Facial Expression Recognition and Face Recognition Using a Convolutional Neural Network

Suci Dwijayanti[1*], Rahmad Rhedo Abdillah[2], Hera Hikmarika[3], Hermawati[4], Zaenal Husin[5], Bhakti Yudho Suprapto[6]
Department of Electrical Engineering
Universitas Sriwijaya
Indralaya, Indonesia
sucidwijayanti@ft.unsri.ac.id

***Abstract*— The human face can be used in various biometrics procedures to identify an individual through face recognition or for facial expression recognition. However, not many studies have addressed the problem of face recognition along with facial expression recognition. In addition, some studies have directed more attention to finding the most suitable feature to extract and feed to a classifier. This study focused on addressing the problem using a convolutional neural network (CNN)-based method. Unlike other methods that require suitable features to be found, this study utilized raw images as the input to the CNN. A total of 16,640 images showing four facial expressions (normal, smiling, surprised, and angry) were used as input data. These data were obtained from 52 people and captured under outdoor conditions (in midday and the afternoon) using a webcam. The CNN-VGG was utilized because it is deep and fast enough for both face recognition and facial expression recognition purposes. The results showed that the VGG-f model architecture could overcome the underfitting and overfitting problems stemming from simpler CNN architectures. The testing results showed that the VGG-f model could recognize faces and facial expressions well. The average accuracies achieved in recognizing 104 faces during the day and in the afternoon were 86.5% and 90.4%, respectively. Additionally, the average accuracies achieved in recognizing the four different facial expressions of 52 people were 72% and 74% during the day and at noon, respectively. Recognition errors may have been caused by similarities between images.**

Keywords*— *Convolutional neural network (CNN), image processing, face recognition, facial expression recognition, VGG-f

I. INTRODUCTION

The term "biometrics" refers to techniques and procedures that utilize human anatomical and behavioral characteristics to identify an individual [1]. The facial pattern is a physiological biometric trait that has attracted the attention of both researchers and practitioners in the field because of its uniqueness and unconstrained acquisition [2]. Different biometric traits are utilized in various security systems, such as fingerprint scanners and iris scanners. Faces are also being utilized as a biometric for security systems.

There are many studies involving face or facial expression recognition. Verma et al. [3] used a Viola and Jones detector and a Gabor filter for feature extraction in a multi-layer perceptron to recognize facial expressions. Abdulrahman et al [4] used a combination of principal component analysis (PCA) and local binary pattern (LBP) to provide the features needed for a support vector machine (SVM). Gang et al. utilized an SVM together with geometric features to recognize facial expression using JAFFE database [5]. Chen et al. [6] performed facial expression recognition using a wavelet energy approach combined with a neural network ensemble based on a bagging algorithm. Guliang and Shoujue [7] proposed a hypersausage neural network that utilized an eigenface for feature extraction. Ebeid, in the face recognition experiment, [8] also utilized an eigenface for feature extraction to compare two neural network-based approaches: a multi-layer perceptron and radial basis function (RBF) neural network.

However, previous methods suffered from low accuracy, depending on the selected set of features to be extracted by the classifier; therefore, several researchers have attempted to overcome the feature dependency by using a convolutional neural network (CNN) [9-12]. Nevertheless, they have addressed face recognition and facial expression recognition tasks separately; for example, [9] discussed the usage of CNN for recognizing facial expressions, and [10] utilized CNN for face recognition. The problem of facial expression recognition remains challenging because of the non-rigid deformations in facial expressions. This study addresses the problems stated above by performing both face recognition and facial expression recognition using the same CNN architecture, thus yielding proper identification of a face with varying facial expressions. Also, this study utilized primary data obtained from the Indonesian faces database, thus providing sufficiently different facial features required for the experiment.

The remainder of this paper is organized as follows: section 2 gives an overview of the general structure of a CNN. Section 3 describes our proposed methodology, including data collection and the details on the CNN application to face recognition and facial expression detection tasks. Section 4 presents, analyzes, and discusses the experimental results obtained. Finally, Section 5 presents concluding remarks.

II. CONVOLUTIONAL NEURAL NETWORK

In this study, CNN was utilized as a classifier to identify both faces and facial expressions. The particular CNN has a 3D layered arrangement (width, height, and depth) and presents a type of deep learning machine combined with a multi-layer perceptron. The width and height refer to the layer size, while depth refers to the layer number, as shown in Fig. 1.

The work was funded by Unversitas Sriwijaya, SAINTEKS Research Grant No. SP DIPA-023.17.2.677515/2020

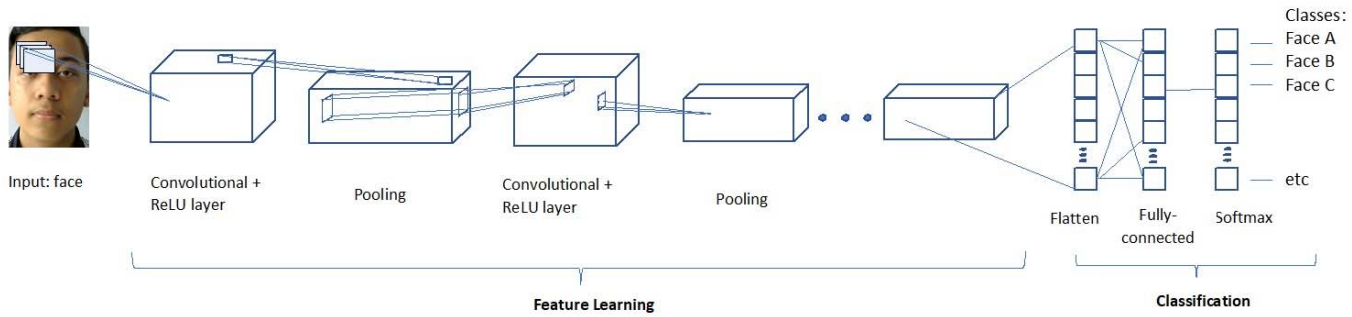


Fig. 1. CNN Architecture

The CNN architecture consists of several layers, namely, convolutional, rectified linear unit (ReLU), pooling, fully connected, and softmax layers.

The convolutional layer uses the convolution operation to refine and sharpen an image and to detect image edges. The convolution operation combines two different arrays to obtain a new array using the following equation:

$$h(x) = f(x) * g(x) = \int_{-\infty}^{\infty} f(a).g(x-a)da \qquad (1)$$

The * symbol represents the convolutional operation, and a is an auxiliary variable (dummy variable).

Since the operations performed in the digital image processing stage are discrete, we can rewrite (1) to the following form:

$$h(x) = f(x) * g(x) = \sum_{-\infty}^{\infty} f(a).g(x-a) \qquad (2)$$

In the above equation, $f(x)$ is the input signal while $g(x)$ is a convolutional kernel or filter kernel.

An ReLU is an activation function that performs a nonlinear operation and employs a rectifier [10]; a ReLU is commonly used in hidden layers to overcome the problem associated with missing gradients. It can be defined mathematically:

$$R(x) = \max(0, x) \qquad (3)$$

A pooling layer is a filter characterized by a certain stride and size; the most commonly used pooling layers implementations are average pooling and max pooling. An example of max pooling can be seen in Fig. 2.

As shown in Fig. 2, when max pooling with a 2×2 filter and stride 2 is performed, for each filter shift, the maximum value in a 2×2 area will be selected. Average pooling, on the other hand, will select an average value. The purpose of the pooling layer is to reduce the dimensions of the feature map in order to reduce computation time and generate a new feature map.

A multi-layer perceptron is said to be fully connected because each neuron from a previous layer is connected to every neuron of the classification layer. The feature extraction layer produces a feature map that is still in the form of a multidimensional array; thus, it needs to be flattened to be used in a fully connected layer.

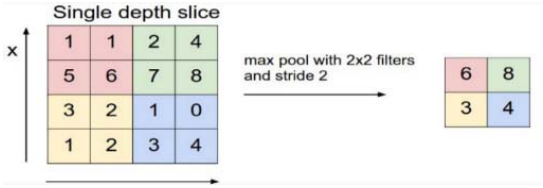


Fig. 2. Max Pooling Layer [13]

Softmax is an activation function used in an artificial neural network that takes an input vector of size K and normalizes it into the output probability distribution, consisting of K probability values. The softmax activation function is defined as follows:

$$\sigma(Z_j) = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}} \qquad (4)$$

The softmax activation function transforms the input into components in the interval (0, 1), all adding up to one.

III. METHODOLOGY

A. Data Collection

The required data were collected at the Control and Robotics Laboratory of Sriwijaya University. Image data in .jpg format were collected from 52 students using a high definition (HD) webcam. Every data element was a face image without any obscuring objects. Each individual was asked to make four different facial expressions: normal, smiling, surprised, and angry. This data collection process was performed in outdoor conditions, during the day (approximately 12 PM) and in the afternoon (approximately 4 PM). In total, 16,640 images were obtained. A data sample of four images showing an individual's expressions can be seen in Fig. 3.

Fig. 3. Sample Images of Facial Expressions (left to the right): Normal, Smiling, Surprised, and Angry

B. *CNN for Face and Facial Expression Recognition*

The proposed system consisted of feature extraction and the classification stage.

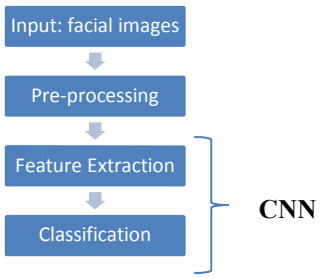


Fig. 4. Face recognition and Facial Expression Recognition Stages

As shown in Fig. 4, the input images were resized to 32 × 32 pixels in the preprocessing stage. The feature extraction was accomplished using the CNN in the convolutional layer with a ReLU activation function; this procedure was followed by max-pooling to filter the output so that the new feature maps could be used as the input of the classification stage. The output of the feature extraction was flattened to transform the feature map into a column vector; then, the column vector was connected, using a fully connected multi-layer perceptron, to a softmax output activation function. We have used Matconvnet [14] for the implementation of the CNN applied both to the face recognition and facial expression recognition processes.

To evaluate the CNN performance in recognizing the face and facial expression, we have calculated the recognition accuracy for testing data as

$$\text{Accuracy}(\%) = \frac{\text{number of positive recognitions}}{\text{Total number of data supplied}} \times 100\% \quad (5)$$

IV. Results and Discussion

A. *Training and Testing Face Recognition*

In the face recognition training process, data were divided into 52 classes that were labeled with the subject's name. Each class contained 250 images from the training dataset and 70 images from the validation dataset.

In the initial stage, we have used a simple CNN architecture, as shown in Table 1.

TABLE I. Simple CNN Architecture

Layer	Kernel Size
Convolutional_1 + ReLU	5 × 5
Max Pooling	2 × 2
Convolutional_2 + ReLU	5 × 5
Max Pooling	2 × 2
Convolutional_3	5 × 5
Fully Connected + Dropout	100
Softmaxloss	52

The parameters used in this architecture are shown in Table 2. In this simple architecture, input images were resized into 32 × 32 pixel format, and a dropout of 0.5 was used for regulation.

The objective of the training process during 100 epochs did not meet the set criteria because overfitting occurred, as shown in Fig. 5. The insufficient depth of the CNN architecture may have caused this problem; therefore, to improve the performance, the CNN depth was increased.

TABLE II. Parameters for the Simple Face Recognition CNN Architecture

Parameter	Value
Image size	32 × 32
Batch Size	256
Epoch	100
Learning Rate	0.001

Therefore, additional layers were introduced, which changed the architecture to resemble that of a CNN-VGG. The VGG-f architecture model was selected because it has the capability of processing large-size images. In addition, it did not cause an increase in device computation costs. The VGG-f architecture model and parameters used in training are presented in Tables 3 and 4, respectively. As shown in Table 4, the input images were resized into 224 × 224 pixels.

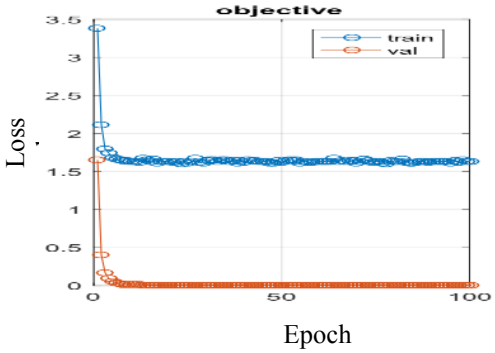


Fig. 5. Objective training of face recognition with simple CNN architecture

TABLE III. VGG-f Architecture Model [11,12]

Layer	Kernel Size
*Convolutional*_1 + ReLU	11 × 11
Max Pooling	3 × 3
*Convolutional*_2 + ReLU	5 × 5
Max Pooling	3 × 3
*Convolutional*_3 + ReLU	3 × 3
*Convolutional*_4 + ReLU	3 × 3
*Convolutional*_5 + ReLU	3 × 3
MaxPooling	3 × 3
*Fully Connected*_1 + *Dropout*	4096
*Fully Connected*_2 + *Dropout*	4096
Softmaxloss	52

TABLE IV. VGG-f CNN Architecture Parameters for Face Recognition

Parameter	Value
Image size	224 × 224
Batch Size	256
Epoch	30
Learning Rate	0.001

The CNN-VGG network was trained with a dropout of 0.5. A representation of the VGG-f model training can be seen in Fig. 6. As shown in the figure, the network loss value during 30 epochs was close to zero. The results of the training using the VGG-f architecture model were satisfactory because the network was well regularized without overfitting or underfitting; this architecture met the criteria of a good network. This model was applied to the test dataset obtained under two conditions: in the middle of the day and in the

afternoon. The total testing dataset included 104 images from 52 students. The results for 10 sample test datasets are shown in Table 5.

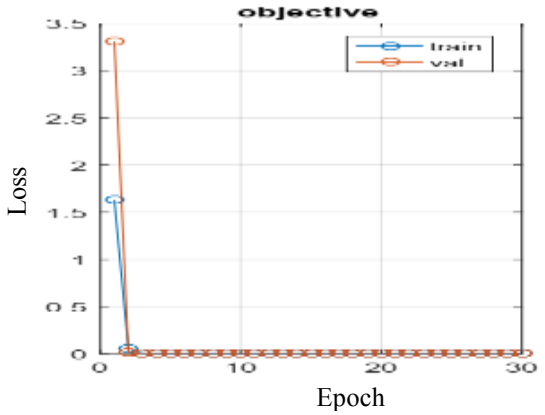


Fig. 6. Objective training of face recognition using the CNN-VGG-f architecture

TABLE V. ACCURACY OF FACE RECOGNITION FOR 10 SAMPLE DATASETS

Name	Day		Afternoon	
	Recognition	Accuracy (%)	Recognition	Accuracy (%)
Sample 1	True	100	True	100
Sample 2	False	42,3	False	41.8
Sample 3	True	98,85	True	93.36
Sample 4	True	99,88	True	99.9
Sample 5	True	84,14	True	95.2
Sample 6	True	100	True	99.9
Sample 7	False	47,54	False	68.7
Sample 8	True	90,9	True	78.4
Sample 9	False	49,1	False	63.6
Sample 10	True	100	True	99.9

As shown in Table 5, the CNN was able to recognize faces well; the proposed CNN demonstrated high recognition accuracy of up to 100%.

However, errors still occurred, as can be seen in Fig. 7; this may be caused by facial resemblances in individuals. As shown in the figure on the right, this facial image was not recognized properly because the face was similar in appearance to the face in the figure on the left. Therefore, this may have caused the CNN to confuse the person in the figure on the right with the person in the figure on the left. The CNN even demonstrated a low accuracy of 49.144%.

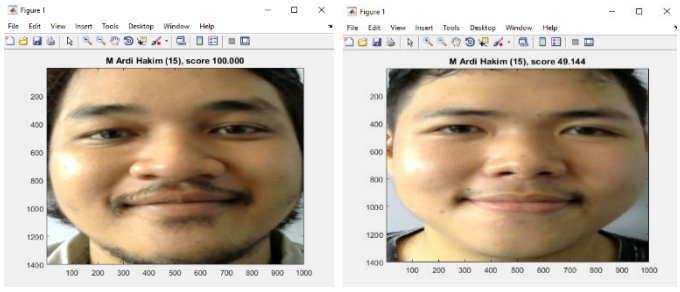


Fig. 7. Errors in recognition: True (left) and False (right)

However, the accuracy of the CNN model in terms of face recognition was typically high. The accuracy of all the testing data was calculated using (5). The results showed that from 52 face image files, the model was able to recognize 45 faces, or 86.5%, photographed in the middle of the day and 47 faces, or 90.4% photographed in the afternoon. This may be caused by the higher photographic quality of images taken in the afternoon, such as the absence of shadows, better lighting conditions, and sharpness.

B. Training and Testing Facial Expression Recognition

For facial expression recognition, the training data consisted of four classes, which contained 3,295 photo images for the training dataset and 865 images for the validation set. The initial training stages used the same architecture and parameters as used in face recognition.

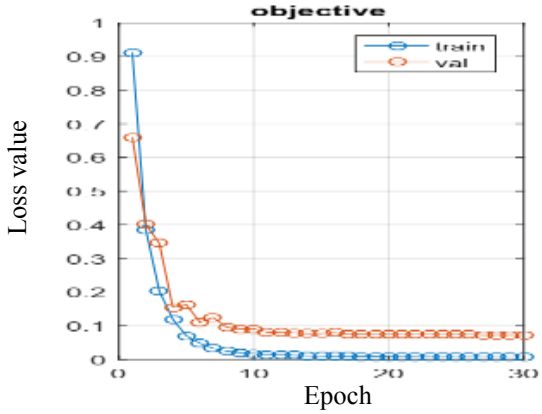


Fig. 8. Objective training of facial expression recognition using CNN-VGG f architecture

The results of objective training using a simple CNN also showed that such CNN suffered from overfitting. The CNN-VGG model VGG-f layer was then used with the same parameters to train facial expression recognition. The softmax layer was adjusted to the number of classes used in the facial expression recognition process.

Fig. 8 shows the training results of the CNN-VGG-f network for the facial expression recognition case. From the figure, it can be seen that the model was adequate because the loss value was small. The training and validation loss values were close, indicating that no overfitting or underfitting occurred in the model.

Tests were run, using images of the participants from the face recognition experiment, but this time showing various facial expressions. The photographs were taken outdoors, in the middle of the day and afternoon. The results from the facial expression recognition can be seen in Table 6.

From Table 6, it can be seen that the proposed facial expression recognition network functioned well in terms of recognizing different facial expressions. Nevertheless, errors occurred in four expressions out of 20 images taken in the middle of the day.

Such errors may occur owing to similarities between given expressions. An example is shown in Fig. 9: in the figure, the normal and smiling expressions are quite similar, so the network may erroneously recognize the expression in the figure on the right as a smile.

TABLE VI. ACCURACY OF FACIAL EXPRESSION RECOGNITION USING 10 SAMPLE DATASETS

Name	Day			Noon		
	Expression	Recognition	Acc. (%)	Expression	Recognition	Acc. (%)
Sample 1	Smile	True	99.9	Smile	True	99.6
	Anger	True	78.0	Anger	True	76.7
Sample 2	Anger	True	99.0	Smile	True	99.9
	Smile	True	99.9	Surprise	True	99.8
Sample 3	Smile	True	99.7	Smile	True	99.9
	Surprise	True	98.9	Surprise	True	92.0
Sample 4	Normal	False	70.0	Normal	False	97.5
	Smile	True	99.9	Smile	True	99.9
Sample 5	Normal	False	41.7	Normal	False	30.0
	Smile	True	99.9	Smile	True	99.9
Sample 6	Normal	False	59.5	Normal	False	82.4
	Smile	True	99.3	Smile	True	99.8
Sample 7	Smile	True	99.7	Smile	True	99.9
	Normal	True	75.7	Normal	False	95.9
Sample 8	Normal	True	69.5	Normal	True	75.1
	Smile	True	99.7	Smile	True	99.5
Sample 9	Smile	True	99.9	Smile	True	99.9
	Anger	True	72.5	Anger	True	75.5
Sample 10	Normal	False	67.7	Normal	False	94.9
	Smile	True	99.9	Smile	True	99.9

After comparing facial expression recognition accuracy for midday and afternoon images, as shown in Fig. 10, we can see that the expression of surprise yields the highest accuracy for both cases. It implies that the expression of surprise contains unique features that were extracted and learned in the feature learning stage. As shown in Fig. 3, the surprise usually makes people open the mouth.

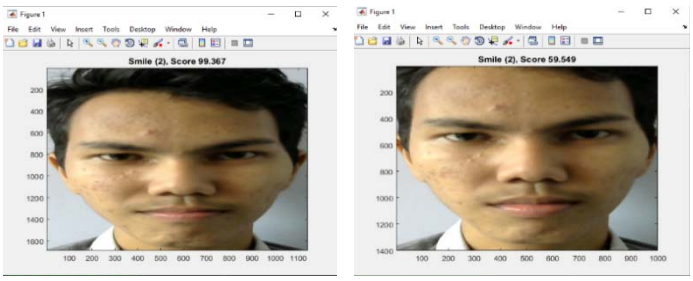
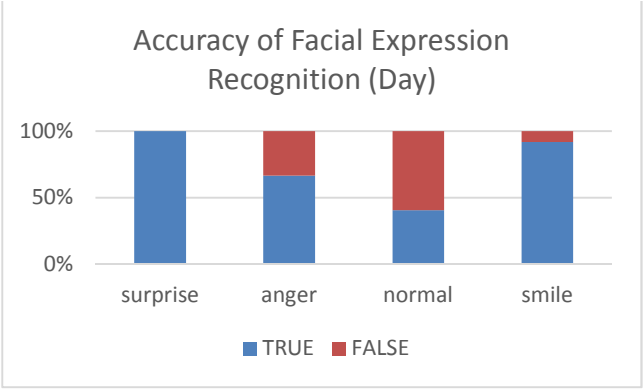


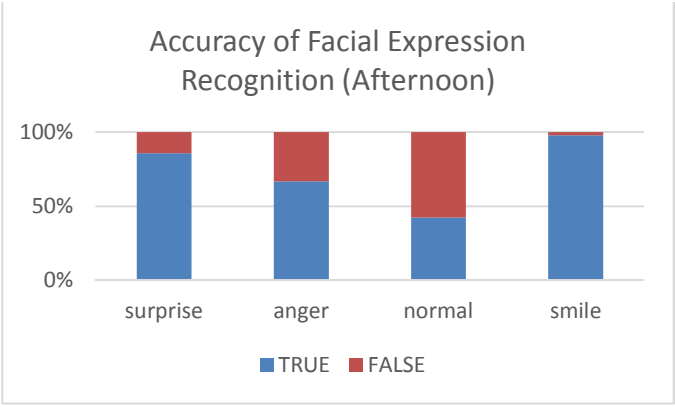
Fig. 9. Error in facial expression recognition: True (left) and False (right)

The facial expression labeled as normal had the lowest recognition accuracy since it is rather difficult to distinguish the characteristic traits of this expression from the smile or anger. Thus, the CNN model may misclassify this expression using the testing data.

However, the CNN was still able to recognize facial expressions adequately. The accuracies in terms of recognition for all facial expressions in midday and the afternoon were 72% and 74%, respectively. In other words, the CNN model was able to recognize 75 expressions in the middle of the day and 77 expressions in the afternoon. The results also indicate that the images taken in the afternoon were of better quality compared to ones taken in the midday.



(a)



(b)

Fig. 10. Accuracy of each facial expression in the midday (a) and the afternoon (b)

From the results of the face recognition and facial expression recognition experiments conducted, it is clear that the CNN-VGG architecture using the VGG-f model is able to overcome the problems that occur with the simpler CNN architectures, such as overfitting. Furthermore, this model is suitable for both face recognition and facial expression recognition tasks. Raw images can be utilized directly without employing a feature extraction stage, which suggests that CNN may reduce the computation cost caused by feature extraction. These results also show that face recognition and facial expression recognition have a high potential for use in a security system utilizing biometrics.

V. CONCLUSION

This study showed that CNN could produce a good performance for face recognition and facial expression recognition tasks using deep VGG-f architecture. In addition to the architecture selected, the parameters used may have also influenced the training process. The CNN was able to extract and capture the features provided by the raw image data. The results showed that the network was able to perform face recognition with an accuracy of up to 100% and facial expression recognition with an accuracy of up to 99.9%. The accuracy in face recognition and facial expression recognition was not affected by the time of day when the data was collected (middle of the day or in the afternoon). Notably, in face recognition, the face similarity features may influence the CNN performance in identifying a person. On the other hand,

facial expression recognition performance may be affected by the similarity of a particular expression in the training data fed to the CNN.

The experiment conducted in this study also presents that this CNN model for face recognition and facial expression recognition can be utilized in a variety of applications, including security systems. This implementation will be performed for the future study of this work.

REFERENCES

[1] A. K. Jain, K. Nandakumar, and A. Nagar, "Biometric template security," EURASIP Journal on Advances in Signal Processing, vol. 2008, pp. 1-17, 2008..

[2] S. Bakshi, S. Kumari, R. Raman, and P.K. Sa, "Evaluation of periocular over face biometric: a case study," Procedia Engineering. vol. 38, pp. 1628–33, 2012

[3] K. Verma and A. Khunteta, "Facial expression recognition using Gabor filter and multi-layer artificial neural network," In International Conference on Information, Communication, Instrumentation and Control (ICICIC), India, pp. 1–5, 2017.

[4] M. Abdulrahman, A. Eleyan, A. Kelimeler, İ. Yüz, and Y. İ. Örüntü, "Facial expression recognition using support vector machines," In 23rd Signal Processing and Communications Applications Conference (SIU), Turkey, pp. 14–17, 2015.

[5] L. Gang, L. Xiao-hua, Z. Ji-liu, and G. Xiao-gang, "Geometric feature based facial expression recognition using multiclass support vector machines," In IEEE International Conference on Granular Computing, China, pp. 1-4, 2009.

[6] C. Feng-jun and W. Zhi-liang, "Facial expression recognition based on wavelet energy distribution feature and neural network ensemble," In WRI Global Congress on Intelligent Systems, China, pp. 122–126, 2009.

[7] Z. Guliang and W. Shoujue, "Hypersausage networks and its application in face recognition," In International Conference on Neural Networks and Brain, China, pp. 1519–1522, 2005.

[8] R. M. Ebeid, Using MLP and RBF neural networks for face recognition: an insightful comparative case study," In International Conference on Computer Engineering & Systems, Egypt, pp. 123–128, 2011.

[9] X. Chen, X. Yang, M. Wang, and J. Zou, "Convolution neural network for automatic facial expression recognition," In International Conference on Applied System Innovation (ICASI), China, pp. 814–817, 2017.

[10] M. Coşkun, A. Uçar, O. Yildirim, and Y. Demir, "Face recognition based on convolutional neural network," In 2017 International Conference on Modern Electrical and Energy Systems (MEES), pp. 376–379, 2017.

[11] P. Wozniak, H. Afrisal, R. G. Esparza, and B. Kwolek, "Scene recognition for indoor localization of mobile robots using deep CNN," In Proceedings of the International Conference on Computer Vision and Graphics (ICCVG), Poland, 2018.

[12] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," Neural Information Processing System, vol. 25, Jan., 2012.

[13] S. Samuel, "Recognition deep learning part 7: convolutional neural network," 2017. [Online]. Available: https//medium.com/@samuelsena/Recognition-deep-learning-part-7-convolutional-neural-network-cnn-b003b477dc94. [Accessed: 27-Feb-2020].

[14] A. Vedaldi and K. Lenc, "Matconvnet: convolutional neural networks for Matlab, "In Proceedings of the 23rd ACM International Conference on Multimedia, pp. 689-92, 2015.