IMPROVING ROBUSTNESS OF FACE EXPRESSION RECOGNITION USING MULTI-CHANNEL LOCAL BINARY PATTERN AND NEURAL NETWORK

Andaru Kharisma Bimantara[1), Nanik Suciati[2)
[1, 2) Informatics Departemen, Institut Teknologi Sepuluh Nopember
Jl. Teknik Kimia, Surabaya, East Java, indonesia
e-mail: andaru.kharisma@gmail.com[1), nanik@if.its.ac.id[2)

ABSTRACT

Facial Expression Recognition (FER) is a subset of Artificial Intelligence (AI) that relates to human non-verbal communication. The development of Convolutional Neural Network (CNN) based FER is subject to noise, mainly because of the usage of RGB Original Image as training data. Many research explored texture feature methods which noise resistant, such as Local Binary Pattern (LBP) and Gray Level Co-occurrence Matrix (GLCM), which mainly worked on grayscale images. Multi-Channel Local Binary Pattern (MCLBP) is derived from LBP which analyzes texture on color images.

This research aims to develop FER using MCLBP as a method of hand-crafted texture feature and NN as a classification method. The combination of MCLBP and Neural Network (NN) is expected more robust to noise. First, preprocessing is applied to the facial image for contrasting with Adaptive Gamma Correction Weighted Distribution (AGCWD). Next, the facial image is converted to MCLBP images. Then MCLBP images are converted to vectors as a NN architecture training data with 5 Fully Connected layers. Batch Normalization and Rectified Linear Unit (ReLu) activation are used in every Fully Connected layer. At the last Fully Connected Layer, ReLu activation was replaced with SoftMax activation. This NN uses Stochastic Gradient Descend (SGD) optimizer with a learning rate of 0.005.

Performance testing was held by comparing the epoch required to reach F1-score 1 and F1-Score from many scenarios in FER with LBP + NN with 140 × 190 image size, LBP + NN with 70 × 85 image size, and MCLBP + NN with 70 × 85 image size approaches. From all scenarios we have tried, the best method is MCLBP with F1-Score =1 in 22 epochs. The method of hand-crafted texture feature with NN can increase the desirable FER performances.

***Keywords**: Local Binary Pattern, Multi-Channel LBP, Neural Network, Face Expression Recognition, Gamma Correction*

MENINGKATKAN KEHANDALAN PENGENALAN EKSPRESI WAJAH MENGGUNAKAN MULTI-CHANNEL LOCAL BINARY PATTERN DAN NEURAL NETWORK

Andaru Kharisma Bimantara[1), Nanik Suciati[2)
[1, 2) Departemen Informatika, Institut Teknologi Sepuluh Nopember
Jl. Teknik Kimia, Surabaya, Jawa Timur, indonesia
e-mail: andaru.kharisma@gmail.com[1), nanik@if.its.ac.id[2)

ABSTRAK

Pengenalan Ekspresi Wajah (FER) adalah bagian dari Kecerdasan buatan (AI) yang terkait dengan komunikasi non-verbal manusia. Pengembangan FER berbasis Convolutional Neural Network (CNN) rentan terhadap noise, pada umumnya karena penggunaan citra original RGB sebagai data training. Banyak penelitian mengeksplor fitur tekstur yang kebal terhadap noise, termasuk Local Binary Pattern (LBP) dan Gray Level Co-occurrence Matrix (GLCM), yang umumnya bekerja pada citra grayscale. Multi-Channel Local Binary Pattern (MCLBP) diturunkan dari LBP yang menganalisis tekstur pada citra berwarna.

Penelitian ini bertujuan untuk mengembangkan FER menggunakan MCLBP sebagai metode fitur tekstur hand-crafted dan Neural Network (NN) sebagai metode klasifikasi. Kombinasi MCLBP and NN diharapkan lebih handal terhadap noise. Pertama, preprocessing diterapkan pada citra wajah untuk mengatur kontras dengan Adaptive Gamma Correction Weighted Distribution (AGCWD). Kemudian, citra wajah dikonversi ke citra MCLBP. Lalu, citra MCLBP dikonversi ke vektor sebagai data training arsitektur NN dengan 5 Layer Fully Connected. Batch Normalization dan aktivasi Rectified Linear Unit (ReLu) *digunakan pada setiap layer Fully Connected. Pada layer terakhir Fully Connected, Aktivasi ReLu diganti dengan aktivasi SoftMax. Pada NN ini digunakan optimisasi Stochastic Gradient Descend (SGD) dengan learning rate 0.005.*

Pengujian performa diadakan dengan membandingkan epoch untuk mencapai F1-score 1 dan F1-Score dari berbagai skenario pada FER dengan pendekatan LBP + NN dengan ukuran citra 140 × 190, LBP + NN dengan ukuran citra 70 × 85, dan MCLBP + NN dengan ukuran citra 70 × 85. Dari berbagai scenario yang diuji coba, hasil terbaik diperoleh MCLBP dengan F1-Score =1 dalam 22 epoch. Metode fitur tekstur hand-crafted dengan NN dapat meningkatkan performa FER yang diinginkan.

***Kata Kunci**: Local Binary Pattern, Multi-Channel LBP, Neural Network, Pengenalan Ekspresi Wajah, Koreksi Gamma*

I. INTRODUCTION

HUMAN convey expression to other individual via communications. Human communications are divided into verbal and non-verbal communications. The verbal communications are applied as speech or writing, whereas the non-verbal communications are based on expression or sign appeared on face, leg, or hand. Recently, Artificial Intelligence (AI) method has massively developed for human communication recognition. The example of AI in verbal communications are speech recognition and text recognition. Then, the example of AI in non-verbal communications are pose estimation, expression recognition and sign recognition. One of the examples of human expression recognition is facial expression recognition (FER) which detect various expression on face such as afraid, happy, hated, surprised, disgusted, and calmed. Research on recognition based on fist hand and tiptoeing foot are rarely common due to lack documentation of dataset rather than FER.

Many FER are developed using Deep Learning, mostly are based on derivative of CNN method. The CNN orchestrate convolution between a pixel and its neighbor inside specified filters. The CNN is able to detect important features in image through a learning cycle. However, the features extraction using CNN is prone to noise, due to the usage of color rather than texture information. In addition to noise problem, FER usually requires texture information on some parts of the face such as lip, chin, head, and eye. Beside the deep learning method, the conventional method, which proposed a feature extraction process by design, not by learning, is widely used in FER. The conventional methods include texture feature extraction using Local Binary Pattern (LBP) and Gray Level Co-occurrence Matrix (GLCM) (Imani and Montazer, 2017). GLCM extracts texture by orchestrating all angle between 2 pixels. Meanwhile, LBP uses the neighbors around a pixel by thresholding those with the pixel itself. Several LBP-derived methods have been implemented in FER such as Compound LBP (Ahmed et al., 2011), Local Derivative Pattern (LDP)(Zhang et al., 2010), Orthogonal Difference LBP (OD-LBP) (Karanwal and Diwakar, 2021), Center-Symmetric LBP (Sun et al., 2018), LBP + Histogram of Oriented Gradient (HOG) in Random Forest (Mady and Hilles, 2018) and Local Vector Pattern (LVP) (Fan and Hung, 2014). The hybrid method that combines CNN and LBP have been proposed to strengthen CNN (Zhang et al., 2010; Tang et al., 2020; Karanwal and Diwakar, 2021). All mentioned method uses LBP result before fed to CNN. The utilization of texture feature in FER case mostly extracts the textures from an authentic grayscale image or a grayscale converted-RGB image. Such an approach cannot capture the interconnectedness of texture information in different color spaces because it only compares pixels and their neighbors in the same color channel.

Color LBP research repairs the drawback of RGB image that does not need to be converted to grayscale image had been available. But the problem of Color LBP is inter-channel texture relationship remain ignored. So, the texture differences are in the middle of nowhere due to the relationship with the neighbor on the same channel does not have texture differences involved but the relationship with the neighbor on the different plane channel not involved might have texture differences. Then, Multi-Channel LBP (MCLBP)[9] appeared which has advantages because of the relationship with the neighbor on the different channel involved which correcting color LBP. Moreover, MCLBP could strengthening CNN which uses textural feature extraction because FER requires a textural comparison between skin and eye, lip texture, etc. instead of color feature comparison.

Differences in contrast and the presence of noise in the input image also affect the performance of FER. Several Gamma Correction-based methods have been proposed to improve the condition of the input image, such Adaptive Gamma Correction (AGC), AGC Weighted Distribution (AGCWD)(S. C. Huang, Cheng and Chiu, 2013), Gabor transformation and Gamma Correction (Zhu, Su and Wang, 2015), RGB Normalization and Gamma Correction (Chude-Olisah *et al.*, 2013), Raised Cosine AGC (RS-AGC) (Deivalakshmi, Saha and Pandeeswari, 2017), Range-Limited Bi-Histogram Equalization and AGC Combination (Gautam and Tiwari, 2015).

This research proposed Multi-Channel LBP and NN for the FER case. First, the facial image passes preprocessing stage by gamma correction using Adaptive Gamma Correction Weighted Distribution (AGCWD)[15]. Then the facial image is converted to MCLBP images which are deployed as NN training data. After that is the NN stage

Inventor	Method	Image Color
Ahmed *et al.*, 2011	Compound LBP	Grayscale
Karanwal and Diwakar, 2021	Orthogonal Difference LBP (OD-LBP)	Grayscale
Sun *et al.*, 2018	Center-Symmetric LBP (CS-LBP)	Grayscale
Zhang *et al.*, 2010	Local Derivative Pattern (LDP)	Grayscale
Fan and Hung, 2014	Local Vector Pattern (LVP)	Grayscale
Choi, Plataniotis and Ro, 2010	Color LBP	Colored but have no relation
Tang *et al.*, 2020	LBP + CNN Ensemble	Grayscale
Zhang *et al.*, 2017	LBP+CNN	Grayscale
Sawardekar, Sowmiya and Naik, 2018	LBP+CNN	Grayscale
Mady and Hilles, 2018	Random Forest and LBP+HOG	Grayscale
Ahmed *et al.*, 2011	Compound LBP	Grayscale
Karanwal and Diwakar, 2021	Orthogonal Difference LBP (OD-LBP)	Grayscale
Sun *et al.*, 2018	Center-Symmetric LBP (CS-LBP)	Grayscale
Zhang *et al.*, 2010	Local Derivative Pattern (LDP)	Grayscale
Fan and Hung, 2014	Local Vector Pattern (LVP)	Grayscale

with 5 Fully Connected Layers with respective Batch Normalization. This NN uses Stochastic Gradient Descent optimization (SGD) and a 0.005 learning rate.

Table 1. FER use LBP methods

II. Related Works

Commonly, FER is based on Deep Learning developed from the CNN method which is orchestrated by a convolution between its pixel and its neighbor inside the filter, and a traditional method also exists. CNN is a pretty straightforward method because architecture and hyperparameter are the only gamechanger in CNN but CNN is subject to noise. Textural Based FER also exists such as LBP and Gray Level Cooccurrence Matrix (GLCM). GLCM is orchestrated by statistics between 2 pixels at all angles. In the meantime, LBP uses textural features orchestrated by thresholding between pixel and its neighbors. Several methods of LBP in FER have been developed. Several LBP research methods at FER are described in Table (1). FER cases such as Compound LBP [2], LDP[3], OD-LBP[4], CS-LBP [5], LBP+HOG at Random Forest [6], and LVP [7] use authentic grayscale images or grayscale converted RGB image. The drawback is could not catch the color-based texture due to grayscale converted images requiring RGB accumulation. Moreover, this method compares only between pixel and its neighbor in the same channel.

Many research merges CNN and LBP methods for strengthening CNN. But the problem remained to exist if sampled images use grayscale converted RGB images. Color LBP research has been available which no need for strengthening CNN. But the problem remained to exist if sampled images use grayscale converted RGB images. Color LBP research has been available which no need for grayscale converted RGB image but don't have the textural connection between pixel channel. So, the texture differences are in the middle of nowhere due to the relationship with the neighbor on the same channel does not have texture differences but the relationship with the neighbor on the different channel which is not involved might have texture differences. Then, appear Multi-Channel LBP (MCLBP) which has advantages because of the relationship with many neighbors over many different channels involved which corrects color LBP.

Besides textural feature problems, the problem could happen with input data. Narrow Image contrast could weaken FER algorithm performance and require Gamma Correction. One Gamma Correction Method is Adaptive Gamma Correction (AGC). Many research on Gamma Corrected at FER has been developed such as Gabor transformation and Gamma Correction [11], RGB Normalization dan Gamma Correction[12].

III. System Design

This FER research with MCLBP and NN method flow consists of input, 3 stages, and output. Started from input image, preprocessing stage, MCLBP stage, and NN stage and output. Each stage is described in each sub-chapter. System Design Flow could be seen at Figure (1).

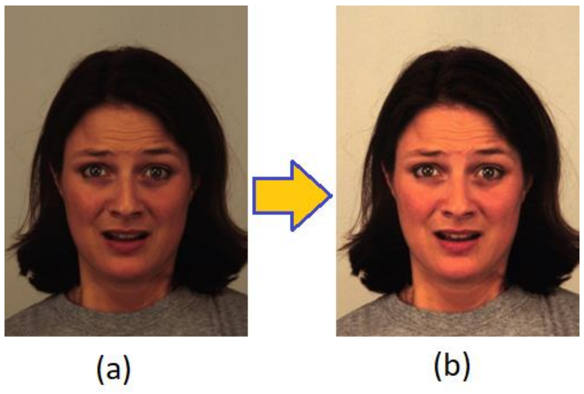

Fig. 2 Preprocessing Result using AGCWD. (a) before preprocessing, (b) after preprocessing

A. *Preprocessing*

Preprocessing is the data optimization stage before entering the core method. Preprocessing is required for repairing problematic images such as noise, narrow contrast, etc. Sometimes, FER contrast is problematic and requires AGC as a solution. Many research at AGC are AGC Weighted Distribution (AGCWD)[10] which measures probability density, Gabor transformation and Gamma Correction [11], RGB Normalization dan Gamma Correction [12], Raised Cosine AGC (RS-AGC)[13], Range-Limited Bi-Histogram Equalization and AGC Combination [14].

For preprocessing, this research follows AGCWD [15]. AGCWD uses AGC which utilizes contrast for strengthening or weakening texture because narrow contrast is subject to noise. Different from regular Gamma Correction, AGCWD is organized by Cumulative Density Frequency (CDF) which has relation to Probability Distribution Function (PDF). The result of preprocessing using AGCWD could be seen at Figure (2). Where Figure (2.a) before preprocessing, Figure (2.b) after preprocessing. Image relation of AGCWD could be seen in Equation (1), PDF could be seen in Equation (2), CDF could be seen in Equation (3).

$$T(l) = l_{max}(1/l_{max})^{\gamma} = l_{max}(1/l_{max})^{1-cdf(l)} \tag{1}$$

Where l is luminance, γ is the inverse value of Cumulative Density Frequency (CDF).

$$Pdf_w(l) = pdf_{max}\frac{(pdf(l) - pdf_{min})^{\alpha}}{(pdf_{max} - pdf_{min})} \tag{2}$$

Where pdf is Probability Distribution Function (PDF), α is manually setting parameter.

$$Cdf_w(l) = \sum_{l=0}^{l_{max}} Pdf_w(l) / \sum Pdf_w \tag{3}$$

Where Cdf is Cumulative Density Frequency (CDF).

B. *MCLBP*

MCLBP[9] is a method derived from LBP which uses texture orchestrated by thresholding between pixel and its neighbors. The majority of LBP use authentic grayscale images or grayscale converted-RGB images which could not catch the color-based texture due to grayscale converted images requiring RGB accumulation. Moreover, LBP is only able to use in one color channel. LBP use thresholding between LBP could be seen in Figure (3). Measurements of LBP are described in Equation (4) and Equation (5).

Where r_c is the center pixel, r_{n*} is neighbor around the center pixel with the different respective in Equation (5).

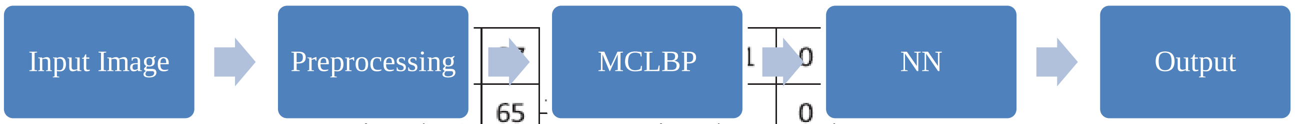

Fig. 1 System Design Flow

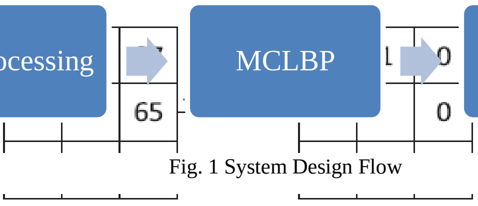

Fig. 3. Local Binary Pattern (LBP) [2]

L (4)

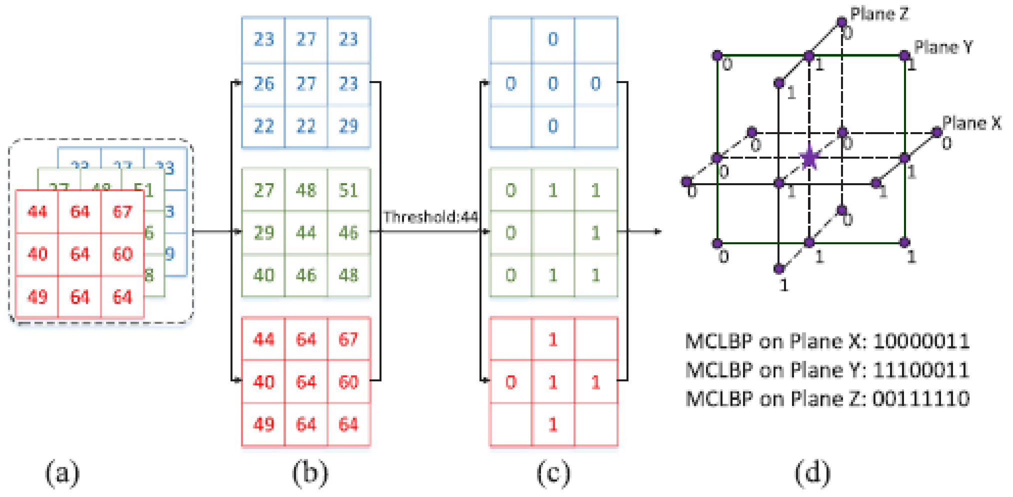
Threshold:44
Plane Z
Plane Y
Plane X
MCLBP on Plane X: 10000011
MCLBP on Plane Y: 11100011
MCLBP on Plane Z: 00111110
(a)
(b)
(c)
(d)

Fig. 4. Multi-Channel LBP (MCLBP) [9]

$$\delta(x) = \begin{cases} 1, X \geq 0 \\ 0, X < 0 \end{cases} \quad (5)$$

LBP then developed into several method such as Complete LBP(CLBP)[19], Local Ternary Pattern (LTP) which use 3 variable instead of LBP which use 2 variable, Robust LTP(RLTP)[20], Local Derivative Pattern (LDP)[3] which derivative of LBP, Center-Symmetric LBP (CS-LBP)[5], Orthogonal Difference LBP (OD-LBP) [4], Local Vector Pattern (LVP) [7], dan Local Tetra Pattern (LTeP)[21].

MCLBP [9] is an LBP development that looks forward to the pixel from across the channel. MCLBP doesn't need to be converted from an RGB color image to a grayscale image. The flows of MCLBP are MCLBP looks forward neighbors around the pixel itself on the same plane channel and looks forward neighbor from across the channel in axis X plane and Z plane by use Equation (6) and Equation (7). Then, MCLBP is measured from LBP measurements of neighbors around the pixel itself on the same plane channel, neighbor from across the channel in axis X plane and neighbor from across the channel in axis Z plane. Image then shifted between R-G-B, G-B-R, B-R-G plane and the result is concatenated 9 color channels texture image. Figure (4.a) is an image of 1 pixel and its neighbor from all channel. Then, the channel is divided into a single channel in Figure (4.b), After that, LBP is doing the job from each X, Y, and Z axis in Figure (4.c) and the result is Figure (4.d) with concatenated 9 color channels.

$$\delta_f(x) = \begin{cases} 1, X \geq 0 \\ 0, X < 0 \end{cases} \quad (7)$$

Where r_c is the center pixel, $r_{n*}$is neighbor around the center pixel with the different respective in Equation (7)

$$Multichannel\ LBP_{f,P,R}(i,j) = \sum_{n=1}^{P} \delta_f(r_n - r_c)2^{n-1} \quad (6)$$

For MCLBP, this research follows the [9]. Preprocessed image from previous stage passes MCLBP flows. The result is 9 channel textural images. In this research, Histogram does not create.

C. *Neural Network*

Neural Network is one of the deep learning methods for classification. It is still better than CNN from Neural Network derivatives. Because Neural Network takes all input instead of CNN which takes input among kernel size. Neural Network Architecture consists of the Fully Connected Layer (FC Layer) and Classification. Batch Normalization is optionally inserted into Neural Network. The flows in Neural Network for each epoch is divided into forward propagation and backward propagation. Forward propagation is process to predicts the classification output from the given inputs, trainable weights and hyperparameters. Backward propagation is process to update the trainable weights from output loss obtained. Neural Network is illustrated at Equation (8).

$$output = \sum_{n=1}^{N} x_n w_n + b \quad (8)$$

Where X is input, B is bias and W is trainable weight.

For this research, we use 5 layers of Neural Network coupled with Batch Normalization. The size of vector in before Layer 1 input is similar to image vector size and after Layer 1 output is 48 × Input Layer 1 vector. The vector size after Layer 2 is cut in half from Layer 1 output, and so Layer 3, Layer 4, and Layer 5. Non-linear ReLu activation is used for each layer except for the last layer which uses Softmax activation. Stochastic Gradient Descent (SGD) is used as an optimizer and the learning rate is set at 0,05.

IV. EXPERIMENTS

A. *Experiment Setup*

We use The Taiwanese Facial Expression Image Database (TFEID) and Karolinska Directed Emotional Face (KDEF)[22] for this research. TFEID consists of 980 images from happy, sad, contemplated, angered, disgusted, afraid, neutral, and surprised classes. KDEF consists of 4900 images of happy, sad, angered, disgusted, afraid, neutral, and surprised classes from 2 scenes of 35 male and female respondents with many angles. We take the sample from 336 TFEID images and 980 front-side KDEF images. For the experiment, we use 2 image sizes. 140 × 190, and 70 × 85. The ratio of images between training data and testing data is 60:40 and 80:20. We set the learning rate at 0,005 and the Stochastic Gradient Descent (SGD) optimizer. We use how many Epochs for gaining F1-score 100% parameter. Epochs don't meet the criteria if gaining F1-Score over 100 epochs or failed. We also measure the time needed for 100 epochs and RAM.

B. *KDEF dataset experiments*

From the experiment, the best experiment is MCLBP with 70 × 85 image size is the best with 22 epochs with training and testing data ratio 60:40. We tested Epoch at F1-Score 1, F1-Score at epoch =100, time consumption and memory consumption from the comparison of LBP and MCLBP methods, comparison of 140 × 190 image size and 70 × 85 image size, comparison of front face direction and all face direction and comparison of preprocessing

Table 2. KDEF Performance Result

Method	Image Size	Face Direction	Preprocessing Inserted	Ratio	Epoch at F1-Score 1	F1-Score at epoch = 100	Time (second)	RAM (GB)
LBP	140 × 190	Front	Yes	60:40	24	1	510	2,08
				80:20	24	1	707	
			No	60:40	26	1	510	
				80:20	25	1	707	
		All	Yes	60:40	*	0,99	1889	3,02
				80:20	*	0,99	2597	
	70 × 85	Front	Yes	60:40	66	1	15	0,08
				80:20	78	1	18	
			No	60:40	72	1	15	
				80:20	77	1	18	
		All	Yes	60:40	*	0,71	80	0,32
				80:20	*	0,71	78	
MCLBP	70 × 85	Front	Yes	60:40	22	1	735	3,02
				80:20	26	1	1012	
			No	60:40	26	1	735	
				80:20	27	1	1012	
		All	Yes	60:40	*	0,99	3698	5,08

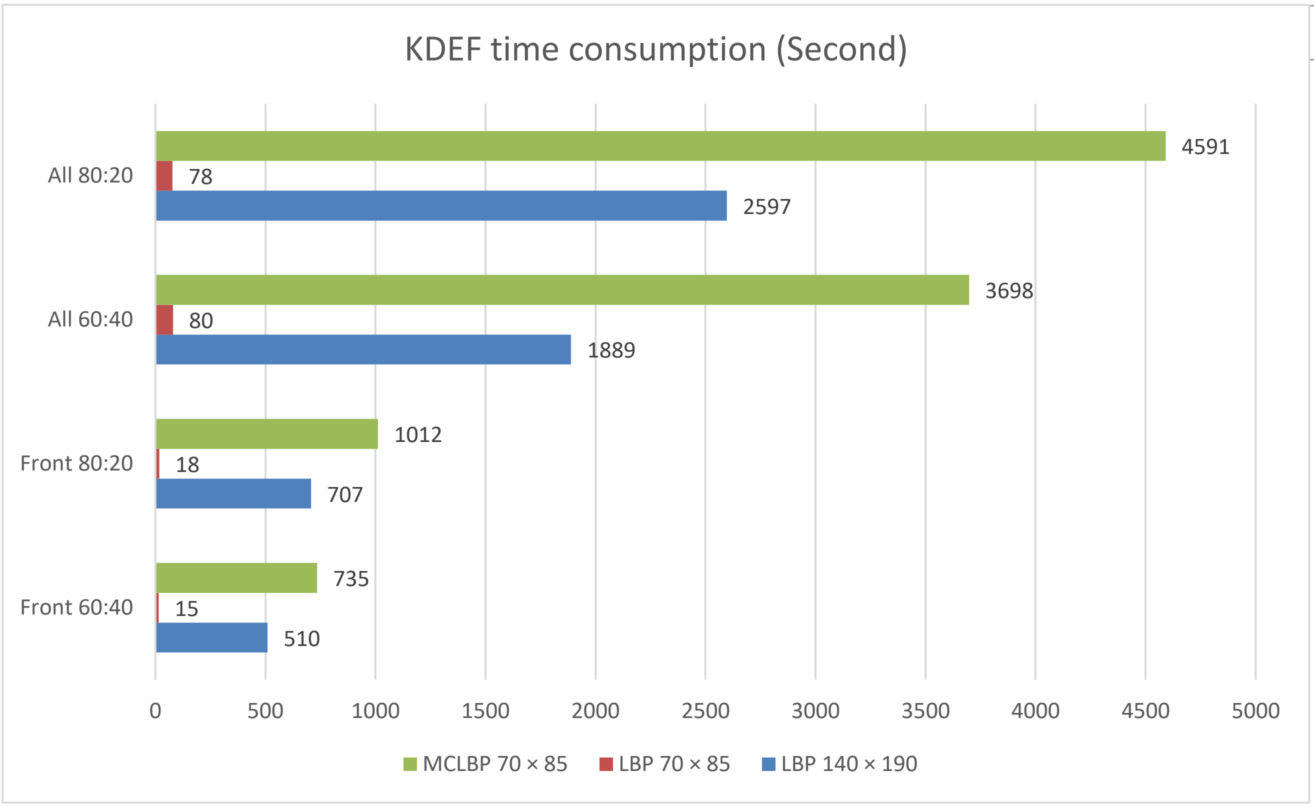

Fig. 5 KDEF Time Consumption

inserted or not inserted. KDEF performance result could be seen at Table (2), KDEF time consumption could be seen at Figure (5), and KDEF memory consumption could be seen at Figure (6).

In the comparison of the LBP and MCLBP method, MCLBP surpasses LBP in terms of epochs required for F1-Score 1. MCLBP method with 70 × 85 image size front face direction require 22 epochs for F1-Score 1 at training data and testing data ratio 60:40 and require 26 epochs for F1-Score 1 at training data and testing data ratio 80:20 beat the LBP method with 70 × 85 image size front face direction require 66 epochs for F1-Score 1 at training data and testing data ratio 60:40 and require 78 epochs for F1-Score 1 at training data and testing data ratio 80:20. MCLBP method with 70 × 85 image size all face direction failed to reach F1-Score 1 but mark 0,99 F1-Score for F1-Score 1 at both training data and testing data ratio 60:40 and 80:20 beat the LBP method with 70 × 85 image size all face direction failed to reach F1-Score 1 but mark 0,71 F1-Score at both training data and testing data ratio 60:40 and 80:20. MCLBP method with 70 × 85 image size front face direction without preprocessing require 26 epochs for F1-Score 1 at training data and testing data ratio 60:40 and require 27 epochs for F1-Score 1 at training data and testing data ratio 80:20 beat the LBP method with 70 × 85 image size front face direction without preprocessing require 72 epochs for F1-Score 1 at training data and testing data ratio 60:40 and require 77 epochs for F1-Score 1 at training data and testing data ratio 80:20. LBP surpasses MCLBP in terms of memory and time consumption. For time consumption, the LBP method with 70 × 85 image size front face direction requires 15 seconds at training data and testing data ratio 60:40 and require 18 seconds at training data and testing data ratio 80:20 beat the MCLBP method with 70 × 85 image size front face direction require 735 seconds at training data and testing data ratio 60:40 and require 1012 seconds at training data and testing data ratio 80:20. LBP method with 70 × 85 image size all face direction require 80 seconds at training data and testing data ratio 60:40 and require 78 seconds at training data and testing data ratio 80:20 beat MCLBP method with 70 × 85 image size front all direction require 3698 seconds at training data and testing data ratio 60:40 and require 4591 seconds at training data and testing data ratio 80:20. For memory consumption, LBP method with 70 × 85 image size front face direction require 0,08 GB beat MCLBP method with 70 × 85 image size front face direction require 3,02 GB. LBP method with 70 × 85 image size all-face direction require 0,32 GB beat MCLBP method with 70 × 85 image size all-face direction require 5,08 GB.

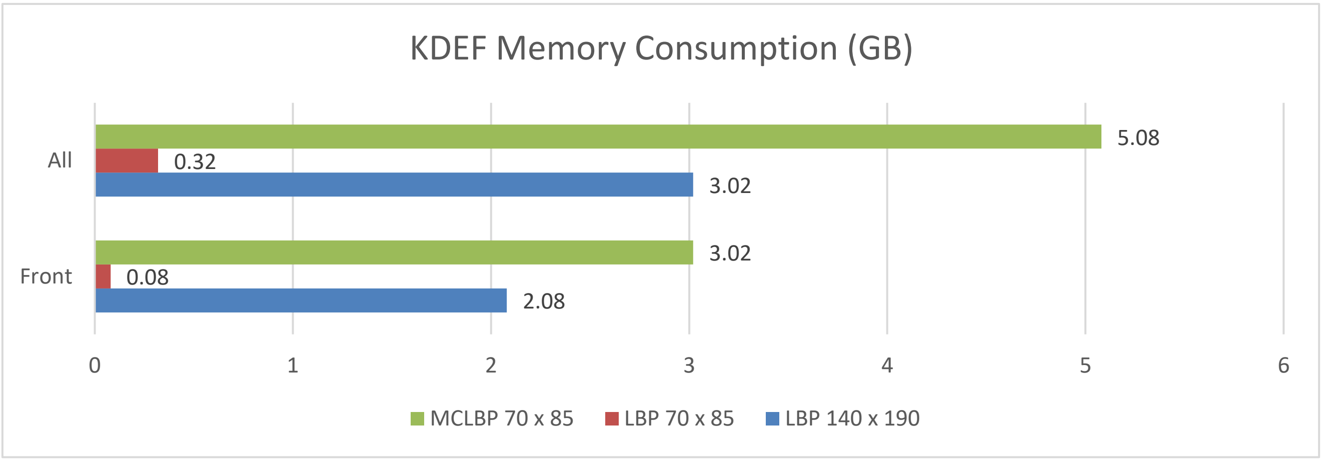

Fig. 6 KDEF Memory Consumption

In the comparison of 140 × 190 image size and 70 × 85 image size, LBP with 140 × 190 image size is better than LBP with 70 × 85 image size but cannot beat MCLBP with 70 × 85 image size in terms of epochs required for F1-Score 1. LBP method with 140 × 190 image size front face direction require 24 epochs for F1-Score 1 at both training data and testing data ratio 60:40 and 80:20 beat LBP method with 70 × 85 image size front face direction require 66 epochs for F1-Score 1 at both training data and testing data ratio 60:40 and require 78 epochs for F1-Score 1 at training data and testing data ratio 80:20 but cannot beat MCLBP with 70 × 85 image size front face direction require 22 epochs for F1-Score 1 at training data and testing data ratio 60:40 and require 26 epochs for F1-Score 1 at training data and testing data ratio 80:20. LBP method with 140 × 190 image size all-face direction failed to reach F1-Score 1 but mark 0,99 F1-Score at both training data and testing data ratio 60:40 and 80:20 beat LBP method with 70 × 85 image size all face direction failed to reach F1-Score 1 but mark 0,71 F1-Score at both training data and testing data ratio 60:40 and 80:20 but cannot beat MCLBP with 70 × 85 image size all-face direction failed to reach F1-Score 1 but mark 0,99 F1-Score at both training data and testing data ratio 60:40 and 80:20. LBP method with 140 × 190 image size front face direction without preprocessing require 26 epochs for F1-Score 1 at training data and testing data ratio 60:40 and require 25 epochs for training data and testing data ratio 80:20 beat LBP method with 70 × 85 image size front face direction without preprocessing require 72 epochs for F1-Score 1 at both training data and testing data ratio 60:40 and require 77 epochs for F1-Score 1 at training data and testing data ratio 80:20 but cannot beat MCLBP with 70 × 85 image size front face direction without preprocessing require 26 epochs for F1-Score 1 at training data and testing data ratio 60:40 and require 27 epochs for F1-Score 1 at training data and testing data ratio 80:20. For time consumption and memory consumption, LBP method with 70 × 85 image size surpasses LBP method with 140 × 190 image size, but LBP method with 140 × 190 image size can surpasses MCLBP method with 70 × 85 image size. LBP method with 70 × 85 image size front face direction require 15 seconds at training data and testing data ratio 60:40 and require 18 seconds at training data and testing data ratio 80:20 beat LBP method with 140 × 190 image size front face direction require 510 seconds at training data and testing data ratio 60:40 and require 707 seconds at training data and testing data ratio 80:20 and MCLBP method with 70 × 85 image size front face direction require 735 seconds at training data and testing data ratio 60:40 and requires 1012 seconds at training data and testing data ratio 80:20. LBP method with 70 × 85 image size all face direction requires 80 seconds at training data and testing data ratio 60:40 and requires 78 seconds at training data and testing data ratio 80:20 beat LBP method with 140 × 190 image size all face direction require 1889 second at training data and testing data ratio 60:40 and requires 2597 seconds at training data and testing data ratio 80:20 and MCLBP method with 70 × 85 image size front all direction requires 3698 seconds at training data and testing data ratio 60:40 and require 4591 seconds at training data and testing data ratio 80:20. For memory consumption, LBP method with 70 × 85 image size front face direction requires 0,08 GB beat LBP method with 140 × 190 image size front face direction requires 2,08 GB and MCLBP method with 70 × 85 image size front face direction requires 3,02 GB. LBP method with 70 × 85 image size all-face direction requires 0,32 GB beat LBP method with 140 × 190 image size all-face direction requires 3,02 GB and MCLBP method with 70 × 85 image size all-face direction requires 5,08 GB.

In the comparison of all-face direction vs front face direction, Front face direction surpasses all-face direction in terms of epochs required for F1-Score 1, memory consumption, and time consumption. MCLBP method with 70 × 85 image size front face direction requires 22 epochs for F1-Score 1 at training data and testing data ratio 60:40 and requires 26 epochs for F1-Score 1 at training data and testing data ratio 80:20 beat MCLBP method with 70 × 85 image size all-face direction failed to reach F1-Score 1 but mark 0,99 F1-Score at both training data and testing data ratio 60:40 and 80:20. LBP method with 70 × 85 image size front face direction requires 66 epochs for F1-Score 1 at training data and testing data ratio 60:40 and requires 78 epochs for F1-Score 1 at training data and testing data ratio 80:20 beat LBP method with 70 × 85 image size all-face direction failed to reach F1-Score 1 but mark 0,71 F1-Score at both training data and testing data ratio 60:40 and 80:20. LBP surpassed MCLBP in terms of memory and time consumption. LBP method with 140 × 190 image size front face direction requires 24 epochs for F1-Score 1 at both training data and testing data ratio 60:40 and 80:20 beat LBP method with 140 × 190 image size all-face direction failed to reach F1-Score 1 but mark 0,99 F1-Score at both training data and testing data ratio 60:40 and 80:20. For time consumption, MCLBP method with 70 × 85 image size front face direction requires 735 second at training data and testing data ratio 60:40 and requires 1012 second at training data and testing data ratio 80:20 beat MCLBP method with 70 × 85 image size all-face direction requires 3698 second at training data and testing data ratio 60:40 and requires 4591 second at training data and testing data ratio 80:20. LBP method with 70 × 85 image size front face direction requires 15 second at training data and testing data ratio 60:40 and requires 18 second at training data and testing data ratio 80:20 beat LBP method with 70 × 85 image size all-face direction requires 80 second at training data and testing data ratio 60:40 and require 78 second at training data and testing data ratio 80:20. LBP method with 140 × 190 image size all-face direction require 510 second at training data and testing data ratio 60:40 and require 707 second at training data and testing data ratio 80:20 beat LBP method with 140 × 190 image size all-face direction require 1889 second at training data and testing data ratio 60:40 and require 2597 second at training data and testing data ratio 80:20. For memory consumption, MCLBP method with 70 × 85 image size front face direction require 3,02 GB beat MCLBP method with 70 × 85 image size all-face direction require 5,08 GB. LBP method with 70 × 85 image size front face direction require 0,08 GB beat LBP method with 70 × 85 image size all-face direction require 0,32 GB. LBP method with 140 × 190 image size all-face direction requires 2,08 GB beat LBP method with 140 × 190 image size all-face direction requires 3,02 GB.

In the comparison of the method with preprocessing and method without preprocessing, the method with processing surpasses the method without preprocessing. LBP method with 140 × 190 image size front face direction with preprocessing require 24 epochs for F1-Score 1 at both training data and testing data ratio 60:40 and 80:20 beat LBP method with 140 × 190 image size front face direction without preprocessing require 26 epochs for F1-Score 1 at training data and testing data ratio 60:40 and require 25 epochs for F1-Score 1 at training data and testing data ratio 80:20. LBP method with 70 × 85 image size front face direction with preprocessing require 66 epochs for F1-Score 1 at training data and testing data ratio 60:40 and require 78 epochs for F1-Score 1 at training data and testing data ratio 80:20 beat LBP method with 70 × 85 image size front face direction without preprocessing require 72 epochs for F1-Score 1 at training data and testing data ratio 60:40 and require 77 epochs for F1-Score 1 at training data and testing data ratio 80:20. MCLBP method with 70 × 85 image size front face direction with preprocessing require 22 epochs for F1-Score 1 at training data and testing data ratio 60:40 and require 26 epochs for F1-Score 1 at training data and testing data ratio 80:20 beat MCLBP method with 70 × 85 image size face direction without preprocessing require 26 epochs for F1-Score 1 at training data and testing data ratio 60:40 and require 27 epochs for F1-Score 1 at training data and testing data ratio- 80:20.

C. TFEID dataset experiments

From the experiment, the best experiment is MCLBP with 70 × 85 image size is the best with 24 epochs with training and testing data ratio 60:40. We tested Epoch at F1-Score 1, F1-Score at epoch =100, time consumption

Table 3 TFEID Performance Result

TFEID	Image Size	Ratio	Epoch at F1-Score 1	F1-Score at epoch = 100	Time (second)	RAM (GB)
LBP	140 × 190	3:2	25	1	217	1,79
		4:1	27	1	278	
	70 × 85	3:2	33	1	8	0,05
		4:1	39	1	9	
MCLBP	70 × 85	3:2	24	1	321	2,67
		4:1	26	1	416	

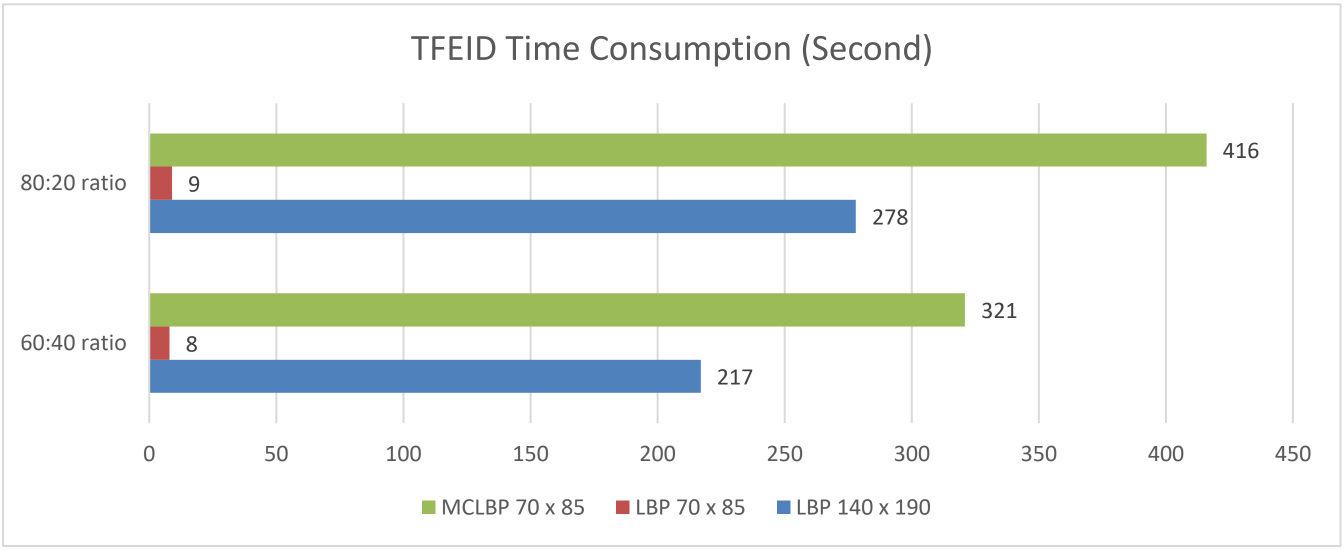

Fig. 7 TFEID Time Consumption

and memory consumption from the comparison of LBP and MCLBP methods, and comparison of 140 × 190 image -size and 70 × 85 image size. TFEID performance result could be seen at Table (3), TFEID time consumption could be seen at Figure (7), and TFEID memory consumption could be seen at Figure (8).

In the comparison of MCLBP and LBP, MCLBP surpasses LBP in terms of epochs required for F1-Score 1. MCLBP method with 70 × 85 image size requires 24 epochs for F1-Score 1 at training data and testing data ratio 60:40 and requires 26 epochs for F1-Score 1 at training data and testing data ratio 80:20 beat LBP method with 70 × 85 image size requires 33 epochs for F1-Score 1 at training data and testing data ratio 60:40 and requires 39 epochs for F1-Score 1 at training data and testing data ratio 80:20. LBP surpassed MCLBP in term of memory and time consumption. For time consumption, the LBP method with 70 × 85 image size requires 8 seconds at training data and testing data ratio 60:40 and requires 9 seconds at training data and testing data ratio 80:20 beat MCLBP method with 70 × 85 image size required 321 seconds at training data and testing data ratio of 60:40 and requires 416 seconds at training data and testing data ratio 80:20. For memory consumption, LBP method with 70 × 85 image size requires 0,05 GB beat MCLBP method with 70 × 85 image size front face direction require 2,67 GB.

In Comparison to 140 × 190 image size and 70 × 85 image size, MCLBP surpasses LBP in terms of epochs required for F1-Score 1. LBP with 140 × 190 image size is better than LBP with 70 × 85 image size but cannot beat MCLBP with 70 × 85 image size in terms of epochs required for F1-Score 1. LBP method with 140 × 190 image size requires 25 epochs for F1-Score 1 at training data and testing data ratio 60:40 and requires 27 epochs for F1-Score 1 at training data and testing data ratio 80:20 beat LBP method with 70 × 85 image size front face direction requires 33 epochs for F1-Score 1 at both training data and testing data ratio 60:40 and requires 39 epochs for F1-Score 1 at training data and testing data ratio 80:20 but cannot beat MCLBP with 70 × 85 image size requires 24 epochs for F1-Score 1 at training data and testing data ratio 60:40 and requires 26 epochs for F1-Score 1 at training data and testing data ratio 80:20. For time consumption and memory consumption, LBP method with 70 × 85 image size surpasses LBP method with 140 × 190 image size, but LBP method with 140 × 190 image size can surpass MCLBP method with 70 × 85 image size. LBP method with 70 × 85 image size requires 8 seconds at training data and testing data ratio 60:40 and requires 9 seconds at training data and testing data ratio 80:20 beat LBP method with 140 × 190 image size requires 217 seconds at training data and testing data ratio 60:40 and

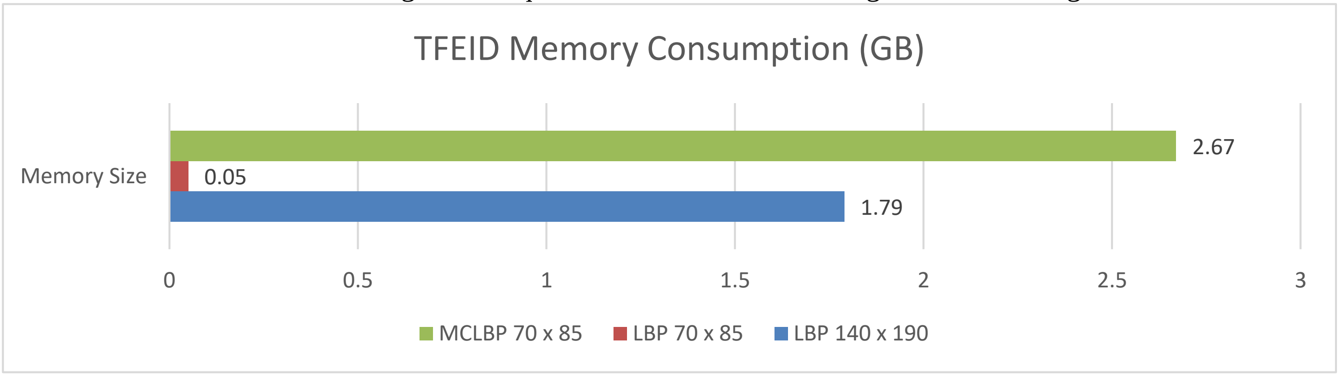

Fig. 8 TFEID Memory Consumption

requires 278 seconds at training data and testing data ratio 80:20 and MCLBP method with 70 × 85 image size requires 321 seconds at training data and testing data ratio 60:40 and requires 416 seconds at training data and testing data ratio 80:20. For memory consumption, LBP method with 70 × 85 image size front face direction requires 0,05 GB beat LBP method with 140 × 190 image size front face direction requires 1,79 GB and MCLBP method with 70 × 85 image size front face direction requires 2,67 GB.

D. Discussion

In the comparison of LBP and MCLBP method, MCLBP surpasses LBP because of the broader features. MCLBP has 9 color channel image and LBP have only 1 color channel image. Although MCLBP can beat LBP, LBP can win in time consumption and memory consumption. In the comparison of 140 × 190 image size and 70 × 85 image size, LBP with 140 × 190 image size is better than LBP with 70 × 85 image size but cannot beat MCLBP with 70 × 85 image size. Because bigger image size means bigger feature. While the drawback of bigger image size is the time consumption and memory consumption. MCLBP can assist in accuracy increasing in FER while decreasing the feature size. In the comparison of all-face direction vs front face direction, front-face direction surpasses all-face direction. Because the lips, eyes, and mouth are fixed positioned in front-face direction rather than all-face direction. In the comparison of the method with the preprocessing and method without preprocessing, the method with processing surpasses method without preprocessing. Preprocessing method could help the image computation by broaden luminance value.

V. Conclusion

We have improved the FER using MCLBP and NN. From the experiments, MCLBP result could affected by comparison of LBP and MCLBP methods, comparison of 140 × 190 image size and 70 × 85 image size, and comparison of front face direction and all face direction. The best result gained by MCLBP method with 70 × 85 image size front face direction with 22 epochs required for F1-Score 1. The time required for 100 epochs is 735 second. The use of MCLBP for Neural Network coupled with preprocessing also improve the epoch required for F1-Score 1. However, FER is subject to easily manipulated. Human could control or trigger face expression consciously.

VI. FUTURE WORK

We expect this MCLBP research could be applied to other image-based research scopes such as autonomous vehicles, video-based research, and etc. This MCLBP FER Image research is also recommended to be applied to psychology and sociology subject. To reduce manipulation of human FER, lot of research about MCLBP FER using both hands or both legs could be required.

References

[1] M. Imani and G. A. Montazer, “GLCM features and fuzzy nearest neighbor classifier for emotion recognition from face,” *2017 7th Int. Conf. Comput. Knowl. Eng. ICCKE 2017*, vol. 2017-Janua, no. Iccke, pp. 8–13, 2017.

[2] F. Ahmed, E. Hossain, A. S. M. H. Bari, and A. Shihavuddin, “Compound local binary pattern (CLBP) for robust facial expression recognition,” *12th IEEE Int. Symp. Comput. Intell. Informatics, CINTI 2011 - Proc.*, pp. 391–395, 2011.

[3] B. Zhang, Y. Gao, S. Zhao, and J. Liu, “Local derivative pattern versus local binary pattern: Face recognition with high-order local pattern descriptor,” *IEEE Trans. Image Process.*, vol. 19, no. 2, pp. 533–544, 2010.

[4] S. Karanwal and M. Diwakar, “OD-LBP: Orthogonal difference-local binary pattern for Face Recognition,” *Digit. Signal Process. A Rev. J.*, vol. 110, p. 102948, 2021.

[5] K. Sun, X. Yin, M. Yang, Y. Wang, and J. Fan, “The face recognition method based on CS-LBP and DBN,” *Math. Probl. Eng.*, vol. 2018, 2018.

[6] H. Mady and S. M. S. Hilles, “Face recognition and detection using Random forest and combination of LBP and HOG features,” *2018 Int. Conf. Smart Comput. Electron. Enterp. ICSCEE 2018*, 2018.

[7] K. C. Fan and T. Y. Hung, “A novel local pattern descriptor - Local vector pattern in high-order derivative space for face recognition,” *IEEE Trans. Image Process.*, vol. 23, no. 7, pp. 2877–2891, 2014.

[8] J. Tang, Q. Su, B. Su, S. Fong, W. Cao, and X. Gong, “Parallel ensemble learning of convolutional neural networks and local binary patterns for face recognition,” *Comput. Methods Programs Biomed.*, vol. 197, p. 105622, 2020.

[9] X. Shu, Z. Song, J. Shi, S. Huang, and X. J. Wu, “Multiple channels local binary pattern for color texture representation and classification,” *Signal Process. Image Commun.*, vol. 98, no. December 2020, p. 116392, 2021.

[10] S. C. Huang, F. C. Cheng, and Y. S. Chiu, “Efficient contrast enhancement using adaptive gamma correction with weighting distribution,” *IEEE Trans. Image Process.*, vol. 22, no. 3, pp. 1032–1041, 2013.

[11] D. X. Zhu, Z. Su, and J. Wang, “Face recognition method combined with gamma transform and Gabor transform,” *2015 IEEE Int. Conf. Signal Process. Commun. Comput. ICSPCC 2015*, 2015.

[12] C. C. Chude-Olisah, G. Sulong, U. A. K. Chude-Okonkwo, and S. Z. M. Hashim, “Illumination normalization for edge-based face recognition using the fusion of RGB normalization and gamma correction,” *IEEE ICSIPA 2013 - IEEE Int. Conf. Signal Image Process. Appl.*, no. 08, pp. 412–416, 2013.

[13] S. Deivalakshmi, A. Saha, and R. Pandeeswari, “Raised cosine adaptive gamma correction for efficient image and video contrast enhancement,” *IEEE Reg. 10 Annu. Int. Conf. Proceedings/TENCON*, vol. 2017-Decem, pp. 2363–2368, 2017.

[14] C. Gautam and N. Tiwari, “Efficient color image contrast enhancement using Range Limited Bi-Histogram Equalization with Adaptive Gamma

Correction," *2015 Int. Conf. Ind. Instrum. Control. ICIC 2015*, no. Icic, pp. 175–180, 2015.

[15] S.-C. Huang, F.-C. Cheng, and Y.-S. Chiu, "Efficient Contrast Enhancement Using Adaptive Gamma Correction With Weighting Distribution," *IEEE Trans. Image Process.*, vol. 22, no. 3, pp. 1032–1041, 2013.

[16] J. Y. Choi, K. N. Plataniotis, and Y. M. Ro, "Using colour local binary pattern features for face recognition," *Proc. - Int. Conf. Image Process. ICIP*, pp. 4541–4544, 2010.

[17] H. Zhang, Z. Qu, L. Yuan, and G. Li, "A face recognition method based on LBP feature for CNN," *Proc. 2017 IEEE 2nd Adv. Inf. Technol. Electron. Autom. Control Conf. IAEAC 2017*, pp. 544–547, 2017.

[18] S. Sawardekar, P. Sowmiya, and R. Naik, "Facial Expression Recognition using Efficient LBP and CNN," *Int. Res. J. Eng. Technol.*, no. June, pp. 2273–2277, 2018.

[19] Z. Guo, L. Zhang, and D. Zhang, "A Completed Modeling of Local Binary Pattern Operator for Texture Classification," *IEEE Trans. Image Process.*, vol. 19, no. 6, pp. 1657–1663, 2010.

[20] X.-H. Han, G. Xu, and Y.-W. Chen, "Robust local ternary patterns for texture categorization," in *2013 6th International Conference on Biomedical Engineering and Informatics*, 2013, pp. 846–850.

[21] S. Murala, R. P. Maheshwari, and R. Balasubramanian, "Local tetra patterns: A new feature descriptor for content-based image retrieval," *IEEE Trans. Image Process.*, vol. 21, no. 5, pp. 2874–2886, 2012.

[22] D. Lundqvist, A. Flykt, and A. Ohman, *The Karolinska directed emotional faces (KDEF)*. Department of Clinical Neuroscience, Psychology section, Karolinska Institutet, 1998.