

PAPER • OPEN ACCESS

Multi-face Recognition for the Detection of Prisoners in Jail using a Modified Cascade Classifier and CNN

To cite this article: G S M Diyasa *et al* 2021 *J. Phys.: Conf. Ser.* **1844** 012005

View the [article online](#) for updates and enhancements.

You may also like

- [Gate Bolt](#)

M. Senthamil Selvi, S. Jansi Rani, A. Nisha et al.

- [Implementation of facial mask detection and verification of vaccination certificate using Jetson Xavier kit](#)

Ragavesh Dhandapani and Sara Marhoon Humaid Al-Ghafri

- [Design of printed circuit board detection with image information technology](#)

Tibyani, N Hayati, I Sutrisno et al.



The Electrochemical Society
Advancing solid state & electrochemical science & technology

243rd ECS Meeting with SOFC-XVIII

Boston, MA • May 28 – June 2, 2023

**Abstract Submission Extended
Deadline: December 16**

[Learn more and submit!](#)

Multi-face Recognition for the Detection of Prisoners in Jail using a Modified Cascade Classifier and CNN

I G S M Diyasa¹, A Fauzi^{2,*}, M Idhom³ and A Setiawan⁴

^{1,2,3} Department of Data Science, Universitas Pembangunan Nasional “Veteran” Jawa Timur, Raya Rungkut Madya Gunung Anyar, Surabaya, East Java, Indonesia

² Department of Air Transportation Management, Aviation polytechnic of Surabaya, Indonesia.

Email: igsusrama@upnjatim.ac.id¹

Abstract. Identifying the identity of a prisoner in a detention cell, through facial recognition automatically is a big, exciting problem and there are many different approaches to solve this problem because it must detect multiple faces (multi-face). Especially in uncontrolled real-life scenarios, faces will be seen from various sides and not always facing forward, which makes classification problems more difficult to solve. In this research, one method is combined deep neural networks that is Convolutional Neural Networks (CNN) and Haar Cascade Classifier as real-time facial recognition, which has proven to be very efficient in face classification. Methods are implemented with assistance library Open-CV for multi-face detection and 5MP CCTV camera devices. In preparing the architectural model Convolutional Neural Networks Do configuration parameter initialization to speed up the network training process. Test results on 51 test data using constructs Convolutional Neural Networks VGG16 models up to a depth of 16 layers of convolution layers with input from the extraction of the Haar Cascade Classifier resulting in facial recognition system performance reaching an accuracy rate of about 87%.

1. Introduction

The growth in the number of detainees requires reconsideration of staff numbers and distribution, in large part requiring an increase in the level of staffing for supervising and managing prisoners [1]. However, staff resources usually do not match the number of detainees. As a result, the staff per prisoner ratio has decreased.

To overcome this, one solution is to need a monitoring system or surveillance of prisoners using CCTV cameras online or in real-time [2]. The purpose of this CCTV camera application is to monitor the presence of detainees in detention cells. Still, because the number of prisoners in detention cells exceeds the quota/detention cell limit, this will make it difficult for the supervisory staff to check the detainees because they have to mark the names of the detainees simultaneously [3]. Therefore this system is considered inefficient so that a facial recognition system of detainees is needed as a presence or a statement that the prisoner is still in a detention cell.

Facial recognition is a challenging problem [4]. One of the factors is due to the various positions of the face image. The camera can capture the part of the face from the front, side, or a certain angle,

causing some facial features such as the eyes or nose to become completely invisible. Another factor is the presence or absence of structural components such as beards, moustaches, or with/without glasses in the facial image. The structural components have a lot of variability's, including shape, colour and size [5]. Other factors that can affect accuracy include lighting,

occlusion and facial expressions. Illumination is a change in light distribution due to the reflectance properties of the skin and the built-in camera controls that cast shadows on certain parts of the face. Occlusion is the result of an object covering aspects of the face, such as a scarf, turban, and others; examples of facial expressions that are smiling, laughing, angry, sad, surprised, and afraid [6].

Several studies related to facial recognition using a Deep Neural Network include research conducted by [7], namely conducted research on facial recognition systems using the Deep Neural Network and Haar Cascade methods to extract facial features to help reduce the complexity of the neural network. The use of DNN makes the process lighter and faster and also has good accuracy, but in this study, what was detected was an image and only a mono face. Another research is about Real-Time Multiple Face Recognition with Deep Learning on the Embedded GPU System [8]. This research consists of tracking techniques and using minimal model weights, which can reduce

processing time and network parameters to learn to recognize multiple facial features in real-time; however, this study does not explain the computation time required to recognize faces. Other research on Face Pose Recognition with Equivalent Mapping [9]. In this study, it was explained that the Deep Residual Equivariant Mapping (DREAM) block could improve facial recognition features and is easy to implement. The results on the dataset are CFP, IJB-A, and MS-Celeb-1M that show the application of the block to various CNN types, including ResNet-18, ResNet- 50, and Center-Loss models. The advantage of this research is that the proposed partnership is not limited to facial recognition with a variety of poses, but can be used for face recognition across ages. Still, this study does not do multiple facial recognition. Research on attendance systems based on facial recognition in the classroom [10]. In this study, an automatic attendance management system with facial recognition was added to attend class attendance online. Researchers use the concept of facial recognition to implement a system that marks the presence of a particular person and recognizes faces and is made portable for more comfortable use. This system functions to observe various facial expressions, lighting and poses of the person.

From the above studies that face Detection and recognition are carried out on one person. They are image-based; this paper discusses multi-detection for face recognition by combining the Haar Cascade method to perform face detection, and one of the Deep Neural Network methods, namely Convolution Neural Network (CNN), based on the VGG16 and Classifier [11], for process facial recognition. This system is made using Python, and tested in real-time using a CCTV camera converted to MP4 format, the age range of object detection (prisoners) is between the ages of 18 to 24 years, the maximum number of objects is five people, the item is facing the camera, and The maximum distance between the object and the camera is 4 meters. This multi-face detection and recognition process begins with collecting data sets (image acquisition), pre-processing, image extraction, classification and identification of face recognition of prisoners, whose final purpose is to monitor the presence of prisoners in the detention cell.

2. Method

In conducting this research, the first step is to collect a dataset of video sets and facial images, which are divided, into two types of groups. The first set of videos and pictures is a set of face images without lights (based on sunlight entering the room) or videos/images of faces in low light conditions. In contrast, the second set of videos/pictures is a face with lights or a face image in bright lighting conditions. Then the architectural design is carried out starting from determining the network depth, layer arrangement, and selecting the type of layer that will be used to get the model based on the input dataset and name label index. Following are the steps in carrying out multiple face detection and recognition, as shown in Figure 1.

At the training stage, facial training videos were taken according to the number of prisoners available.

Then the image is processed-processing, namely conversion to image grey-scale. The next step, Detection and segmentation of faces from video/training images and video/test images. The Detection is done with the haar-cascade classifier method make use of the library OpenCV and the python programming language, to get feature extraction from the face image to be segmented against the background, each face will be saved in a database which is a value, the next step in the face identification process is the face image in the training data database look for the smallest or closest similarity value. The method of calculating the similarity value uses CNN; if the face has been identified. An internal check will be carried out the database, and change the status of the detainee's presence to 1, which means that the person is in the detention room/cell during the surveillance process.

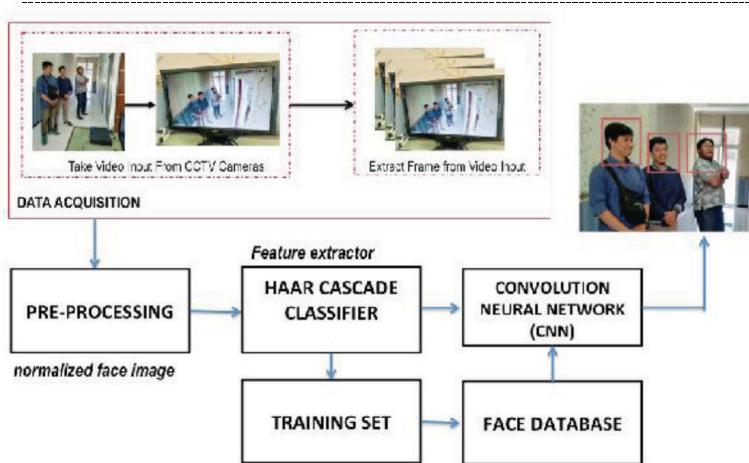


Figure 1. Method for Multi detection and face recognition of prisoners.

2.1. Data Acquisition

The input data in this system is in the form of video in MP4 or AVI format taken directly from a room using a CCTV camera with an average video quality of 30 fps, the footage is grouped into two, namely video with lights and without lights, from video data, these are converted into images per frame. This data will later be used as training data and also testing data for data testing in this study using images in JPG, JPEG, or PNG formats, while videos are in MP4 or AVI formats (Masdiyasa, Purnama and Purnomo, 2018).

The more data used, the more training programs will be so that it has a better level of accuracy. These data are in the form of videos/images of the faces of the prisoners, which were taken several times. The data is collected in a folder called dataset. Each dataset is divided into several folders according to the name of the prisoner. The video/image data were taken as a sample of eleven (11) prisoners, then grouped into 11 classes of prisoners. The amount of data in each class ranges from 100 image data to 1000 image data. The image in this dataset has an image size in each type of 224 x 224 pixels. The picture is taken from various aspects, one of which is the aspect of the category of facial direction and facial expression. The face direction category is looking up, down, right and left. Elements of the image category are also from facial expressions such as smile, frown, glare, squint, and so on. The dataset is separated from faces that wear glasses and those who don't, so Face recognition can capture and validate looks when wearing glasses and not. The training dataset will be used to conduct Training.

2.2. Pre-processing

The pre-processing process is the process of processing the original data or image before the Convolutional Neural Network (CNN) algorithm processes the data or image [12]. Pre-processing has

several ways, namely changing a colour image (RGB) to gray-scale, image binarization, image cropping [13]. Pre-processing has several objectives, including reducing or eliminating noise, clarifying data or image features, enlarging or reducing the size of the data or image converting the original data or image (picture) into what is needed.

Then taking the unique characteristics to determine an idea carries out the process of determining the features of the image. The feature extraction process is a step of separating or solving a vision based on the features of the image. Feature extraction produces a variety of characteristic values from each individual. The type of feature values also depends on the number of partitions. The greater the number of sections, the more numerous and varied the characteristic values will be obtained. In the Face Recognition process, a face is required to be taken by the camera so that it can be processed and determined as someone's identification. Generally, looks have a different shape from other body organs, so there are benchmarks for several organs that can determine that the image or image taken is a face. Some facial features have eyes, nose, and mouth. The purpose of feature extraction includes: reducing the amount of data or images, separating images according to their characteristics, retrieving the most crucial information from processed data or pictures, making face recognition easier to identify data or images

Cropping performed on the detected object so that only the detected object remains. This is necessary for the items to be used for feature extraction and classification processes. In this process, the image will be cropped with a size of 224 x 224 pixels. This process is done with the help of the Open-CV library. The cropping process will help to get a complete object area like in the image.

2.3. Face Detection

Object detection using the haar cascade classifiers feature is a useful object detection method [14]. It is a machine learning-based approach where the cascade functions are trained from multiple positive and negative images. Each part is a single value obtained by subtracting the number of pixels under the white box from the number of pixels under the black box.

The program will search automatically to determine haar features according to the lines and edges of the dark and light images. There are several kinds of haar features, namely edge features, line features, and centre features [15]. After determining the haar feature in a particular image area, the image integral will be calculated in that pixel area, and the image essential calculation process can be seen in equation one about the Haar Cascade Classifier [16].

$$s(x, y) = i(x, y) + s(x, y) + s(x - 1, y) - s(x - 1, y - 1) \quad (1)$$

Information :

$s(x, y)$ = is the sum of the values for each pixel

$i(x, y)$ = is the intensity value obtained from the pixel value of the input image

$s(x - 1, y)$ = is the pixel value on the x-axis

$s(x, y - 1)$ = is the pixel value on the y-axis

$s(x - 1, y - 1)$ = is the diagonal pixel value

In one image, the image will have a lot of haar features in it so that there will also be a lot of image integral calculations in an image. After that, the image will be processed by cascade classification. *Cascade classifier* performs the process of many features by organizing in a stratified classification form. There are three classifications to determine whether or not there are facial features on the features that have been selected.

1. In the first filter, a classifier feature was selected with a detection rate of 100% and an error rate of about 50%.

2. In the second filter, five classifier features were selected with a detection rate of 100% and an error rate of 40% (20% cumulative).
3. In the third filter, 20 classifier features were selected with a detection rate of 100% with an error rate of 10% (2% cumulative).

When the multilevel classification process is carried out, then, on *image* It will be marked with a rectangle in the detected face area, and if no face is seen, the image will not be drawn by a rectangle.

2.4. CNN Design

At the facial recognition stage using the Convolution Neural Network (CNN) [17], an array-shaped image will be processed on the existing layer on CNN to determine the accuracy of the method in identifying facial recognition in a person. In the CNN training process, there are several stages, including the processes in the Convolution Layer, Pooling Layer, and Fully Connected [18]. The CNN structure of the real-time facial recognition program is, as shown in Figure 2.

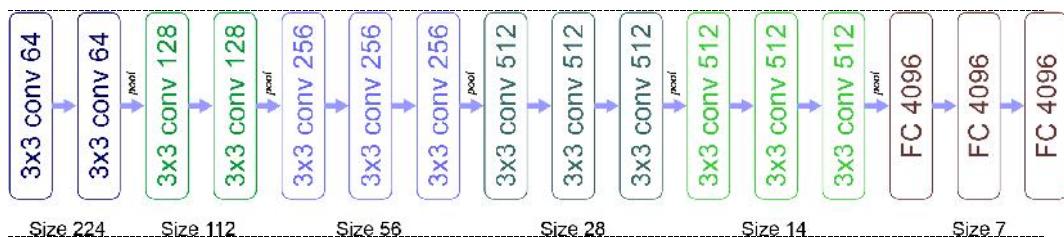


Figure 2. CNN structure

In Figure 2, it is explained that the structure used in this study has an input image at the beginning of the process with a size of 224 x 224 pixels. Then the input in the form of a vector will be processed in the first convolutional layer with a length of 3 x 3 with several filters of 64. Then it is reprocessed in the second convolution layer with 64 filters of 3 x 3. Then the subsampling of the first layer is 2 x 2 in size with the number of strides/displacement 2. When the image is subsampled or max pooled, the image size is halved to 112 x 112 pixels. Then it is processed in the third convolution layer with 128 3 x 3 size filters. Then the fourth layer with a size of 3 x 3 with 128 filters will be entered. The image will return to the second max poll so that the image size becomes 56 x 56 pixels. Then the image is processed again in the fifth convolution layer with 256 filters of 3 x 3. After that, the idea is processed into the sixth and seventh convolution layers with 256 filters and a size of 3 x 3. Then it will be max polled in the third so that the image size becomes 28 x 28 pixels. Then the convolution image is processed again by entering the eighth, ninth, and tenth convolution layers respectively with 512 filters of 3 x 3. After that, the max pool process will join the 2 x 2 size with stride two so that the image this time will be 14 in length. Will enter convolution layers eleven, twelve, and thirteen respectively with 512 filters of 3 x 3. Then the last subsampling layer will join with a size of 2 x 2 so that the image size becomes 7. After that, the first fully connected layer will enter with a 7 x 7 size filter 4096, the second one with a 4096 filter size 1 x 1, and the third one with a 4096 filter size 1 x 1 sequentially. Then there will be three hidden layers, each measuring 100, 10, and according to the number of classes, namely 11. So in this CNN process, it has a total of 19 layers, namely 13 convolutional layers, three fully connected layers, and three hidden layers.

2.5. Trial Scenarios

The trial scenario is carried out to get results following the design, but before heading to the system test scenario stage, first, perform a functional test of the components that have been prepared. After that, the dataset that has been collected will be carried out a training trial to get the face recognition model correctly according to the entered dataset. At the model training stage, the input is required in the form of a dataset and label name. The following is a test scenario:

- a. Conduct training with an image size of 224x224 pixels,
- b. Train training data using optimal weights that have been tried before with around 11000 training data,
- c. Performed trials on the 51 data described in Table 1.
- d. Carry out the testing process with CNN architectures as in Table 2.
- e. Welly's class with one photo of the test data is used for comparisons across all categories of people,
- f. Testing with the Convolution Neural Network method to recognize faces is expected to get the best level of accuracy.

Table 1. Data and Class

Name	Training Data	Test Data	Name	Training Data	Test Data
Hasan	1000	5	Nadiah	1000	5
Taruna	1000	5	Ilham	1000	5
Bimo	1000	5	Thomas	1000	5
Tachul	1000	5	Welly	1000	1
Lilik	1000	5	Winda	1000	5
			Zdikri	1000	5

Table 2. CNN Layer

Layer	Dimension Size	Layer	Dimension Size
Convolution Layer + ReLU	64 3x3 Size filter	Convolution Layer + ReLU	512 3x3 Size filter
Convolution Layer + ReLU	64 3x3 Size filter	Convolution Layer + ReLU	512 3x3 Size filter
Maxpooling	Size (2x2) Stride 2	Convolution Layer + ReLU	512 3x3 Size filter
Convolution Layer + ReLU	128 3x3 Size filter	Maxpooling	Size (2x2) Stride 2
Convolution Layer + ReLU	128 3x3 Size filter	Convolution Layer + ReLU	512 3x3 Size filter
Maxpooling	Size (2x2) Stride 2	Convolution Layer + ReLU	512 3x3 Size filter
Convolution Layer + ReLU	256 3x3 Size filter	Convolution Layer + ReLU	512 3x3 Size filter
Convolution Layer + ReLU	256 3x3 Size filter	Maxpooling	Size (2x2) Stride 2
Convolution Layer + ReLU	256 3x3 Size filter	Convolution Layer + ReLU	4096 filter (7x7),
		ReLU	Dropout 50%
Maxpooling	Size (2x2) Stride 2	Convolution Layer + ReLU	4096 filter (1x1),
		ReLU	Dropout 50%
		Convolution Layer + ReLU	4096 filter (1x1)

3. Result

3.1. Data Preparation

The data used in this study were taken in a room using a handheld mobile device. This training data is in the form of MP4 video format with an average of 30 frames per second. This data will later be divided into 11 classes of people's faces with the names Alfian, Taruna, Bimo, Tachul, Lilik, Nadiah, Ilham, Thomas, Welly, Winda and Zdikri. Data processing is carried out in python language with the help of OpenCV and Tensorflow libr.

3.1.1. Video extraction

In the video extraction process, the data will be made into frames, if the video is two seconds long, 60 structures are obtained, this separation is done by opening the video file and capturing each frame found. After that, the results will be saved in the data folder with the name of the photo (current structure). If in the first process, the current frame is 0, so the image will have the name foto0 with a JPG extension. This process will stop when there are no more readable images.

3.1.2. Grayscale

Change the image from RGB colour to grey-scale. The video/image that will be tried to be greyed out is tes.png. Then initialize grey to turn the image grey. In Open-CV using the term "imshow" means displaying a grey image with the result title, Figure 3 (a, b)

3.1.3. Change Image Size

The image size obtained from the video still has a size that is wide enough so that it will prolong the data training process. Therefore the image needs to be resized to be smaller. The measure used is 224x224 pixels. In the Open-CV framework cv2.resize is a program code to cut or crop an image. The image will be cropped to 224x224 pixels with the pre-defined area at img_crop. After that, the image will be saved in the dataset folder. Figure 3 (c) cropping results.

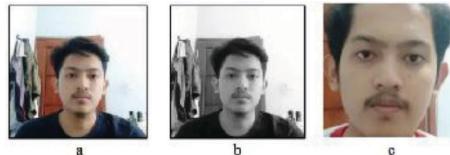


Figure 3. Image Gray-scale Conversion Results (b) on RGB Image Images (a) Image Cropping Results (c).

3.2. Preprocess Training

Before data to be trained, it is necessary to group the data in a folder with each class name. Videos that have been converted into images will be included in a folder called samples and will be used as a dataset in the training process. In the OS module os.listdir is the code to return a list containing the names in the directory list given by path_samples. Here all directory listings in Samples will be copied to the Dataset directory. After that, it displays how many classes there are in the Samples. Not only the name of the folder will be returned the list but also the files in the folder class. The files in the classes folder will be read one by one and then processed with face detection using the Haar cascade classifier

3.3. Haar Cascade Classifier Process

The test will run when the video_capture or camera can be opened. If you cannot open the camera, the program will stop and show that it cannot open the camera. After that take or read the frame by frame on the video webcam. After that, apply a grey-scale filter to the frame. The next process is initializing the faces for the detectMultiscale function, and here the grey image will be processed with a scale factor of 1.3 and with a 30x30 filter. When a look is detected, the program will create a face box with the top point (x, y) and the bottom issue (w, h) with the colour BGR (0, 255, 255), which is yellow. After that, save the detected image in the data folder with the name photo + current frame. Figure 4.a. is an example of a face detection result using the Haar Cascade Classifier, a and b, the development of multi-face detection.

3.4. CNN Process

In this CNN process, first, call the stored pickle function in the pre-training process; contact the sequential procedure on Tensor Flow. Then the next step is to create a layer with the extraction function by calling the conv2D function with the number of filters 64 and size 3 x 3 and using ReLU activation, then calling the MaxPooling2D function with a filter size of 2 x 2. There are 16 convolutional processes in this program, which have been described in Table 2.

The next step for the flatten function is to make the input into one dimension, then the fully connected layer function with three hidden functions layer. The first layer numbered 100, the second layer numbered 10, and the third layer numbered the same as the number of classes, namely 11. Then update all the total weights according to the number of objects using the softmax activation function. Using compile model (sparse categorical cross-entropy) and optimizer (nadam) and metrics (accuracy). The output of this process is the weighted result of a person's photo in a class.

3.5. Training Data and Test Data

To perform classification, first, the training and testing data is made. Data testing in this trial uses 51 photo data with five photos in each class except for Welly's type, which only has one image. In comparison, the training data will be taken in the amount of 100 to 1000 data for each class of the same person. The class label between the training data and the testing data has the same class label.

The training trial dataset was collected from 11 volunteers in the form of facial selfie video data which would later be converted into face-cut images with various lighting conditions and facial positions of each individual. The following is a detailed description of the list of training data presented in Table 3.

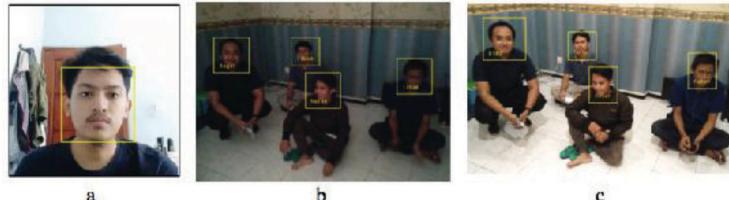


Figure 4. Face Detection Results, (a) mono faces, (b) multi-faces Less lighting, (c) multi-faces Bright lighting

Table 3. Training Dataset

Data	amount	Condition
4 Male Faces	5000 pictures	Bright lighting
3 Female Faces	3000 pictures	Bright lighting
3 Male Faces	3000 pictures	Less lighting
1 Male Face	1000 pictures	Bright lighting

Table 3. explains that the result data will be divided into four groups, namely

1. Four male faces with bright lighting, namely Hasan, Taruna, Bimo, and Zdikri,
2. Three faces of women with bright lighting, namely Lilik, Nadiah, and Winda,
3. 3 Male faces with low lighting, namely Tachul, Thomas, and Ilham.
4. 1 The face of a man with bright lighting is Welly. The test data in the Welly class is used for comparison in all categories, which will later be called test 6.

The dataset that has been collected a training trial will be conducted to get the correct facial recognition model according to the entered dataset. At the model training stage, the input is required in the form of a dataset and label name. The dataset will automatically match the label name.

3.6. Testing

Model testing is done by implementing facial recognition with images or in real-time. The following are the results of all trials using Figs.

3.6.1. Experiment Using Pictures

The following is a description of the test results using a photo which will be translated using a confusion matrix table and the following is a calculation of the accuracy value calculated using equation (2)

$$\text{Accuracy} = \frac{(TP + TN)}{TP + FP + FN + TN} \quad (2)$$

Information:

TP = the prediction result of the system figures an in figure a, TN = the prediction result of the system is figure b in figure b, FP = the prediction result of the system figures in figure b, FN = the prediction result of the system is figure b in figure a. Table 4. 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 2, the total accuracy value for all test data is $59 + 53 + 55 + 61 + 55 + 55 + 60 + 59 + 58 + 60 = 575$ then divided by the total $FN + FP + TN + TP$, namely 600. So that $575 / 660 * 100 = 87,1\%$.

Table 5. explains that the results of testing accuracy if the data are grouped into table 4. Then the image with bright lighting will have more maximum results because the facial features seen by the system will be more.

In testing the system on photos during the recognition process, it has an average duration of about 10-15 seconds. The training process has time. 18 minutes on 100 training data for each class, 24 minutes on 200 training data for each class, 37 minutes at 300 training data per class, 42 minutes on 400 training data for each class, 48 minutes on 500 training data for each class, 61 minutes on 600 training data for each class, 70 minutes on 700 training data for each class, 77 minutes on 800 training data for each class, 88 minutes on 900 training data for each class, 95 minutes on 1000 training data for each class.

Table 4. Accuracy Value of Each Training

Name	100	200	300	400	500	600	700	800	900	1000
Hasan	5	3	5	6	6	6	6	6	6	5
Taruna	4	6	6	4	6	4	5	4	5	5
Bimo	6	5	5	6	5	5	5	5	4	5
Tachul	5	5	6	6	5	5	6	6	5	6
Lilik	6	3	5	6	5	5	5	6	6	5
Nadiah	5	5	6	6	6	6	5	6	6	6
Ilham	5	6	3	6	4	3	5	6	5	5
Thomas	6	5	6	6	6	6	6	6	6	6
Welly	6	6	3	5	2	5	5	2	5	5
Winda	5	4	5	5	4	6	6	6	4	6
Zdikri	6	5	5	5	6	4	6	6	6	6
Total	59	53	55	61	55	55	60	59	58	60
Accuracy%	89,39	80,30	83,33	92,42	83,33	83,33	90,91	89,39	87,88	90,91

Table 5. Accuracy of Lighting Types

Gender	Recording Conditions	Accuracy
Male	Bright lighting	85%
Women	Bright lighting	78%
Male	Less lighting	60.5%

3.6.2. 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 done with one person. When the system runs online, the results from the webcam will automatically record all processed images, which will later be saved as a video with AVI format. The following is the result of implementing a real-time system.

4. Conclusion

Based on the results of research that has designed and made a multi-face recognition detection system and through the tests that have been carried out, it can be concluded that, among others, multi-face recognition technology for monitoring prisoners in detention cells can be done using the Convolutional Neural Network (CNN). The process of making this application with the stages of making Face Recognition, namely image acquisition, preprocessing, extraction, classification, and identification of image data. These stages are made in the python programming language. This study succeeded in using multi Face Recognition by 11 people, a dataset of people's faces consisting of 100-10000 facial data for each person. The dataset is separated into 3 data stages, namely, train data, validation data, and test data. The results of testing the three datasets were able to identify the faces captured by the camera with an accuracy of 87%. The program in this study was successfully used by the surveillance of prisoners in a room in a detention cell. In further research, the Face Recognition program in this study can be used as a reference and used in a more extensive security system as well as a broader scope of opportunities in the development of the world of technology.

5. References

- [1] Chin, V. (2010) Handbook for prison leaders: A basic training tool and curriculum for prison managers based on international standards and norms. Available at: https://www.unodc.org/documents/justice-and-prison-reform/UNODC_Handbook_for_Prison_Leaders.pdf.

- [2] Dharanya, S. et al. (2013) ‘Embedded based 3G security system for prison’, Indian Journal of Science and Technology, 6(5), pp. 4492–4495. doi: 10.17485/ijst/2013/v6i5.17.
- [3] Penal Reform International (2013) ‘Balancing security and dignity in prisons : a framework for preventive monitoring’, in.
- [4] Ekenel, H. K. and Stiefelhagen, R. (2009) ‘Why is facial occlusion a challenging problem?’, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 5558 LNCS(June), pp. 299–308. doi: 10.1007/978-3-642-01793-3_31.
- [5] Singh, S. and Prasad, S. V. A. V. (2018) ‘Techniques and challenges of face recognition: A critical review’, Procedia Computer Science, 143, pp. 536–543. doi: 10.1016/j.procs.2018.10.427.
- [6] Ohlyan, S., Sangwan, M. S. and Ahuja, T. (2013) ‘A Survey On Various Problems & Challenges In Face Recognition’, International Journal of Engineering Research and Technology, 2(6), pp. 2533–2538. Available at: www.ijert.org.
- [7] Gupta, P. et al. (2018) ‘Deep Neural Network for Human Face Recognition’, International Journal of Engineering and Manufacturing, 8(1), pp. 63–71. doi: 10.5815/ijem.2018.01.06.
- [8] Saypadith, S. and Aramvith, S. (2019) ‘Real-Time Multiple Face Recognition using Deep Learning on Embedded GPU System’, 2018 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference, APSIPA ASC 2018 - Proceedings, (November), pp. 1318–1324. doi: 10.23919/APSIPA.2018.8659751.
- [9] Cao, K. et al. (2018) ‘Pose-Robust Face Recognition via Deep Residual Equivariant Mapping’, Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 5187–5196. doi: 10.1109/CVPR.2018.00544.
- [10] Bhattacharya, S. et al. (2018) ‘Smart attendance monitoring system (SAMS): A face recognition based attendance system for classroom environment’, Proceedings - IEEE 18th International Conference on Advanced Learning Technologies, ICALT 2018, pp 358–360. doi: 10.1109/ICALT.2018.00090
- [11] Yamashita, R. et al. (2018) ‘Convolutional neural networks: an overview and application in radiology’, Insights into Imaging, 9(4), pp. 611–629. doi: 10.1007/s13244-018-0639-9.
- [12] G Alaslani, M. and Elrefaei, L. A. (2019) ‘Transfer Learning with Convolutional Neural Networks for IRIS Recognition’, International Journal of Artificial Intelligence & Applications, 10(5), pp. 49–66. doi: 10.5121/ijaia.2019.10505.
- [13] Masdiyasa, I. G. S. et al. (2018) ‘Modified Background Subtraction Statistic Models for Improvement Detection and Counting of Active Spermatozoa Motility’, Lontar Komputer : Jurnal Ilmiah Teknologi Informasi, 9(1), p. 28. doi: 10.24843/lkjiti.2018.v09.i01.p04.
- [14] Useche Murillo, P. C., Moreno, R. J. and Pinzon Arenas, J. O. (2017) ‘Comparison between CNN and Haar classifiers for surgical instrumentation classification’, Contemporary Engineering Sciences, 10(28), pp. 1351–1363. doi: 10.12988/ces.2017.711157.
- [15] Hussain, S. A. and Salim Abdallah Al Balushi, A. (2020) ‘A real time face emotion classification and recognition using deep learning model’, Journal of Physics: Conference Series, 1432(1). doi: 10.1088/1742-6596/1432/1/012087.
- [16] Phuc, L. T. H. et al. (2019) ‘Applying the haar-cascade algorithm for detecting safety equipment in safety management systems for multipleworking environments’, Electronics (Switzerland), 8(10). doi: 10.3390/electronics8101079.
- [17] Triantafyllidou, D. and Tefas, A. (2016) ‘Face detection based on deep convolutional neural networks exploiting incremental facial part learning’, Proceedings - International Conference on Pattern Recognition, 0(December), pp. 3560–3565. doi: 10.1109/ICPR.2016.7900186.
- [18] Syafeeza, A. R. et al. (2014) ‘Convolutional neural network for face recognition with pose and illumination variation’, International Journal of Engineering and Technology, 6(1), pp. 44–57.