the field of affective computing, computer animation, cognitive science, and perceptual sciences etc. [2]. The major mode of exchanging feelings and emotions in daily life is facial expressions. Small facial gestures are strong enough to pass on the message to another person about one's feelings. Facial emotion is more important than verbal communication, as the emotions are an actual spontaneous reflection of a person's feelings. Lots of research is being carried out to develop such robots that are capable of understanding the facial emotions of human beings and could understand different moods of people [3]. To automate the facial emotion recognition system, various techniques have been used. With the help of such systems, facial expressions can be detected and the system has been applied during interviews [4], for surveillance systems [5], and for detection of aggression [6]. When it comes to computer vision and artificial intelligence, facial emotion recognition is one of the most important topics. For detection of facial emotions, various sensors can be used but facial images are more important because they carry enough information to understand interpersonal communication. Various research has been conducted over the last couple of years, out of which deep-learning-based FER approaches along with detailed algorithms have been proposed. In addition there are various hybrid and deep-learning approaches that are a combination of convolutional neural networks that combine the spatial and temporal features of frames [7].

Facial emotion recognition systems have gained popularity over the decades because of their diverse applications, and the majority of those applications are applicable in real-world activities like smart supervision for suspicious activities, marketing, group emotion analysis, etc. In the same field, a cost-effective system has been proposed by Muhammad Sajjad et al. that will help to implement a smart security system for law enforcement. This system has been proposed using Raspberry-Pi with Pi-cam, that also makes it cost effective and compact [8]. With the advancement of technology and the availability of various compact devices like Raspberry-Pi, it becomes easy to equip police and security officers with compact systems that can detect facial images in real-time. In addition to that, with the development of cloud-based technology, the captured images can be sent to cloud for future action. Such a cloud-assisted facial recognition framework has been proposed by Muhammad Sajjad et al. that can help to identify criminals and provide ease for police and security people to identify criminals quickly and easily with this proposed framework [9].

A smart home based on Raspberry-Pi has also been proposed that is completely automated. Chowdhury et al. has proposed this system to automate and provide access via web to carry out everyday work [10]. An energy efficient and smart water management system has been proposed to provide cost-effective solutions over existing irrigation systems. Agrawal et al. has proposed a system that could initially water around 50 pots kept in the garden and proposed a strong system that could further be extended for bigger fields [11]. Another application based on face recognition has been proposed by [12]. This system provides access only to those who were identified by the system and makes the system more secure. The proposed system is also based on Raspberry-Pi and Pi-Camera that again is a cost-effective and user-friendly hardware to work with. Various challenges like pose mismatching [13] and usage of strong descriptors [14] have been mentioned in the literature and several alternatives have been provided in the literature to handle such issues. Various classification techniques are available in the literature and a few of them are amazingly effective to extract important information from those images like HFR (Hybrid face regions) [15]. Implementation of deep networks for smartphones has allowed facial detection along with gender classification, and a pixel pattern-based gender classification has also been proposed on the FERRET dataset, with an accuracy of 90% with frontal faces [16]. Various architectures are proposed in the literature to implement emotion recognition in real-time with multiple faces in videos [17]. It becomes a challenge to detect faces and emotions in critical situations. To manage such situations, a fast and accurate

system based on ORFs has been proposed that provides results by working on multiple components like backgrounds, pose estimation, face patches, etc. [18] The contributions of study are as follows:

- The implementation of hardware prototype for real time facial emotion detection with Raspberry-Pi.
- A deep convolutional neural network known as Mini-Xception is used for training, validation, and testing of emotive facial images.
- Support vector machine (SVM) classification is implemented in the Raspberry-Pi hardware for classifying the persons.

The organization of the study is as follows: Section 2 presents the overview of AI and CNN; Section 3 presents the proposed methodology where the methodology for real time face emotion detection is covered. Section 4 presents the hardware setup, experimental results, and comparison of proposed hardware with previous studies. Finally, the study concluded in the final section.

2. Background

Artificial Intelligence (AI) refers to the representation of human intelligence of machines designed to think and recreate human actions [19]. The concept can also be extended to any system that demonstrates characteristics consistent with a human mind such as learning and problem-solving. AI is interdisciplinary, however advancements in machine learning and deep learning are triggering a shift in perspective for nearly all sectors [20]. Computer vision is an artificial intelligence area that concentrates on image issues. Convolutional neural networks (CNNs) and computer vision combined can perform complex operations from image recognition to resolving scientific problems [21]. CNNs are well known for the capability of image recognition and classification. In general, a basic convolutional neural network consists of neurons connected via multiple layers. These layers collect the input images and process them in different layers. A simple CNN consists of three types of layers: the convolutional layer, the max-pooling layer, and the fully connected layer. The first two layers are responsible for feature extraction, introducing non-linearity and feature reduction to reduce overfitting. The last layer, known as a fully connected layer, helps in the classification based on the features that are extracted in the previous layers. The fully connected layer contains the majority of the parameters. The number of parameters has also been reduced, presented in architectures like Inception V3 [22] in which the last layer is added, i.e., Global Average Pooling operation. This layer reduces the feature map by taking the average and converting the feature map into a scalar value. To further reduce the parameters, modern CNN architectures have presented the use of residual modules [23] and depth wise-separable convolutions [5]. The depth wise-separable CNNs works by separating the task of feature extraction and combining it within the convolutional layer, hence the parameters are further reduced. So, we have used a Mini-Xception CNN proposed by [24] which reduces the parameters by using depth wise-separable convolution layers instead of simple convolution layers and eliminates fully connected layers.

3. Proposed Methodology

This section discerns the proposed framework. The elucidation of each step is elaborated on in the next sections. The entire process is divided into three tightly coupled tasks. The first task is to train the pertained deep network after dividing the dataset into training, validation, and testing. The entire dataset of emotive facial images is divided into an 8:1:1 ratio. A dataset with N images will be divided into $\mathcal{O}_{\mathrm{TR}}$ for training, $\mathcal{O}_{\mathrm{VD}}$ for cross validation and $\mathcal{O}_{\mathrm{T}}$ for testing purposes. This means a training set of N number of images I will consist of $\mathcal{O}_{TR} = \left\{ I_1^{TR}, I_2^{TR}, I_3^{TR} \ldots I_{4N/5}^{TR} \right\}$ as training images, $\mathcal{O}_{VD} = \left\{ I_{(4N/5)+1}^{VD} \ldots I_{9N/10}^{VD} \right\}$ as validation images and $\mathcal{O}_{T} = \left\{ I_{(9N/10)+1}^{T} \ldots I_{N}^{T} \right\}$ as testing images. A deep convolutional neural network known as Mini-Xception is used for training, validation, and testing of

emotive facial images. Training, validation, and testing is done on Google-CoLab with 12GB NVIDIA Tesla K80 GPU using FER 2013 dataset.

The entire architecture is divided into two tightly coupled tasks, i.e., face recognition and facial emotion recognition in real time. For face recognition, a pre-trained deep network known as OpenFace is used. To begin with, a real time image from video is captured as I_1^r. The total number of images captured in real time is $\delta_R = \left\{ I_1^r, I_2^r \ldots I_N^r \right\}$. To train the deep network, 6 images of each subject have been used and the network is trained with single triplet method of training for 20 different people with N images and denoted as δ_{TR}. Once the training is complete, for the real time captured image, I_1^r, the very first task is to find the face inside the captured image and discard the unwanted information. The description of parameters used in the proposed architecture is given in Table 1.

Table 1. Description of parameters of proposed architecture.

Ժ Emotive Facial Dataset	Φ_n^{embd} 128 Feature vectors as face embeddings
$Ժ_{TR}$ Training Images	I Real time image representation
$Ժ_{VD}$ Validation Images	I_1^r First real-time facial image
$Ժ_T$ Testing Images	I_{pr}^r Preprocessed real-time facial image
δ_R Real Time Captured Images	I_{cr}^r Cropped real-time facial image
I_1^{TR} First Training Image	I_n^r n real-time facial images
$I_{4N/5}^{TR}$ 80% Training Images	Š SVM Classifier
$I_{9N/10}^{VD}$ 10% Validation Images	ε_c Classifier with labels having c = 0 to 6 classes
$I_N^T N$ number of testing images	I Image representation
δ_{TR} Training Dataset of facial images	Ď Face detection database images

A well-known method known as HOG (Histogram of Oriented Gradients) has been used to find the faces. After the detection of the face, the facial image I_1^r has been cropped, which is further preprocessed in order to remove the effects of bad lighting, tilted face, and skewness, etc., in the cropped image I_{cr}^r. The cropped image is preprocessed with the face landmark estimation algorithm. This algorithm locates 68 landmarks on the cropped image I_{cr}^r and, with the help of simple affine transformations, the image is preprocessed I_{pr}^r using rotation, shear, and scale to center the eyes and mouth of the cropped image I_{cr}^r at best. The preprocessed image I_{pr}^r is fed to the pre-trained network to extract the features from I_{pr}^r image and generate 128 embeddings that are measurements of the face. The feature of the preprocessed image I_{pr}^r is generated with the help of the neural network that will generate a feature vector of 128 embeddings as Φ_n^{embd}. The last step is to classify the image by measuring the closest match of 128 embeddings Φ_n^{embd} by comparing it with the database images. The feature vector Φ_n^{embd} is passed through the simple SVM classifier Š, to recognize the face. The entire architecture with its detailed framework is represented in Figure 1.

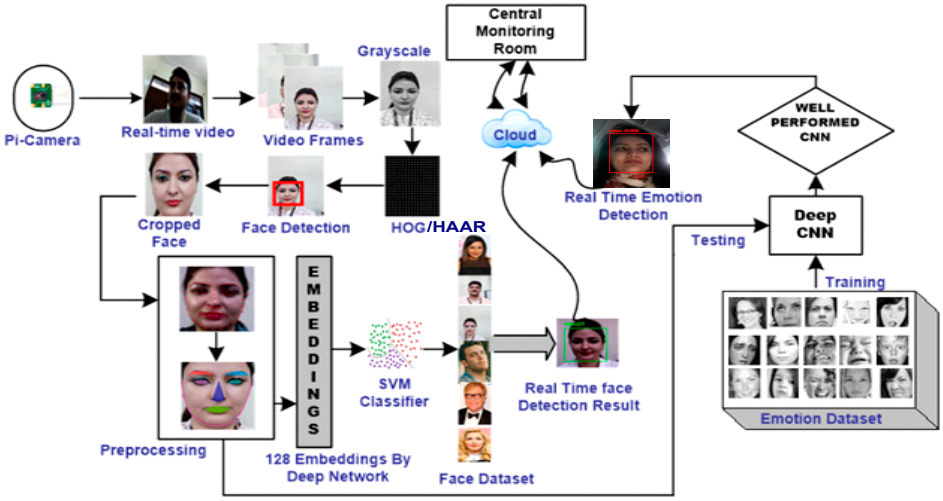


Figure 1. Architecture for face recognition and facial emotion recognition in real time using Raspberry-Pi (Dataset from [25]).

The second task is to transmit the output of preprocessing stage I^r_{pr} to the cloud where a pre-trained network of emotive facial recognition system is already available. This image I^r_{pr} is again passed through all the layers of the mini-Xception deep network, which is a fast and depth-wise separable convolutional neural network for the recognition of emotion captured in I^r_1 image. The deep network on the cloud is trained on seven basic emotions and is labelled as $\{\varepsilon_c = \varepsilon_1\varepsilon_{2\ldots}\varepsilon_7\}$ where the basic classes of seven emotions is represented as c.

3.1. Face Detection

After capturing the facial images in real time using Pi-cam, the first and foremost task is to separate the faces. To remove the unwanted and redundant information from the facial images like the background, a variety of methods are available. The most well-known method was the Viola Jones algorithm that was invented in the early 20s. We are using another method known as Histogram of Oriented Gradients, or HOG, to detect the facial images. Raspberry-Pi captures the emotive facial images in real-time via Pi-Cam from real time video frames. This image is real-time captured input image I^r which is converted to grayscale to extract HOG features to extract the facial part in the input image I^r. Finally the facial image is cropped, I^r_{cr}, and is fed to a feature extraction unit to recognize faces in real time. Figure 2a shows the basic steps of face detection using HOG. As shown in Figure 2, the gradients are calculated for the entire grayscale image, and this is done by calculating the gradients for 16 $\times$ 16 pixels at a time.

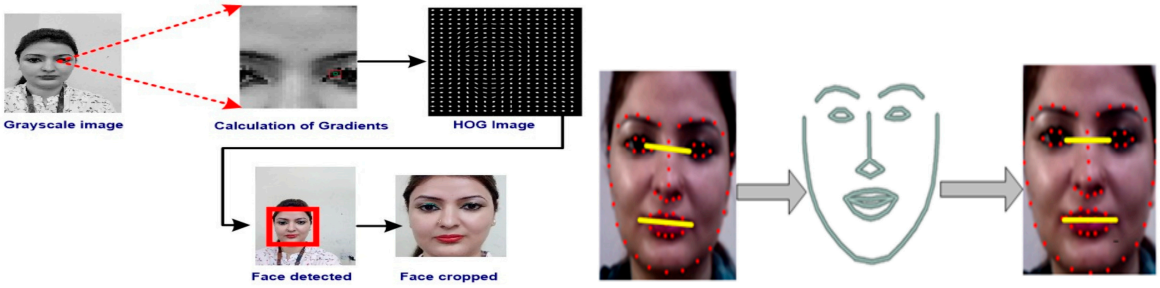


Figure 2. (**a**) Face detection using Histogram of Oriented Gradients (HOG) and (**b**) Face is aligned in real time using ensemble of regression trees algorithm.

This calculation is repeated for the entire grayscale image, and we will end up with an image of gradients. The next step is to calculate the strongest gradients in 16 × 16 windows of pixels and replace the gradients in that window with the strongest gradient. These will result in a basic image that consists of basic structure of face. To locate the face in the real-time captured input image or on real time video, we only located that part of the image which looks remarkably like a known HOG pattern and crop that part of the input image I^r, as a result we get the cropped facial image I^r_{cr}.

3.2. Face Alignment

As the face is captured in real-time, the image captured can have faces turned in different direction. To deal with such situations, we wrapped each picture so that our system can locate the eyes and lips in a sample place. To perform this operation, we have used an algorithm proposed by [23] which is known as face landmark estimation. The main work of this algorithm is to locate 68 specific points known as facial landmarks, as shown in Figure 2b, of all faces.

These landmarks locate the eyes, nose, chin, lips, and eyebrows etc., on any face. As explained by [19], $S = \left(x_1^T, x_2^T, \ldots, x_p^T\right)^T \in R^{2p}$ is the vector that represents the p number of facial landmarks in image I. The main aim is to perform the estimation of S to the best possible estimate, which is nearest to the true shape and is denoted by $S\prime^{(t)}$. This is done with the help of a cascade of regressors, where each regressor keeps on predicting and continuously updating the vector so that the estimation is accurate. $S\prime^{(t+1)} = S\prime^{(t)} + r_t S\prime^{(t)}$ is the method in which the regressor $r_t(.,.)$ is being used in a cascade for prediction and updating the vector $S\prime^{(t+1)}$.

3.3. Face Encoding

The next important step is to extract the features from the exactly centered image. The best way to get the unique features of any facial image is to measure the face. The dimensions of each face are different. The main challenge is about which measurement plays a vital role in recognition of captured image. This task can be difficult to achieve if performed with the traditional method of feature extraction. To achieve accuracy and raise the speed, a deep network is trained as machines have been proven to be better than humans when it comes to prediction. Training a deep network requires a lot of computation and power from a system. So, we used a pre-trained network which is provided by OpenFace [20]. Now, we just give the input and the Deep network that measures the 128 measurements for each face instead of single face; the network has been trained on 3 facial images at an instant as shown in Figure 3. This is achieved by training the first image (Image anchor) of person with a second image (positive) of the same person, with a completely different image (Image negative) of another person, as shown in Figure 4. The main purpose is to have the image anchor closer to image positive, as compared to any other image, called image negative. The selection of a triplet to carry out 128 measurements

is important. Machine learning experts call these measurements of every individual face "embedding". Training on face embedding using a large set of images called dataset will improve the accuracy and decrease the error rate eventually. This process requires huge CPU power and lot of time. To understand triplet loss, consider the representation as $f(y) = I^s$ which is representing an image y into s-dimensional Euclidean space. We oblige this implanting to live on the s-dimensional hypersphere. $f(y)_2 = 1$. As shown in Figure 3a, the main aim is to achieve minimum distance between y_j^m (Anchor) of a specific person with all the other images y_j^p(Positive) of the same person as compared to the image of any other person y_i^n(Negative).

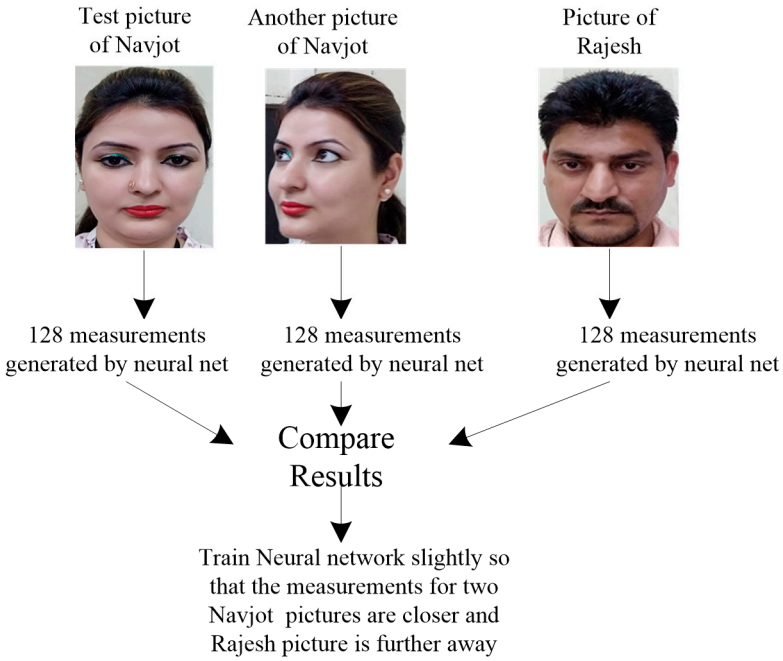


Figure 3. Generation of 128-dimensional data from triplet.

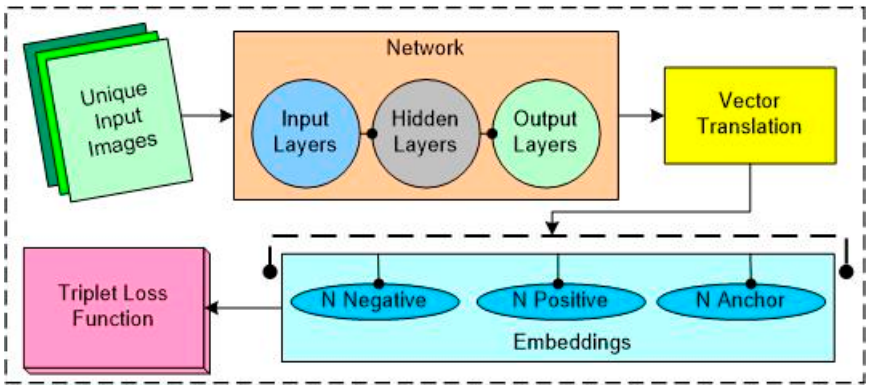


Figure 4. Network training flow for M unique images [Adapted from 5].

So, we want to have the following:

$$\|y_j^m - y_j^p\|_2^2 + \beta\|y_j^m - y_j^n\|_2^2 \forall \left(y_j^m, y_j^n, y_j^p\right) \in \zeta \tag{1}$$

where β is the enforced margin between negative and positive pair of images and ζ is the set of all the possible triplets and has numbers equal to number P.

$$\sum_j^P \left[\|f(y)_j^m - f(y)_j^p\|_2^2 - \|f(y_j^m) - f(y_j^n)\|_2^2 + \beta\right]_+ \tag{2}$$

Generation of multiple triplets will help to overcome the issue faced in Equation (1) and selection of suitable and complex triplets will result in the improvement of the deep learning model.

3.4. SVM Based Classification

The last step is the most important step of finding the names of persons from the encodings. Different techniques have been presented that will help in the evaluation of various classifiers. A variety of machine learning classification algorithms can be used to classify the faces but the most simple and efficient one has been used for classification of faces, known as support vector machine (SVM). We kept it simple because we only want the output to be the face with the name of the person. Moreover, we are implementing this on Raspberry-Pi, so we want our system to be fast and accurate. Running this classifier on hardware takes milliseconds, which what we want, and the result of this classifier is the name of the person.

3.5. Dataset

The training of the network is illustrated in the Figure 4, here we have mapped the unique images from a single network into triplets. The gradient of the triplet loss is back propagated to the unique images through the mapping. The dataset that we have used consists of 35,888 images of facial emotions with seven categories (0 = Angry, 1 = Disgust, 2 = Fear, 3 = Happy, 4 = Sad, 5 = Surprise, 6 = Neutral). The FER 2013 dataset consist of 48 × 48 pixel gray scale images (https://www.kaggle.com/msambare/fer2013 (accessed on 11 May 2021)). The dataset that we have used is in csv format, consisting of only two columns, i.e., "emotions" and "pixels" and is kept in Google drive. The entire data is divided into an 8:1:1 ratio for training σ_{TR}, validation σ_{VD} and testing σ_{T}.

The number of images available in the dataset has been categorized as per the expression, and the total number of images under each category is shown in Table 2 and the graphical representation of dataset is shown in Figure 5. The FER 2013 dataset is not a uniform dataset, and it does not contain a uniform number of images under each category. Figure 6 shows the sample images from the FER 2013 dataset. A large number of datasets is available to detect facial emotions.

Table 2. Data set classes.

Class	Emotion	Number	After Augmentation
0	Angry	4953	4953
1	Disgust	547	6564
2	Fear	5121	5121
3	Happy	8989	8989
4	Sad	6077	6077
5	Surprise	4002	4002
6	Neutral	6198	6198

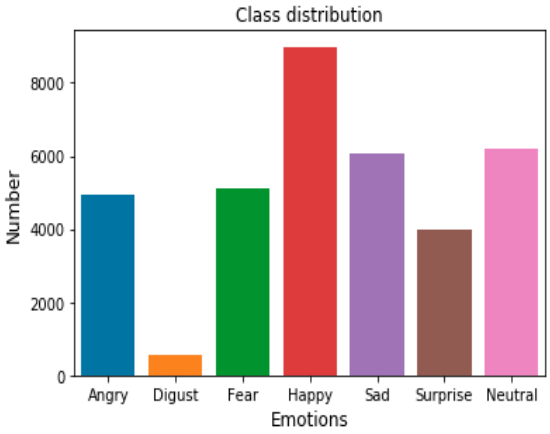


Figure 5. Graphical representation of class distribution.

Figure 6. Sample data from FER 2013 dataset.

3.6. Training CNN Model: Mini Xception

The dataset has been kept on Google drive and the training has been done on Google-CoLab with 12GB NVIDIA Tesla K80 GPU. The CNN has been trained with 80% of training data from the FER dataset and the remaining 10% of dataset is kept for validation. The architecture of Mini-Xception proposed by [24] is shown in Figure 7. Testing has been done on the remaining 10% of data as shown in Figure 8 and on the input given by Raspberry-Pi after detecting the face and converting the cropped and pre-processed images into 48 $\times$ 48 sizes. This architecture is trained on the FER 2013 dataset because we want the response to be quick and the proposed architecture of Mini-Xception has been proved to be quick and light because of its unique architecture and replacement of convolutional layers with depth wise convolutional layers, which will reduce the number of parameters and make it reliable to implement for real time emotion recognition. The results of training are shown in Figure 9 and training loss is represented in Figure 10. As our system is based on Raspberry-Pi, which has certain constraints in-terms of memory and processing capability, so a smaller number of parameters will be helpful in future advancement of this system. We have achieved an accuracy of 66% on FER 2013 dataset (without augmentation), as mentioned in the state-of-the-art, and 68% after data augmentation. The main reason behind this accuracy is variation in the dataset and non-uniformity of images under each category.

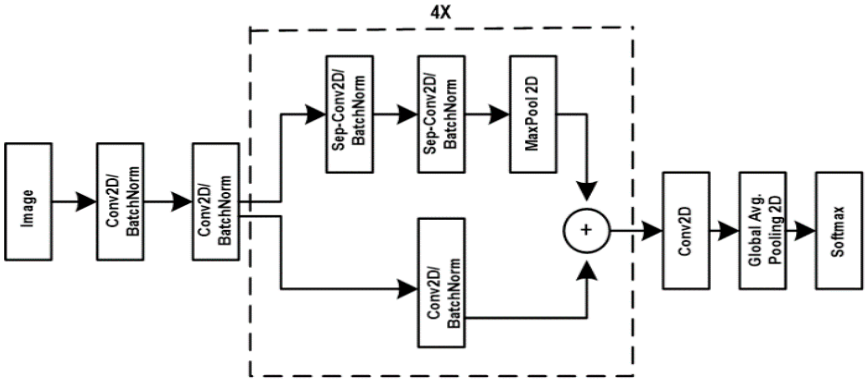
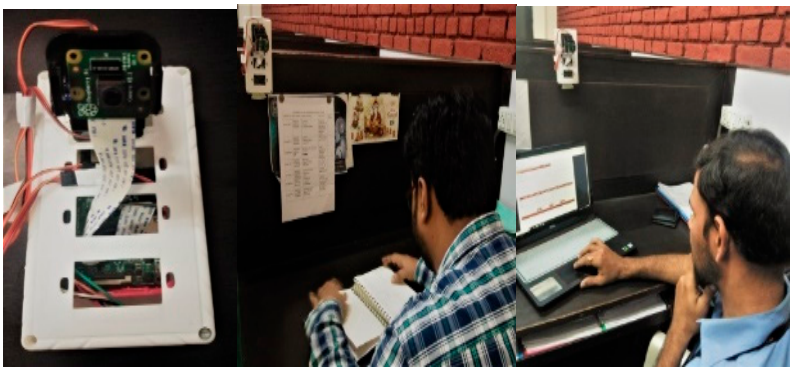
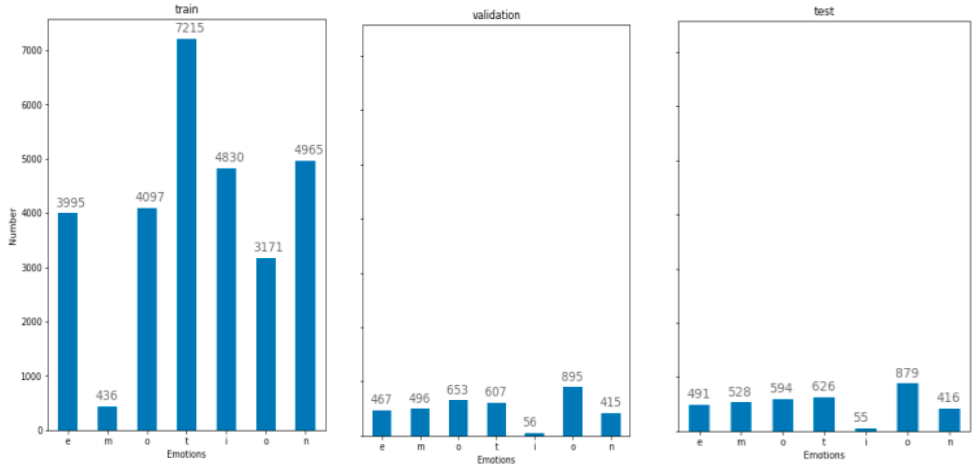


Figure 7. Mini-Xception Architecture.



(**a**) (**b**) (**c**)

Figure 8. (**a**) Hardware setup (**b**,**c**) Hardware setup in Employees Cubical.



(**a**) (**b**) (**c**)

Figure 9. Graphical representation of data segregated for (**a**) Training (**b**) Validation and (**c**) Testing.

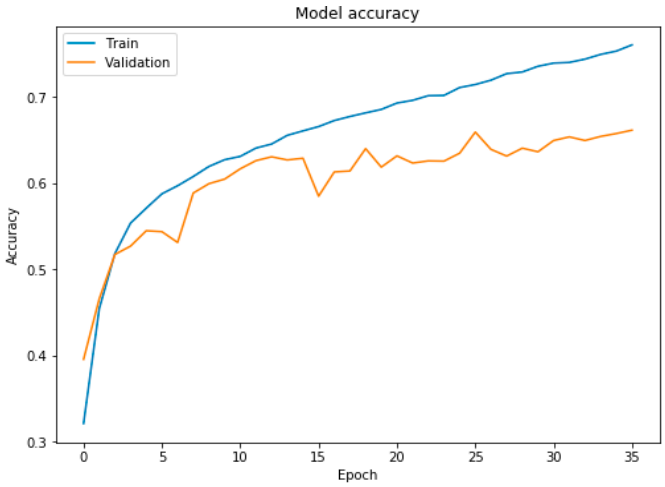


Figure 10. Graphical representation of training.

4. Experimental Results

The detailed results are shown in this section. We have divided the entire architecture to carry out two main tasks. The first task of face recognition has achieved an accuracy of 100%, as all the faces that were trained are recognized correctly, as open face has provided near-human accuracy [20] on the LFW benchmark. So, we have used it and implemented it in real time video with the help of the Raspberry-Pi 3 B+ model. We have used Python with OpenCV and with the help of Pi-Cam we acquired the live video and recognize the faces in those live videos. The proposed framework can locate multiple faces in the frames but is able to recognize only those that are already trained and present in the dataset. Face recognition has given correct results even with different objects like spectacles. We have installed the setup along with the biometric attendance area so that, at the time of punching in and punching out, the expression on the faces of employees can be recorded and, after collecting the data of the fortnight, the recorded faces with names and recognized expressions can be analysed. This analysis can be useful to recognize the consistent behavior of the employees in private organizations. For example, an employee with a constant sad or disgusted expression on his face can be identified and can be reported to a happiness cell or psychological support cell for helping such employees and making them feel good and comfortable in the workplace. The hardware setup of the proposed system is shown in Figure 8.

The comaparison with various models is shown in Table 3 and specifications of the system are given in Table 4, and the detailed algorithm is explained in Algorithm 1 and Algorithm 2. After detecting the face in real time, the cropped and pre-processed image is given to the pre-trained deep network which is trained on the FER 2013 dataset using Python and Keras. The results after classification are shown in the confusion matrix, as shown in the Table 5. Several misclassifications have been found as “disgust” is misclassified as “Angry”. From the dataset one can easily locate that the count of disgusted faces in the dataset is least 547. This simply indicates that the number of features that the network has been trained to classify as a disgusted face is less compared to other classes. Hence, the misclassification took place.

Table 3. Comparison with various Models.

Model	Accuracy	Learning Rate	Test Accuracy	Optimizer	Regularization	Activation Function
Mini_Xception	73%	0.005	68%	Adam, SGD	L1	ReLU
Densenet161	59%	0.001, 0.001, 0.005	43%	Adam, SGD	L2	Sigmoid
Resnet38	68%	0.0001	60%	SGD, AdaGrad	L1	Sigmoid
Mobilenet_V2	72.5%	0.0001, 0.001	64%	AdaGrad, Adam	L2	ReLU

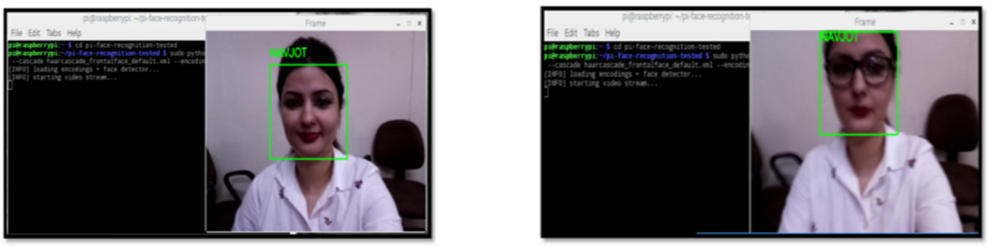
Table 4. System configuration.

Name	Configuration
Imaging Libraries	OpenCV 2.4.11, imutils, dlib v18.16, Scikit-Learn, Scikit-Image, OpenFace
Libraries	Matplotlib, RPI.GPIO, Numpy, SciPy, PyLab,
Programming Languages	Python 2.7
Operating System	NOOBS

Table 5. Normalized Confusion Matrix of the testing dataset without augmentation.

True Label		Predicted Label						
	Angry	**0.68**	0.01	0.05	0.04	0.08	0.03	0.10
	Disgust	0.47	**0.44**	0.02	0.02	0.00	0.04	0.02
	Fear	0.20	0.00	**0.37**	0.03	0.15	0.12	0.12
	Happy	0.03	0.00	0.01	**0.89**	0.01	0.03	0.03
	Sad	0.14	0.00	0.09	0.06	**0.45**	0.02	0.24
	Surprise	0.04	0.00	0.07	0.05	0.01	**0.81**	0.02
	Neutral	0.08	0.00	0.03	0.06	0.08	0.02	**0.73**
		Angry	**Disgust**	**Fear**	**Happy**	**Sad**	**Surprise**	**Neutral**

The total number of parameters for which the network has been trained is 2,134,407. When tested on real time video, 110 out of 120 images with expressions are recognized correctly. Figure 10 is the graphical representation of model accuracy and Figure 12 is the graphical representation of model loss. Figure 11 shows the real time face recognition result. Table 3 shows the comparison of available models.



(**a**) (**b**)

Figure 11. Face recognition result in real-time (**a**) without spectacles (**b**) with spectacles.

Algorithm 1. Face Detection in Real-Time.

\ Input: Real time video of subjects I
I. Capture the real time image I from the real time video frames
II. Facial Dataset creation
For k = 1 to size of $(\delta|TR)$= $\mathbb{N}\left(\delta_R = \left\{I_1^r, I_2^r \ldots I_n^r\right\}\right)$, $\mathbb{N}$ is the count of each subjects sample and n is the total number of subjects captured
  a. Select an image I_1^r from δ_R
  b. Convert the image I_1^r to grayscale image: I_G^r Gray Scale $(I|1^r)$
  c. Detect the face region using Histogram of Oriented Gradients (HOG): I_H^r HOG I_G^r
  d. Crop the facial region I_{CR}^r Cropped I_H^r
  e. Preprocess the cropped image I_{CR}^r by applying facial landmark and affine transform: I_{pr}^r Pre-processing I_{CR}^r
  f. Repeat the sub steps a to e of II of database creation for n x Ď times
End
III. Label all $nD\left(I_{pr}^r\right)$ images of dataset with the name of the subjects: $(\delta|TR)^l$ Labeled $nD\left(I_{pr}^r\right)$
IV. Training and feature extraction:
For k = 1 to size of $(\delta|TR)^l$,where l represents labelled data
  a. Select the anchor image $y_j^m = \left(1.I_1^r\right)$ of first subject
  b. Select the positive image $y_j^p = \left(2.I_1^r\right)$ of the same subject
  c. Select the negative image.$y_i^n = \left(1.I_2^r\right)$ of the second subject
  d. Feed the images y_j^m, y_j^p, y_i^n to the pre-trained deep network
  e. Repeat the training to achieve $y_j^m - y_j^{p}{}_2^2 + \beta y_j^m - y_j^{n}{}_2^2 \forall \left(y_j^m, y_j^n, y_j^p\right) \in \zeta$,where β is the enforced margin between negative and positive pair of images and ζ is the set of all the possible triplets and has numbers equal to number M
  f. Generate multiple triplets to improve deep learning by $\sum_j^M$
  g. Generate the feature vector Φ_n^{embd},where n is the number of embedding's generated by deep network
End
V. Cross validate by capturing image I^r in real time
  i. Repeat the substeps a to e of II Facial Dataset creation for I^r
  ii. Repeat the substeps a to f of IV Training step and feature extraction
VI. Feed the image to classifier to classify the image I^r in real time by passing all the feature vectors Φ_n^{embd} called embedding's that were generated in the substeps a to g of IV Training step and the Output: Prediction of the face of the subject with name in real-time\

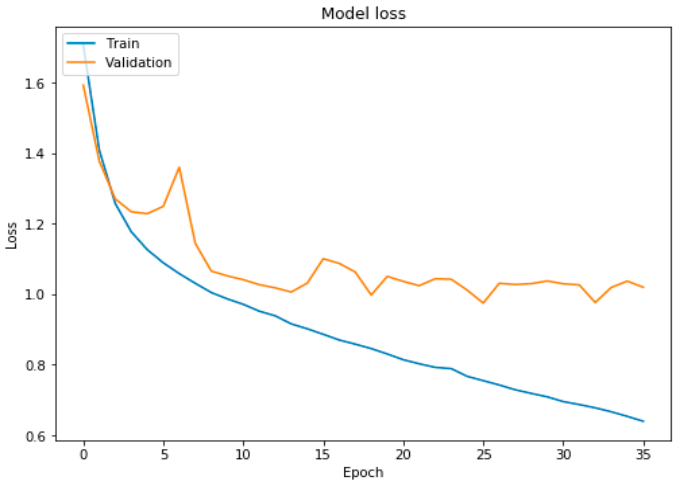


Figure 12. Graphical representation of loss while training.

Algorithm 2. Emotion Detection in Real-Time.

\Input Emotive facial dataset $\mathcal{F}$
I. Divide the dataset into training $\mathcal{F}_{TR}$,validation $\mathcal{F}_{VD}$ and testing $\mathcal{F}_T$
II. Training and feature extraction:
For j = 1 to size of) = $I^{TR}_{4N/5}$,where N is the total number of images in $\mathcal{F}$
  a. Select the image I^{TR}_1 from $\mathcal{F}_{TR}$
  b. Feed the input I^{TR}_1 to the deep network
  c. Train the network by passing the *images*) with their labels and let the network extract all the parameter
End
III. Cross Validate:
For j = 1 to size of ($\mathcal{F}_{VD} = \left\{ I^{VD}_{(4N/5)+1} \dots I^{VD}_{9N/10} \right\}$
  a. Select the image $I^{VD}_{(4N/5)+1}$ from $\mathcal{F}_{VD}$
  b. Repeat the substeps b and c from II Training and feature extraction for images $\mathcal{F}_{VD}$
  c. Use validation images $\mathcal{F}_{VD} = \left\{ I^{VD}_{(4N/5)+1} \dots I^{VD}_{9N/10} \right\}$ the reduction of overfitting
End
IV. Testing:
For j = 1 to size of $\mathcal{F}_T = \left\{ I^{T}_{(9N/10)+1} \dots I^{T}_{N} \right\}$
  i. Select the image $I^{T}_{(9N/10)+1}$ from $\mathcal{F}_T$
  ii. Repeat the substeps b and c from II Training and feature extraction for images $\mathcal{F}_T$
  iii. Use testing images $\mathcal{F}_T = \left\{ I^{T}_{(9N/10)+1} \dots I^{T}_{N} \right\}$ to test the trained network for efficiency
End
V. Real time testing:
Take the input from subset e of II from Algorithm 1
For k = 1 to size of $(\delta|TR) = \mathbb{N}(\delta_R = \{I^r_1, I^r_2 \dots I^r_n\})$, $\mathbb{N}$ is the count of each subjects sample and nis the total number of subjects captured
  i.Resize the image I^r_{pr} Pre-processing I^r_{CR}
I^r_{RS} Resize I^r_{pr}
  ii. Repeat the substeps b and c from II Training and feature extraction for images I^r_{RS}
End
VI. Predict the facial expression $\{\varepsilon_c = \varepsilon_1 \varepsilon_{2 \dots} \varepsilon_7\}$ where the basic classis of seven emotions is represented as c = [Angry, disgust, fear, Happy, Sad, Surprise and Neutral].Output: Prediction of facial emotion of subject in real-time\

To evaluate the performance and effectiveness of proposed edge device, we have compared with the previous studies on facial emotional detection. It has been realized that the hardware implementation for facial emotion detection and recognition is less implemented. A few studies that have implemented this hardware recorded lower accuracy, like 51.28% and 47.44%, when compared with the proposed model, i.e., 68%.

As discussed in the above section, the FER 2013 dataset is not a balanced dataset. A total of 35,887 images of 7 classes are present in this dataset. The unbalanced dataset gave the results which are very low, specifically for disgust, fear, and sad emotion. So, a data balancing technique is used for balancing the data. Keras API helps to increase the data set by applying various techniques by using the Image Data Generator function. This mainly includes five functions, i.e., rotation at a certain angle, shearing, zooming, rescale, and horizontal flip. Before data augmentation, a total of 35,887 images were used, out of which only 547 images were of disgusted expressions. After applying data augmentation, a total of 41,904 images were used, of which 6564 were of disgusted faces. The confusion matrix after data augmentation is shown in Table 6.

Table 6. Confusion Matrix of the testing dataset after data augmentation.

True Label	Angry	Disgust	Fear	Happy	Sad	Surprise	Neutral
Angry	**0.68**	0.01	0.05	0.04	0.08	0.03	0.10
Disgust	0.47	**0.54**	0.02	0.02	0.00	0.04	0.02
Fear	0.20	0.00	**0.50**	0.03	0.15	0.12	0.12
Happy	0.03	0.00	0.01	**0.89**	0.01	0.03	0.03
Sad	0.14	0.07	0.05	0.06	**0.45**	0.02	0.24
Surprise	0.04	0.01	0.07	0.03	0.01	**0.81**	0.02
Neutral	0.08	0.01	0.03	0.06	0.08	0.02	**0.73**
	Angry	**Disgust**	**Fear**	**Happy**	**Sad**	**Surprise**	**Neutral**
	Predicted Label						

Table 6 shows the confusion matrix after data augmentation. It can be noticed that prediction for disgust and fear has improved. The overall efficiency of system after data augmentation has been raised by 2% and came out as 68%. Table 7 shows the comparison of proposed edge device with previous studies.

Table 7. Comparison of proposed edge device with previous studies.

Research	Objective	Hardware Based Device	Cloud Server	Algorithm	Accuracy
[26]	Face emotion recognition	No	No	Hybrid CNN-RNN	94.91
[27]	Facial expression recognition	No	No	CNN	NA
[28]	Emotional Recognition in the Wild	Yes	No	CNN	NA
[29]	Facial Expression Emotion Detection	AtlysTM Spartan-6FPGA development board	No	SVR (Support Vector Regression)	MATLAB Simulink: 51.28% Xlinix simulation: 47.44%
Proposed	Facial emotion & detection	Raspberry-Pi based standalone edge device	Yes	CNN + SVM	Raspberry-Pi-based edge device: 68%

5. Conclusions

Real time detection of any kind of activity that is suspicious in nature is difficult to identify without any actual interaction with the subject or suspect. Reading the face of a person in real time is a challenging task. With the help of compact and portable devices, it becomes easy for the majority of organizations to understand the behavior of their employees and resolve some of the minor and major issues at an early stage. To achieve that, a framework has been tested and proposed that can be implemented in any organization to understand employee behavior. The proposed framework is a cost-effective and compact alternative over all those heavy and bulky systems that are difficult to implement in real time. The system has been tested for 20 different people with all 7 emotions, and out of total 120 images, 110 images were identified with correct emotions in real time. The proposed framework has been implemented using the Mini-Xception Deep Network because of its computational efficiency in a shorter time as compared to other networks.

Facial expression representation plays an important role in facial expression recognition. It can be viewed as generating good features for describing the appearance, structure, and motion of facial expressions. More specifically, facial expression features attempt to effectively describe the facial muscle or facial motion for static or dynamic facial images. Numerous works have already done this and, although different proposed methods for

facial expression recognition have achieved good results, there remain different problems that need to be addressed by the research community. The most important one is face variability in a single person. There are many factors that can cause two pictures from the same person to look totally different, such as light, face expression, or occlusion. Another problem to be taken into account is the environment. Except in controlled scenarios, face pictures have very different backgrounds, which can make the problem of face recognition more difficult. To address this issue, many of the most successful systems focus on treating the face alone, discarding all the surroundings. Smart meeting, video conferencing, and visual surveillance are some of the real-world applications that require a facial expression recognition system that works adequately on low resolution images. There exist lots of methods for facial expression recognition but very few of those methods provide results or work adequately on low resolution images. More research effort is required to be put forth for recognizing more complex facial expressions than the six classical ones, such as fatigue, pain, and mental states such as agreeing, disagreeing, lying, frustration, thinking, as they have numerous application areas. Other problems include expression intensity estimation, spontaneous expression recognition, micro expression recognition (brief, involuntary facial expression, lasts only 1/25 to 1/15 of a second), mis-alignment problems, illumination, and face pose variation. Moreover, studies proved that visual captures of facial expressions alone are not sufficient to identify the exact human emotions discussed in this section. This research can be further carried out by combining FER systems with various physiological sensors to identify the exact mental state of a person.

Author Contributions: N.R., A.G. and R.S. made contributions to conception and manuscript writing; A.S.A. and S.S.A. examined and supervised this research and outcomes; M.R. and Z.K. revised and polished the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by Taif University Research Supporting Project number (TURSP-2020/311), Taif University, Taif, Saudi Arabia.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data used in this research will be made available on request to corresponding author.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Wiener, M.; Mehrabian, A. *Language within Language: Immediacy, a Channel in Verbal Communication*; Ardent Media: Lake Geneva, WI, USA, 1968; ISBN 0891972684.
2. Kaulard, K.; Cunningham, D.W.; Bülthoff, H.H.; Wallraven, C. The MPI facial expression database—A validated database of emotional and conversational facial expressions. *PLoS ONE* **2012**, *7*, e32321. [CrossRef]
3. Zeng, J.; Shan, S.; Chen, X. Facial expression recognition with inconsistently annotated datasets. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 222–237.
4. Edwards, J.; Jackson, H.J.; Pattison, P.E. Emotion recognition via facial expression and affective prosody in schizophrenia: A methodological review. *Clin. Psychol. Rev.* **2002**, *22*, 789–832. [CrossRef]
5. Amos, B.; Ludwiczuk, B.; Satyanarayanan, M. Openface: A general-purpose face recognition library with mobile applications. *CMU Sch. Comput. Sci.* **2016**, *6*, 1–20.
6. Ashraf, A.B.; Lucey, S.; Cohn, J.F.; Chen, T.; Ambadar, Z.; Prkachin, K.M.; Solomon, P.E. The painful face–pain expression recognition using active appearance models. *Image Vis. Comput.* **2009**, *27*, 1788–1796. [CrossRef] [PubMed]
7. Ko, B.C. A brief review of facial emotion recognition based on visual information. *Sensors* **2018**, *18*, 401. [CrossRef]
8. Sajjad, M.; Nasir, M.; Ullah, F.U.M.; Muhammad, K.; Sangaiah, A.K.; Baik, S.W. Raspberry Pi assisted facial expression recognition framework for smart security in law-enforcement services. *Inf. Sci.* **2019**, *479*, 416–431. [CrossRef]
9. Sajjad, M.; Nasir, M.; Muhammad, K.; Khan, S.; Jan, Z.; Sangaiah, A.K.; Elhoseny, M.; Baik, S.W. Raspberry Pi assisted face recognition framework for enhanced law-enforcement services in smart cities. *Futur. Gener. Comput. Syst.* **2020**, *108*, 995–1007. [CrossRef]
10. Chowdhury, M.N.; Nooman, M.S.; Sarker, S. Access Control of Door and Home Security by Raspberry Pi Through Internet. *Int. J. Sci. Eng. Res.* **2013**, *4*, 550–558.

11. Agrawal, N.; Singhal, S. Smart drip irrigation system using raspberry pi and arduino. In Proceedings of the International Conference on Computing, Communication & Automation, IEEE, Greater Noida, India, 15–16 May 2015; pp. 928–932.
12. Chowhan, R.S.; Tanwar, R. Password-Less Authentication: Methods for User Verification and Identification to Login Securely Over Remote Sites. In *Machine Learning and Cognitive Science Applications in Cyber Security*; IGI Global: Hershey, PA, USA, 2019; pp. 190–212.
13. Srikote, G.; Meesomboon, A. Face Recognition Performance Improvement using a Similarity Score of Feature Vectors based on Probabilistic Histograms. *Adv. Electr. Comput. Eng.* **2016**, *16*, 107–113. [CrossRef]
14. Bashar, F.; Khan, A.; Ahmed, F.; Kabir, H. Face recognition using similarity pattern of image directional edge response. *Adv. Electr. Comput. Eng.* **2014**, *14*, 69–77. [CrossRef]
15. Lajevardi, S.M.; Hussain, Z.M. Feature extraction for facial expression recognition based on hybrid face regions. *Adv. Electr. Comput. Eng.* **2009**, *9*, 63–67. [CrossRef]
16. Haider, K.Z.; Malik, K.R.; Khalid, S.; Nawaz, T.; Jabbar, S. Deepgender: Real-time gender classification using deep learning for smartphones. *J. Real-Time Image Process.* **2019**, *16*, 15–29. [CrossRef]
17. Lu, H.; Huang, Y.; Chen, Y.; Yang, D. Automatic gender recognition based on pixel-pattern-based texture feature. *J. Real-Time Image Process.* **2008**, *3*, 109–116. [CrossRef]
18. Greche, L.; Akil, M.; Kachouri, R.; Es-Sbai, N. A new pipeline for the recognition of universal expressions of multiple faces in a video sequence. *J. Real-Time Image Process.* **2020**, *17*, 1389–1402. [CrossRef]
19. Yoon, J.; Kim, D. An accurate and real-time multi-view face detector using orfs and doubly domain-partitioning classifier. *J. Real-Time Image Process.* **2019**, *16*, 2425–2440. [CrossRef]
20. Lu, Y. Artificial intelligence: A survey on evolution, models, applications and future trends. *J. Manag. Anal.* **2019**, *6*, 1–29. [CrossRef]
21. Pannu, A. Artificial intelligence and its application in different areas. *Artif. Intell.* **2015**, *4*, 79–84.
22. Lemley, J.; Bazrafkan, S.; Corcoran, P. Deep Learning for Consumer Devices and Services: Pushing the limits for machine learning, artificial intelligence, and computer vision. *IEEE Consum. Electron. Mag.* **2017**, *6*, 48–56. [CrossRef]
23. Kazemi, V.; Sullivan, J. One millisecond face alignment with an ensemble of regression trees. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 1867–1874.
24. Schroff, F.; Kalenichenko, D.; Philbin, J. Facenet: A unified embedding for face recognition and clustering. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 815–823.
25. Sambare, M. FER-2013: Learn Facial Expressions from An Image. Available online: https://www.kaggle.com/msambare/fer2013 (accessed on 11 May 2021).
26. Bența, K.-I.; Vaida, M.-F. Towards real-life facial expression recognition systems. *AECE* **2015**, *15*, 93–102. [CrossRef]
27. Jain, N.; Kumar, S.; Kumar, A.; Shamsolmoali, P.; Zareapoor, M. Hybrid deep neural networks for face emotion recognition. *Pattern Recognit. Lett.* **2018**, *115*, 101–106. [CrossRef]
28. Zhang, H.; Jolfaei, A.; Alazab, M. A face emotion recognition method using convolutional neural network and image edge computing. *IEEE Access* **2019**, *7*, 159081–159089. [CrossRef]
29. Riaz, M.N.; Shen, Y.; Sohail, M.; Guo, M. Exnet: An efficient approach for emotion recognition in the wild. *Sensors* **2020**, *20*, 1087. [CrossRef] [PubMed]