IMPLEMENTATION ALGORITHM OF HIDDEN MARKOV MODELS FOR FACIAL GESTURE RECOGNITION

Roberto Pangihutan Situmeang

Teknik Informatika – Universitas Komputer Indonesia
Jl. Dipatiukur 112-114 Bandung
E-mail : robertopasitumeang2@gmail.com

ABSTRACT

The human face provides a lot of information, a lot of interesting things that can be considered and studied intensively. Humans interact with one another using a variety of facial gesture. Gesture from facial can describe the expressions. In this research is the implementation of the Hidden Markov Models method for facial gesture recognition.

The stages of this research process that is done every image preprocessing is resize 128x128, grayscale, canny edge detection. Threshold image is divided into 4 sections with a size of 64x64. Discrete feature extraction Cousine Transform (DCT) on each part, and each part DCT matrix is converted into vectors. Feature extraction Principle Component Analysis (PCA) generate a 4x4 matrix and then the matrix is formed into a sequence. Sequence PCA value is normalization process. The results of the normalization process is used for classification Hidden Markov Model (HMM).

The results of this study is use 120 data classification training of image data to detect facial expressions in real time for testing, and generate output types are recognizable facial expressions. Based on test results Confussion Matrix using 166 times the real-time detection accuracy value then obtained an average of 50% correct classification and 50% wrong.

Keywords: *Hidden Markov Model, Discrete Cousine Transform, Principle Component Analysis, image classification, image processing, facial gesture*, expression.

1 INTRODUCTION

The face is a part of the human body that are the focus in social interaction. The face is also one of the unique parts of the human body that have different characteristics. The face has an important role to show emotion / expression. The human face provides a lot of information, a lot of interesting things that can be considered and studied intensively. Several studies that gather information from the human face is a face recognition and facial gesture recognition. When humans interact with each other, they use a wide range of facial cues to convey information. Facial cues that are formed can convey certain facial expressions.

In research on recognition of facial expressions based on shape and texture, concluded that the method used to produce the percentage of the truth by the percentage value that is the expression of a smile 42.86%, regular 0%, sullen expression 0%, the blinking 28.58%, the expression gleaming 42.86%, a shocked expression 28.58 %, so the research on facial expression recognition based on shape and tekst ur still possible to do further using other methods, the research uses a method SMOM to describe the expression and methods ESTM to seek common image [1].

Based on these studies, the study conducted by the method implementations Hidden Markov Model (HMM) for the introduction of facial cues (Face Gesture). Due to the research on Hand Gesture Recognition using a real-time tracking method and Hidden Markov Models, which concluded that the method of Hidden Markov Model (HMM) is well used in recognizing hand gestures (Hand Gesture) [2]. Hidden Markov Model is modeling the probability of a system to search for the parameters Markov unknown to acquire the systems analysis. Method Hidden Markov Model (HMM) is able to handle the change statistics of the image, the model elements using probabilities. One application is in image processing [3].

In this research will be the introduction of facial cues (face gesture). Generally, this study was designed in several stages of the face detection, face extraction and facial gesture recognition. Face detection is the stage to define and separate the parts of a still image is defined as the face using Haar cascade. The face extraction is the stage for taking part of the facial features. The introduction of facial gesture was a stage in facial gesture recognition using Hidden Markov Model (HMM).

2 RESEARCH

2.1 Overview

The main objective of this thesis is to see whether the method of Hidden Markov Models can recognize facial cues in real time in terms of how accurately cue recognizable face by building a system that can

perform face recognition gestures. The process flow diagram for the system architecture and the training data flow testing of the system architecture, can be seen in Figure 1 as follows:

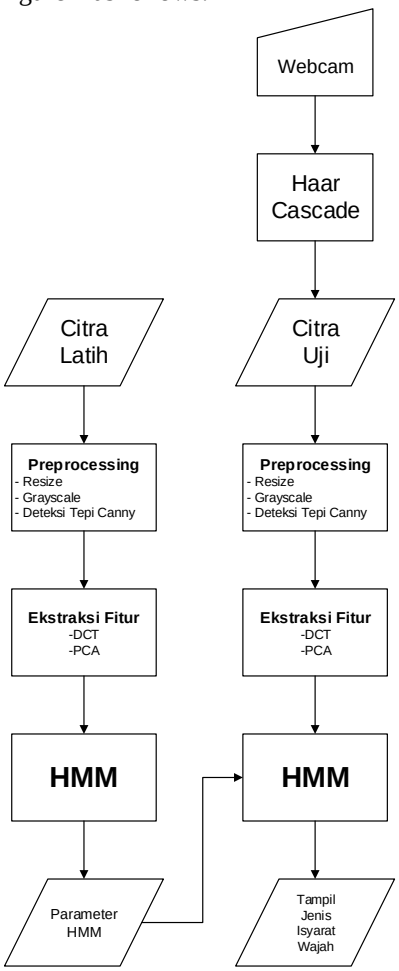


Figure 1. System Architecture

In the groove system architecture for imagery trainer explained there are several stages of training data cue face done preprocessing, namely Resize facial image with 128x128 and the image process is carried Grayscale, then the Edge Detection Canny, then the result of edge detection Canny feature extraction DCT and PCA further processed on HMM algorithm, HMM algorithm results in the form of parameters, which are then stored HMM parameters. In the groove system architecture test images described are several stages of face detection and image capture is defined as the face using the Haar Cascade via webcam in real time, then the image becomes the test data. The next stage of test data entered into the process of preprocessing, subsequent feature extraction, then processed using HMM algorithm by taking the training data as a reference parameter types of gestures facial recognition, then the system displays the type of facial cues.

2.2 Data Input

Data input will be taken on this system is an image taken during a motion on the face. So clearly the image of a gesture, and in this study determines the category of type of movement contained in the image. Each process on image processing using pixel values of the image for calculation. Each pixel in the image is composed of three elements of basic colors namely Red (R), Green (G) and Blue (B), which is often called RGB, and each element has a value between 0-255 that into code color depth of each element. For example, the calculation of the processes of image processing, image used angry and happy faces.



Figure 2. “Normal” Face



Figure 3. “Angry” Face



Figure 4. “Happy” Face



Figure 5. “Sad” Face

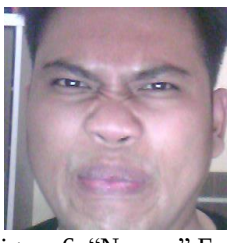


Figure 6. “Nausea” Face



Figure 7. “Shocked” Face

2.2.1 Preprocessing

To make the process of image matrix calculations, counting example using Figure 2 is the image of "Face Image Ordinary" (the dimensions of 345 x 345 pixels) which will be used as a matrix in advance. Will then generate a matrix-forming image is converted into a matrix which has a matrix 3 colors, namely Red, Green and Blue.

1. Resize

After taking the matrix of the image, resize it done. Resize can be called the stage of normalization dimensional image that is a process of changing the image size be 128x128 pixels. The goal, to equalize dimensional image of each image input, so that the extraction process there is no difference dimensional image of the face image data matrix. The method is used on the resize bicubic interpolation method [4].

2. Grayscale

Grayscale is an image that has a pixel value of every single sample. Imagery displayed is gray, black color variation in the intensity of the weakest part and white on the intensity of the strongest. Then the analysis conducted that make the matrix grayscale using the formula Luminosity [5].

$$Grayscale = 0.2989 \times R + 0.5870 \times G + 0.1141 \times B \quad (1)$$

Example calculations can be seen in the manual grayscale grayscale calculation [1.1] below:

$$grayscale[1,1] = (0.2989\ x\ R) + (0.5870\ x\ G) + (0.1141\ x\ B)$$

$$grayscale[1,1] = (0.2989\ x\ 253) + (0.5870\ x\ 246) + (0.1141\ x\ 254)$$

$$grayscale[1,1] = (75.6217) + (144.402) + (28.9814)$$

$$grayscale[1,1] = 249.0051$$

Then rounded to 249.

3. Canny Edge Detection

The first step on the canny edge detection is the multiplication result matrix grayscale with a Gaussian filter with the aim of eliminating the existing noise in the image by implementing convolution with a Gaussian filter equation [5]:

$$Gaussian = \begin{bmatrix} 0.0121 & 0.0261 & 0.0337 & 0.0261 & 0.0121 \\ 0.0261 & 0.0561 & 0.0721 & 0.0561 & 0.0261 \\ 0.0337 & 0.0721 & 0.0935 & 0.0721 & 0.0337 \\ 0.0261 & 0.0561 & 0.0721 & 0.0561 & 0.0261 \\ 0.0121 & 0.0261 & 0.0337 & 0.0261 & 0.0121 \end{bmatrix}.$$

$$Grayscale \quad (2)$$

How to calculate gaussian can be explained in the example Gaussian [1,1] as follows:

Gaussian[1,1] =(0.0935x249 + 0.0721x248 + 0.0337x249 + 0.0721x249 + 0.0561x248 + 0.0261x250 + 0.0337x249 + 0.0261x248 + 0.0121x250)

Gaussian[1,1] = 106.0199

Than rounder to 106.

The next step edge detection process is executed using Sobel operator is used to calculate the color difference in pixel being calculated with the surrounding pixels horizontally (Gx) and vertical (Gy). Here formula of Gx [5]:

$$Gx = \text{Sobel } Gx \text{ . Gaussian Result} \quad (3)$$

And equation of sobel Gx as below:

$$Sobel\ Gx = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad (4)$$

How to calculate the Gx Sobel can be explained in the example Sobel Gx [1,1] where the result of the following calculation [7]:

Gx[1,1] = (2 x 145) + (1 x 106)
Gx[1,1] = 487

Formula of Gy as follows::

$$Gy = \text{Sobel } Gy\text{. Gaussian Result} \quad (5)$$

And equation of sobel Gy as below:

$$\text{Sobel } Gy = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad (6)$$

How to calculate Gy Sobel can be explained in the example Sobel Gy [1,1] where the result of the following calculation:

Gy (1,1) = ((-2) x 144) + ((-1) x 197)

Gy (1,1) = -485

Having in mind all the results of Gx and Gy, can be calculated next step is determining the gradient magnitude (edge strength).

To calculate the value of each coordinate then $|G|_{(x,y)}$ take the value of each coordinate $Gx_{(x,y)}$ and the value of $Gy_{(x,y)}$.

Example calculation for value $|G|_{(1,1)}$, then the value is taken $Gx_{(1,1)}$ =487, and the value of $Gy_{(1,1)}$ = (-485). So that the calculation examples can be seen as follows [7]:

$$|G|_{(1,1)} = \sqrt{(487)^2 + (-485)^2} = \sqrt{237{,}169 + 235{,}225} = \sqrt{472{,}394}$$

$|G|_{(1,1)}$ = 687,3092.

Than rounded to 687.

Then the next step is to determine the direction of the edge by using the equation,

$$\theta = \boldsymbol{arctan}\left(\frac{Gy}{Gx}\right) \quad (7)$$

Then to determine the angle of the gradient by dividing into four directions can be seen in Figure 8 below [7]:

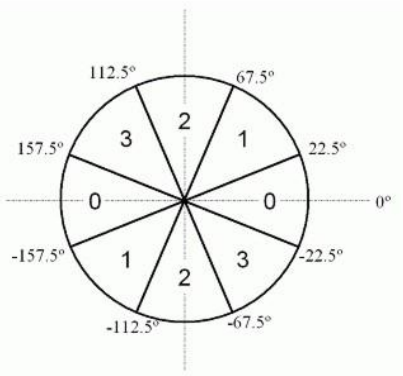


Figure 8 Angle of the gradient

The division into four directions are as follows:

1. If θ>0 and θ<22.5 or θ>157.5 and θ< -157.5 than the gradient corner is 0.
2. If θ>22.50 and θ<67.5 or θ> -157.5 and θ< -112.5 than the gradient corner is 45.
3. If θ>67.50 and θ<112.5 or θ> -112.5 and θ< -67.5 than the gradient corner is 90.
4. If θ>112.50 and θ<157.5 or θ> -67.5 and θ< -22.5 than the gradient corner is 135.

In the following calculation example using $Gx_{(1,1)}$ and $Gy_{(1,1)}$ as follows:

$$\theta = \tan^{-1}\left(\frac{-485}{487}\right) = \tan^{-1}(-0.99589322) = -44.8821$$

Than the gradient corner $= 135°$

Then the elimination of non-maximum suppression (non-maximum pressure) with the following steps [7]:

1. For each pixel (x, y), specify the nearest gradient direction in accordance with the direction 8 of neighborhood. Starting from the gradient magnitude values (2, 2) in order to have the direction of eight neighborhoods. Hereinafter called the gradient magnitude mg.
2. For the gradient angle = 0, if mg (x, y)> mg (x, y - 1) and mg (x, y)> mg (x, y + 1), then the non-maximum value suppression (x, y) is mg (x, y). If not, then the non-maximum value suppression (x, y) is 0.
3. For the gradient angle = 45, if mg (x, y)> mg (x + 1, y - 1) and mg (x, y)> mg (x - 1, y + 1), then the value of non-maximum suppression (x, y) is mg (x, y). If not, then the non-maximum value suppression (x, y) is 0.
4. For the gradient angle = 90, if mg (x, y)> mg (x - 1, y) and mg (x, y)> mg (x + 1, y), then the non-maximum value suppression (x, y) is mg (x, y). If not, then the non-maximum value suppression (x, y) is 0.
5. For the gradient angle = 135, if mg (x, y)> mg (x - 1, y - 1) and mg (x, y)> mg (x + 1, y + 1), then the value of non-maximum suppression (x, y) is mg (x, y).
6. If not, then the non-maximum value suppression (x, y) is 0.

Then do thresholding or floating, which is the process of separation of pixels based on the degree of gray is smaller than its threshold value. In the calculation method used in this study to determine the Otsu threshold value, which will generate a value. Then the threshold value is used to convert the image data into binary data with the aim to further process becomes easier.

The principle of Otsu method are described as follows. First of all, the probability of intensity values i in the histogram calculated by the following equation 8 [4]:

$$p(i) = \frac{n_i}{N}, p(i) \geq 0, \sum_{i=1}^{t} p(i) = 1 \quad (8)$$

With the stated number of pixel intensity ni i and N denotes the number of all pixels in the image. If the histogram is divided into two classes (object and background), the weighting of the second class is declared in each of the following equations 9 and 10:

$$w_1(t) = \sum_{i=1}^{t} p(i) \quad (9)$$

$$w_2(t) = \sum_{i=t+1}^{L} p(i) = 1 - w_1(t) \quad (10)$$

In this case, L stated number of gray levels. Average second grade is calculated by an equation 11 and equation 12 below:

$$m_1(t) = \sum_{i=1}^{t} i.\, p(i)/w_1(t) \quad (11)$$

$$m_2(t) = \sum_{i=1}^{t} i.\, p(i)/w_2(t) \quad (12)$$

The second variant of the class represented by the equation 13 and equation 14 below:

$$\sigma_1{}^2(t) = \sum_{i=1}^{t} (1 - m_1)^2 . \frac{p(i)}{w_1(t)} \quad (13)$$

$$\sigma_2{}^2(t) = \sum_{i=t+1}^{L} (1 - m_2)^2 . \frac{p(i)}{w_2(t)} \quad (14)$$

The total variance can be expressed by equation 15 below:

$$\sigma^2(t) = \sigma_W{}^2(t) + \sigma_B{}^2(t) \quad (15)$$

In this case, $\sigma_w{}^2$ named as *Within-Class Variance* (WCV) and $\sigma_B{}^2$ named as *Between-Class Variance* (BCV). WCV can be expressed by the equation 16 below:

$$\sigma_W{}^2(t) = W_1(t) . \sigma_1(t)^2 + W_2(t) . \sigma_2(t)^2 \quad (16)$$

The above formula shows that the WCV is the number of individual classes variances that have been weighted by the probability of each class. The BCV is expressed by equation 17 below:

$$\sigma_B{}^2(t) = W_1 . [m_1(t) - m_T]^2 + W_2 . [m_2(t) - m_T]^2 \quad (17)$$

In this case, m_T is the average total ($m_T = \sum_{i=1}^{N} i.\, P(i)$).

The optimum threshold value can be obtained in 2 ways. The first way is done by minimizing WCV. The second way is done by maximizing the BCV. However, based on these two ways, the second way is to save computing[4].

4. Feature Extraction

The next stage is the extraction of features from the image. The purpose of this feature extraction to get the hash value of the image. In this study used two feature extraction, feature extraction first is Discrete Cosine Transform (DCT), then the resulting value in the feature extraction feature extraction is processed by Principal Component Analysis (PCA) [8].

Before the DCT process, the distribution of the image into sub-images with size 64x64, then formed four sub-image matrix, DCT process is then performed. Then do the extraction of features or feature extraction with a discrete cosine transform [9].

Then the payoff matrix DCT transformed into vectors to construct a matrix into one line down. The results of each feature extraction DCT next section combine to form a matrix. Then the DCT matrix will be calculated by the feature extraction Principal Component Analysis (PCA). then take the value of the coefficient matrix of variables coeff PCA [10], as follows:

Coefficients PCA

$$\begin{bmatrix} 0.2714 & 0.9199 & -0.1679 & -0.2280 \\ 0.1034 & 0.0997 & 0.9712 & -0.1900 \\ 0.9540 & -0.2915 & -0.0695 & 0.0107 \\ 0.0747 & 0.2427 & 0.1539 & 0.9549 \end{bmatrix}$$

Then the PCA coefficients are organized so as to form a vector, such as the following:

Coefficients PCA =

$$[0.2714 \quad 0.1034 \quad 0.9540 \quad \cdots \quad 0.9549]$$

2.2.2 Modelling, Normalization Data Sequence and Training

Process modeling is the process of providing a sign / symbol on a train image facial expression will do the training, modeling examples can be seen in table 1 below:

Table 1 Modelling Data Training

Image Expression	Symbol
Normal	N
Happy	B
Angry	H
Nausea	M
Sad	S
shocked	T

While the process of training data sequence is a process for converting PCA matrix of each image trainer to shape facial expression vectors. The process of training data sequence performed in conjunction with the establishment of training data label. In this study, using 120 datasets with 20 datasets for each type of expression.

Further training data sequences PCA performed indexing value (normalized) for all training data. This process is done because the process sequences HMM only receive the value of non-negative integers. As for how the normalization used are as follows [11]:

1. The length of the interval of each index = (maximum value - minimum value) / total index.
2. Then each value sequences on the value of PCA compiled into an index according to the index that has been created.
3. In addition, each value sequences that have been compiled sequences written into the table.

Then the value of the normalization is used as an observation in the HMM.

2.2.3 Hidden Markov Model

Hidden Markov Model or better known as the Hidden Markov Model (HMM) is a statistical model of a system that is assumed to be a process of Markov with a parameter that is not known, and the challenge

is to determine the parameters of the hidden (state) of parameters that can be observed (observer). The parameters set can then be used for further analysis, for example for pattern recognition applications (Pattern Recognition). A HMM can be considered as a dynamic Bayesian Network simplest.

In general Markov models (Vanilla / Visible Markov Model), its state can be observed directly, therefore the probability of the state transition (state) being the only parameter. Inside the hidden Markov model, its state can not be observed directly, but can be observed are the variables that are affected by the state. Each state has a probability distribution over the tokens output that may arise. Therefore, a series of tokens generated by HMM gives some information about the sequence of state-state [12].

Here is a picture of the flow of flowchart HMM for the training can be seen in Figure 9.

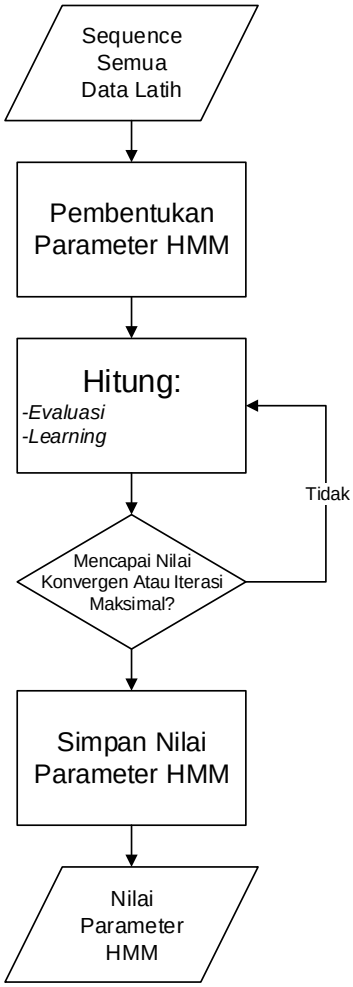


Figure 9 Flowchart HMM for training

Here is a picture of the flow of flowchart HMM for testing can be seen in Figure 10.

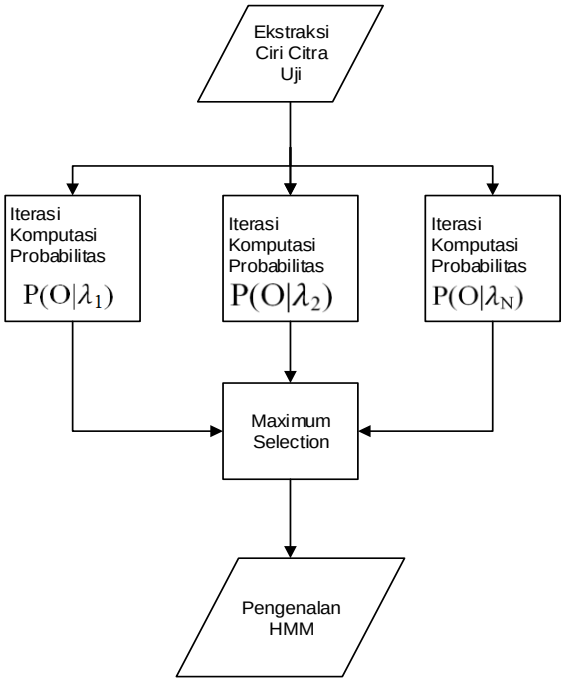


Figure 10 Flowchart HMM for Testing

2.2.3.1 Formation of HMM Parameter

HMM has three main parameters that have been obtained, the HMM parameters can be written in the form λ = (N, M, A, B, π), these parameters can be written in terms of probability to be analyzed [13].

Where:
M = number of state = 6
N = long Sequence = 16

$A_{MM} = Transition = \frac{1}{M}, \forall\, m, m = \frac{1}{6} = 0.16667$

$B_{MN} = Emission = \frac{1}{N}, \forall\, m, m = \frac{1}{16} = 0.0625$

$\pi_0 = 1$

1. Parameter π

Is the probability of occurrence of a state in the beginning. π parameter is written as follows:

$\pi_0 = 1$

2. Parameter A

Is the probability of the position of a state against all existing state. Initial transition probabilities before each iteration of the training data will be worth the same expression. The calculation of probabilities early transition can be seen as below:

$$A(MM) = \begin{bmatrix} 0.1667 & 0.1667 & 0.1667 & 0.1667 & 0.1667 & 0.1667 \\ 0.1667 & 0.1667 & 0.1667 & 0.1667 & 0.1667 & 0.1667 \\ 0.1667 & 0.1667 & 0.1667 & 0.1667 & 0.1667 & 0.1667 \\ 0.1667 & 0.1667 & 0.1667 & 0.1667 & 0.1667 & 0.1667 \\ 0.1667 & 0.1667 & 0.1667 & 0.1667 & 0.1667 & 0.1667 \\ 0.1667 & 0.1667 & 0.1667 & 0.1667 & 0.1667 & 0.1667 \end{bmatrix}$$

3. Parameter B

Is the probability of occurrence of a state in the entire state of existing rows. Initial probability in HMM can be written in matrix form, it can be determined by the following equation:

B(M, N) =

0.0625	0.0625	0.0625	0.0625	0.0625	...	0.0625	0.0625
0.0625	0.0625	0.0625	0.0625	0.0625	...	0.0625	0.0625
0.0625	0.0625	0.0625	0.0625	0.0625	...	0.0625	0.0625
0.0625	0.0625	0.0625	0.0625	0.0625	...	0.0625	0.0625
0.0625	0.0625	0.0625	0.0625	0.0625	...	0.0625	0.0625
0.0625	0.0625	0.0625	0.0625	0.0625	...	0.0625	0.0625

2.2.3.2 Training of HMM

HMM in order to do the training data, there are two fundamental problems with the HMM to be resolved, namely [14]:

1. Evaluation

Ie it will look for the P (O | λ) or probability of the sequence of observation if given HMM: λ = (A, B, π). These opportunities can be determined induction algorithms using inductive forward. Than P (O | λ) can be calculated as follows [15]:

a. Initialization

For state (t) =1, which included observations (1) = 1

$\alpha_i(m) = \pi(m) b_m(O_1)$ (18)

$\alpha_1(1) = \pi(1) b_1(O_1)$ = (1)(0.0625) = 0.0625
$\alpha_2(1) = \pi(1) b_1(O_1)$ = (1)(0.0625) = 0.0625
$\alpha_3(1) = \pi(1) b_1(O_1)$ = (1)(0.0625) = 0.0625
$\alpha_4(1) = \pi(1) b_1(O_1)$ = (1)(0.0625) = 0.0625
$\alpha_5(1) = \pi(1) b_1(O_1)$ = (1)(0.0625) = 0.0625
$\alpha_6(1) = \pi(1) b_1(O_1)$ = (1)(0.0625) = 0.0625

b. Induction

At this stage it will be calculated $\alpha_2(j)$and $\alpha_3(j)$ probability on t = 2 as follows :

$\alpha_{i+1}(j) = \left[\sum_{i=1}^{n} \alpha_1(i)\alpha_{ij}\right] b_j(O_2)$ (19)

$\alpha_2(1) = \left[(\alpha_1(1)\alpha_{12}) + (\alpha_1(2)\alpha_{12}) + (\alpha_1(3)\alpha_{12}) + (\alpha_1(4)\alpha_{12}) + \left(\alpha_1(5)\alpha_{12} + (\alpha_1(6)\alpha_{12})\right)\right] b_1(O_2)$

$\alpha_2(j) = [(0.0625)(0.1667) + (0.0625)(0.1667) + (0.0625)(0.1667) + (0.0625)(0.1667) + (0.0625)(0.1667) + (0.0625)(0.1667)](0.0625)$

$\alpha_2(j) = [(0.01041875) + (0.01041875) + (0.01041875) + (0.01041875) + (0.01041875) + (0.01041875)](0.0625)$

$\alpha_2(j) = [(0.0625125)](0.0625)$= 0.003907

Probability t = 3 as follows :

$\alpha_3(j) = [(0.0625)(0.1667) + (0.0625)(0.1667) + (0.0625)(0.1667) + (0.0625)(0.1667) + (0.0625)(0.1667) + (0.0625)(0.1667)](0.0625)$

$\alpha_3(j) = [(0.01041875) + (0.01041875) + (0.01041875) + (0.01041875) + (0.01041875) + (0.01041875)](0.0625)$

$\alpha_3(j) = [(0.0625125)](0.0625)$= 0.003907

⋮

until probability t = 6 as follows:

$\alpha_6(j) = [(0.0625)(0.1667) + (0.0625)(0.1667) + (0.0625)(0.1667) + (0.0625)(0.1667) + (0.0625)(0.1667) + (0.0625)(0.1667)](0.0625)$

$\alpha_6(j) = [(0.01041875) + (0.01041875) + (0.01041875) + (0.01041875) + (0.01041875) + (0.01041875)](0.0625)$

$\alpha_6(j) = [(0.0625125)](0.0625)$= 0.003907

c. Termination

From the termination stage, the algorithm produces the settlement as follows:

$P(O \mid \lambda) = \sum_{i=1}^{n} a_T(i)$ (20)

$P(O \mid \lambda) = a_1(1) + a_2(1) + a_3(1) + a_4(1) + a_5(1) + a_6(1) + a_1(2) + .. + a_5(6) + a_6(6)$

$P(O \mid \lambda) = 0.140653125$

2. Learning

To resolve this issue use Baum-Welch algorithm. In the Baum-Welch algorithm, defined three variables, namely variable forward, backward variables, and variables ξ(i, j). Variable forward and backward variables will be used in the calculation of the variable ξ(i, j) and variable $\gamma_t(i)$.

The first variable, variable forward has been defined in equation (18), as well as the induction phase is given by equation (19).

The second variable, which is defined as the backward variables, $\beta_t(i)$= P$(O_{t+1}, O_{t+2,\ldots,} O_T | Q_t = S_i, \lambda)$ ie, the probability of the partial observation sequence $O = O_1, O_2, \ldots, O_T$, given state S_i at time t and the model λ. Next, $\beta_t(i)$ can be solved using induction as follows [14]:

a. Initialization

$\beta_T(i)$=1 (21)

b. Induction

$\beta_t(i) = \sum_{j=1}^{N} a_{ij}\, b_j(O_{t+1})\beta_{t+1}(j), \quad t = T-1, T-2, \ldots, 1, \; 1 \le j \le N$ (22)

The third variable, namely variable $\xi(i, j)$ defined as,

$\xi(i, j)$= P$(Q_t = S_i, Q_{t+1} = S_j, O | \lambda)$ (23)

namely, the opportunity at the time t are in state S_i and at time t + 1 are in the state S_j, if given sequence O and the model λ.

Equation (24) writable:

$$\xi(i,j)=\frac{P(Q_t=S_i,Q_{t+1}=S_j,O|\lambda)}{P(O|\lambda)} \quad (24)$$

or can be expressed by the forward-backward variables as:

$$\xi_t(i,j)=P(q_t=S_i,q_{t+1}=S_j\mid O,\lambda)=\frac{\alpha_t(i)a_{ij}b_j(O_{t+1})\beta_{t+1}(i)}{\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_t(i)a_{ij}b_j(O_{t+1})\beta_{t+1}(i)} \quad (25)$$

2.2.3.3 Face detection (Haar Cascade Classifier)

The method used in this study to detect the presence of a face from the webcam is Haar Cascade Classifier. Where the function of the detector face each location will receive the image in order to classify the area of the face or not the face area.

Haar Cascade Classifier stored in an XML file with an extension that serves to decide how to classify each of the locations where the face. Classifier that can be downloaded there are several kinds, including: classifying the face of pond front (frontal face), the classification of facial profile (profile face), classifying the entire body (full body / pedestrian), the classification for the upper body (upper body) and classification of body the bottom (lower body).

In OpenCV, Haar Cascade Classifier classifier to detect faces looking ahead there are several options that bias is used, namely [16]:

1. "haarcascade_frontalface_default.xml"
2. "haarcascade_frontalface_alt.xml"
3. "haarcascade_frontalface_alt2.xml"
4. "haarcascade_frontalface_alt_tree.xml"

The fourth classification will provide detection results vary, depending on the program and the local environment.
In this study used "Haarcascade_frontalface_alt_tree.xml".

2.2.3.4 Facial Expression Recognition

After the process of training HMM models do, is the process of expression recognition in real time. Facial expression recognition solved by decoding.

Will look for the optimal state sequence $Q^*=\{Q_1^*,Q_2^*,....,Q_T^*\}$ if given the observation sequence $O=\{O_1, O_2,....,O_T\}$ and model λ = (A B, ,π). Rows of the best state to be determined in the form of a single track that is connected from $t=1,2,.....,T$.

To solve the problem of Viterbi decoding algorithm is used. Steps in the Viterbi algorithm to determine the best state sequence, namely [14]:

a. Initialization

$$\delta(i)=\pi_i.b_i(O_1) \quad (26)$$
$$\psi_t(i)=0$$

b. Recursion

$$\delta_t(j)=\max_{1\le i\le N}\left[\delta_{t-1}(i).a_{ij}\right].b_j(O_t),\quad 2\le t\le T \quad 1\le j\le N \quad (27)$$

$$\psi_t(j)=\operatorname{argmax}_{1\le i\le N}\left[\delta_{t-1}(i).a_{ij}\right],\quad 2\le t\le T \quad 1\le j\le N \quad (28)$$

1. best state at the time T (Q_T)

$$P^*=\max\left[\delta_T(i)\right]\quad 2\le t\le T \quad (29)$$
$$q_T{}^*=\arg\max\left[\delta_T\right]\quad 1\le i\le N \quad (30)$$

2. The best row state at $t=T-1,T-2,\ldots,1$

$$q_T{}^*=\psi_{t+1}(q_{t+1}{}^*)\quad t=T-1,T-2,...1 \quad (31)$$

The initial step faces are detected using the Haar Cascade, then do image preprocessing and feature extraction, then the results do HMM The process flow facial expression recognition can be seen in Figure 11 below:

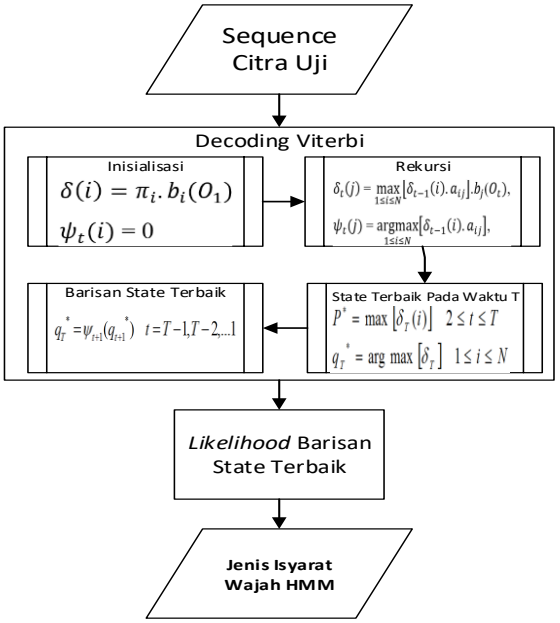


Figure 11. The process flow facial expression recognition.

2.3 Application testing

Performance testing is used to determine the accuracy of which is done using methods Confussion Matrix. Confussion Matrix is a table that visualize the performance of an algorithm. Each column in the matrix represent the predicted class, while each row represents an actual class [17].

Here's a table performance testing methods of Hidden Markov Model on the introduction of facial gestures using the confusion matrix.

Table 2. Testing Performance Confussion Matrix

True	Prediction					
	Normal	Happy	Angry	Nausea	Sad	Shocked
Normal	15	4	2	2	3	2
Happy	2	13	3	2	2	3
Angry	3	3	14	3	2	2
Nausea	2	3	2	12	4	2

Sad	3	2	3	3	13	2
Shocked	2	3	4	3	2	11

From the table, said last calculated total accuracy and precision for a review class EVERY. The result is as follows:

Accuracy = $(15 + 12 + 11 + 9 + 11 + 9)/(15 + 4 + 2 + 2 + 3 + 2 + 2 + 13 + 3 + 2 + 2 + 3 + 3 + 3 + 14 + 3 + 2 + 2 + 2 + 3 + 2 + 12 + 4 + 2 + 3 + 2 + 3 + 3 + 13 + 2 + 2 + 3 + 4 + 3 + 2 + 11) = 0.5$

Normal = $\frac{15}{15+4+2+2+3+2}$ =0.5357

Happy = $\frac{13}{2+ 13+3+2+2+3}$ =0.52

Angry = $\frac{14}{3+3+14+3+2+2}$ =0.5185

Nausea = $\frac{12}{2+3+2+12+4+2}$ =0.48

Sad = $\frac{13}{3+2+3+3+13+2}$ =0.5

Shocked = $\frac{11}{2+3+4+3+2+11}$ =0.44

Conffusion matrix of testing uses 166 times the detection, obtained an average accuracy of 50% correct classification and 50% wrong. Factors that cause errors in classifying using Hidden Markov Model (HMM) because it depends on the characteristics of the image and the amount of training data is used.

3 CONCLUSIONS AND SUGGESTIONS

3.1 Conclusions

Conclusions from the study entitled: "Implementation of Hidden Markov Model Algorithm for Face Recognition Cues" are as follows:

1. Research on image processing facial expressions can be used to identify which type of facial expression facial expressions "happy", "angry", "sick", "sad", "shocked" and a normal face.
2. Method of Hidden Markov Models can be applied in the classification process introduction of this kind of facial expression "Happy", "Angry", "Sick", "sad", "Shocked" and a normal facial image processing is done on facial expression with an average accuracy results 50% correct in recognizing..

3.2 Suggestions

1. Research using images of facial expressions has not been able to detect the specific characteristics of the facial expressions due to the technique of "edge detection preprocessing" detected face is a form of edges, thus color, shape and texture is ignored for image classification process.
2. It is expected that further research is able to recognize facial expressions based on shape, texture and other features characteristic.

BIBLIOGRAPHY

[1] S, P Marhutala dan Soelaiman, M.Kom, R. "Pengenalan Ekspresi Wajah Berdasarkan Bentuk dan Tekstur". Teknik Informatika, ITS.

[2] Feng-Sheng Chen, Chih-Ming Fu, dan Chung-Lin Huang. "*Hand Gesture recognition using a real-time tracking method and Hidden Markov Models*". Institute of Engineering, National Tsing Hua University.

[3] Zaenal Abidin, "Pengembangan Sistem Pengenalan Ekspresi Wajah Menggunakan Jaringan Syaraf Tiruan Backpropagation (Studi Kasus Pada Database MUG).", Matematika, Universitas Negeri Semarang, 2011.

[4] Shreyas Fadnavis "Image Interpolation Techniques in Digital Image Processing: An Overview", International Journal of Engineering Research and Applications, 2014.

[5] Abdul Kadir, dan Adhi Susanto, "Teori dan Aplikasi Pengolahan Citra", Andi, 2013.

[6] Mazid kamal, dan Ruki Suko Basuki., M. Kom, "Segmentasi Citra Daun Tembakau Berbasis Deteksi Tepi Menggunakan Algoritma Canny", Teknik Informatika, Universitas Dian Nuswantoro Semarang.

[7] Chrisopher Danil, "Edge Detection dengan Algoritma Canny", STMIK IBBI.

[8] SameerKais Jameel, "Face Recognition System Using PCA and DCT in HMM", University of Raparin, International Journal of Advanced Research in Computer and Communication Engineering, 2015.

[9] Syafruddin Syarif , Nadjamuddin Harun, Muh. Tola, M. Wihardi Tjaronge, Zulfajri B. Hasanuddin, Zulkarnain Jamid, Rifad Zulfikar Asgar., "Sistem Cerdas Deteksi Citra Dengan Metode Discrete Cosine Transform.", Universitas Hasanuddin, 2012.

[10] Ramen Atonov Purba, "Implementasi Algoritma Principal Components Analysis Dalam Sistem Pengenalan Citra Wajah.", LP3M Medan, 2013

[11] Han, J., Kamber, M. and Pei, J., "Data Mining: Concepts and techniques (3nd ed.), Waltham USA : Morgan Kaufmann Publisher, 2012.

[12] Sepritahara, "Sistem Pengenalan Wajah (*Face Recognition*) Menggunakan Metode Hidden Markov Model (HMM)" ,Teknik Elektro, UI.

[13] S.A. Santosh Kumar, D.R. Deepti, and B. Prabhakar, "*Face Recognition using Pseudo-2D Ergodic Hidden Markov Model*",Central Research Laboratory

[14] P Dymarski. 2011."*Hidden Markov Model, Theory and Aplications"*. India: InTech.

[15] Xiaolin Li, Marc Parizeau, and Rejean Plamondon., "Training Hidden Markov Models With Multiple Observations – A combinatorial Method.", IEEE Computer Society, 2000.

[16] Hadi Santoso, dan Agus Harjoko, "Haar Cascade Classifier dan Algoritma Adaboost Untuk Deteksi Banyak Wajah Dalam Ruang Kelas", Sistem Informasi. STIMIK Atma Luhur.

[17] Marini, Yennie., Emiyati., Hawariyah, Siti., dan Hartuti, Maryani., "Perbandingan Metode Klasifikasi Suvervised Maximum Likelihood Dengan Klasifikasi Berbasis Objek Untuk Inventarisasi Lahan Tambak Di Kabupaten Maros", Pusat Pemanfaatan Penginderaan Jauh, 2014.