Happy and Sad Classification using HOG Feature Descriptor in SVM Model Selection

1st Derry Alamsyah
Informatics
Faculty of Computer Science and Engineering
MDP University
Palembang, Indonesia
derry@mdp.ac.id

2nd Muhammad Fachrurrozi
Informatics
Faculty of Computer Science
Sriwijaya University
Indralaya, Indonesia
mfachrrz@unsri.ac.id

2021 International Conference on Informatics, Multimedia, Cyber and Information System (ICIMCIS) | 978-1-6654-2733-3/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICIMCIS53775.2021.9699183

***Abstract*—Facial Expression Recognition (FER) of the image is one of the potential research fields. It remains some open problems to be solved such as various head positions, backgrounds, occlusion, face attribute etc., where the FER 2013 dataset give such conditions. In this research, the small balanced dataset used to recognize two common fundamental expression, happy and sad face image as our set conditions. Using SVM as classifier and HOG as feature expression method, this research shows best performance, that is 72% accuracy, in quadratic polynomial kernel with intercept constant b = 1 and tolerance constant C = 0.1. By using such conditions, minimized pose variant, a conventional approach in FER such SVM and HOG has shown fair performance in the FER 2013 dataset.**

Keywords—SVM, HOG, FER

I. Introduction

Human expression can be described by facial expression, hand gestures, and voice, while 55%, as a big one, was component come from facial expression [1]. Commonly, there are six fundamental expression such as happiness, sadness, fear, disgust, surprise, and anger [2]. It has given some benefit to classify a human expression. Particularly, in Health care, better serve from analyze inner psychological human [3], Human-robotic interactions (HRI), and human computer interaction (HCI) [1].

Facial Expression Recognition (FER), as an area of research, focus on how the system can recognise human expression by their face or facial aspect. FER system can be classified into two approaches, they are *machine learning* (conventional) and *deep learning* approach [4]. The difference between them is how they obtain their feature extraction. Machine learning or conventional approaches utilize an algorithm to get manually specific feature, for instance texture, shape, feature point or even in frequent domain such as gabor feature. while another get they feature automatically.

For the dataset, there are many researcher had created this one. Started in 1998, JAFFE dataset begin this area of research using seven facial emotion posed by Japanese female [5], while KDEF dataset offer five different angles [6] in same year. After that, there are video and 3D dataset in earlier 2000 such as MMI [7] and BU-3DFE dataset [8] respectively. For illumination aspect, GENKI-4K dataset offer wide range of illumination conditions [9]. CK+ dataset conducted still image and video sequence from 18 to 30 subject age range [10]. For some obstacle such as glasses, NVIE dataset conducted that condition [11] and Multi-PIE dataset set extend range of viewpoints and illumination conditions [12]. In pain aspect UNBC-Mc dataset had conducted that condition [13]. Oulu-CASIA give 23 to 58 subject age ranges with sequence image for expression [14]. For multi modal aspect, GEMEP-FERA give audio and video record [15]. For the wild scenario or real word conditions AFEW [16] and FER2013 [17] give an issues need to be solved. The MPI dataset guarantee for natural face expression [18], while DISFA dataset given variety of ethnicities in different gender [19]. Brighamton university expand their dataset before (BU-3DFE) to BP4D-spontaneous dataset [20]. For some race included, CE dataset offer their dataset in several subject age, 23 years old in rate [21]. In recent, there is RAF-DB dataset for real world situation with more variety aspect in subject [22][23].

There are some issues in this area of research as a challenge and opportunity. In [4], their research reveal that they are five issues which are wild environmental conditions, the lack of high-quality publicity available data, the pressure of high volume data processing, multi-modal affect recognition, and visual privacy. For the wild environmental conditions they are some challenge, for instance illumination aspect, pose variant, any occlusion (glasses, hat, etc.), view point and many others. In FER 2013 Dataset there are multiple pose variation collected from many subject with many occlusion. This dataset give challenge to us that need to be solved.

Some research used FER2013 dataset, in particular, [2] used Extreme Learning Machine (ELM) with two different feature extractors and give 63.86% for the HOG feature and 55.11% for the LBP feature. Meanwhile, by using the deep learning approach, [24] used CNN to recognize facial expression with 51.1% success and [25] give better result 72.10%. Another research, [26] used DNN give 66.4% success.

By using SVM Classifier and HOG feature extraction, in their empirical evaluation for FER2013 dataset, [27] give a better performance for cubic kernel as 57.17% accuracy. In their research, they used all the seven basic human expression. As mention before, there are a pose-variant issue, that come in real world condition dataset, included this FER2013 dataset. From that issues, how if we reduce some pose variant, then this research focus on two basic or fundamental expression, happy and sad, to reduce some pose variant in wild condition and looking for how good the SVM and HOG in this condition. Moreover, this research using small number of dataset in reduce pose-variant condition (in wild conditions) and same number of dataset for balanced dataset aspect.

II. Histogram of Oriented Gradient (HOG)

HOG is one of feature descriptor used gradient along with its orientation (magnitude and direction) as an important information to describe or represent an image. This feature descriptor is widely used in computer vision to determine an object [28][29].

Given a grayscale image I has two gradient image I_x and I_y based on x-axis and y-axis respectively. These two gradient image are computed by (1) and (2) to all pixel trough row r and column c. As the name is Histogram of Oriented Gradient (HOG) than these two image are substantial part of this method. The gradient image are given by subtraction from original image I based on their x-axis shown by change of column in (1), an the same analogy for y-axis. The concept is using numerical 1st order derivative function. HOG feature extraction method collect gradient feature of image trough the group of local part of image I called *cell* (shown by Figure 1.a : Green box). This feature μ as magnitude of image gradient (3) are set by their orientation θ (4).

$$I_x(r,c) = I(r,c+1) - I(r,c-1) \quad (1)$$

$$I_y(r,c) = I(r-1,c) - I(r+1,c) \quad (2)$$

$$\mu = \sqrt{I_x + I_y} \quad (3)$$

$$\theta = \frac{180}{\pi}\left(\tan_2^{-1}(I_y, I_x) \bmod \pi\right) \quad (4)$$

They are B group for this feature called *bin* varied on certain number, typically 9 bins used in [0, 180] (shown by Figure 2). The index of bin i is lie on $[0, B-1]$ where the range of each bin are in $[wi, w(i+1))$ for w is computed by (5) and the center c_i given by (6). They are such components in each bin j voted by (7) and (8), where $j = \left\lfloor \frac{\theta}{w} - \frac{1}{2} \right\rfloor$. If j is over $B-1$ than j turn back to the first index (shown by figure 2) or rather $(j+1)\ mod\ B$.

$$w = \frac{180}{B} \quad (5)$$

$$c_i = w\left(i + \frac{1}{2}\right) \quad (6)$$

$$v_j = \frac{\mu}{w}(c_{j+1} - \theta) \quad (7)$$

$$v_{j+1} = \frac{\mu}{w}(\theta - c_j) \quad (8)$$

The cell in the image may consist of several number of pixels, particularly 9 pixels, where there is no overlap pixel on each formed cell. Furthermore, each cell is grouped as a block and may have overlap cell in there. These all shown by figure 1.b as a cell and block from original image. The histogram of this feature extraction method is come out from histogram of every magnitude μ grouped by such bins in each cell (see figure 1.a). All the cell histogram features are grouped once more as a block and become the last form of this feature method or rather call the feature of HOG (9). This feature needs to be normalized by (10) for every cell and block. In addition, H is a block for normalize every cell and a feature for normalize every block.

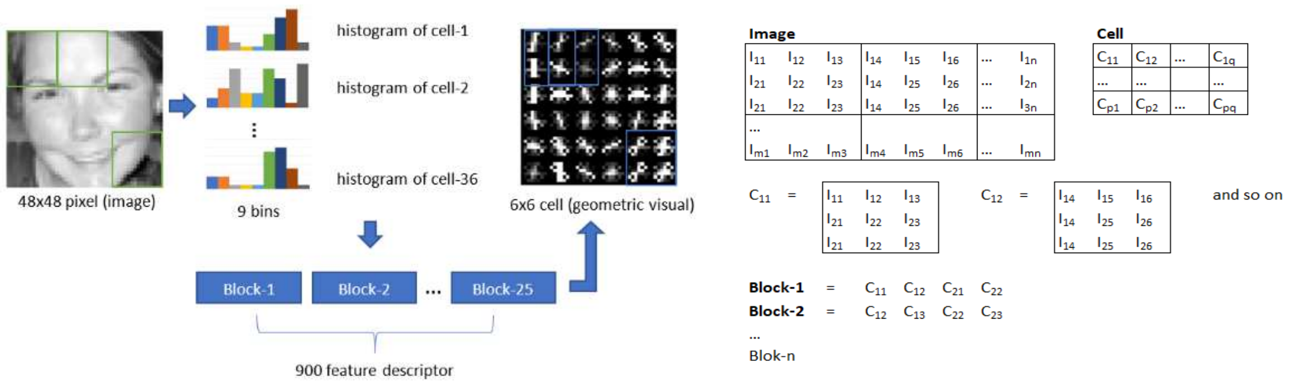
$$HOG_{feature} = [b_1, b_2, \dots b_k]$$

where

$$b_k = [C_1^k, C_2^k, \dots, C_r^k] \text{ and } C_r^k$$

is corresponding cell in b_k block (9)

$$H = \frac{H}{\sqrt{|H|^2 + \varepsilon}} \quad (10)$$



(a) (b)

Fig. 1. Hog (a) Feature Extraction and (b) cell-block orientation

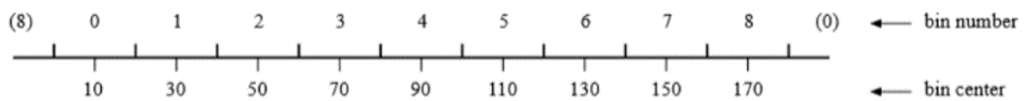


Fig. 2. Bin in HOG

III. Support Vector Machine (SVM)

SVM is a method to classify two class by split feature vector from sample space using a line in R^2 or rather called hyperplane in R^n .This hyperplane assosiated with some vector to support optimal margin, described by Figure 3 [30][31].

Let say we have two group of classes $y \in \{-1, +1\}$ for each data points $x \subset R^2$. Notice from the figure 3, they are separated by black line as two group. This line called SVM *hyperplane*. By using the sign of hyperplane result, SVM determine the class of data point x (11).

$$f_{svm}(x) = sign(w.x + b) \quad (11)$$

$$\mathcal{L}(x, \lambda) = \frac{1}{2}||w||^2 - \Sigma_{i=1}^{n} \alpha_i (y_i(w.x_i + b) - 1) \quad (12)$$

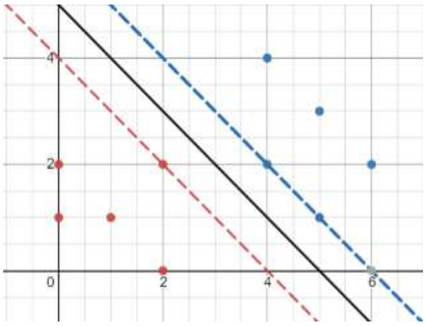


Fig. 3. SVM Hyperplane

SVM work with the boundaries line (fig. 1: red and blue line) that lie on such data point $\boldsymbol{x}_i$ called *support vector*. The maximum gap between two classes (boundaries) is achieved by minimize $\frac{1}{2}||\boldsymbol{w}||^2$ with subject to $y_i(\boldsymbol{w}.\boldsymbol{x}+b) \geq 1$ as a constraint. By using Lagrange function, the SVM hyperplane parameter weight $\boldsymbol{w}$ and bias b are discovered (12), where the corresponding α_i called *Lagrange Multiplier*.

The given data may not linearly separated as figure 1 showed us. By this condition, we have to transform our data point into a certain space R^d (shown by figure 4). In order to transform data, we use Kernel function. This may lead us to "linearly separated data", where the hyperplane not in line form, moreover it can be in the plane or higher dimension (this why *hyperplane* term came out). Several kernel function are shown by table 1.

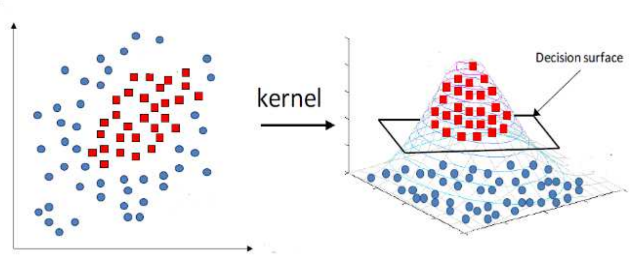


Fig. 4. Data Transformation by Kernel Function

Even we use such kernel to make the data linearly separated, in fact the data may still in a non-linearly separated form. By this condition, SVM hyperplane has change into (13) with the parameter given by (14) as a dual form of (12), with $C \in R$ as soft margin parameter in SVM. The constant value C compromise between margin maximization and training error minimization. A high value for C give a high penalization on errors that lead to hard-margin SVM, while the other is conversely. That is how SVM work for classifier, in particular binary classifier since it classifies two class naturally.

$$f(x_t) = sign\left(\sum_{i=1}^{N} y_i \alpha_i K(x_t, x_i) - b\right) \quad (13)$$

$$\underset{\alpha \in R^N}{\text{minimize}}\, \psi(\alpha) = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} y_i y_j K(x_i, x_j)\alpha_i \alpha_j - \sum_{i=1}^{N}\alpha_i$$

subject to $\quad 0 \leq \alpha_i \leq C, \forall i$

and $\quad \sum_{i=1}^{N} y_i \alpha_i = 0 \quad (14)$

TABLE I. KERNEL FUNCTION

Function	Name
$K(x_t, x_i) = x_t . x_i$	Linear Kernel
$K(x_t, x_i) = (b + x_t . x_i)^n$	Polynomial Kernel
$K(x_t, x_i) = \exp(-\gamma \lVert x_t - x_i \rVert^2)$	RBF Kernel
$K(x_t, x_i) = \exp(-\gamma \lVert x_t - x_i \rVert)$	Exponential Kernel
$K(x_t, x_i) = \tanh(k x_t . x_i - \delta)$	Sigmoid Kernel
$K(x_t, x_i) = (p + x_t . x_i)^q \exp(-\gamma \lVert x_t - x_i \rVert^2)$	Hybrid Kernel

Notice, for Radial Basis Function (RBF) kernel with $\gamma = 1/2\sigma^2$ it also called Gaussian Kernel.

IV. SEQUENTIAL MINIMAL OPTIMIZATION (SMO)

SMO is a simple algorithm to find optimal value of Lagrange multipliers in quadratic programming. It is only use two Lagrange multipliers as a sub-problem and iteratively solve the optimization problem by using equation (14) [11].

$$\underset{\alpha_2}{\text{minimize}}\, \psi(\alpha_2) = \frac{\chi}{2}\alpha_2^2 - \zeta\alpha_2 + \kappa$$

$$\text{subject to} \quad \begin{array}{ll} 0 \leq \alpha_2 \leq C, & \\ \gamma - C \leq \alpha_2 \leq \gamma, & (\text{if } s = 1), \\ -\gamma \leq \alpha_2 \leq -\gamma + C, & (\text{if } s = -1) \end{array} \quad (14)$$

with

$$s = y_1 . y_2, \gamma \subset R \quad (15)$$

$$\zeta = s\gamma K_{11} - s\gamma K_{12} - s + 1 \quad (16)$$

$$\chi = K_{11} + K_{22} - 2K_{12} \quad (17)$$

$$\kappa = \frac{\gamma^2 K_{11}}{2} - \gamma \quad (18)$$

where K_{ij} equal to $K(x_i, x_j)$.

By using first order differentiation this objective function (14) can be analytically solved.

V. EXPERIMENT AND RESULT

This research conducted three experiments scheme in order to find best SVM model to classify happy (labeled as 0) and sad (labeled as 1) face image by using some hyperparameters. By using cross validation (with 5 fold) in train data every hyperparameter were test and select one optimum hyperparameter as a best model to test data. The scheme experiment detail shown by Table II and Figure 5.

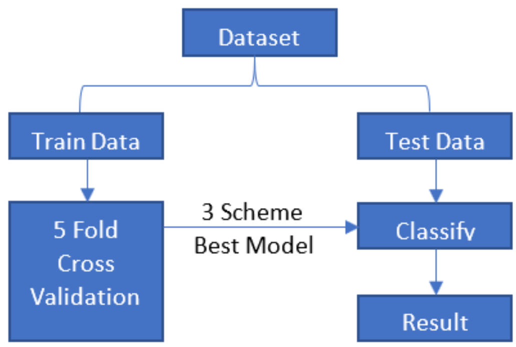


Fig. 5. Research Scheme

TABLE II. KERNEL FUNCTION

Scheme	Kernel	Hyperparameter
1	RBF	C = 0.1 and γ = {1/‖feature‖, 1/(‖feature‖.var_feature (σ^2))
		C = 1 and γ = {1/‖feature‖, 1/(‖feature‖.var_feature (σ^2))
		C = 10 and γ = {1/‖feature‖, 1/(‖feature‖.var_feature (σ^2))
2	Polynomial	C = 0.1, n = 2 and b ={ 0.1, 1, 10}
		C = 1, n = 2 and b ={ 0.1, 1, 10}
		C = 10, n = 2 and b ={ 0.1, 1, 10}
3	Linear	C = {0.1, 1, 10}

Happy and sad image in this research was taken from facial expression recognition 2013 (FER-2013) dataset. There are 100 grayscale face images for each happy and sad expression with 48 x 48 pixels was used. These images has various face pose, some example shown by Figure 6. The dataset then randomly set as train and test data where 150 images are set into train data and 50 images set into test data. Same proportion of happy and sad image for both train and test data.

In order to gain a SVM model (as an output of training phase), sequential minimal optimization (SMO) applied to find support vector along with the associated weight as a model. The model used feature vector as an input gained from feature extraction using HOG feature descriptor $\in \boldsymbol{R}^{900}$. This feature are transformed using radial basis function kernel (RBF) with γ, polynomial kernel with n and b, and linear kernel. In every hyperparameter set $C = \{0.1, 1, 10\}$ in each scheme, cross validation was conducted. Afterward, there is 18 model gained from 5 repetition in cross validation showed their performance in Table (III, IV, … IX). Also, the detail of training and testing phases shown by Figure 7.

Fig. 6. FER-2013 Image Dataset

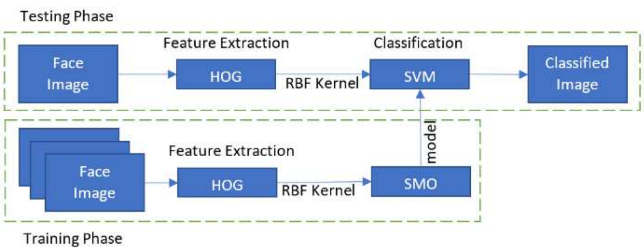


Fig. 7. Training and Testing Phase

Feature vectors are HOG feature descriptor set by 25 block x 4 cell x 9 bin. It has 8 x 8 pixel in every cell and 2 x 2 cell in every block with stride equal to 1. Starting from the grayscale image, all the magnitude value of image was grouped by gradient set into 9 bin in every cell. This group of magnitude formed histogram called histogram of oriented gradient. Afterward, every cell group into block and was normalized by L2 norm. The feature extraction proses shown by Figure 1.a.

First scheme of this research used RBF kernel with two different values of γ. Shown by Table III, for $C = 0.1$, both of them has a similar result, 0.70 and 0.73 where σ^2 is variant of feature vector (from HOG feature descriptor). However, second γ model has low recall value indicating low sad emotion recognition compare to first γ model.

For the rest C value in this scheme (shown by Table IV & V) give same similar pattern, second γ value (the lowest value) gives a better result than other. It indicates the higher curved transformed features, the better result. Best result in this scheme it is shown by second γ value with high "tolerance" C value, 10. That is, 0.79 accuracy value with fair precision and recall value (0.79 & 0.72).

TABLE III. RBF KERNEL FOR $c = 0.1$

Fold	$C = 0.1$ and $\gamma = \frac{1}{\|feature\|}$				$C = 0.1$ and $\gamma = \frac{1}{\|feature\|\sigma^2}$			
	Prec	*Rec*	*f1-score*	*Acc*	*Prec*	*Rec*	*f1-score*	*Acc*
1	0.71	0.80	0.75	0.73	0.75	0.80	0.77	0.77
2	0.75	0.80	0.77	0.77	0.91	0.67	0.77	0.80
3	0.83	0.67	0.74	0.77	0.91	0.67	0.77	0.80
4	0.62	0.53	0.57	0.60	0.64	0.47	0.54	0.60
5	0.61	0.73	0.67	0.63	0.73	0.53	0.62	0.67
average	0.70	0.71	0.70	0.70	0.79	0.63	0.69	0.73

TABLE IV. RBF KERNEL FOR $c = 1$

Fold	$C = 1$ and $\gamma = \frac{1}{\|feature\|}$				$C = 1$ and $\gamma = \frac{1}{\|feature\|\sigma^2}$			
	Prec	*Rec*	*f1-score*	*Acc*	*Prec*	*Rec*	*f1-score*	*Acc*
1	0.71	0.80	0.75	0.73	0.75	0.80	0.77	0.77
2	0.75	0.80	0.77	0.77	0.92	0.80	0.86	0.87
3	0.83	0.67	0.74	0.77	0.85	0.73	0.79	0.80
4	0.62	0.53	0.57	0.60	0.62	0.53	0.57	0.60
5	0.61	0.73	0.67	0.63	0.75	0.60	0.67	0.70
average	0.70	0.71	0.70	0.70	0.78	0.69	0.73	0.75

TABLE V. RBF KERNEL FOR $c - 10$

Fold	$C = 10$ and $\gamma = \frac{1}{\|feature\|}$				$C = 10$ and $\gamma = \frac{1}{\|feature\|\sigma^2}$			
	Prec	*Rec*	*f1-score*	*Acc*	*Prec*	*Rec*	*f1-score*	*Acc*
1	0.71	0.80	0.75	0.73	0.75	0.80	0.77	0.77
2	0.75	0.80	0.77	0.77	0.79	0.73	0.76	0.77
3	0.83	0.67	0.74	0.77	0.80	0.80	0.80	0.80
4	0.62	0.53	0.57	0.60	0.62	0.67	0.65	0,63
5	0.61	0.73	0.67	0.63	1.00	0.60	0.75	0.80
average	0.70	0.71	0.70	0.70	0.79	0.72	0.75	0.79

Second scheme shown by Table VI-VII, where polynomial kernel is used with $b = \{0.1, 1, 10\}$ (intercept) and $n = 2$ (degree). In this scheme, all SVM model give similar result around 0.70 accuracy. However, best performance was given by $C = 0.1$ and $b = 1$ SVM model with 0.77 accuracy and fair f1-score 077.

Last scheme, shown by Table IX used linear kernel. The best performance in this scheme given by $C = 1$ or $C = 10$

with same f1 score, as they shared same performance. All the scheme used in this research has low performance in fold 4.

TABLE VI. POLYNOMIAL KERNEL FOR $c = 0.1$

Fold	$C = 0.1$ and $\{b = 0.1, n = 2\}$				$C = 0.1$ and $\{b = 1, n = 2\}$				$C = 0.1$ and $\{b = 10, n = 2\}$			
	Prec	*Rec*	*f1-score*	*acc*	*Prec*	*Rec*	*f1-score*	*Acc*	*Prec*	*Rec*	*f1-score*	*Acc*
1	0.75	0.80	0.77	0.77	0.75	0.80	0.77	0.77	0.71	0.80	0.75	0.73
2	0.92	0.80	0.86	0.87	0.92	0.80	0.86	0.87	0.73	0.73	0.73	0.73
3	0.83	0.67	0.74	0.77	0.79	0.73	0.76	0.77	0.79	0.73	0.76	0.77
4	0.67	0.67	0.67	0.67	0.65	0.73	0.69	0.67	0.60	0.60	0.60	0.60
5	0.75	0.60	0.67	0.70	0.79	0.73	0.76	0.77	0.91	0.67	0.77	0.80
average	0.78	0.71	0.74	0.76	0.78	0.76	0.77	0.77	0.75	0.71	0.72	0.73

TABLE VII. POLYNOMIAL KERNEL FOR $c = 1$

Fold	$C = 1$ and $\{b = 0.1, n = 2\}$				$C = 1$ and $\{b = 1, n = 2\}$				$C = 1$ and $\{b = 10, n = 2\}$			
	Prec	*Rec*	*f1-score*	*acc*	*Prec*	*Rec*	*f1-score*	*Acc*	*Prec*	*Rec*	*f1-score*	*Acc*
1	0.72	0.87	0.79	0.77	0.72	0.87	0.79	0.77	0.72	0.87	0.79	0.77
2	0.75	0.80	0.77	0.77	0.75	0.80	0.77	0.77	0.75	0.80	0.77	0.77
3	0.76	0.87	0.81	0.80	0.76	0.87	0.81	0.80	0.76	0.87	0.81	0.80
4	0.61	0.73	0.67	0.63	0.59	0.67	0.62	0.60	0.61	0.73	0.67	0.63
5	0.91	0.67	0.77	0.80	0.91	0.67	0.77	0.80	0.91	0.67	0.77	0.80
average	0.75	0.79	0.77	0.75	0.75	0.78	0.75	0.75	0.75	0.79	0.76	0.75

TABLE VIII. POLYNOMIAL KERNEL FOR $c = 10$

Fold	$C = 10$ and $\{b = 0.1, n = 2\}$				$C = 0.1$ and $\{b = 1, n = 2\}$				$C = 0.1$ and $\{b = 10, n = 2\}$			
	Prec	*Rec*	*f1-score*	*acc*	*Prec*	*Rec*	*f1-score*	*Acc*	*Prec*	*Rec*	*f1-score*	*Acc*
1	0.72	0.87	0.79	0.77	0.72	0.87	0.79	0.77	0.72	0.87	0.79	0.77
2	0.75	0.80	0.77	0.77	0.75	0.80	0.77	0.77	0.75	0.80	0.77	0.77
3	0.76	0.87	0.81	0.80	0.76	0.87	0.81	0.80	0.76	0.87	0.81	0.80
4	0.61	0.73	0.67	0.63	0.59	0.67	0.62	0.60	0.59	0.67	0.62	0.60
5	0.91	0.67	0.77	0.80	0.91	0.67	0.77	0.80	0.91	0.67	0.77	0.80
average	0.75	0.79	0.77	0.75	0.75	0.78	0.75	0.75	0.75	0.78	0.75	0.75

TABLE IX. LINEAR KERNEL

Fold	$C = 0.1$				$C = 1$				$C = 10$			
	Prec	*Rec*	*f1-score*	*acc*	*Prec*	*Rec*	*f1-score*	*Acc*	*Prec*	*Rec*	*f1-score*	*Acc*
1	0.71	0.80	0.75	0.73	0.72	0.87	0.79	0.77	0.72	0.87	0.79	0.77
2	0.75	0.80	0.77	0.77	0.71	0.80	0.75	0.73	0.71	0.80	0.75	0.73
3	0.83	0.67	0.74	0.77	0.76	0.87	0.81	0.80	0.76	0.87	0.81	0.80
4	0.62	0.53	0.57	0.60	0.61	0.73	0.67	0.63	0.61	0.73	0.67	0.63
5	0.61	0.73	0.67	0.63	0.91	0.67	0.77	0.80	0.91	0.67	0.77	0.80
average	0.70	0.71	0.70	0.70	0.74	0.79	0.76	0.75	0.74	0.79	0.76	0.75

After all scheme are conducted, then the unseen data (feature that not used for train and test in cross-validation) will be recognized by best model achieved before. As result, Shown by Table X, RBF kernel and linear kernel has decreasing result compare to their scheme before, along with linear kernel. However, polynomial consider to be the best SVM model to recognize sad and happy face image in fer2013 dataset.

In addition, comparing our proposed method with [27] by using specific parameter b = 1 and C = 0.1 on quadratic polynomial kernel, in only two expressions happy and sad, we gain better result. Also, in [27] show, for overall performance in all basic expression (7 expression), the accuracy is 0.57 and then by using only two common expression it increased by 0.64 accuracy as a pose-variant reduced. Moreover, by using small and balanced dataset it increased by 0.72 accuracy as our proposed method.

TABLE X. TEST DATA

Kernel	**C**	**Prec**	**Rec**	**f1-score**	**Acc**
RBF: $\gamma = \frac{1}{\|feature\|\sigma^2}$	10	0.72	0.60	0.65	0.62
Polynomial $n = 2$ and $b = 1$	0.1	0.72	0.72	0.72	0.72
Linear	1	0.76	0.61	0.68	0.64
Linear	10	0.76	0.61	0.68	0.64

TABLE XI. TEST DATA

Kernel	**b, C**	**Acc. For Happy and Sad Expression**
Qubit Polynomial Kernel [27]	-	0.64
Proposed Method: Quadratic Polynomial Kernel	b=1, C=0.1	0.72

VI. Conclusion

In this research the best model is given by polynomial kernel, 0.72 accuracy where $b = 1$ and $n = 2$ with $C = 0.1$. however, for each kernel such as RBF kernel, best model given by lowest γ value with $C = 10$ and for linear kernel, best model given by $C = 1$ or $C = 10$.

In addition, reducing some pose variant by involving only two common expression and using small balanced dataset give a fair performance. This can be another task for FER in real world (wild condition) dataset, such FER2013, to select some feature extraction/descriptor that can handle pose-variant in future. It happens not only in conventional approach but also in deep learning approach. And not to be forgotten, there is also a problem in unbalanced dataset.

References

[1] Samadiani N, Huang G, Cai B, Luo W, Chi CH, Xiang Y, He J. A Review on Automatic Facial Expression Recognition Systems Assisted by Multimodal Sensor Data. Sensors (Basel). 2019 Apr 18;19(8):1863. doi: 10.3390/s19081863. PMID: 31003522; PMCID: PMC6514576.

[2] S. S. Shafira, N. Ulfa, H. A. Wibawa and Rismiyati, "Facial Expression Recognition Using Extreme Learning Machine," 2019 3rd International Conference on Informatics and Computational Sciences (ICICoS), Semarang, Indonesia, 2019, pp. 1-6, doi: 10.1109/ICICoS48119.2019.8982443.

[3] Y. He and X. He, "Facial Expression Recognition Based on Multi-Feature Fusion and HOSVD," 2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), Chengdu, China, 2019, pp. 638-643, doi: 10.1109/ITNEC.2019.8729003.

[4] Y. Huang, F. Chen, S. Lv, and X. Wang, “Facial Expression Recognition: A Survey,” Symmetry, vol. 11, no. 10, p. 1189, Sep. 2019.

[5] Lyons, M.; Akamatsu, S.; Kamachi, M.; Gyoba, J. Coding facial expressions with gabor wavelets. In Proceedings of the Third IEEE International Conference on Automatic Face and Gesture Recognition, Nara,Japan, 14–16 April 1998; pp. 200–205.

[6] Lundqvist, D.; Flykt, A.; Öhman, A. The Karolinska directed emotional faces (KDEF). CD ROM Dep. Clin. Neurosci. Psychol. Sect. Karolinska Inst. 1998, 91, 630.

[7] Pantic, M.; Valstar, M.; Rademaker, R.; Maat, L. Web-based database for facial expression analysis. In Proceedings of the IEEE International Conference on Multimedia and Expo, Amsterdam, The Netherlands, 6 July 2005, p. 5.

[8] Yin, L.;Wei, X.; Sun, Y.;Wang, J.; Rosato, M.J. A 3D facial expression database for facial behavior research. In Proceedings of the 7th International Conference on Automatic Face and Gesture Recognition (FGR06), Southampton, UK, 10–12 April 2006; pp. 211–216.

[9] Whitehill, J.; Littlewort, G.; Fasel, I.; Bartlett, M.; Movellan, J. Toward practical smile detection. IEEE Trans. Pattern Anal. Mach. Intell. 2009, 31, 2106–2111.

[10] Lucey, P.; Cohn, J.F.; Kanade, T.; Saragih, J.; Ambadar, Z.; Matthews, I. The extended cohn-kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops, San Francisco, CA, USA, 13–18 June 2010; pp. 94–101.

[11] Wang, S.; Liu, Z.; Lv, S.; Lv, Y.; Wu, G.; Peng, P.; Chen, F.; Wang, X. A natural visible and infrared facial expression database for expression recognition and emotion inference. IEEE Trans. Multimed. 2010, 12, 682–691.

[12] Gross, R.; Matthews, I.; Cohn, J.; Kanade, T.; Baker, S. Multi-PIE. Image Vis. Comput. 2010, 28, 807–813.

[13] Lucey, P.; Cohn, J.F.; Prkachin, K.M.; Solomon, P.E.; Matthews, I. Painful data: The UNBC-McMaster shoulder pain expression archive database. In Proceedings of the Face and Gesture 2011, Santa Barbara, CA, USA, 21–25 March 2011; pp. 57–64.

[14] Zhao, G.; Huang, X.; Taini, M.; Li, S.Z.; Pietikäinen, M. Facial expression recognition from near-infrared videos. Image Vis. Comput. 2011, 29, 607–619.

[15] Valstar, M.F.; Mehu, M.; Jiang, B.; Pantic, M.; Scherer, K. Meta-analysis of the first facial expression recognition challenge. IEEE Trans. Syst. Man Cybern. Part B (Cybern.) **2012**, 42, 966–979.

[16] Dhall, A.; Goecke, R.; Lucey, S.; Gedeon, T.; others. Collecting large, richly annotated facial-expression databases from movies. IEEE Multimed. **2012**, 19, 34–41.

[17] Carrier, P.L.; Courville, A.; Goodfellow, I.J.; Mirza, M.; Bengio, Y. FER-2013 Face Database; Universit de Montral: Montreal, QC, Canada, 2013.

[18] Kaulard, K.; Cunningham, D.W.; Bülthoff, H.H.; Wallraven, C. The MPI facial expression database—A validated database of emotional and conversational facial expressions. PLoS ONE **2012**, 7, e32321.

[19] Mavadati, S.M.; Mahoor, M.H.; Bartlett, K.; Trinh, P.; Cohn, J.F. Disfa: A spontaneous facial action intensity database. IEEE Trans. Affect. Comput. **2013**, 4, 151–160.

[20] Zhang, X.; Yin, L.; Cohn, J.F.; Canavan, S.; Reale, M.; Horowitz, A.; Liu, P.; Girard, J.M. Bp4d-spontaneous: A high-resolution spontaneous 3d dynamic facial expression database. Image Vis. Comput. **2014**, 32, 692–706.

[21] Du, S.; Tao, Y.; Martinez, A.M. Compound facial expressions of emotion. Proc. Natl. Acad. Sci. USA **2014**, 111, E1454–E1462.

[22] Li, S.; Deng, W.; Du, J. Reliable crowdsourcing and deep locality-preserving learning for expression recognition in the wild. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, Hawaii,USA, 22–25 July 2017; pp. 2852–2861.

[23] Li, S.; Deng, W. Reliable crowdsourcing and deep locality-preserving learning for unconstrained facial expression recognition. IEEE Trans. Image Process. **2019**, 28, 356–370.

[24] V. Tümen, Ö. F. Söylemez and B. Ergen, "Facial emotion recognition on a dataset using convolutional neural network," 2017 International Artificial Intelligence and Data Processing Symposium (IDAP), Malatya, 2017, pp. 1-5, doi: 10.1109/IDAP.2017.8090281.

[25] Breuer, R.; Kimmel, R. A deep learning perspective on the origin of facial expressions. arXiv **2017**, arXiv:1705.01842.

[26] Mollahosseini, A., Chan, D., and Mahoor, M. H., “Going Deeper in Facial Expression Recognition using Deep Neural Networks”, *arXiv e-prints*, 2015

[27] Saeed, S.; Baber, J.; Bakhtyar, M.; Ullah, I.; Sheikh, N.; Dad, I.; Sanjrani, A.A. Empirical Evaluation of SVM for Facial Expression Recognition. Int. J. Adv. Comput. Sci. Appl. **2018**, 9, 670–673.

[28] N. Dalal and W. Triggs. “Histogram of Oriented Gradients for Human Detection”. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR). Vol. 1. No. 3. pp 886-893. 2005.

[29] D. Alamsyah. “Pengenalan Mobil pada Citra Digital Menggunakan HOG-SVM”. Jurnal Teknik Informatika dan Sistem Informasi (JATISI). Vol. 1. No. 2. pp.162-168. 2017.

[30] P. Negri. “A MATLAB SMO Implementation to Train a SVM Classifier: Application to Multi-Style License Plate Number Recognition”. Image Processing Online (IPOL). pp 51-70. 2018. https://doi.org/10.5201/ipol.2018.173

[31] Cortes C, Vapnik V. “Support-vector networks”. Machine learning. 1995;20(3):273–97.