INTERNATIONAL JOURNAL
ON INFORMATICS VISUALIZATION

journal homepage : www.joiv.org/index.php/joiv

Face Recognition Using Convolution Neural Network Method with Discrete Cosine Transform Image for Login System

Ari Setiawan [a], Riyanto Sigit [b,*], Rika Rokhana [a]

[a] *Department of Electrical Engineering, Politeknik Elektronika Negeri Surabaya, Surabaya, 60111, East Java, Indonesia*
[b] *Department Informatic and Computer Engineering, Politeknik Elektronika Negeri Surabaya, Surabaya, 60111, East Java, , Indonesia*
*Corresponding author: *riyanto@pens.ac.id*

***Abstract*— These days, the application of image processing in computer vision is becoming more crucial. Some situations necessitate a solution based on computer vision and growing deep learning. One method continuously developed in deep learning is the Convolutional Neural Network, with MobileNet, EfficientNet, VGG16, and others being widely used architectures. Using the CNN architecture, the dataset consists primarily of images; the more datasets there are, the more image storage space will be required. Compression via the discrete cosine transform technique is a method to address this issue. We implement the DCT compression method in the present research to get around the system's limited storage space. Using DCT, we also compare compressed and uncompressed images. All users who had been trained with each test 5 times for a total of 150 tests were given the test. Based on testing findings, the size reduction rate for compressed and uncompressed images is measured at 25%. The case study presented is face recognition, and the training results indicate that the accuracy of compressed images using the DCT approach ranges from 91.33% to 100%. Still, the accuracy of uncompressed facial images ranges from 98.17% to 100%. In addition, the accuracy of the proposed CNN architecture has increased to 87.43%, while the accuracy of MobileNet has increased by 16.75%. The accuracy of EfficientNetB1 with noisy-student weights is measured at 74.91%, and the accuracy of EfficientNetB1 with imageNet weights can reach 100%. Facial biometric authentication using a deep learning algorithm and DCT-compressed images was successfully accomplished with an accuracy value of 95.33% and an error value of 4.67%.**

***Keywords*— Computer vision; convolutional neural network; discrete cosine transform; face recognition; authentication.**

Manuscript received 4 Jan. 2023; revised 25 Jan. 2023; accepted 20 Feb. 2023. Date of publication 30 Jun. 2023.

I. INTRODUCTION

A system's authentication process can be carried out in several ways, ranging from using passwords, RFID [1] to biometrics [2], [3]. RFID is a system capable of automatically transmitting identification (ID) utilizing radio waves [4]. Classify the applications of RFID into the following categories: RFID-based smart fitting room [5], public transit, access control, animal identification, container identification, athletic events, industrial automation, and medicinal uses [6]. Biometrics is the identification of an individual based on their bodily traits. This identifying procedure can utilize fingerprints, faces, iris, and arm veins [3], [7]. The financial sector was a pioneer in developing and implementing biometrics at the outset of its emergence. With the advancement of technology, biometrics have permeated the commercial, educational, and healthcare sectors.

Passwords have the problem of being forgotten, whereas RFID has the disadvantage of not being carried over or losing the card. Although each method has its benefits and drawbacks, the biometric method is a particularly special technique. Biometrics are regarded as unique since they are the natural characteristics of humans. In other words, biometry can solve system authentication issues caused by passwords and RFID cards. Biometric research has focused on the iris, palm veins, fingerprints, human face identification, and facial emotion [8]. Several researchers have discovered human face recognition in their efforts to find human faces. Today, the technology and the use cases for face recognition are expanding dramatically. It is employed, among other things, to classify gender or authenticate systems. In a study by Lin and Xie [9], they experimented with gender classification of facial recognition, and the results in the Asian celebrity face dataset reached 97.4%.

In addition, facial recognition is utilized to protect the system's or device's stored data or assets. Its uses and

applications are vast, spanning from banking and payment to health and staff attendance. Continued research into the necessity and advancement of face-related biometric technologies is intriguing. A technique has been developed to recognize faces using the Haar cascade method to detect faces [10].

As face detection technology advances, scientists are also developing facial recognition. This face recognition procedure is an example of classifiable image processing. The method that is currently known and being developed is deep learning. In discriminatory tasks, Deep Learning has made progress. This is enabled by advancements in Deep Learning architecture, powerful computation, access to vast amounts of data, as well as an ever-changing reset. Deep Learning Networks have been applied successfully to Computer Vision tasks for image classification, object detection, and image segmentation. Convolutional Neural Network (CNN), a deep learning algorithm that may be applied to computer vision problems, is currently the most prevalent technique.

Many CNN architectures exist and are developing. This study will use transfer learning to retrain the MobilNet, VGG16, and EfficientNetB1 architectural models. The quantity, image aspect ratio, and image size of the datasets are issues that arise while employing CNN. Due to these issues, using CNN for face recognition requires a large number of datasets, resulting in a huge file size for the dataset as a whole. Using CNN for image classification will necessitate a colossal picture storage capacity, which becomes problematic for storing facial images for face recognition. A method is required to compress the picture dataset since the image size can be lowered while maintaining the image's characteristics. Discrete Cosine Transform (DCT) is one of the many ways to minimize image size.

II. Materials and Method

Currently, digital picture processing is required. Nonetheless, inadequate memory for data processing is one of the challenges faced. A method for compressing images is fundamental. Image compression is required to ensure that images transported or processed do not consume too much memory or hard drive space. The compression method is anticipated to lower the file size without altering the image's characteristics. Discrete Cosine Transform (DCT) offers a solution to such a problem, and DCT technique converts the spatial domain to the frequency domain.

For storage efficiency, a picture compression method is required to minimize the image's size. Xiao et al. [11] research apply the Discrete Cosine Transform (DCT) compression algorithm to video. Such a method was applied to underwater picture compression, employing the DCT method to reduce image size while maintaining accuracy. This occurs because the DCT method compresses photos by reducing the number of bits but does not compromise image features. According to research by Setiawan, Sigit and Rokhana [12], the DCT compression approach, in addition to lowering the image's file size, can also accelerate the learning process with a performance value of more than 90%.

In the present study, the case to be raised is the effect on the accuracy value of compressed datasets with uncompressed datasets. Based on existing research, implementing DCT in deep convoluted neural networks can increase the classification level to 10% [13]. Despite using compression, the condition of the features of the image does not change. We compared a dataset of compressed images using the DCT method with images that were not compressed. The dataset collection in this experiment was obtained by taking pictures directly using a webcam and from the author's image gallery. We collect the dataset using the Haar cascade algorithm to perform face detection. The method of collecting images is in accordance with the block diagram in Fig. 2, with a total of 1.500 photos separated into 30 classes collected for this research. The dataset consists of 50 face images for each class.

When capturing images using a webcam, the device utilizes an external webcam with the specs stated in Table 1.

TABLE I
WEBCAM SPECIFICATION

Parameter	Specification
Type	Logitech C270
Resolution	720p/30fps
Focus	Fix focus
Camera	0.9 Megapixel

The hardware in this study uses an AMD Ryzen 5 4500G with 32 GB RAM Display NVIDIA GeForce GTX 1650. The system diagram used is shown in Fig. 1.

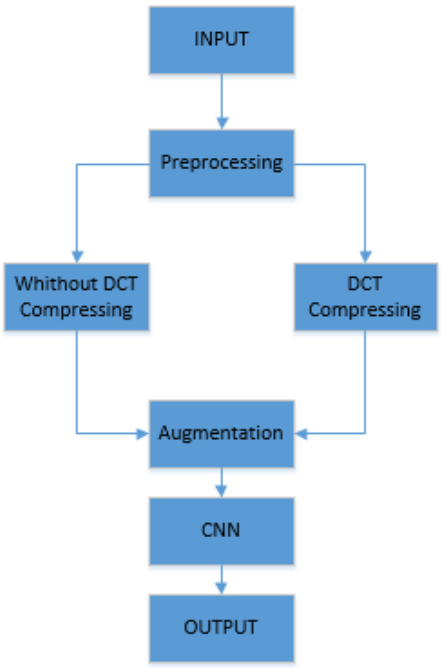


Fig. 1 System diagram

A. Preprocessing

Modifying the resulting model to execute face recognition with optimal accuracy is crucial. In order to prepare the dataset, it is of paramount importance to perform image preprocessing. In this work, face-area cropping and resizing were performed as preprocessing. Face detection can be used to locate and index images and videos with background, size, and position. There are several face detection technologies [10], including Haar cascade.

The Haar cascade method was utilized in this preprocessing step to determine the facial region. A previous study presented an efficient object recognition method based on Haar feature-based cascade classification [14]. The

method is based on research demonstrating that the cascade function greatly boosts the detector's speed by concentrating on the most promising areas of an image.

The face is specified in order toward to webcam throughout data collection. The image preparation procedure uses the system diagram shown in Fig. 2.

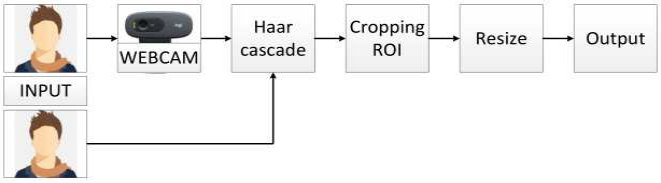


Fig. 2 Preprocessing system diagram

This method was applied to datasets obtained through image galleries or by direct webcam retrieval. Following face detection, the image was cropped using Region Of Interest (ROI) and resized to 300x300 pixels. Fig 3 illustrates an example of a face crop.

Fig. 3 Image cropping

Fig. 4 represents the results of the dataset retrieval performed using a webcam camera.

Fig. 4 Sample dataset collection results

B. Discrete Cosine Transform (DCT)

Discrete cosine transformation (DCT) separates an image into spectral parts or sub bands for different functions related to the image's visual quality. DCT converts the signal or image from the spatial domain to the frequency domain, where this method is one of the techniques in image compression. DCT has different types, and all DCTs have excellent energy compaction properties [15]

Compression with the DCT method can be used to perform bit reduction that is not detrimental to the details or features of the image. Mukti [16] compared the compression of jpg and png images on underwater images, with compression results reaching 96%. In the case of an image, the image will be separated by certain blocks so that the block will depend on making a 2D-DCT transformation [17].

One-dimensional DCT equation from real numbers with the following formula [18], [19], [20]:

$$C(u) = \sqrt{\frac{2}{N}}\alpha\,(u)\sum_{x=0}^{n-1} f\,cos\left(\frac{\pi u(2n+1)}{1}\right) \quad (1)$$

$$whare\ C(u) = \begin{cases}\frac{1}{2}, & u = 0\\ 1, & otherwise\end{cases} \quad (2)$$

The values of these blocks can be represented in the function f(x, y) and the function F (u , v), then [21], [22]:

$$F(u) = C(u)\sum_{x=0}^{n-1} f(x)\,cos\left[\frac{(2x+1)u\pi}{2n}\right] \quad (3)$$

$$F(v) = C(v)\sum_{y=0}^{n-1} f(y)\,cos\left[\frac{(2y+1)v\pi}{2n}\right] \quad (4)$$

$$f(x) = \sum_{u=0}^{n-1} F(u)\,C(u)\,cos\left[\frac{(2x+1)u\pi}{2n}\right] \quad (5)$$

$$f(y) = \sum_{v=0}^{n-1} F(v)\,C(v)\,\cos\left[\frac{(2y+1)v\pi}{2n}\right] \quad (6)$$

$$whare\ C(u,v) = \begin{cases}\frac{1}{2}, & u,v = 0\\ 1, & otherwise\end{cases} \quad (7)$$

C. Augmentation

In performing image classification, especially in deep learning, they often experience the problem of too few datasets [23]. When the training data set is unbalanced (the amount of available data is not the same between different categories), image classification accuracy often decreases significantly [24]. For this reason, data augmentation techniques are usually used to expand the dataset for better training performance [25]. The augmentation process modifies the dataset to have more variations [26]. Some commonly used augmentations are rotation, flipping, zooming, and contrast transformation [27]. Rotation augmentation was performed by rotating the image clockwise or counterclockwise between 1 and 359. Rotational augmentation is primarily determined by the degree of rotation parameter. The rotation carried out in this experiment is 400.

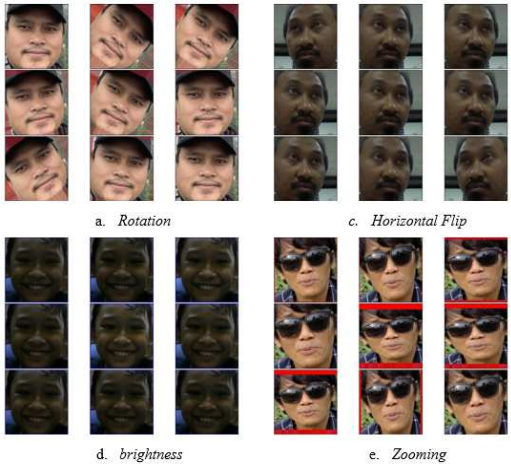


Fig. 5 Sample augmentation process

Augmentation flipping was performed by horizontally flipping the image, as horizontal flipping is more prevalent than vertical flipping. This is one of the simplest augmentations to apply, and it has shown to be useful in datasets, e.g., CIFAR-10 and ImageNet [27]. Pictured below is an example of the augmentation we performed.

D. Convolutional Neural Network

CNN is a feed-forward artificial neural network because it extracts spatial data from images and is highly effective for image classification issues. In 1998, researchers Yann LeCun, Leon Bottou, Joshua Bengio, and Patrick Haffner utilized Deep Convolutional Neural Networks for number recognition following the creation of neural networks in 1943 by Warren S. McCulloch and Walter Pitts and the advancement of technology. Researchers continue to enhance the accuracy and performance of each design. Current CNN architecture includes various modifications, such as MobilNet, LeNet-5, AlexNet, ZFNet, GoogLeNet VGGNet, etc. The primary architecture components based on the existing model are the convolution layer, pooling layer, fully connected layer, and dropout. The convolutional layer is a collection of neurons that form a filter with dimensions of length and height (pixels).

The placement of the pooling layer in the CNN architecture occurs after the convolutional layer. Pooling later consists of a filter with a particular size and stride that shifts throughout the activation map. This layer reduces the number of parameters that can be trained to prevent overfitting; thus, it is critical to implement correct pooling methods so that the reduced feature map keeps all the important features [28].

In the application, either maximum or average pooling may be used. The pooling layer accelerates computation because fewer parameters must be updated and overfitted.

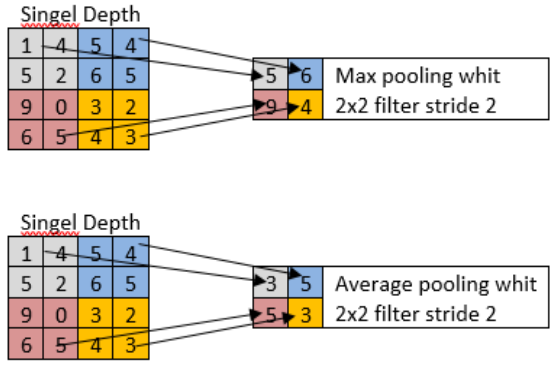


Fig. 6 Illustration of maximum and average pooling

The activation map produced by the feature extraction layer is still in the form of a multidimensional array; therefore, it must be reshaped into a vector before it can be used as input by the fully linked layer. This layer comprises a hidden layer, an activation function, an output layer, and a loss function. It is typically employed in applying multilayer perceptron's to alter the data dimensions. Before each neuron in the convolution layer can be entered into a fully linked layer, it must be turned into one-dimensional data because it causes the data to lose spatial information and is irreversible. In contrast, the fully linked layer can only be deployed on the networks.

The first thing to consider is how to reduce overfitting because we only expect a small amount of data to be collected in the scenario we are considering. The most common method for lowering overfitting in the learning process is to increase dropout rates. By randomly dropping features from feature maps, many cutting-edge Deep Neural Networks (DNNs) use dropout to alleviate the overfitting problem [29].

Dropout is a regularization technique for neural networks that prevent the usage of randomly selected neurons during training. The neurons are randomly removed. The introduction of dropout can speed up learning in addition to reducing overfitting. Dropouts are typically used in completely connected layers, where most parameters are located. We specifically set the dropout rate at 0.2.

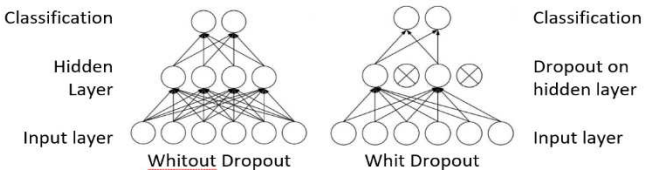


Fig. 7 Layer dropout diagrams

CNN uses multiple configurations of multilayer perceptrons to reduce the amount of preprocessing required by the learning algorithm [30]. In recent years, many classifiers based on CNN networks have been proposed, such as VGG16 [31], EfficientNet [32], and MobileNet [33]. The CNN model has an exceptionally high level of detection accuracy for classification performance. The CNN architecture we used is MobileNet, EfficientNetB1, and VGG16, according to this study. This study utilized EfficientNetB1 pre-training on noisy students and EfficientNetB1 and MobileNet models pre-training on ImageNet. Fig. 8 shows the proposed CNN architecture, with the parameters presented in Table 2.

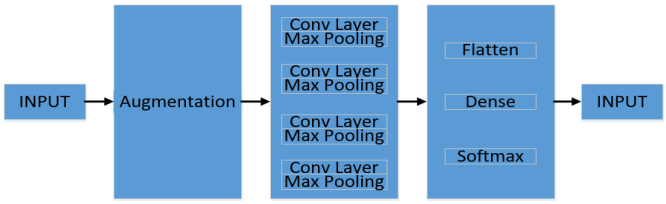


Fig.8 Proposed CNN architecture

TABLE II
PARAMETERS LAYER ARCHITECTURE

Layer	Parameters
Conv Layer	Kernel 3x3 Strides 2 Activation Function relu Input shape 150x150x3
Max Pooling	-
Conv Layer	Activation Function relu
Max Pooling	-
Conv Layer	Activation Function relu
Max Pooling	-
Conv Layer	Activation Function relu
Max Pooling	-
Drop Out	Dropout 0.5
Flatten	
SoftMax	Activation Function Softmax

E. Testing Authentication System

The following Fig. 9 shows the concept design of the system authentication program's flowchart.

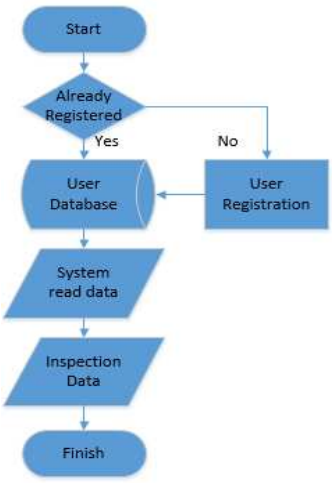


Fig. 9 Flowchart authentication system

For evaluating the performance of the model, the confusion matrix is utilized. The formula for determining accuracy is as follows [34]:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (8)$$

Where TP is true positive, TN is true negative, FP is false negative, and FN is false negative.

III. Results and Discussion

A. Preprocessing dataset retrieval and collection.

The collection of datasets was performed according to block figure 4, with a total of 1.500 photos separated into 30 classes collected for this research. The dataset consists of 50 face images for each class.

When capturing images using a webcam, the device utilizes an external webcam with the specs stated in Table 3.

TABLE III
WEBCAM SPECIFICATION

Parameter	Specification
Type	Logitech C270
Resolution	720p/30fps
Focus	Fix focus
Camera	0.9 Mega pixel

Figure 10 shows the results of the data collection method.

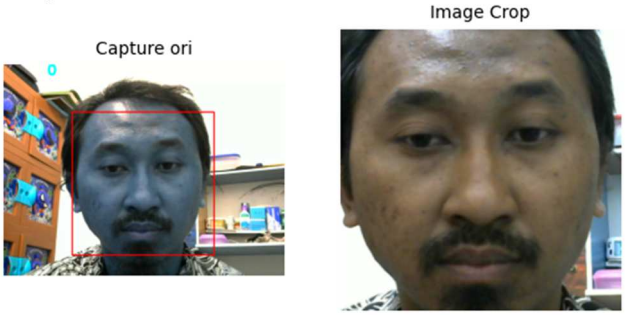


Fig. 10 Sample image faces

The author used Haar cascade frontal face and Haar cascade eye to identify the face and eye areas. All data recognized by Haar cascade is scaled to 300 by 300 pixels and then stored on the hard drive. Fig. 11 is the outcome of storing the completed dataset.

Fig. 11 Sample dataset

B. Image compression.

Before proceeding with the learning process, we compress the dataset that has been collected so that we get two groups of datasets, namely datasets with compression and datasets without compression with DCT. Fig. 12 illustrates an example of a comparison between compressed and uncompressed images. Meanwhile, Fig. 13 shows the file size before and after compression, and it can be seen that there is a decrease in the image size, which will undoubtedly save storage space.

From Fig. 12 and Fig. 13, the DCT compression method that we employed is not only able to reduce the image file size by 25%, but this method also does not damage the features of the original image. After collecting two groups of datasets, we conducted experiments by proposing CNN architecture and conducting transfer learning for the CNN MobilNet, VGG16, and EfficientNetB1 architectures.

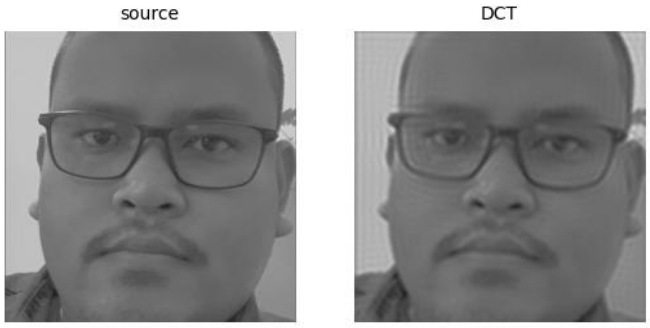


Fig. 12 Compressed image comparison

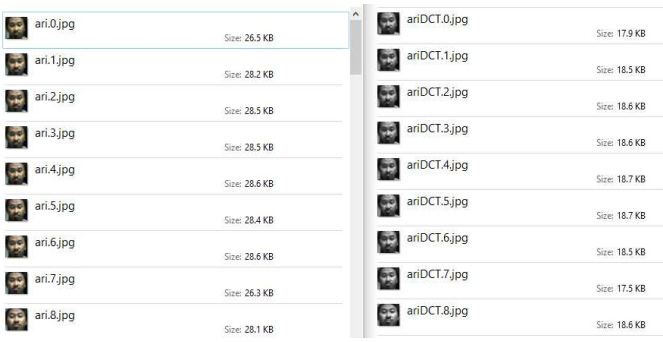


Fig. 13 File size before and after compression DCT

C. CNN's Training Process

The overall dataset used is 1,500 face images, with 80% as training, 10% as validation, and 10% as testing data. As a result, the composition of the dataset as training is 1,200 images, 150 images as validation, and 150 images as testing. Our learning process is carried out on datasets with two groups, the first is datasets without compression using DCT, and the second is compression using the DCT method. The results of the training are provided in Tables 4 and 5.

TABLE IV
RESULTS OF ACCURACY TRAINING WITHOUT COMPRESSING

Epoch Value	CNN Structure (Accuracy %)				
	Proposed CNN	Mobile NetV2	Efficient NetB1 ImageNet	EfficientNetB1 Noisy-Student	VGG16
10	91.08	5.58	99.75	97.17	75.17
20	97.00	14.75	98.67	98.5	90.58
50	99.08	33.08	99.5	99.92	96.33
100	99.42	82.75	99.75	99.5	97.17
200	99.92	71.25	99.33	99.92	97.58
300	99.58	98.75	99.83	99.83	98.17
400	99.92	99.42	99.42	100	96.08
500	100	97.17	96.83	99.92	99.33

Based on the table above, the accuracy of each architecture will continue to increase as the number of epochs increases. In the MobilNet and EfficientNetB1 ImageNet architectures, there is a similarity in the accuracy values at the 400th epoch, with an accuracy value of 99.42%. For the proposedCNN architecture on the 200th and 400th epochs with an accuracy value of 99.92%. The VGG16 architecture in the 400th epoch experienced a decrease in performance from the previous epoch with a decrease in accuracy from the previous reaching 2.09 points with a percentage decrease of 2.13%. The EfficientNetB1 architecture with the noisy weight student is the architecture with the best performance value with an accuracy value of 100% in the 400th epoch.

Figs 14, 15, and 16 depict graphs showing accuracy and loss of training results using uncompressed datasets.

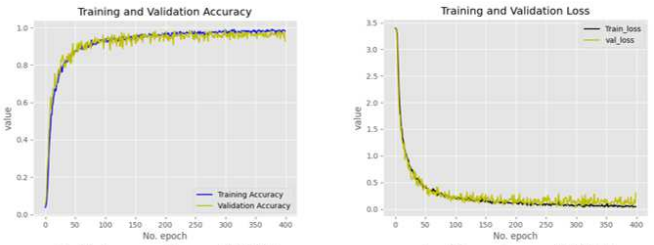


Fig. 14 Accuracy and loss proposed CNN

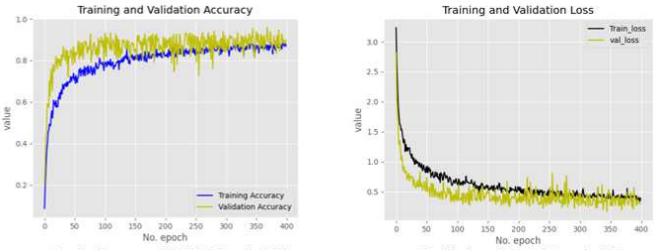


Fig. 15 Accuracy and loss VGG16

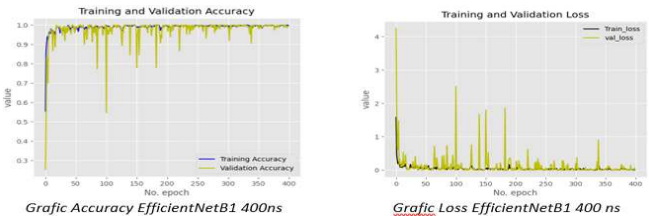
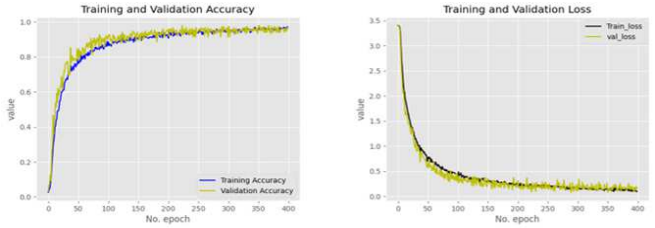


Fig. 16 Accuracy and loss Efficient net noisy-student

Fig. 17, 18, and 19 represent the accuracy and loss training graphs for a DCT-compressed dataset.



Fig, 17 Accuracy and loss proposed CNN

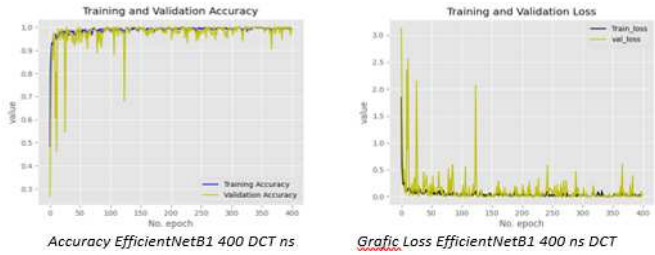


Fig.18 Accuracy and loss Efficient net noisy student

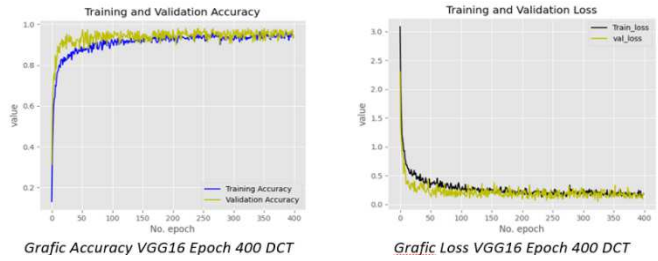


Fig. 19 Accuracy and loss VGG16

The best results are obtained at epochs between 300 and 400, as shown in Table 3, with accuracy values ranging from 98.17% to 100%. The best accuracy value is provided by the EfficiencyNetB1 architecture, which has an accuracy value of up to 100%. A comparison graph of accuracy values during training is shown in Fig 20.

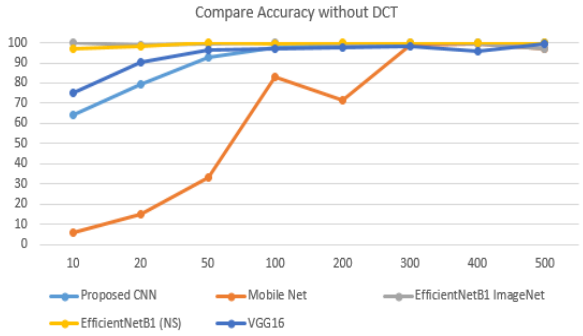


Fig. 20 Training accuracy without DCT

TABLE V
RESULT OF ACCURACY TRAINING WITH COMPRESSING

Epoch Value	CNN Structure (Accuracy %)				
	Proposed CNN	Mobile NetV2	Efficient NetB1 ImageNet	Efficient NetB1 Noisy-Student	VGG16
10	89.42	3.33	98.92	98.92	84.25
20	99.17	13.17	95.50	98.92	80.33
50	99.50	37.67	98.75	98.08	96.42
100	99.92	71.58	86.92	99.67	97.58
200	100	97.75	99.58	99.92	96.42
300	98.75	98.50	100	98.42	96.92
400	89.50	91.33	99.83	99.83	98.67
500	100	97.67	100	100	99.75

Table 4 provides the result of experiments conducted on datasets compressed using the DCT method, with the result that at epochs between 300 to 500, all architectures reach optimal levels. Based on these results, it can be seen that EfficientNetB1 is the best architecture in the case we raised, with the accuracy value of the training results reaching 100%. Fig. 21 is a graph representing the outcomes of comparing the training process using a compression dataset.

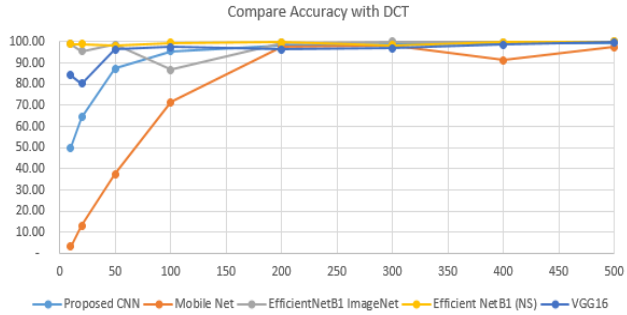


Fig. 21 Training accuracy with DCT image.

The graph in Fig. 21 shows that, on average, for each architecture used in training, the accuracy value will frequently increase with an increase in the number of epochs used. We tested the model on the 300th and 400th epoch using a testing dataset of 150 images, and the findings are as follows:

TABLE VI
TESTING RESULT

CNN Architecture	% Error Image without Compression DCT		% Error Image with Compression DCT	
	Epoch 300	Epoch 400	Epoch 300	Epoch 400
ProposedCNN	5.33	3.33	0.67	3.33
MobileNet	4	0.67	3.33	6.67
EfficientNet B1 (Imagenet)	3.33	0.67	0	0.67
EfficientNet B1 (Noisy-Student)	2.67	0.67	0.67	0
VGG16	0.67	10.67	6	6

From the test results, the structure of CNN EfficientNetB1 using Noisy-Student weights with image compression using the DCT method can reduce the error value during testing. The accuracy of the proposed architecture increased, reaching 87.43% for the mobile net, 16.75% for efficientnetb1, 74.91% for efficient netb1 with noisy-student weights, and 100% for efficientnetb1 with weights for imagenet accuracy. The 300th epoch was when this occurred. The error comparison graph when testing for some of the architectures is shown in the following Fig. 22.

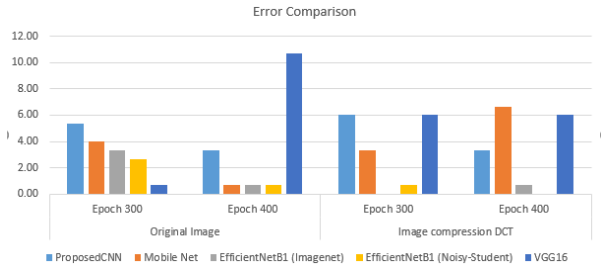


Fig. 22 Error comparison dataset

The confusion matrix is utilized to determine the number of prediction errors during testing. The graphic below presents the EfficientNet B1 (Noisy-Student) architecture's confusion matrix. Figure 23 shows architecture EfficientNet B1 (Noisy-Student) with the original image, whereas Figure 24 shows architecture EfficientNet B1 (Noisy-Student) with a DCT image.

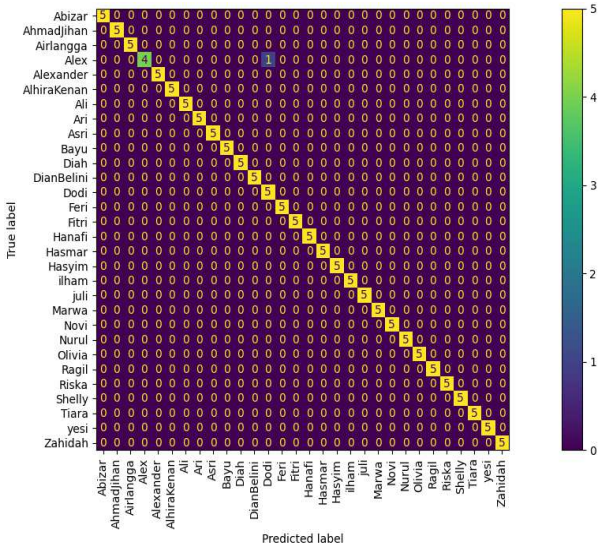


Fig. 23 Confusion matrix EfficientNet B1 (Noisy-Student) with the original image

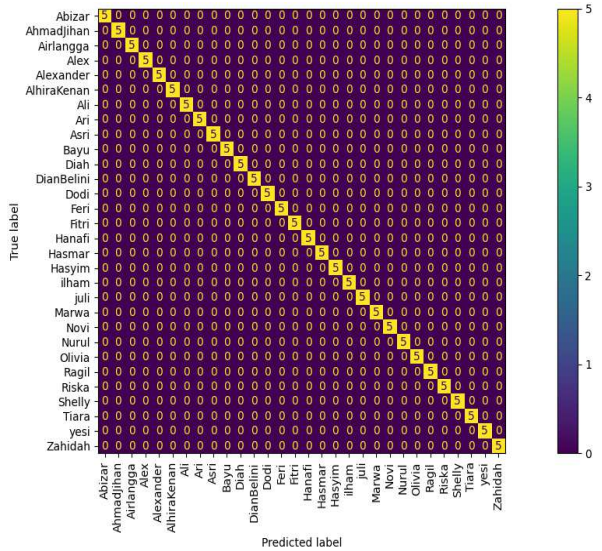


Fig. 24 Confusion matrix EfficientNet B1 (Noisy-Student) with a DCT image.

The use of CNN in the EfficientNetB1 architecture with the DCT image compression method on the dataset can be used as an alternative to increasing the accuracy of testing in the case of face recognition.

D. Testing authentication on the system.

The test scenario consists of capturing the subject's face from 30 cm, 50 cm, 70 cm, 90 cm, and 110 cm away from the webcam. The test data findings are shown in the following table.

TABLE VII
THE RESULTS OF DISTANCE TESTING

No	Distance (cm)	*Capture*	User ID	Status
1	30		8	Success
2	50		8	Success
3	70		8	Success
4	90		8	Success
5	110		8	Success

Following that, we tested all of the trained users 5 times, for a total of 150 tests. The test results are shown in the following Fig. below:

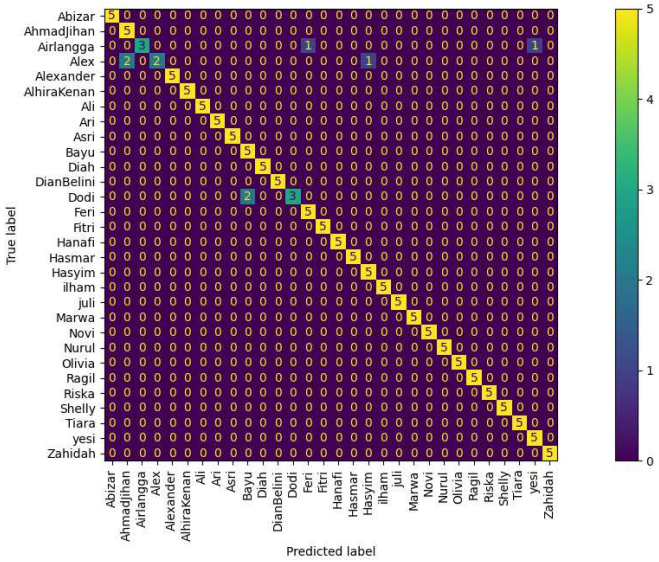


Fig. 25 Confusion matrix

As shown in fig.25 above, the results of the login authentication test into the system were performed on registered users with 5 tests. The accuracy was 95.33%, with 143 successes and 7 errors, and the error was 4.67%.

IV. CONCLUSION

Accuracy can be improved through testing utilizing compressed datasets and the DCT approach shown in Table 7. The accuracy of the proposed architecture increased, reaching 87.43% for MobileNet, 16.75% for EfficientNetB1, 74.91% for EfficientNetB1 with noisy-student weights, and 100% for EfficientNetB1 with weights for imagenet accuracy. The 300th epoch was when this occurred. When using images compressed with the DCT method, accuracy is increased in the 400th epoch compared to images that are not compressed. The accuracy value reaches 100% when using the noisy-student weights in the EfficientNetB1 architecture.

In this study, facial biometric authentication using the Convolutional Neural Network deep learning algorithm and DCT-compressed images can be successfully accomplished up to 143 times and erroneously up to 7 times, with an accuracy value of 95.33% and an error value of 4.67%. In order to develop improvements and conduct future research, it is necessary to use the invariant illumination method in order to eliminate errors caused by lighting that were discovered during testing.

ACKNOWLEDGMENT

The authors are grateful to all academicians at the State Electronics Polytechnic of Surabaya (PENS) for supporting this research and providing research resources. We are obliged to all colleagues in the Signal Vision and Graphic laboratories for their assistance in completing this research.

REFERENCES

[1] Inkenzeller, RFID Handbook–Fundamentals and Applications in Contactless Smart Cards and Identification–2 nd Edition, West Sussex England: John Wiley & Sons Ltd, 2003, ISBN: 0-470-84402-7.

[2] E. P. Kukula, M. J. Sutton and S. J. Elliott, "The Human–Biometric-Sensor Interaction Evaluation Method: Biometric Performance and Usability Measurements," *IEEE Transactions on Instrumentation and Measurement,* vol. 59, no. 4, pp. 784-791, 2010, doi: 10.1109/TIM.2009.2037878.

[3] K. Hongo and H. Takano, "Personal Authentication with an Iris Image Captured Under Visible-Light Condition," in *2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, Miyazaki, Japan, 2018, doi: 10.1109/SMC.2018.00389.

[4] G. Álvarez-Narciandi, A. Motroni, M. R. Pino, A. Buffi and P. Nepa, "A UHF-RFID gate control system based on a Convolutional Neural Network," in *2019 IEEE International Conference on RFID Technology and Applications (RFID-TA)*, Pisa, Italy, 2019, doi: 10.1109/RFID-TA.2019.8892080.

[5] M. Hauser, M. Griebel and F. Thiesse, "A hidden Markov model for distinguishing between RFID-tagged objects in adjacent areas," in *2017 IEEE International Conference on RFID (RFID)*, Phoenix, AZ, USA, 2017, doi: 10.1109/RFID.2017.7945604.

[6] Y. Á. López, J. Franssen, G. Á. Narciandi, J. Pagnozzi, I. G.-P. Arrillaga and F. L.-H. Andrés, "RFID Technology for Management and Tracking: e-Health Applications," *Sensor,* vol. 18, no. 8, 2018, doi: 10.3390/s18082663.

[7] J. Wang and G. Wang, "Quality-Specific Hand Vein Recognition System," *IEEE Transactions on Information Forensics and Security,* vol. 12, no. 11, pp. 2599 - 2610, November 2017, doi: 10.1109/TIFS.2017.2713340.

[8] H. B. Khoirullah, N. Yudistira and F. A. Bachtiar, "Facial Expression Recognition Using Convolutional Neural Network with Attention

Module," *International Journal on Informatics Visualization,* vol. 6, no. 4, pp. 897-903, 2022, doi: doi: 10.30630/joiv.6.4.963.

[9] Y. Lin and H. Xie, "Face Gender Recognition based on Face Recognition Feature Vectors," in *2020 IEEE 3rd International Conference on Information Systems and Computer Aided Education (ICISCAE)*, Dalian, China, 2020, doi: 10.1109/ICISCAE51034.2020.9236905.

[10] D. T. P. Hapsari, C. G. Berliana, P. Winda and M. A. Soeleman, "Face Detection Using Haar Cascade in Difference Illumination," in *2018 International Seminar on Application for Technology of Information and Communication*, Semarang, Indonesia, 2018, doi: 10.1109/ISEMANTIC.2018.8549752.

[11] X. Xiao, Y. Zhuang, Z. Wang and X. Zhang, "A reconstruction algorithm based on 3D tree-structure Bayesian compressive sensing for underwater videos," in *2015 IEEE International Conference on Information and Automation*, Lijiang, China, 2015, doi: 10.1109/ICInfA.2015.7279411.

[12] A. Setiawan, R. Sigit and R. Rokhana, "Implementation of Face Recognition Using Discrete Cosine Transform on Convolutional Neural Networks," in *2022 International Electronics Symposium (IES)*, Surabaya, Indonesia, 2022, doi: 10.1109/IES55876.2022.9888295.

[13] M. Haque, "A two-dimensional fast cosine transform," *IEEE Transactions on Acoustics, Speech, and Signal Processing,* vol. 33, no. 6, pp. 1532 - 1539, December 1985, doi: 10.1109/TASSP.1985.1164737.

[14] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Computer Society Conference on Computer Vision and Pattern Recognition*, USA, 2001, doi: 10.1109/CVPR.2001.990517.

[15] W. Cao and Y. Zhou, "4 × 4 parametric integer discrete cosine transforms," in *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, Banff, AB, Canada, 2017, doi: 10.1109/SMC.2017.8123167.

[16] K. Mukti, "Analisa Discrete Cosine Transform Pada Kompresi Underwater Image," Universitas Dian Nuswantoro Semarang, Indonesia, 2015.

[17] N. Ahmed, T. Natarajan and K. Rao, "Discrete Cosine Transform," *IEEE Transactions on Computers,* Vols. C-23, no. 1, pp. 90 - 93, January 1974, doi: 10.1109/T-C.1974.223784.

[18] V. G. Reju, S. N. Koh and I. Y. Soon, "Convolution Using Discrete Sine and Cosine Transforms," *IEEE Signal Processing Letters,* vol. 14, no. 7, pp. 445 - 448, July 2007, doi: 10.1109/LSP.2006.891312.

[19] S. Martucci, "Symmetric convolution and the discrete sine and cosine transforms," *IEEE Transactions on Signal Processing,* vol. 42, no. 5, pp. 1038 - 1051, May 1994, doi: 10.1109/78.295213.

[20] C.-W. Kok and W.-S. Tam, "Transform Domain," in *Digital Image Interpolation in Matlab*, Wiley-IEEE Press, 2019, doi: 10.1002/9781119119623.ch5, pp. 123 - 159.

[21] M. A. Abdalla, A. A. Abdo and A. O. Lawgali, "Utilizing Discrete Wavelet Transform and Discrete Cosine Transform for Iris Recognition," in *2020 20th International Conference on Sciences and Techniques of Automatic Control and Computer Engineering (STA)*, Monastir, Tunisia, 2020, doi: 10.1109/STA50679.2020.9329312.

[22] Y. Wei, C. Zhu, Y. Yang and Y. Liu, "A Discrete Cosine Model of Light Field Sampling for Improving Rendering Quality of Views," in *2020 IEEE International Conference on Visual Communications and Image Processing (VCIP)*, Macau, China, 2020, doi: 10.1109/VCIP49819.2020.9301838.

[23] L. Perez and J. Wang, "The Effectiveness of Data Augmentation in Image Classification using Deep Learning," in *arXiv:1712.04621v1 [cs.CV] 13*, Dec 2017, doi: 10.48550/arXiv.1712.04621.

[24] S. Liu, J. Zhang, Y. Chen, Y. Liu, Z. Qin and T. Wan, "Pixel Level Data Augmentation for Semantic Image Segmentation Using Generative Adversarial Networks," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Brighton, UK, 2019, doi: 10.1109/ICASSP.2019.8683590.

[25] H. Xu and e. al, "An Enhanced Framework of Generative Adversarial Networks (EF-GANs) for Environmental Microorganism Image Augmentation With Limited Rotation-Invariant Training Data," *IEEE Access,* vol. 8, pp. 187455 - 187469, 2020, doi: 10.1109/ACCESS.2020.3031059.

[26] Abdurrahman and A. Purwarianti, "Effective Use of Augmentation Degree and Language Model for Synonym-based Text Augmentation on Indonesian Text Classification," in *2019 International Conference on Advanced Computer Science and information Systems (ICACSIS)*, Bali, Indonesia, 2019, doi: 10.1109/ICACSIS47736.2019.8979733.

[27] C. Shorten and T. M. Khoshgoftaar, "A survey on Image Data Augmentation for Deep Learning," *Journal of Big Data,* vol. 6, 2019, doi: 10.1186/s40537-019-0197-0.

[28] N. Devi and B. Borah, "Cascaded pooling for Convolutional Neural Networks," in *2018 Fourteenth International Conference on Information Processing (ICINPRO)*, Bangalore, India, 2018, doi: 10.1109/ICINPRO43533.2018.9096860.

[29] K.-B. Nguyen, J. Choi and J.-S. Yang, "Checkerboard Dropout: A Structured Dropout With Checkerboard Pattern for Convolutional Neural Networks," *IEEE Access,* vol. 10, pp. 76044 - 76054, 2022, doi: 10.1109/ACCESS.2022.3190643.

[30] Y. Lecun, L. Bottou, Y. Bengio and P. Haffner, "Gradient-based learning applied to document recognition," in *Proceedings of the IEEE*, Vol. 86, No. 11, pp. 2278-2324, 1998, doi: 10.1109/5.726791.

[31] S. Liu and W. Deng, "Very deep convolutional neural network based image classification using small training sample size," in *2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR)*, Kuala Lumpur, Malaysia, 2015, doi: 10.1109/ACPR.2015.7486599.

[32] M. Tan and Q. V. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," in *36th International Conference on Machine Learning, ICML 2019*, Long Beach, 2019, doi: 10.48550/arXiv.1905.11946.

[33] M. R. Islam, N. Tasnim and S. B. Shuvo, "MobileNet Model for Classifying Local Birds of Bangladesh from Image Content Using Convolutional Neural Network," in *2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, Kanpur, India, 2019, doi: 10.1109/ICCCNT45670.2019.8944403.

[34] A. Lawi and F. Aziz, "Classification of Credit Card Default Clients Using LS-SVM Ensemble," in *2018 Third International Conference on Informatics and Computing (ICIC)*, Palembang, Indonesia, 2018, doi: 10.1109/IAC.2018.8780427.