International Seminar of Research Month 2021
Volume 2022
http://dx.doi.org/10.11594/nstp.2022.2432

NST PROCEEDINGS

OPEN ACCESS

Conference Paper

Feature Extraction for Face Recognition Using Haar Cascade Classifier

I Gede Susrama Mas Diyasa[1]*, Alfian Hendika Putra[2], Mohammad Rafka Mahendra Ariefwan[1], Primus Akbar Atnanda[1], Fetty Trianggraeni[2], Intan Yuniar Purbasari[2]

[1]Department of Data Science, Faculty of Computer Science, Universitas Pembangunan Nasional "Veteran" Jawa Timur, Surabaya 60294, Indonesia
[2]Department of Informatics, Faculty of Computer Science, Universitas Pembangunan Nasional "Veteran" Jawa Timur, Surabaya 60294, Indonesia

Corresponding author:
E-mail:
igsusrama.if@upnjatim.ac.id

ABSTRACT

This is an era where sophisticated technology can develop rapidly, one of which is technology in the field of computer vision, such as facial recognition systems that can be used in security systems, access control systems, smart cards, and surveillance systems. In developing a face recognition system, the level of recognition accuracy can be influenced by several factors, namely lighting factors, facial expressions, facial positions, and changes in facial attributes. This study uses the Haar Cascade Classifier method in facial extraction and is assisted by using CNN for facial classification. This research uses python programming and the Open CV library.

Keywords: Feature extraction, face recognition, haar cascade classifier

Introduction

With the development of multimedia technology, the use of cameras and image/video processing has increased. One of the uses of cameras and image/video processing is face detection and recognition. Security cameras (Patter et al, 2011), attendance (Wagh et al., 2015), security applications on smartphones (Kremic et al., 2012), and use for online games (Zhan et al., 2008) are some of the applications of facial recognition. Biometric recognition techniques using cameras have various advantages over conventional recognition techniques such as using cards or passwords. Facial recognition cannot be duplicated, stolen, or forgotten.

The face recognition system or Face Recognition (FR) (Arain et al., 2018) identifies faces by matching the face database to the image. There have been significant advances regarding Face Recognition in recent years in learning the design and features of facial recognition models (Sari & Utaminingrum, 2019). The researchers' goal in designing Face recognition is to create a facial recognition system that can match or surpass human recognition. Problems with facial recognition systems are challenging issues. Many issues can affect the accuracy, one of which is the position of the face image. Images by the camera in real-time can capture the part of the face from the front, side, and top so that some components in the face area, namely the eyes, nose, and mouth, are cut off or not seen optimally.

Some researchers use various methods to detect and recognize faces, such as Principal Component Analysis or PCA (Poon et al., 2011) or Scale Invariant Feature Transform (SIFT) (Lowe, 2004). However, these methods require much computation for processing (Wilson & Fernandez, 2006). To speed up the computational process, the haar cascade algorithm is used. As for facial recognition, the Local Binary Pattern (LBP) algorithm is used for facial recognition because it can store important information in the image and work in low lighting sources (Ahonen et al., 2014).

How to cite:
DIyasa et al. (2022). Feature extraction for face recognition using haar cascade classifier. *International Seminar of Research Month 2021*. NST Proceedings. pages 197-206. doi: 10.11594/ nstp.2022.2432

The Haar cascade algorithm combined with LBP has been investigated, but it is still an image, has not used a camera, and is real-time (Tang et al., 2010).

This study uses a webcam and can distinguish facial and non-face objects, as well as people who have been registered in the database in real-time. The haar cascade algorithm and the LBP algorithm will be explained in more detail in the next section. Then proceed with the experimental method in the third part, and in the fourth part, the experiment results are given. In the end, conclusions can be drawn from the experimental results.

Literature Review

Haar feature

Haar Feature is a feature based on the Haar Wavelet. A Haar wavelet is a single square wave (one high and one low). For two dimensions, one is light, and one is dark. Furthermore, combinations of boxes are used for better detection of visual objects (Diyasa et al., 2016). Each Haar-like feature consists of a variety of black and white boxes, as shown in Figure 1, namely three types of box (rectangular) features, including the two-rectangle feature type (horizontal/vertical), the three-rectangle feature type, and the vertical feature type. Four-rectangle feature (Diyasa et al., 2017).

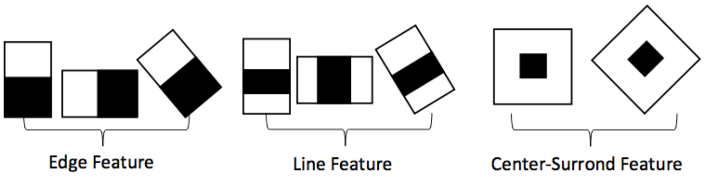


Figure 1. Feature variations on haar

The presence of the Haar feature is determined by subtracting the average pixel in the dark area from the average pixel in the bright spot. If the value of the difference is above the threshold or threshold, it can be said that the feature exists. The value of the Haar-like feature is the difference between the number of gray level pixel values in the black box area and the white box area.

$$F(x) = \text{SumBlack rectangle} - \text{SumWhite rectangle} \quad (1)$$

Where for boxes on Haar-like features can be calculated quickly using an integral image to determine the presence or absence of hundreds of Haar features in an image and at different scales efficiently. In general, such integration means adding small units together. In this case, the small units are pixel values. The integral value for each pixel is the sum of all the pixels from top to bottom. The entire image can be summed with several integer operations per pixel from the top left to the bottom right. The value at the pixel location (x,y) contains the sum of all the pixels in the rectangle area from the top left to the location (x,y) or the shaded area.

$$ii(x, y) = \sum_{x' \leq x, y' \leq y} i(x', y') \quad (2)$$

Information:
ii(x,y) = integral image at location x,y
i(x',y') = pixel value in the original image

Cascade classifier

For the face detection process, the haar cascade algorithm is used. In general, a haar-like feature is used to detect objects in digital images. The term Haar denotes a mathematical function (Haar Wavelet) in a box, and the principle is the same as in the Fourier function. Initially, image processing was only by looking at the RGB value of each pixel, but this method turned out to be ineffective (Rahim, 2013). Viola and Jones then developed it to form the Haar-Like feature. Haar-like feature processes images in boxes, wherein in one box, there are several pixels. Each box is then processed and produces different values that indicate dark and light areas. These values will be used as the basis for image processing (Putra et al., 2012).

The way to calculate the value of this feature is by subtracting the value of the pixels in the white area by the pixels in the black area. To simplify calculating feature values, the Haar algorithm uses a medium in the form of an Integral Image. An integral image is an image whose value for each pixel is the sum of the pixel values from the top left to the bottom right. For example, pixels (a,b) have an accumulative value for all pixels (x, y). Where x a and y b. In using the haar cascade method, several images can be processed, one of which is grayscale. Cascade Classifier is a step to get more accurate results by calculating the Haar Feature value multiple times and repeatedly. Figure 2 shows the workflow of the Cascade Classifier. In stage 1 classification, each sub-image will be classified with one feature, and if the results do not meet the criteria, the results are rejected. In stage 2 classification, each sub-image will be reclassified. If the desired threshold value is obtained, proceed to the next filter stage (classification stage 3). Until the sub-image that passes will decrease until it is close to the image in the sample.

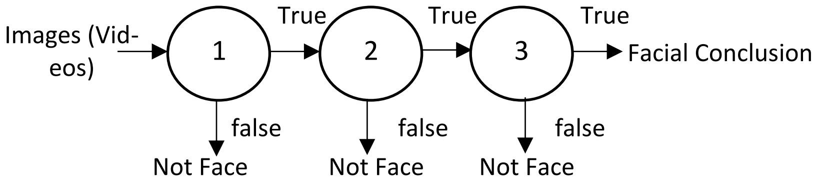


Figure 2. Cascade classifier method flow

Local binary pattern histogram

Face recognition is a continuation of the face detection process. In face detection, which detects a person's face, the face can be obtained from images or videos by utilizing the training results from the haar cascade (Santoso & Ariyanto, 2018). Then the results of this process are combined with the Image Matching process with the Local Binary Pattern Histogram algorithm. With this method, the photos that have been studied will be matched with the detection results from the streaming camera, wherein the streaming later several images in the database are then matched by utilizing the histogram values that have been extracted from the images by using the Local Binary Pattern Histogram equation.

The main characteristic of face recognition using this method is the composition of the micro-texture pattern, which is a nonparametric operator that describes the local spatial layout of the image. LBPH is defined as the ratio of the binary value of the pixel at the center of the image to the 8-pixel values around it. For example, in a 3x3 image, the binary value at the center of the image is compared to the surrounding values. By subtracting the pixel value at the center of the image with the pixel values around it, if the result is more or equal to 0, it is given a value of 1, and if the result is less than 0, then it is given a value of 0. After that, eight binary values are arranged clockwise or vice versa and changed eight binary bits into a decimal value to replace the pixel value at the center of the image (Wahyudi et al., 2018).

$$\textit{Threshold LBPH} = \text{surround value center pixel value} \qquad (3)$$

After compiling the binary clockwise, if one of the binary threshold boxes is 1, then enter the binary value according to its rank, but if it is 0, then the result is also 0. Finally, add the LBP value. To match the owner's face, an equation is used to get an approximate histogram value which will be used as a predictive value to identify the owner of the face (Equation 4). The following is the equation to find the histogram value approach, with the value of D is used to compare the faces contained in the database and the faces detected by the camera.

$$D = \sqrt{\sum_{i=1}^{n} (hist\ 1_i - hist2_i)^2} \quad (4)$$

Material and Methods

This research methodology contains how the steps used in this research can be understood and structured correctly. The research flow used in the face recognition system using the Haar Cascade Classifier method uses the waterfall model (Kusyadi, 2018). The waterfall model was chosen because it went through sequential planning steps from top to bottom in making the system so that anyone quickly understood it. The following are the steps in research, among others, Literature studies, namely at this stage, it is done by reading articles on the internet, journals, and books as references for designing and what needs to be prepared. The literature study obtained is the primary reference source in the process of working on this research.

Then Needs Analysis Collect training data needs and training data by asking volunteers to send videos or photos containing their faces. In addition, the research uses a laptop with Intel Core-i5 specifications, 8 Gb RAM, NVIDIA GEFORCE 930M, 0.3 Megapixel Webcam Camera.

In the system design, a face recognition system uses the Convolutional Neural Network with the VGG16 architecture. System Design at this stage is the stage where the system is built with Python programming. This step is the implementation of the results of the entire system design in the form of program code scripts that the system can read. The VGG16 architecture was applied to train the recognition of 11,000 photos of training data in 11 classes of people for system training. And finally, testing was carried out on 51 pictures of test data to see how the system's accuracy level was at 100, 200, 300, 400, 500, 600, 700, 800, 900, and 1000 training data. While the method used is as shown in Figure 3.

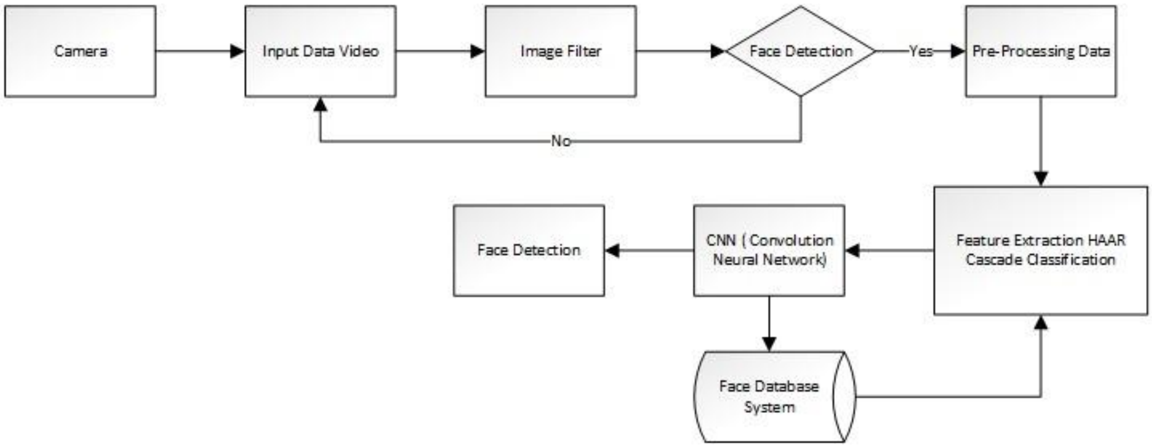


Figure 3. Method for Face detection

Pre-process data

Preprocessing is the initial stage carried out before starting the classification process. The first stage in the preprocessing will convert the training data in the form of video into an image frame by frame (Diyasa et al., 2016). After becoming an image, the following process is to change the RGB image into a gray image. The last step is to resize the image to an asymmetrical size of 224 x 224 pixels. The higher the size of the image, the harder it will be to carry out the classification process so that it also increases the time of recognition, but the information data obtained in the image is getting bigger and can affect the accuracy received. Figure 4 is a flowchart of the preprocessing.

Figure 4. The flow of the Data Pre-processing method

Face detection

Detection Face is the main object that this robot system will detect. The initial process carried out by the system is to capture images from the camera. Then the image will be converted from an RGB image into a grayscale image. It is intended that the image can be processed using the Haar Cascade method, where this method will detect faces. Once a face is detected, the system will mark the face with a green box. This is done to distinguish objects that do not face. For this experiment, six Indonesians, four men and two women aged between 18-30 years, were selected as experimental subjects. The VGG16 architecture was applied to train the recognition of 11,000 photos of training data in 11 classes of people for system training. And finally, testing was carried out on 51 pictures of test data.

Object detection using the haar cascade classifiers feature is a practical object detection method. It is a machine learning-based approach where the cascade function is trained from many positive and negative images (Hossen et al., 2017).

Haar-like feature or commonly called Haar Cascade Classifier, is a rectangular (square) feature. The process of recognizing objects in the Haar Like Feature is based on a simple value of a feature in an image. This method only depends on the number of pixels in a square or feature and not the value of each pixel of an image, so this method has the advantage of a speedy computational process.

The presence of the Haar feature is determined by subtracting the average pixel in the dark area from the average pixel in the bright spot. If the value of the difference is above the threshold or threshold, it can be said that the feature exists (Diyasa, 2017). Figure 5. is the process flow of the haar cascade classifier method in face detection.

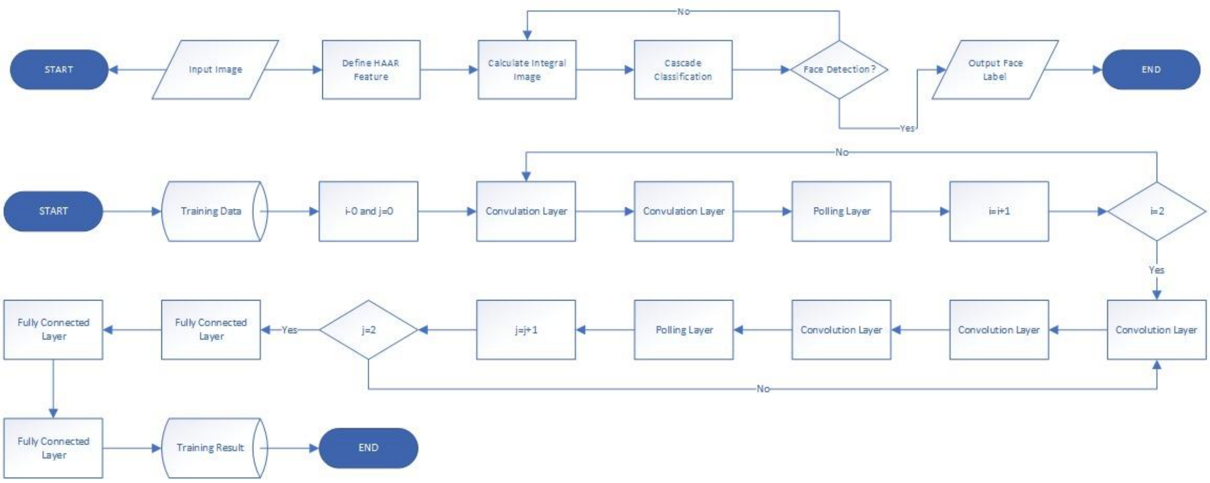


Figure 5. Haar cascade classifier method for face detection

Face database

Before the system can recognize a face, the face to be recognized is first stored in the database to find out the LBPH (Local Binary Pattern Histogram) value or the histogram value of an image. To recognize faces properly, the system requires at least 51 image input from the camera.

Each subject is taken data with various poses and angles as much as 100, 200, 300, 400, 500, 600, 700, 800, 900, and 1000 times. The data will be saved as an image database. Database creation is needed to get facial data to be recognized. The samples are then put in a folder, with each person having their unique number. The number is used to identify the facial model of the person. The sample image is then converted into a grayscale image before being stored in the database.

After that, the training process is carried out. In this process, the histogram value will be extracted from the database image. The value is stored in the form of a data array and stored along with the identification number of each subject.

Face recognition

Then in the face detection process, the camera will get an input of the detected face image, and the histogram value will be known. So the system can compare the histogram value of the input image from the camera with the histogram value of the face image in the database. The face will be recognized as the face of the owner of the histogram value closest to the histogram value in the database.

We are applying the LBPH algorithm to the detection system in real-time. In this step, the LBPH algorithm that has been trained will produce a histogram value which will then be compared with the value that will be detected directly by the camera (real-time). To get the match between the images and the values stored in the database, it is necessary to compare two histograms between the detected images and the images in the database and find the distance to the closest histogram values. To solve this, we use (1) So, the algorithm's output is the identification number of each image which is changed to the name of the owner of the face. Then, in the end, it is displayed on the monitor. For this experiment, each subject is within range of the camera, with the number of issues varying.

Convolution layers are a layer that will calculate the output of neurons connected to the local region in the input image, and each neuron performs dot product multiplication between the small areas connected to the input image with a shift filter. In the convolution process, two variables can determine the size of the output matrix, namely padding (Dwarampudi & Reddy, 2019) and stride (Suartika et al., 2016). Padding is a pixel with a value of 0 outside the input matrix, while stride is a value that determines how many kernel pixels are shifted. At the face recognition stage using the Convolution Neural Network (CNN), an array-shaped image will be processed at the layer on CNN. There are several stages in the CNN training process, including the Convolution Layer, Pooling Layer, and Fully Connected processes. Figure 6. It is a flow of steps to determine the identification of face names.

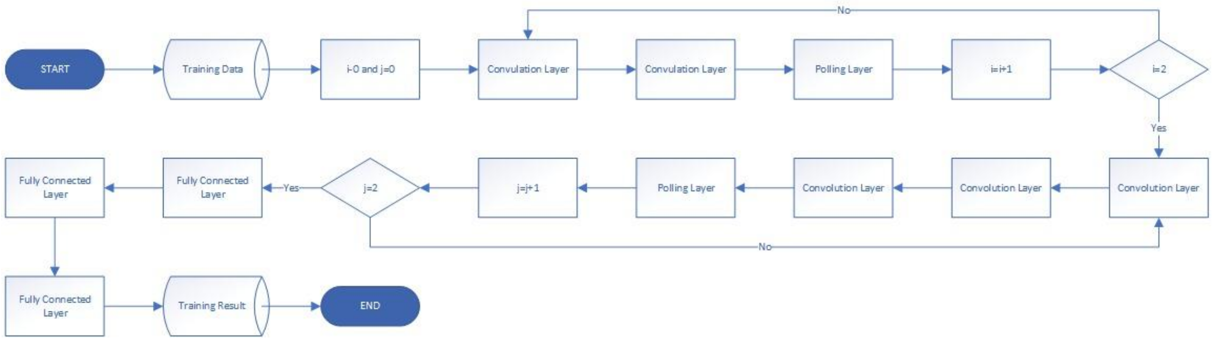


Figure 6. Face recognition flow

Pooling layer

The pooling layer (Nagi et al., 2011) is the layer that will perform the downsampling operation on the spatial dimensions (width, height). Pool Layer is a layer that shrinks the dimensions of the previous layer. This layer is a 2-dimensional kernel with a size of mxn. Usually, the kernel used is 2 x 2, so that the size of the image dimensions will be reduced by half. The process that occurs in the max-pooling layer is to return the input size. The trick is to find the location of the maximum value in the forward process and then return the length. The design of this pooling layer has 2 x 2 filters, and the displacement value is 2. This form will reduce the feature map by 50% of its original size. Figure 7. It is an illustration of the max-pooling process.

Flattening

Flattening (Jin et al., 2015) is an operation that converts a matrix into a one-dimensional vector. The flattening process converts the feature map that has been obtained from the previous layer into a one-dimensional vector so that the feature map can be classified with fully connected layers and softmax. The flattening process is illustrated in Figure 8. At the backpropagation stage, this layer converts the one-dimensional vector back into a matrix with the original dimensions so that the filter weight change process can then be carried out.

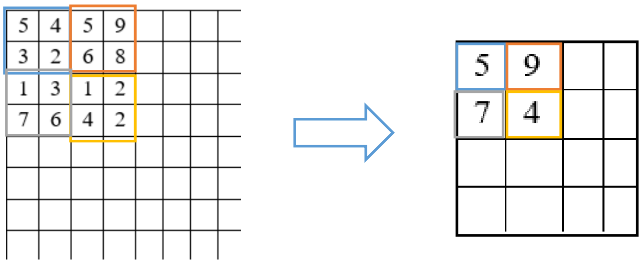


Figure 7. Max pooling process

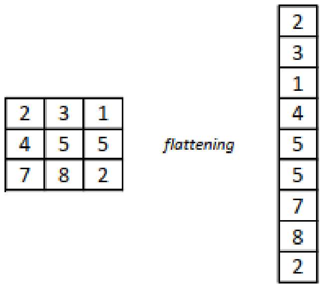


Figure 8. Flattening process

Fully connected layer

This layer is composed of several layers, and each layer is formed of several neurons which are all fully connected. The process in Fully Connected Layer (Basha et al., 2019) functions to carry out the classification process following Figure 9. In which Layer 1 will be fed forwarding to layer 2 using the ReLU activation function. Layer 2 will be classified using softmax.

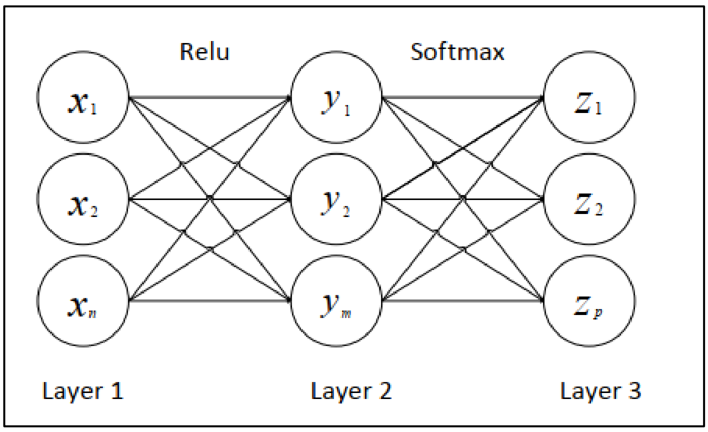


Figure 9. Fully connected layer process

Activation function

The activation function is a non-linear function that allows an ANN to transform input data into a higher dimension so that simple hyperplane cuts can enable classification. There are several activation functions used in this study, namely

1. Activation of the Rectified Linear Unit, commonly called ReLU, is an activation function that eliminates negative values. Forward and backward processes through ReLU only use if conditions. If the element is negative, the value is set to 0, with no exponential, multiplication, or division operations. With such characteristics, the ReLU activation function has the advantage of reducing training and testing time significantly when dealing with networks with many neurons.
2. Activation of Tanh, Hyperbolic tangent function or what is often called TANH is generally faster to reach convergence than sigmoid and logistic activation functions and can produce higher accuracy. The performance offered by the TANH activation function is almost the same as the classification performance generated by the RELU activation function.
3. Softmax activation (Dewaa & Afiahayati, 2018) is an algorithm model that can be used to classify more than two classes. The softmax activation works from the stored class label; a vector value with absolute value will be taken and converted into a vector with a value between zero and one, which will be worth one when added up.

Results and Discussion

Photo trial

The trials were conducted on six photo test data in Alfian, Bagus, Fadil, Hamzah, Lilik, Nadiah, Panji, Thomas, Winda, Zdikri. The training data carried out also varies from 100 to 1000 photo data. Table 1 is the result of the total correct prediction value on the system.

Table 1. Photo test results

Name	100	200	300	400	500	600	700	800	900	1000
Alfian	5	3	5	6	6	6	6	6	6	5
Good	0	0	2	1	2	2	5	4	3	3
Eminent	3	5	5	6	5	5	5	5	4	5
Hamza	5	5	6	6	5	5	6	6	5	6
Like	4	3	5	6	5	5	5	6	6	5
Nadia	5	5	6	6	6	6	5	6	6	6
Banner	0	0	3	3	4	3	2	5	5	5
Thomas	2	5	6	6	6	6	6	6	6	6
Winda	2	3	3	2	2	5	5	2	5	5
Zdikri	3	3	5	5	4	3	3	4	4	6
Total	29	32	46	47	45	46	48	50	50	52
% accuracy	48	53	76	78	75	76	80	83	83	86

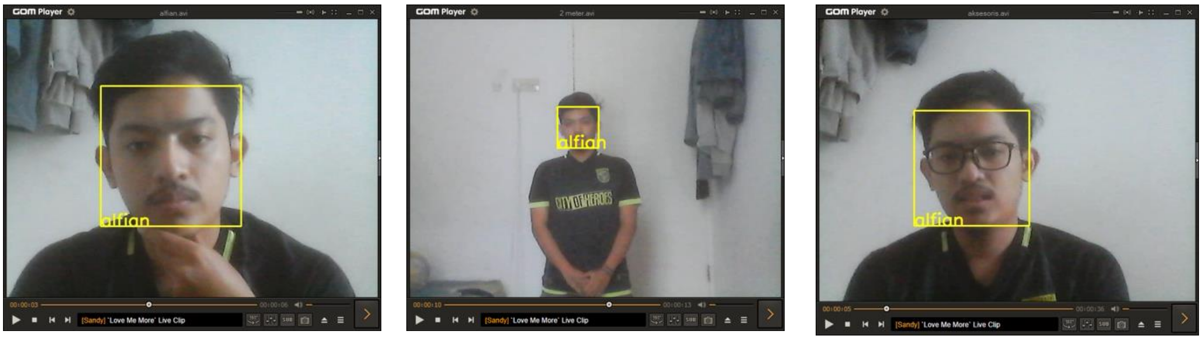
Table 1 is an accuracy table with the sum of TP and TN in each class of people from the training process using 100 training data to 1000 training data. If calculated using equation 1, the total accuracy value for all test data is 29+32+46+47+45+46+48+50+50+52=445, then divided by the total FN+FP+TN+TP, which is 600. So 445 /600*100=74%.

$$Accuracy = ((TP + TN))/(TP + FP + FN + TN) \quad (5)$$

Information:
TP=The result of system prediction is a picture an in the picture a,
TN=The result of system prediction is image b on image b,
FP=The result of system prediction is a picture an on picture b,
FN=The result of system prediction is picture b on the picture a.

Real-time trial

Although the system can recognize the faces of more than one person, in the trial model, the real-time facial recognition system is only carried out with one person. When the system is running online, the results from the webcam will automatically record all processed images, which will later be saved as video in AVI format. The following is the result of implementing the system in real-time.



(a) Real-time Trial Results (b) Test Results 2 Meter Distance (c) Test Results Using Glasses

Figure 10. Real-time test with 2 meters distance and using glasses

Figure 10 (b) is the result of a real-time trial. This study also tested how when a person is about 2 meters from the camera. The following figure 6 is the result of the experiment. In addition, the author also tested how the object was wearing accessories such as glasses and hats. The following Figure 10 (c) is an example of the results of object testing images using accessories.

Conclusion

Based on the research conducted, from needs analysis, system design, system design, to training and testing. So it can be drawn some conclusions that The architectural form and parameters of the Convolutional Neural Network with a depth of 16 convolution layers used in this study are pretty appropriate because they can extract and classify features quite well. This can be seen from the pretty good accuracy results, the results of the accuracy research using the Convolution Neural Network architecture VGG16 is 74% in the testing process of 600 data, the image processing technique for facial recognition used is entirely appropriate because it can recognize someone even though he is wearing accessories such as sunglasses and hat. Using this CNN architecture, the results will be maximized if the training data used has bright lighting so that the system can capture more facial features for the training process.

The author can give several suggestions for further research, namely: Looking for more varied forms of training data with various facial expressions and facial positions so that more details for training data on facial features, Looking for architecture or other parameters so that the program can run faster like an architecture-google net which was the best architecture in 2014 during the ImageNet Large Scale Visual Recognition Competition (ILSVRC), and use a laptop device with higher specifications than Intel Core-i5 and 8 Gb RAM so that when processing data faster, in this study a 0.3-megapixel webcam camera is used so that the image quality obtained is less than optimal. The author suggests using a camera with a resolution of 5 megapixels to get maximum results, look for plug-ins that support GPU-based program runtimes to run any program faster.

Acknowledgment

Thank you to LPPM UPN "Veteran" Jawa Timur for the research that has been given so that we can publish papers in several journals.

References

Ahonen, T., Pietikäinen, M., Hadid, A., & Mäenpaä, T. (2004). *Face recognition based on the appearance of local regions*. Machine Vision Group, InfoTech. University of Oulu, Finland.

Arain, R. H., Maitlo, A., Ruk, S. A., & Shaikh, H. (2018). Study of face recognition techniques: A survey. (IJACSA) International Journal of Advanced Computer Science and Applications, 9(2), 42-49. Doi: 10.14569/IJACSA.2018.090606

Basha, S. H., Dubey, S. R., Pulabaigari, V., & Mukherjee, S. (2019). *Impact of fully connected layers on performance of convolutional neural networks for image classification.* Computer Science https://www.researchgate.net/publication/330955396.

Dewaa, C. K., & Afiahayati. (2018). Suitable CNN weight initialization and activation function for javanese vowels classification. *Procedia Computer Science, 144*(2018), 124-132.

Diyasa, S. M., Purnama, K. E., & Purnomo, M. H. (2016). Automated analysis of human sperm number and concentration (oligospermia) using the Otsu threshold method and labeling. *IOP Conference Series: Materials Science and Engineering, 105*, 012038.

Diyasa, S. M., Setiawan, A., & Khalis, M. (2017). Design and build an application "W-Mass (Weight Monitor Assistant)" Based on android studio with native java language. *Research Journal of Poltekbang, 4*(2), 1-9.

Dwarampudi, M., & Reddy, N. V. S, (2019). Effects of padding on LSTMS and CNNS. *Computer Science.*

Hossen, A. M. A., Ogla, E. A. A., & Ali, M. M. (2017). Face detection by using openCV's Viola-Jones Algorithm based on coding eyes. *Iraqi Journal of Science, 58*(2A), 735–745.

ISuartika E. P. I. W., Wijaya, A. Y., & Soelaiman, R. (2016). Image classification using Convolutional Neural Network (CNN) at Caltech 101. *Journal Of Its Engineering, 5*(1), A65-A69.

Jin, J., Dundar, A., & Culurciello, E. (2015). Flattened convolutional neural networks for feedforward acceleration. *International Conference on Learning Representations (ICLR) 2015*. https://www.researchgate.net/publication/269722508.

Kremic, E, Subasi, A., & Hajdarevic, K. (2012). Face recognition implementation for client server mobile application using PCA. *Proceedings of the ITI 2012 34th International Conference on Information Technology Interfaces*, 435-440. doi: 10.2498/iti.2012.0455

Kusyadi, I. (2018). *Penerapapan waterfall method for information systems BCF 1.5.* ResearchGate https://www.researchgate.net/publication/323118868

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journalof Computer Vision, 60*, 91-110.

Nagi, J., Ducatelle, F., Caro, G. A. D., Ciresan, D., Meier, U., Giusti, A., Nagi, F., Schmidhuber, J., & Gambardella, L. M. (2011). Max-pooling convolutional neural networks for vision-based hand gesture recognition. *IEEE International Conference on Signal and Image Processing Applications (ICSIPA2011).*

Peter, K. J., Glory, G. G. S., Arguman, S., G. Nagarajan, Devi, V. V. S., & Kannan, K. S. (2011). Improving ATM security via face recognition. *2011 3rd International Conference on Electronics Computer Technology*, Kanyakumari, 373-376.

Poon, B., Amin, M. A., & Yan, H. (2011). Performance evaluation and comparison of PCA based human face recognition methods for distorted images, *International Journal of Machine Learning and Cybernetics, 2*(4), 245–259.

Putro, M. S., Adji, T. B., & Winduratna, B. (2012). Sistem deteksi wajah dengan menggunakan metode viola-jones. *Seminar Nasional "Science, Engineering and Technology"*, Yogyakarta.

Rahim, A. (2013). Face recognition using Local Binary Patterns (LBP). *Global Journal of Computer Science and Technology Graphics & Vision, Pabna University of Science and Technology Bangladesh, 13*(4), 1-9.

Santoso, A., & Ariyanto, G. (2018). Implementation of hard-based deep learning for face recognition. *Emitter Journal. 18*(01), 1-5.

Santoso, H., & Harjoko, A. (2012). HAAR cascade classifier dan algoritma adaboost untuk deteksi banyak wajah dalam ruang kelas. *Jurnal Teknologi, 6*(2), 108–115.

Sari, Y. A., & Utaminingrum, F. (2019). Face recognition with unique pose using learning method vector quantization. *Journal of Information Technology and Computer Science Development, 3*(1), 8473-8480.

Tang, H., Sun, Y., Yin, B., & Ge, Y. (2010). Face recognition based on Haar LBP histogram. *2010 3rd International Conference on Advanced Computer Theory and Engineering (ICACTE)*, Chengdu, V6-235-V6-238.

Wagh, P., Thakare, R., Chaudhari, J., & Patil, S. (2015). Attendance sistem based on face recognition using eigen face and PCA algorithms. *2015 International Conference on Green Computing and Internet of Things (ICGCIoT)*, *2015*, 303-308

Wahyudi, E., Wirawan, & Kusuma, H. (2018). *Teknik pengenalan wajah berbasis fitur Local Binary Pattern (LBP).* Eepis final project, Jurusan Teknik Elektro- FTI, Institut Teknologi Sepuluh Nopember.

Wilson, P. I., & Fernandez, J. (2006). Facial feature detection using Haar classifiers. *J. Comput. Sci. Coll., 21*(4), 127-133

Zhan, C., Li, W., Ogunbona, P., & Safaei, F. (2008). A real-time facial expression recognition sistem for online games. *International Journal of Computer Games Technology, 10*, 1-7.