ISSN: 2230-9926

Available online at http://www.journalijdr.com

IJDR

International Journal of Development Research
Vol. 07, Issue, 11, pp.16960-16967, November, 2017

IJDR
International Journal of Development Research
VOL.07, ISSUE 11, NOVEMBER 2017
2017

ORIGINAL RESEARCH ARTICLE OPEN ACCESS

FACIAL EXPRESSION RECOGNITION BY USING MODIFIED CONVOLUTIONAL NEURAL NETWORK (MCNN) AND MODIFIED GABOR FILTER

[1]Lafta Raheem Ali, [2]Dr. Haider Kadhim Homood and [3]Dr. Amer Saleem Elameer

[1]Postgraduate Candidate, Informatics Institute for Postgraduate Studies, ICCI, Baghdad, Iraq
[2]Department of Computer Science, Al- Al-Mustansiriya University, Baghdad, Iraq
[3]Informatic Institute for Postgraduate Studies (IIPS), ICCI, Baghdad, Iraq

ARTICLE INFO

Article History:

Received 14th August 2017
Received in revised form
29th September, 2017
Accepted 03rd October, 2017
Published online 29th November, 2017

Key Words:

Facial expression,
Convolutional Neural Network,
Gabor Filter**.**

Corresponding author

ABSTRACT

Facial expression recognition is an important topic in an image processing and pattern recognition. The expressions work as a mirror for what happened in the human mind, enable individuals to understand what their peers feel. This paper suggest develop a system for recognizing facial expressions, modified Gabor filter has been use for feature extraction, then classifying the facial expressions based on these extracted features by using a modified Convolutional Neural Network (MCNN). the results of using MCNN with Modified Gabor was (93%-99%) accuracy rate with very less time for both multi and single face respectively We have gotten a very good result comparing to the other methods in the same fields (PCA and Gabor filter. This results with Extended Cohn Kanade (CK+) and JAFFE Databases.

Citation: Lafta Raheem Ali, Dr. Haider Kadhim Homood and Dr. Amer Saleem Elameer. 2017. "Facial Expression Recognition by Using Modified Convolutional Neural Network (MCNN) and Modified Gabor Filter.", *International Journal of Development Research*, 7, (11), 16960-16967.

INTRODUCTION

Expression shows the internal changes that are happens inside the human mind at any time [Ekman, 1997]. Facial expressions play a critical role in human communication, where the emotional states for human are convey and perceived through it, besides that, facial expression is consider the stronger way for representing the demeanor of human [Lucey et al., 2010]. The basic emotions are divided into seven emotions (six emotions which are: angry, sadness, happiness, surprise, fear, and disgust, besides the natural expression) [Ekman, 1997]. As indicated by Mehrabian (1968), the contribution of the verbal section of a message is only for 7% from the whole message effect, while the vocal section for 38% and finally facial expressions for 55% to the speaker's message effect [Arumugam, 2011; Rashi Goyal and Tanushri Mittal, 2014]. In this paper, a proposed system for recognizing facial expressions based on micro-expressions has been developed by using a convolutional neural network and modifying it to get high accurate and less average rate time classification method. The input of the system is a facial image while the output is the emotion type.

Related works

This part reviews some previous works which are linked to this research and related to these three components:

Shan et al (2017) proposed a system for recognizing facial expressions, where Haar-like features which represent the first part of the Viola-Jones approach is used to detect the face region, while CNN algorithm is used for extracting the features and classifying the facial expressions. Both the Japanese Female Facial Expression Database (JAFFE) and the Extended Cohn-Kanade Dataset (CK+) are used. The accuracy performance of the proposed system reaches 76.7442% and 80.303% in the JAFFE and CK+, respectively [Shan et al., 2017].

Lopes et al. (2017) proposed a method for achieving facial expression recognition system by using Convolutional neural network (CNN) for feature extracting and classifying facial expressions, the result was 96.76% of accuracy in the CK+ database beside it was fast to train [Lopes et al., 2017]. **Al-Sumaidaee et al. (2015)** proposed a novel method for facial expression recognition based on the combination of using Gabor filters and Local Gradient Code-Horizontal Diagonal (LGC-HD) features, the results on the JAFFE and CK databases showed high recognition rate of 93.33% and 90.62% respectively [Al-Sumaidaee et al., 2015].

Gabor filter

The images contain particular angles have features that aligned with it, and for extracting these features Gabor filters can be applied for this mission. The properties of the optimal localization in both frequency domains and spatial domains are acquired by Gabor filter. The frequency and angle represent the most significant parameters of the Gabor filter. Particular features that share similar frequency or angle can be chosen and used for individualizing among various facial expressions depicted in images [Liu et al., 2015]. The linear filter that used to detect the edge in the spatial domain is a Gabor filter. Gabor filter has two components to represent the orthogonal directions, these components are real and imaginary. Those two components can be formed into a complex or may be used individually. Equation 1 is used for computing real component of Gabor filter [Deepak Verma, 2012]

$$g(x,y,\theta,\lambda,\varphi,\sigma,\gamma) = e^{\left(-\frac{x'^2+\gamma^2 y'^2}{2\sigma^2}\right)} \times \cos\left(2\pi \frac{x'}{\lambda} + \varphi\right) \qquad \text{Eq. 1}$$

While equation 2 is used for computing imaginary component of Gabor filter [Yousra, 2009]:

$$g(x,y,\theta,\lambda,\varphi,\sigma,\gamma) = e^{\left(-\frac{x'^2+\gamma^2 y'^2}{2\sigma^2}\right)} \times \sin\left(2\pi \frac{x'}{\lambda} + \varphi\right) \qquad \text{Eq. 2}$$

x' and y' parameters are given by the following equations 3 and 4:

$$x' = x\cos(\theta) + y\sin(\theta) \qquad \text{Eq. 3}$$

$$y' = -x\sin(\theta) + y\cos(\theta) \qquad \text{Eq. 4}$$

Where the (*x, y*) specify the position of pixel in the spatial domain, while the other samples represent parameters and illustrated in table 1.

Table 1: Gabor's parameters [Yousra, 2009]

Symbol	Parameter	Values
θ	*Specifies the* **orientation** *of the wavelet.*	$\{0, \frac{\pi}{8}, \frac{2\pi}{8}, \frac{3\pi}{8}, \frac{4\pi}{8}, \frac{5\pi}{8}, \frac{6\pi}{8}, \frac{7\pi}{8}\}$
λ	*specifies the* **wavelength** *of the sine wave*	$\{4, 4\sqrt{2}, 8, 8\sqrt{2}, 16\}$
φ	*Specifies the* **phase** *of the sine wave.*	$\{0, \frac{\pi}{2}\}$
σ	*Specifies the* **radius** *of the Gaussian.*	$\sigma = \lambda$
γ	*Specifies the* **aspect ratio** *of the Gaussian.*	1

The Gabor filter bank is exploited to extract the features. The filter bank and other several filters are created and used for extracting multi-scale and multi-orientation features from a given facial image. The filter bank that is commonly used consisting of Gabor filters of 8 orientations and 5 different scales, as shown in figure 1[Ali et al., 2015]

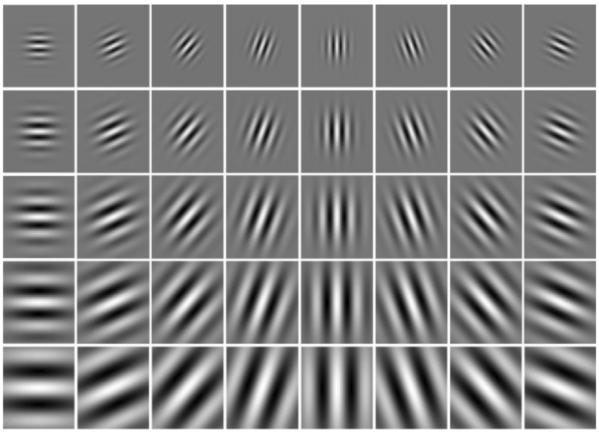


Figure 1: Gabor filter, 8 orientations and 5 different scales [Ali et al., 2015]

The representation of Gabor features for the facial image can be computed by convolution the image I(x, y) with the Gabor filters bank $g_{u,v}(x,y)$ for getting Gabor filter as shown in equation 5. The result of the convolution process is a complicated value, where the decomposition of it to real and imaginary part is possible as shown in equations 6 and 7 respectively [Deepak Verma, 2012; Hafez et al., 2015].

$$G_{u,v}(x,y) = I(x,y) \otimes g_{u,v}(x,y) \qquad \text{Eq. 5}$$

Where $G_{u,v}$ is Gabor filter of size *u* and angle *v*, $\otimes$ represent the convolutional operator.

$$E_{u,v}(x,y) = Re[G_{u,v}(x,y)] \qquad \text{Eq. 6}$$

$$O_{u,v}(x,y) = Im[G_{u,v}(x,y)] \qquad \text{Eq. 7}$$

E refers to even part, while *O* is the odd part. The amplitude $A_{u,v}(x,y)$ and the phase $\varphi_{u,v}(x,y)$ are both resulted from the decomposition of filtering, as shown in equations 8 and 9 respectively [Hafez, 2015]

$$A_{u,v}(x,y) = \sqrt{E^2{}_{u,v}(x,y) + O^2{}_{u,v}(x,y)} \qquad \text{Eq. 8}$$

$$\phi_{u,v}(x,y) = \arctan\left(\frac{O_{u,v}(x,y)}{E_{u,v}(x,y)}\right) \qquad \text{Eq. 9}$$

Convolutional Neural Networks (CNN)

CNN is an efficient recognition algorithm which is widely used in pattern recognition and image processing. It has many features such as simple structure, less training parameters and adaptability. It has become a hot topic in voice analysis and image recognition. Its weight shared network structure make it more similar to biological neural networks. It reduces the complexity of the network model and the number of weights [Liu et al., 2015]. Convolutional neural networks (CNNs) use images as input and convolution kernels, made up out of weights, as connections to the next layer.

These weights are arranged in 3D volumes and transforms the input image into an output of node activations [Habrman, 2016].

Convolutional Neural Networks Architecture

There are four operations for building any block of Convolutional Neural Network. These operations are Convolutional Layer, ReLU (Rectified Linear Unit), Pooling layer (Sub Sampling) and fully Connected Layer or as known (Classification). The work of these operations is as following

Convolutional Layer

This layer forms the basis of the convolutional Neural Networks and accomplishes the basic training operations, and therefore the network neurons are fired. The work of this layer is preserving the spatial relationship among pixels through learning image features by using input data in a form of small squares. Every image in this method can be represented as a matrix of pixel values [Wu, 2017] . The convolutional process can be expressed in more general as there is ($N \; x \; N$) input layer, and by using (m x m) filter f. The output size will be ($N - m + 1) \; x \; (N - m + 1$), so the convolutional can be computed by the equation 10:

$$x_{ij}^{l} = \sum_{a=0}^{m-1} \sum_{b=0}^{m-1} f_{ab} \, y_{(i+a)(j+b)}^{l-1}. \qquad \text{Eq. 10}$$

The overall process is illustrated in figure 2, where (a) represent is 2 x 2 kernel, the stride by 1 pixel each time, and (b) is 4 x 4 matrix which represent the convolutional input at left, at right the output of this process.

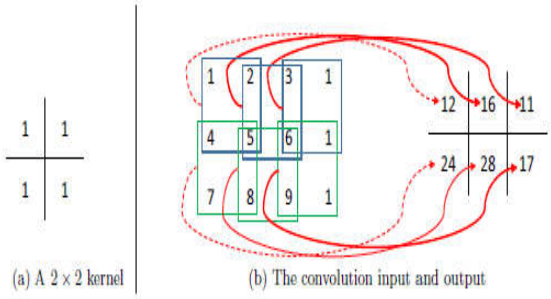


Figure 2: Demonstration of the Convolutional Operation. [Wu, 2017]. Before the beginning of convolutional process, three parameters must be decided for detect the size of activation map. Theses parameters are [Habrman, 2016]

- **Depth**: which represent the number of filters used in the convolution operation.
- **Stride**: represent the number of pixels by which the filter matrix will be slide on the input matrix.
- **Zero-padding**: this parameter provide the ability to add zero's around the border of input matrix.

ReLU (Non Linearity)

ReLU is a non-linear operation and it refers to Rectified Linear Unit. The work of this layer is to replace the values of each negative pixel in the activation map with zero[15]. ReLU provide a non-linearity to the CNN, where the most today's data which need to train by CNN is in form of non-linear data. The size of input will not be changed under this layer, and the output of ReLU is given by the function at 11 equation [Wu, 2017] which is illustrated in figure 3:

$$y_{i,j,d} = \max \left\{0, x_{i,j,d}^{l}\right\} \qquad \text{Eq. 11}$$

Where x^{l}, y represent the input and output respectively whose are the same size, and

$0 \leq i < H^{l} = H^{l} + 1$,
$0 \leq j < W^{l} = W^{l} + 1$, and
$0 \leq d < D^{l} = D^{l} + 1$

Here H, W, D refers to height, width and depth. ($W1 \times H1 \times D1$) is the size of input, (F) is spatial extent and (S) is the stride. The output of this layer is ($W2 \; x \; H2 \; x \; D2$)
Where:

- $W_2 = (W_1 - F)/S + 1$.
- $H_2 = (H_1 - F)/S + 1$.
- $D_2 = D_1$.

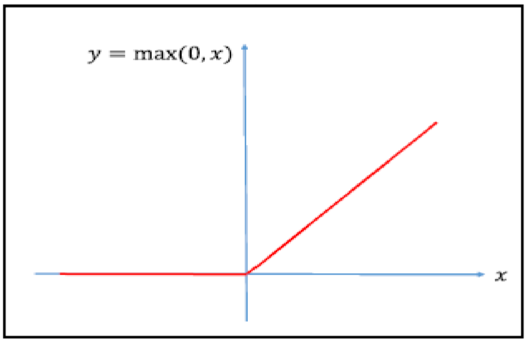


Figure 3. The ReLU function [Wu, 2017]

Pooling Layer

Subsampling (also called spatial pooling or down sampling). The work of this layer is reducing the dimensionality of each activation map with retains only the most significant information. There are various types of spatial pooling like *average, max, sum*, etc. Max pooling will be used in this work. Where at this type of pooling if there is 2 x 2 window and the stride is 2 then the result will be the largest element from the activation map in that window. Average pooling for that window or the sum of overall elements for the same window also can be used instead of max pooling. Equation 12 shows max pooling function[16].

$$a_j = \max_{N \, x \, N}(\, a_i^{n \, x \, n} \;\; u(n, n)) \qquad \text{Eq. 12}$$

Where n x n is the block region will be token for applying max-pooling on it, the output of this process is a single value which represent the biggest value at that block, and $N \times N$ represent the input layer, applies a window function $u(x, y)$ to the input patch, and computes the maximum in the neighborhood [16][17]. Max pooling operation is shown in figure 4.

Fully Connected Layer

This mean that every neuron in the previous layer is connected with each neuron in the next layer [Nash, 2015].

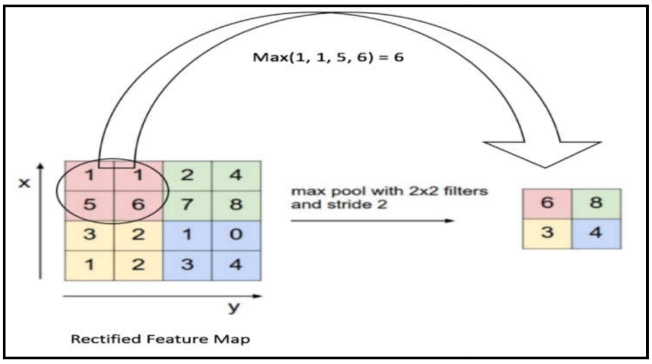


Figure 4. Max Pooling [Valero, 2016]

The output of the previous layers (convolutional and pooling) which represent high-level features of the inputted image. Fully connected layer will use these features to classify the input images into different classes depending on the training dataset. The features that result from convolutional and subsampling layers sometimes are good and equivalent for the classification task, but it is better when those features are combined together. The summation of output probabilities of the FC layer always equal to 1, which ensured by softmax function, where it take values between zero and one and the sum of these values. Equation 13 shows the softmax function [Amara, 2017]:

$$y_c = \varsigma(z)_c = \frac{e^{z_c}}{\sum_{d=1}^{C} e^{z_d}} \qquad for\ c = 1 \dots C \qquad \text{Eq. 13}$$

Where $(\varsigma)$the softmax function, (z) acts is the input, (y) is the output for real values between zero and one, (c) is number of inputs and each input may have one or more dimension (where C is the number of dimensions). The total error at the output layer can be got by using equation 5 [Amara, 2017]:

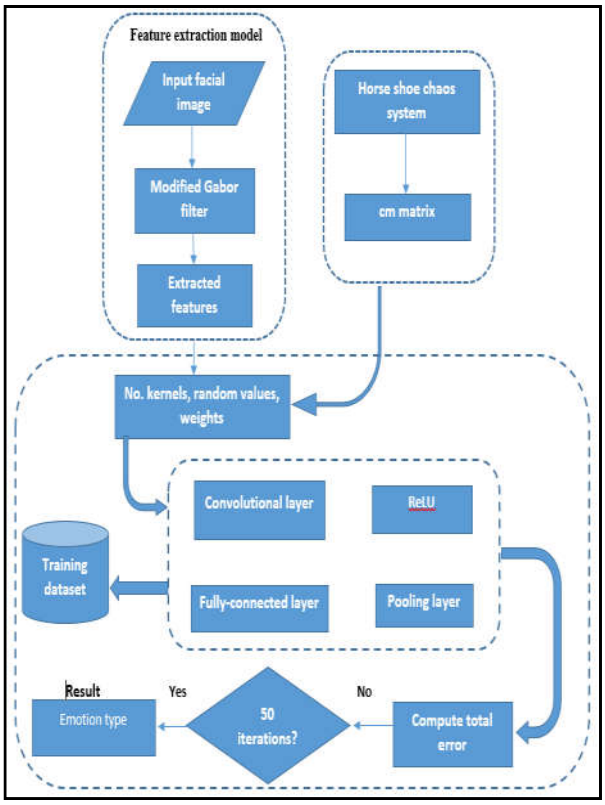


Figure 5. Proposed system architecture

Proposed system design

Modified Gabor filter used to extract the facial features, this method is achieved by applying two stages of modifications. The first proposed modifications are combine between the wvelet trnsform with the Gbor filter to increse the effeciency and speeding of the gbor filter. The second modification is using the control parameters and averaging of the transformed image in equation 5 to adjusted the features extrcted by Gbor filter and increse the accurcy of this opertion. The third modification is using the control parmeter to the equation 8 to adjust the value of the output ampltude tht effect in the next Gabor calculations. Modified Convolutional Neural Network (M-CNN) have been used to classify the type of emotion depending the extracted features by Modified Gabor filter. The modification to the CNN is by using the horseshoe chaos system in the learning phase in order to increase the learning speed and get more efficient classification with stable decisions for different time periods. Figure 5 shows the architectural model of the proposed system.

<u>Modified Gabor feature proposed algorithm</u>

Input: Scale (R, G, B) image I.
Output: Features that are delivered to the classifier.

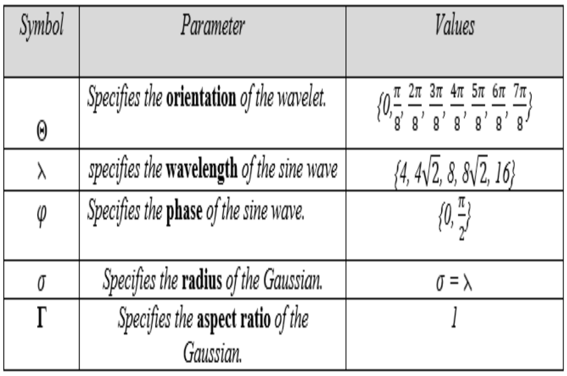
Step 1: Starts
Step 2: Apply 2-levels wavelet transform to the face image.
Step 2: For $scale = 0.1\ to\ 0.5$ do steps 3 to 9 to the LL transform image parts.
Step 3: Resized (I) with scale.
Step 4: For $\theta = 0\ to\ 180\ by\ 30\ do$ steps 5 to 9
Step 5: Apply equations (2.10) and (2.11) to compute xn' and yn'.

$$x' = x\cos(\theta) + y\sin(\theta)$$
$$y' = -x\sin(\theta) + y\cos(\theta)$$

Step 6: Apply equations (2.8) and (2.9) to compute both real-imaginary components of Gabor filter.

$$g(x, y, \theta, \lambda, \varphi, \sigma, \gamma) = e^{\left(-\frac{x\prime^2 + \gamma^2 y\prime^2}{2\sigma^2}\right)} \times \cos\left(2\pi \frac{x'}{\lambda} + \varphi\right)$$
$$g(x, y, \theta, \lambda, \varphi, \sigma, \gamma) = e^{\left(-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}\right)} \times \sin\left(2\pi \frac{x'}{\lambda} + \varphi\right)$$

with following parameters:



Symbol	Parameter	Values
Θ	Specifies the **orientation** of the wavelet.	$\{0, \frac{\pi}{8}, \frac{2\pi}{8}, \frac{3\pi}{8}, \frac{4\pi}{8}, \frac{5\pi}{8}, \frac{6\pi}{8}, \frac{7\pi}{8}\}$
⋋	specifies the **wavelength** of the sine wave	$\{4, 4\sqrt{2}, 8, 8\sqrt{2}, 16\}$
φ	Specifies the **phase** of the sine wave.	$\{0, \frac{\pi}{2}\}$
σ	Specifies the **radius** of the Gaussian.	σ = ⋋
Γ	Specifies the **aspect ratio** of the Gaussian.	1

Step 7: Compute the Gabor features by applying modify equation 2.12.

$$G_{u,v}(x,y) = gr * average(I(x,y)) + ((I(x,y) \otimes g_{u,v}(x,y))$$

$where\ gr\ is\ the\ control\ adjusted\ prameter.$

Step 8: Decompose the result of step 7 to real and imaginary parts by applying equations (2.13) and (2.14).

$$E_{u,v}(x,y) = Re[G_{u,v}(x,y],\ O_{u,v}(x,y) = Im[G_{u,v}(x,y)]$$

Step 9: Compute the amplitude from the equation (2.15).

$$A_{u,v}(x,y) = 2\pi\ gr\sqrt{E_{u,v}^2(x,y) + O_{u,v}^2(x,y)}$$

Where the $A_{u,v}\ (x,y)$ *is the amplitude resulted from the decomposing of iltering.*

$gr\ is\ the\ control\ adjusted\ prameter.$

Step 10: Reducing the dimensions by 25 by resampling the image using sampling filter until features in feature vector between 15 and 20.

Step 11: The extracted features is delivered to classifier.

Step 12: End.

MCNN Proposed Algorithm

Input: Training image (N x N) matrix.

Output: Emotion type with it's probability.
Step 1: Start

Step 2:

Initial the horseshoe chaos system and get the output numbers then stored in the chaos matrix (cm). The horseshoe chaos system are:

$$\begin{cases} \dot{x}_1 = a(x_2 - x_1) + x_2x_3, \\ \dot{x}_2 = (c-a)x_1 - x_1x_3 + cx_2, \\ \dot{x}_3 = ex_2^2 - bx_3 \end{cases}$$

Step 3: Initialize:

- Number of kernels (filters).
- Filters' size.
- A random values for all parameters.

Step 4: Apply convolution process by multiplying and summing each element in the filter matrix with the identical element in the input image according to the detected stride, by applying Eq.1.

$$x_{ij}^{l} = \sum_{a=0}^{m-1} \sum_{b=0}^{m-1} f_{ab}\ y_{(i+a)(j+b)}^{l-1}$$

Step 5: Apply ReLU operation to get non-linearity by replacing the negative values of each pixel in the activation map with zero through using Eq.2.

$$y_{\ i,j,d} = max\ \{0, x_{i,j,d}^{l}\}$$

Step 6: Apply max pooling on the output of the ReLU layer for dimensionality reduction by applying Eq. 3.

$$a_j = \max_{N\ x\ N}(\ a_i^{n\ x\ n}\ u(n,n))$$

Step 7: Apply FC layer, detect the emotion class and for each class find the output probabilities by applying Eq. 4.

$$y_c = \varsigma(z)_c = \frac{e^{z_c}}{\sum_{d=1}^{C} e^{z_d}} \quad for\ c = 1 \ldots C$$

Step 8: Compute the total error at the output layer using Eq. 5.

$$Total\ Error = \sum 1/2\ (target\ probability - output\ probability)^2$$

Step 9: Update:

- Weights values using cm matrix.
- Values of filter matrix using cm matrix.

Step 10: Repeat steps 2-9 until 50 rounds.

Step11: Stored the micro-facial emotions classifications results

Step 12: End.

Databases

This work implemented with two databases, the first is Cohn Kanade database collects from students of more hundred universities whose age from 18 to 30 years, this database is extended to CK+ database. Figure 4 show examples of this database. The second database are used is the Japanese's Females Facial Expressions (JAFFEs) Databases. The databases contain 213 image of 7 facial-expressions (6 basics facial-expressions and neutral) poses by 10 Japanese's female-models. Figure 5 show examples of this database.

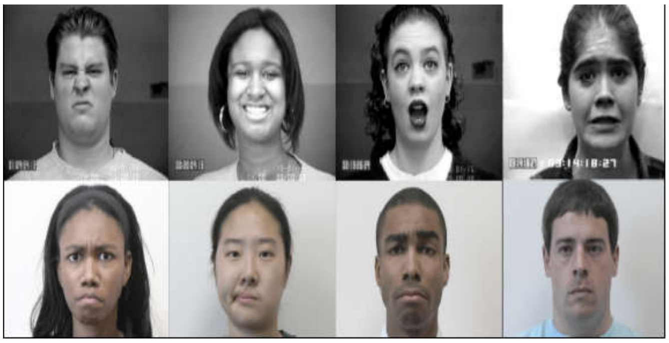


Figure 4. Examples of CK+ database

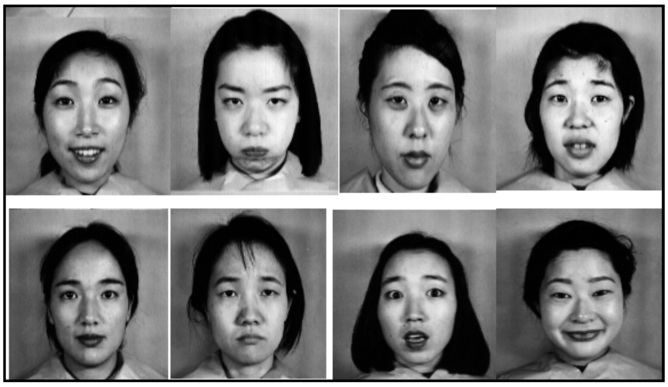


Figure 5. The JAFFE Databases Examples

Experimental result

Frequency domain is more reliable for extraction features from image to ease the effect of scaling, rotating and translation in order to reduce the tedious manual work needed for the normalization process. Figure 6 shows an example of the Modified Gabor filter results.

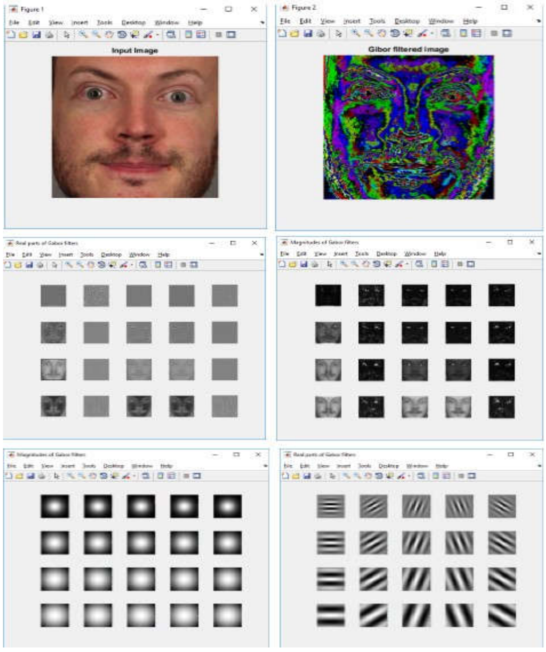


Figure 6. Modified Gabor Filter Output

The results in figure 6 were get from applied the modified Gabor filter using the following information: gamma=0.3; aspect ratio psi=0; phase theta=90; orientationbw=2.8; bandwidth or effective width lambda=3.5; wavelength pi=180; No. of scales= 4; No. of orientations= 5. No. of rows in a 2-D Gabor filter= 20. No. of columns in a 2-D Gabor filter =20. While the second technique used to facial features extraction is the PCA algorithm. The PCA was used for comparison purpose with modified Gabor filter. Figure 7 shows the PCA results. Table 2 and 3 show the results of applied MVJ, VJ, and PCA in features extraction.

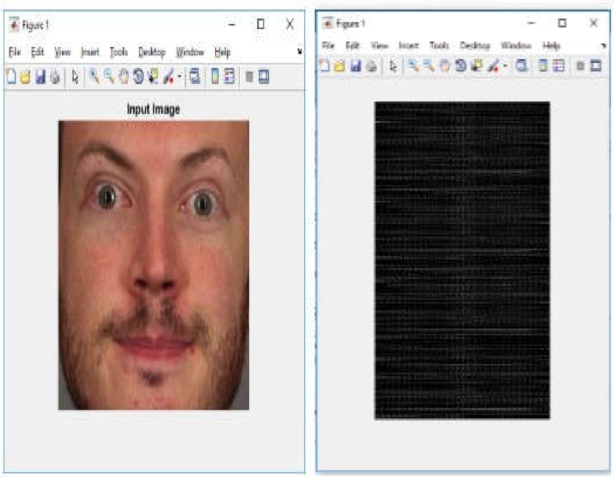


Figure 7. PCA features extraction results

The modified convolution neural network (MCNN) used to classify and take the decisions depend on the extracted features (micro-expressions and facial expressions).

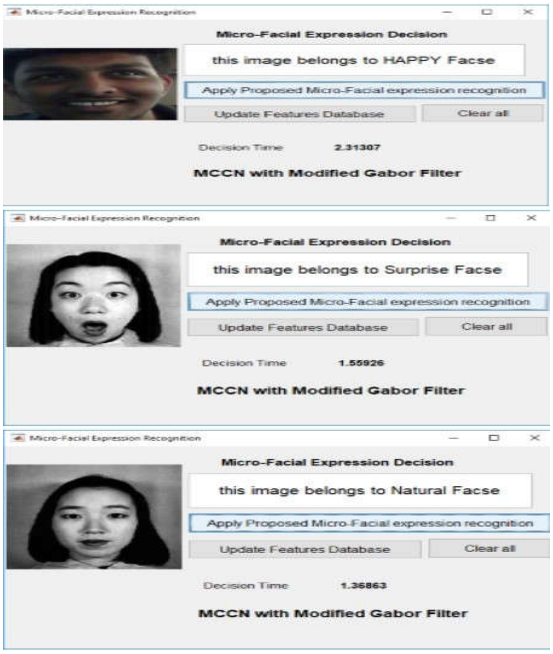
Table 2. The average time and features size of single face detection using the Gabor, MGabor, and PCA

Sample name	Sample numbers	MGabor filter		Gabor filter		PCA	
		Time (sec)	Features size	Time (sec)	Features size	Time (sec)	Features size
Anger	200	6.123	20000-250000	6.876	430000-460000	7.876	100000-102400
Happy	200	6.345	21000-270000	6.854	420000-470000	7.899	100000-102800
Sad	200	6.211	21000-260000	6.890	40000-470000	7.931	100000-103000
Surprise	200	6.331	20000-270000	6.899	41000-480000	8.120	100000-103000
Natural	200	6.232	20000-240000	6.768	400000-450000	7.876	100000-102000
Disgust	200	6.375	20000-270000	6.932	420000-485000	8.230	100000-104000
Fear	200	6.421	20000-270000	6.931	420000-485000	8.342	100000-104000

Table 3. The average time and features size of Multi- face detection using the Gabor, MGabor, and PCA

Sample name	Sample numbers	MGabor filter		Gabor filter		PCA	
		Time (sec)	Features size	Time (sec)	Features size	Time (sec)	Features size
Anger	200	8.333	300000-400000	9.121	67000-800000	11.561	1000000-1002000
Happy	200	8.434	340000-450000	9.341	68000-90000	11.873	1000000-103000
Sad	200	8.321	340000-450000	9.353	70000-940000	11.870	1000000-1060000
Surprise	200	8.303	360000-470000	9.563	70000-850000	11.785	1000000-1080000
Natural	200	8.211	350000-380000	9.240	70000-900000	11.652	1000000-1070000
Disgust	200	8.500	320000-500000	9.334	71000-920000	11.998	1000000-1090000
Fear	200	8.601	320000-500000	9.456	68000-930000	11.989	1000000-1090000

The MCNN content three layers with different size depending on the features size passing to it. But the output is one of the seven emotions for human face with seven facial expressions. The MCNN will classify the Facial and Micro-Facial expression Feature and take decision for expression type (happy, sad, anger, fear, surprise, disgust, and natural). The MCNN designed to take the features from three different features extraction methods (modified Gabor filter, Gabor filter, and PCA). Figures 8-10 show the MCNN result with modified Gabor filter, Gabor filter, and PCA features extraction. Tables 4-4.7 show the result of the MCNN with different features extraction methods.


Micro-Facial Expression Recognition

Micro-Facial Expression Decision

this image belongs to HAPPY Facse

Apply Proposed Micro-Facial expression recognition

Update Features Database | Clear all

Decision Time 2.31307

MCCN with Modified Gabor Filter

Micro-Facial Expression Recognition

Micro-Facial Expression Decision

this image belongs to Surprise Facse

Apply Proposed Micro-Facial expression recognition

Update Features Database | Clear all

Decision Time 1.55926

MCCN with Modified Gabor Filter

Micro-Facial Expression Recognition

Micro-Facial Expression Decision

this image belongs to Natural Facse

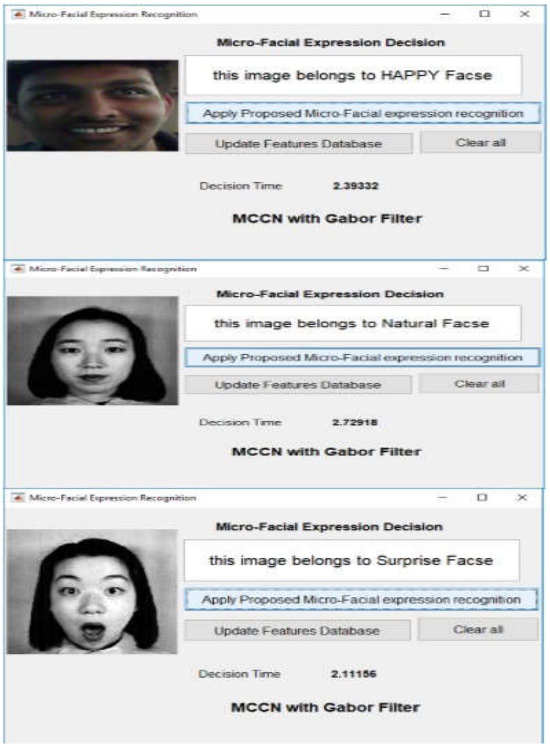
Apply Proposed Micro-Facial expression recognition

Update Features Database | Clear all

Decision Time 1.36863

MCCN with Modified Gabor Filter

Figure 8. MCNN with Modified Gabor Filter results


Micro-Facial Expression Recognition
Apply Proposed Micro-Facial expression recognition
Update Features Database
Clear all

Micro-Facial Expression Decision

this image belongs to HAPPY Facse

Decision Time 2.39332

MCCN with Gabor Filter

Micro-Facial Expression Decision

this image belongs to Natural Facse

Decision Time 2.72918

MCCN with Gabor Filter

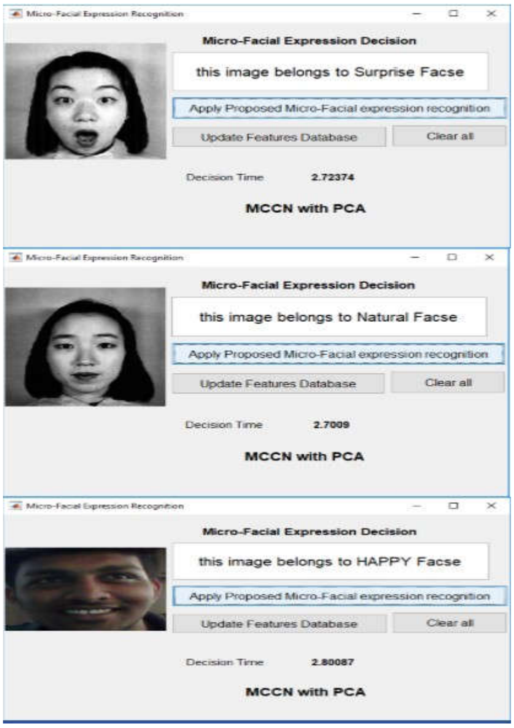
Micro-Facial Expression Decision

this image belongs to Surprise Facse

Decision Time 2.11156

MCCN with Gabor Filter

Figure 9. MCNN with Gabor Filter Results


Micro-Facial Expression Recognition
Apply Proposed Micro-Facial expression recognition
Update Features Database
Clear all

Micro-Facial Expression Decision

this image belongs to Surprise Facse

Decision Time 2.72374

MCCN with PCA

Micro-Facial Expression Decision

this image belongs to Natural Facse

Decision Time 2.7009

MCCN with PCA

Micro-Facial Expression Decision

this image belongs to HAPPY Facse

Decision Time 2.80087

MCCN with PCA

Figure 10. MCNN with PCA Results

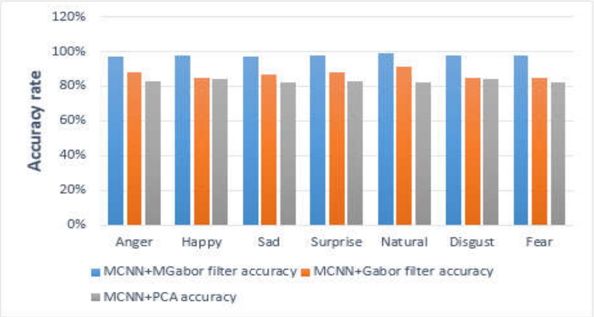


Figure 11. Accuracy rate of Multi- face Human Emotions Recognition using the Gabor, Modified Gabor, and PCA

Table 4: The average time and features size of single Human Emotions Recognition using MCNN and Gabor, Modified Gabor, and PCA

Sample name	Sample numbers	MGabor filter		Gabor filter		PCA	
		Avg. Time (sec)	Features size	Avg. Time (sec)	Features size	Avg. Time (sec)	Features size
Anger	200	1.984	20000-250000	2.100	430000-460000	2.700	100000-102400
Happy	200	1.876	21000-270000	2.250	420000-470000	2.750	100000-102800
Sad	200	1.989	21000-260000	2.200	40000-470000	2.800	100000-103000
Surprise	200	1.990	20000-270000	2.300	41000-480000	2.600	100000-103000
Natural	200	1.690	20000-240000	2.103	400000-450000	2.500	100000-102000
Disgust	200	1.980	20000-270000	2.410	420000-485000	2.650	100000-104000
Fear	200	1.990	20000-270000	2.359	420000-485000	2.640	100000-104000

Table 5: The average training time and features size of single Human Emotions Recognition using MCNN and Gabor, Modified Gabor, and PCA

Sample name	Sample numbers	MGabor filter		Gabor filter		PCA	
		Avg. Time (sec)	Features size	Avg. Time (sec)	Features size	Avg. Time (sec)	Features size
Anger	200	167	20000-250000	240	430000-460000	290	100000-102400
Happy	200	180	21000-270000	250	420000-470000	280	100000-102800
Sad	200	189	21000-260000	245	40000-470000	278	100000-103000
Surprise	200	194	20000-270000	240	41000-480000	280	100000-103000
Natural	200	190	20000-240000	250	400000-450000	275	100000-102000
Disgust	200	192	20000-270000	256	420000-485000	275	100000-104000
Fear	200	191	20000-270000	265	420000-485000	285	100000-104000

Table 4. 1: The decision accuracy of Multi- face Human Emotions Recognition using the Gabor, Modified Gabor, and PCA

Sample name	Sample numbers	MCNN+MGabor filter	MCNN+Gabor filter	MCNN+PCA
		accuracy	accuracy	accuracy
Anger	290	97%	88%	83%
Happy	290	98%	85%	84%
Sad	290	97%	87%	82%
Surprise	290	98%	88%	83%
Natural	290	99%	91%	82%
Disgust	290	98%	85%	84%
Fear	290	98%	85%	82%

Conclusion

The main objective of the proposed system is to recognition the human emotions based on the facial expression and micro-expression for the mobile devices and computers.

We implement the proposed system (evaluated and testing) under different cases and using two face databases. From the results, the proposed modification give us a good results comparing with same methods without modify them. Less than time execution and more accuracy for classification and recognition for seven human emotions expression. We conclude that:

- Modified Gabor filter have been used to extract the facial features with high speed and best the efficiency in features extraction due to the wavelet transform effects.
- Modified Convolutional neural network can be considered as one of the best Facial expression detection due to the ability to deals with many types of features extracted from different features extractions methods with same efficiency for all features types and size.
- MCNN able to classify the emotions of each facial image and it was a very successful method for detecting the emotions which are represented by the micro expressions on the face.

REFERENCES

Ali, L. R., Homood, H. K. and Elameer, A. S. 2017. "Feature Extraction Techniques on Facial Images : An Overview," *Int. J. Sci. Res.*, vol. 6, no. 9, pp. 2015–2018.

Al-Sumaidaee, S. S., Dlay, S. A. M., Woo, W. L. and Chambers, J. A. 2015. "Facial Expression Recognition using Local Gabor Gradient Code-Horizontal Diagonal Descriptor," in *2nd IET International Conference on Intelligent Signal Processing (ISP)*, 2015, pp. 1–6.

Amara, J., Owonibi, M., Kaur, P. and Bouaziz, B. 2017. "Convolutional Neural Network Based Chart Image Classification," in *25th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision*, 2017.

Arumugam, D. and Be, S. 2011. "Emotion classification using facial expression," *Int. J. Adv. Comput. Sci. Appl.*, vol. 2, no. 7, pp. 92–98.

Ekman, P. 1997. "Should We Call it Expression or Communication," *Innovation*, vol. 10, no. 4, pp. 333–344.

Habrman, D. 2016. "Face Recognition with Preprocessing and Neural Networks," Linköping University, Department of Electrical Engineering, Master Thesis.

Hafez, S. F., Selim, M. M. and Zayed, H. H. 2015. "2D Face Recognition System Based on Selected Gabor Filters and Linear Discriminant Analysis LDA," *Int. J. Comput. Sci. Issues*, vol. 12, no. 1, pp. 33–41.

K. O. and Nash, R. 2015. "An Introduction to Convolutional Neural Networks," no. December.

L. K. S. and K. J. Deepak Verma, "Performance Analysis of Feature Extraction Technique for Facial Expression Recognition," *Int. J. Adv. Comput. Technol. ISSN2319-7900*, vol. 2, no. 4, pp. 16–20, 2012.

Liu, T., Fang, S., Zhao, Y., Wang, P.and Zhang, J. 2015. "Implementation of Training Convolutional Neural Networks," University of Chinese Academy of Sciences, Beijing, China.

Lopes, A. T., de Aguiar, E., De Souza, A. F. and Oliveira-Santos, T. 2017. "Facial expression recognition with Convolutional Neural Networks: Coping with few data and the training sample order," *Pattern Recognit.*, vol. 61, no. January, pp. 610–628.

Lucey, P., Cohn, J. F., Kanade, T., Saragih, J., Ambadar, Z. and Matthews, I. 2010. "The extended cohn-kande dataset (CK+): A complete facial expression dataset for action unit and emotionspecified expression," *IEEE Conf. Comput. Vis. Pattern Recognit. Work.*, no. July, pp. 94–101.

Rashi Goyal and Tanushri Mittal, 2014. "Facial Expression Recognition using Artificial Neural Network," *HCTL Open Int. J. Technol. Innov. Res.*, vol. 10, no. July, pp. 1–10.

Scherer, D., Andreas, M. and Behnke, S. "Evaluation of Pooling Operations in Convolutional Architectures for Object Recognition," in *20th International Conference on Artificial Neural Networks (ICANN)*.

Shan, K., Guo, J., You, W., Lu, D. and Bie, R. 2017. "Automatic Facial Expression Recognition Based on a Deep Convolutional-Neural-Network Structure," in *IEEE Computer Society*, pp. 123–128.

Valero, H. G. 2016. "Automatic Facial Expression Recognition," School of Computer Science, The University of Manchester, Master Thesis.

Wu, J. 2017. "Introduction to Convolutional Neural Networks," Nanjing University, China.

Yousra BEN JEMAA, S. K.2009. "Automatic local Gabor features extraction for face recognition," *Int. J. Comput. Sci. Inf. Secur.*, vol. 3, no. 1.

Zeiler, M. D. and Fergus, R. 2014. "Visualizing and Understanding Convolutional Networks," in *Springer International Publishing Switzerland*, pp. 818–833.

****