IAES International Journal of Artificial Intelligence (IJ-AI)
Vol. 14, No. 2, April 2025, pp. 1056~1066
ISSN: 2252-8938, DOI: 10.11591/ijai.v14.i2.pp1056-1066

Enhancing facial recognition accuracy through feature extractions and artificial neural networks

Adhi Kusnadi, Ivranza Zuhdi Pane, Fenina Adline Twince Tobing
Department of Informatics, Faculty of Engineering and Informatcs, Universitas Multimedia Nusantara, Banten, Indonesia

Article Info

Article history:

Received Dec 28, 2023
Revised Oct 26, 2024
Accepted Nov 14, 2024

Keywords:

Backpropagation
Convolutional neural network
Discrete cosine transform
Face recognition
Feature extraction
Gaussian mixture model

ABSTRACT

Facial recognition is a biometric system used to identify individuals through faces. Although this technology has many advantages, it still faces several challenges. One of the main challenges is that the level of accuracy has yet to reach its maximum potential. This research aims to improve facial recognition performance by applying the discrete cosine transform (DCT) and Gaussian mixture model (GMM), which are then trained with backward propagation of errors (backpropagation) and convolutional neural networks (CNN). The research results show low DCT and GMM feature extraction accuracy with backpropagation of 4.88%. However, the combination of DCT, GMM, and CNN feature extraction produces an accuracy of up to 98.2% and a training time of 360 seconds on the Olivetti Research Laboratory (ORL) dataset, an accuracy of 98.9% and a training time of 1210 seconds on the Yale dataset, and 100% accuracy and training time 1749 seconds on the Japanese female facial expression (JAFFE) dataset. This improvement is due to the combination of DCT, GMM, and CNN's ability to remove noise and study images accurately. This research is expected to significantly contribute to overcoming accuracy challenges and increasing the flexibility of facial recognition systems in various practical situations, as well as the potential to improve security and reliability in security and biometrics.

Corresponding Author:

Ivransa Zuhdi Pane
Department of Informatics, Faculty of Engineering and Informatics, Universitas Multimedia Nusantara
Scientia Boulevard Gading St., Tangerang Regency, Banten 15810, Indonesia
Email: ivransa.zuhdi@lecturer.umn.ac.id

1. INTRODUCTION

Facial recognition is an essential system in the digital world that is used to identify a person from digital images [1]. This system is applied as a solution in various fields such as security, biometrics, robotics, image search, and image and video indexing [2]–[5]. As technology develops, recognition provides significant advantages in various contexts [6]. One of its superior features is its solid security because this technology offers a safe and convenient way of authentication [7], reducing dependence on passwords and conventional access cards. Facial recognition technology is widely used in security to identify individuals and control access to restricted areas. For example, many airports use remote recognition systems to check passengers and ensure that they are people registered in the airline's database, which can help improve safety and efficiency. In biometrics, facial recognition is used in identification and verification systems, such as using the face to unlock a smartphone [8]–[10]. This security relies heavily on the system's ability to recognize faces accurately, thereby increasing efficiency and making it easier for users to open their smartphones without entering a password. Despite its many advantages, facial recognition technology faces several challenges that must be overcome. One of the main challenges is the level of accuracy that has yet to reach its maximum potential [11]. Factors such as lighting conditions, facial angles, and varying user demographics can influence the consistency and accuracy of identifying individuals [12], [13]. This raises concerns regarding misidentification and potential bias in this technology, so research and development must be conducted to ensure consistent and reliable performance across all user groups. Various methods have been proposed to improve face recognition accuracy, including feature extraction [14]. The main goal of feature extraction is to extract essential features from facial images to reduce noise during classification and increase accuracy [15].

To overcome this technological challenge, previous research has proposed several algorithms, including discrete cosine transform (DCT) [16], gray level co-occurrence matrix (GLCM) [17], and Gaussian mixture model (GMM) [18]. Previous research using GLCM and backward propagation of errors (backpropagation) showed 89% accuracy with a distance of 1 pixel [19]. The results of convolutional neural networks (CNN) research with the AlexNet architecture provide an accuracy of 98.5% [20]. The research results using DCT have an accuracy of 95% [21]. The research results using low-frequency DCT data for face and palm recognition produced an accuracy of 95%. These studies show significant levels of facial recognition accuracy, but there is still room for improvement, especially in dealing with facial variations involving changes in position and orientation.

A comprehensive literature review was conducted that carefully explores the methodology and theoretical foundations related to face recognition, with a particular focus on several vital approaches, including DCT [22], GMM [23], backpropagation, and CNN [24]. An in-depth analysis is conducted to understand the advantages, weaknesses, and latest developments in each method or theory discussed. Sources of information taken include previous scientific journals, academic theses, essential articles, and relevant digital resources. Source selection is based on strict criteria to ensure the validity and relevance of the information presented. The literature review also covers the latest literature in this field, ensuring that the knowledge presented remains relevant and up-to-date.

This research aims to overcome these obstacles by combining DCT and GMM feature accuracy techniques. This research will also evaluate the potential of artificial neural network (ANN) algorithms such as backpropagation and CNN, which have been proven effective in object recognition. These algorithms will be integrated with feature extraction to increase facial recognition accuracy, especially for facial variations that include facial position and orientation changes. This process will involve a careful training stage to ensure the integrated algorithms can recognize facial variations accurately, including facial position and orientation changes. However, it is essential to note that combining these algorithms can also increase the computational complexity of the system, which can affect processing time.

By combining the DCT and GMM feature extraction methods with the ANN algorithm, this research can significantly contribute to the development of facial recognition technology. The results of this research are expected to increase the accuracy of facial recognition significantly. Thus, this research opens up new opportunities for developing more sophisticated facial recognition technology and can provide more effective solutions in various contexts.

2. METHOD

The method used in this research includes stages, as detailed in Figure 1. This research differs from previous research [20] in that it does not remove the image background. Selection because it is required at the DCT feature extraction stage. DCT inherently focuses on and mitigates high-frequency data, effectively minimizing the influence of background components, so explicit background removal is unnecessary.

This research methodology approach uses feature extraction from images using DCT at low frequencies so that it has the potential to have more information that can be used to identify features in images [25]. Next, the GMM algorithm obtains facial image texture information, which can be used as an identification feature [26]. After feature extraction, facial data is recognized using ANN algorithms, namely backpropagation and CNN. Backpropagation algorithms learn quickly by computing synaptic updates using feedback connections to send error signals [27]. CNN was chosen as a classification method because of its compatibility with image data, where CNN can independently learn and extract features from an image [28]. In addition, the features extracted by DCT and GMM are combined to improve the accuracy of face recognition in the face of variations, such as changes in facial position and orientation. The results of the trained ANN model will be tested, and its accuracy will be calculated.

2.1. Image preprocessing

Figure 2 illustrates a sample of some of the datasets used. The facial dataset used in this research is the Olivetti Research Laboratory (ORL) dataset [29], which consists of 410 facial images from 41 different people, and each person has 10 facial images, an example of which can be seen in Figure 2(a). Each image is 80×70 pixels in size and is in JPG format. The second data set is the Yale dataset. This data has 165 facial images from 15 different people with different facial images; an example can be seen in Figure 2(b); each image is 320×243 pixels and is in GIF format. The third dataset is the Japanese female facial expression

(JAFFE) dataset, which contains 213 facial images with 10 Japanese female faces, an example of which can be seen in Figure 2(c). The size of each image is 256×256 pixels in TIFF format. These three datasets were chosen because they have a variety of subjects, so they have sufficient resources to train the model well. Before the data is used, it is processed to improve its suitability to the model and feature extraction. The image is converted from red, green, blue (RGB) color [29] to grayscale [30]. In this process, the intensity of the gray color is maintained so that the image still contains essential information.

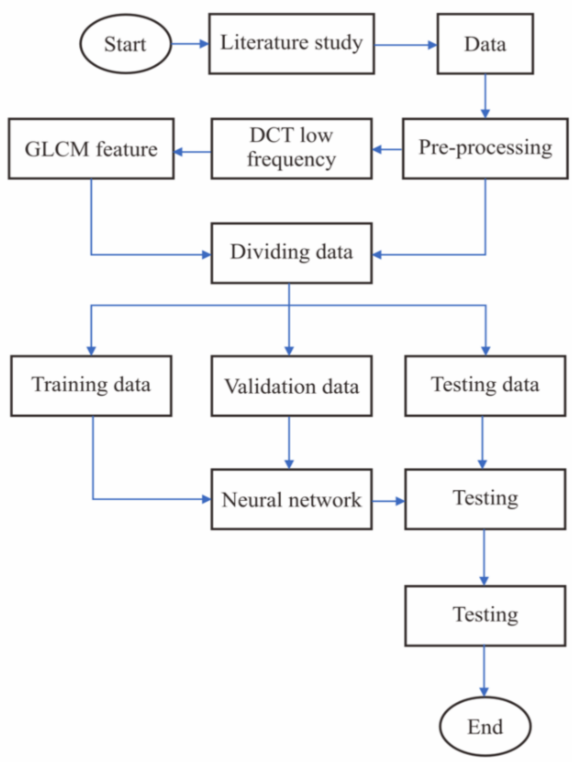
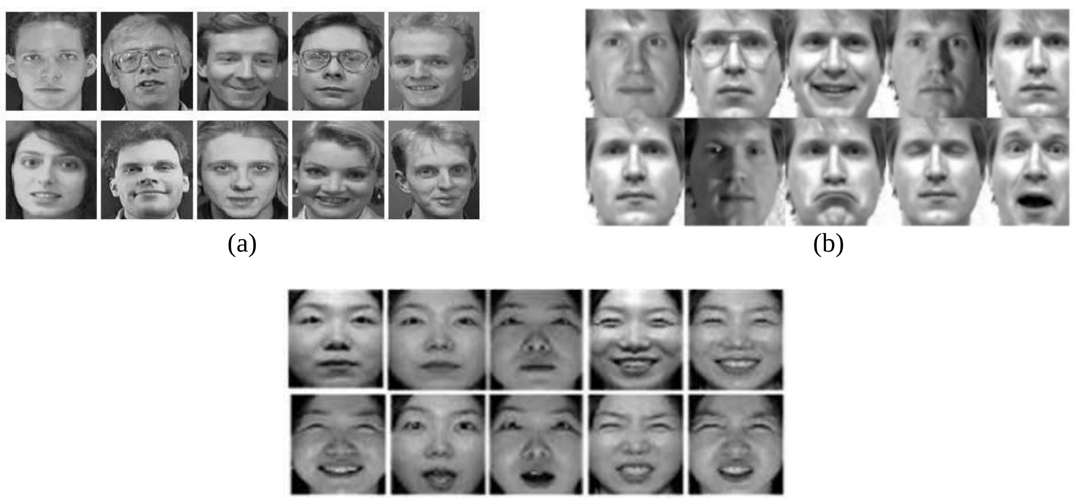


Figure 1. Research method



(a) (b)

(c)

Figure 2. Sample of (a) ORL dataset, (b) Yale dataset, and (c) JAFFE dataset

Method development requires that the data be processed first by converting RGB colors to grayscale [30], [31]. The brightness level represents the pixel intensity value in a grayscale image, measured on a grayscale from 0 (black) to 255 (white). The goal of this stage is to simplify the analysis as it reduces the complexity of the data from three color channels to one color channel and retains essential information about the brightness levels required for facial recognition.

2.2. Feature extraction

2.2.1. Low-frequency discrete cosine transform feature extraction

Low-frequency DCT [31] is a technique used in feature extraction, usually applied in signal processing tasks such as image and audio analysis; a visualization of the DCT coefficient matrix can be seen in Figure 3. It involves converting data into a new representation that combines cosine functions with varying frequencies. In this context, "low frequency" captures slow and significant data variations while eliminating fast fluctuations [32]. This is especially useful in tasks that emphasize basic structures or fundamental characteristics. To use low-frequency DCT for feature extraction, data, such as an image, is divided into blocks, and DCT is applied to each block. The resulting coefficients, which emphasize low-frequency information, are selected and combined into a feature vector. This compact representation preserves important features while reducing dimensions, making it useful for tasks such as image compression, pattern recognition, and data analysis. At this stage, the previously processed dataset is extracted using DCT to produce coefficients with three types of frequencies. The frequency that will be used is low because it is at this frequency that facial features are stored. Low coefficients, only 8×8 pixels in size, are selected again at the top left of the DCT matrix coefficient image [33].

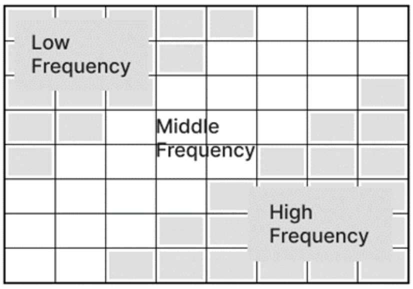


Figure 3. DCT coeficient matrix [34]

Besides, low frequencies are selected based on research [35]. This research tested various combinations of DCT low-frequency percentages on the detector features accuracy level.

$$F(u,v) = \frac{2}{N} C(u)C(v) \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} f(i,j) cos\left[\frac{(2i+1)u\pi}{2N}\right] cos\left[\frac{(2j+1)v\pi}{2N}\right] \quad (1)$$

Where $F(u,v)$ is the DCT value in frequency coordinates (u,v), f(i,j) is the pixel value in spatial coordinates (i,j), N is the DCT block size, and $C(u)$ is a cosine function related to frequency (u).

2.2.2. Gaussian matrix model features

GMM [36] is a probabilistic model that analyzes data with overlapping Gaussian components [37]. This model can be used for data clustering and can also be used to identify the underlying distribution of the data [38]. The basic formula for GMM is as (2) [39].

$$P(\text{X}|\Theta) = \sum_{k=1}^{K} \pi_k \cdot N(\text{X}|\mu_k, \sum k) \quad (2)$$

Where P(X|Θ) is probability of data X given parameter Θ in GMM, K is number of Gaussian components in GMM, π_k is the weight for each Gaussian component, which indicates the proportion or probability of occurrence of that component, and N(X|μ_k, $\sum k$) is the Gaussian density function for component k with mean μ_k and covariance matrix $\sum k$.

The main objective of GMM is to find the optimal Θ parameters that give the highest probability for the provided data. To determine the model parameters, GMM uses the expectation maximization (EM) algorithm, where in the expectation (E) stage, the expected value of each Gaussian component in the mixture is calculated, and in the maximization (M) stage, the model parameters are recalculated using that predicted value [40]. The iteration process continues until convergence occurs when the GMM parameters are stable, or the difference between successive iterations becomes minimal. The EM process in GMM involves two stages [41]: i) E stage is estimates the posterior probability of each Gaussian component (group) for each data point. The formula for calculating the posterior probability (responsibility) for each Gaussian component and ii) M stage is uses the posterior probabilities estimated in the E stage to update the GMM parameters, including the weights, mean, and covariance matrix.

2.3. Training data

2.3.1. Data splitting

After extraction, the data is divided into training, validation, and testing. Training data is used to train facial recognition algorithms so that they can understand the data for classification purposes. Validation data is used to evaluate model performance during the training process but is not used to train the model itself. Data testing is the final stage to test model performance on data that has never been seen before. Data distribution with a proportion of 60% training data, 20% validation data, and 20% testing data. The division is a strategic approach in machine learning and data science aimed at optimizing the model development process. This specific distribution reflects a balanced approach, ensuring sufficient data for training while allocating ample resources for both model tuning and unbiased evaluation [42], [43].

2.3.2. Data processing methods with discrete cosine transform and gaussian matrix model

The method combines the DCT transformation with the GMM model to produce a representation of data features with a focus on low frequencies using DCT and then applying the GMM model for further analysis and data classification. The process begins by changing the data into a one-dimensional (1D) form through a reshaping process, allowing further processing using the DCT transformation. Here is the pseudocode for the combination:

```
// Function to extract low-frequency components of DCT
Function performDCTLowFrequency(inputSignal, lowFrequencyThreshold):
// Apply DCT to the input signal
transformedSignal=DCTAlgorithm(inputSignal)
// Extract low-frequency components based on the specified threshold
lowFrequencyComponents=extractLowFrequency (transformedSignal, lowFrequencyThreshold)
return lowFrequencyComponents
// Function to initialize and train a Gaussian Mixture Model (GMM)
function trainGMM(data, numberOfComponents):
// Initialize a GMM with the specified number of components
gmm=InitializeGMM(numberOfComponents)
// Train the GMM on the provided data
gmm.fit(data)
return gmm
// Main processing function to combine DCT (low frequency) and GMM
function processSignal(inputSignal):
// Step 1: Apply DCT to the input signal and extract low-frequency components
// Define a threshold to identify low-frequency components
lowFrequencyThreshold=defineThreshold()
lowFrequencyDCTOutput=performDCTLowFrequency (inputSignal, lowFrequencyThreshold)
// Optional: Further feature extraction or selection from the low-frequency DCT output
features=extractFeatures(lowFrequencyDCTOutput)
// Step 2: Train a GMM on the low-frequency DCT
// Select the number of GMM components based on application-specific criteria
numberOfGMMComponents=
selectNumberOfComponents()
gmmModel=trainGMM (features, numberOfGMMComponents)
return gmmModel
```

After processing the data through a combination of DCT and GMM, the data is input to the ANN.

2.4. Facial recognition accuracy

2.4.1. Backpropagation

Backpropagation [44], a vital training technique in the context of ANN used in various applications, including facial recognition. Backpropagation was chosen in this research because of its critical ability to train ANN, especially for complex tasks such as face recognition. Backpropagation allows the network to update weights and biases based on prediction errors, enabling error correction and performance improvements over time. Backpropagation has 3 types of layers, namely i) input layer is a part consisting of units where the units start from 1 to n; ii) a hidden layer is a layer that consists of at least one layer, where each layer consists of several units; and iii) the output layer is each neuron unit in the input layer connected to all units in the hidden layer below it. Vice versa, every unit in the hidden layer is connected to all units in the output layer.

In Figure 4, the backpropagation architecture is presented, illustrating the structure and relationships among the three types of layers. These layers consist of the input layer, the hidden layer, and the output layer. Each layer plays a specific role in the backpropagation process by adjusting the weights based on the calculated error, enabling the model to learn more accurately.

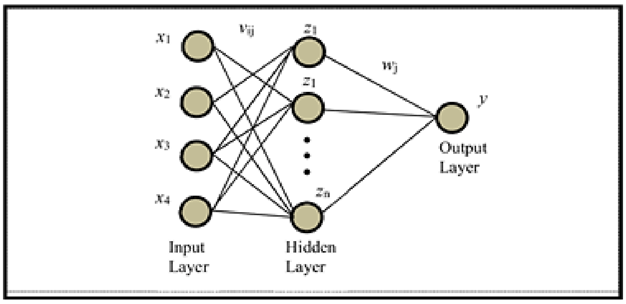


Figure 4. Architecture of backpropagation [45]

Backpropagation algorithms are key in improving network performance for complex tasks [46]. This algorithm works because a neural network can improve by understanding and correcting prediction errors during training. The training begins with initializing the weights and biases for each neuron in the network. Next, training data in the form of facial images or facial examples is presented to the network. This data flows through the network in a series of steps called feedforward, where each neuron performs calculations based on the weights and input signals it receives. At the end of the feedforward process, the network produces predictions of features or characteristics of the extracted faces. Next, a comparison is made between the predicted results and the correct labels, representing the person's identity in the image. The prediction error is measured as an error; the next step is returning (backpropagating) this error through the network. This involves calculating the error gradient against the weights and biases in each neuron.

The weights and bias are updated by subtracting the error gradient from the current weights and bias, and this process is repeated repeatedly for each training example in the dataset. The backpropagation algorithm tries to find a set of weights that optimizes the network's ability to recognize faces with high accuracy. This can take time and many factors, such as the learning rate, the number of neurons in the hidden layer, and the number of iterations required. This iterative process gradually improves the neural network's ability to recognize patterns and features on faces until it finally reaches a sufficient level of accuracy. Therefore, backpropagation is a critical foundation in developing sophisticated and efficient ANN in various applications, including facial recognition. With a deep understanding of these algorithms, developers and researchers can achieve optimal results in complex facial recognition tasks.

2.4.2. Convolutional neural network model

The CNN model is a deep learning architecture designed to tackle image and image processing tasks [47]. CNN was chosen in this research because of its excellent ability to handle image processing tasks, including face recognition. CNNs are specifically designed to extract hierarchical features from image data, enabling a deeper understanding of visual structures and patterns. CNN consists of several layers, including convolutional layers that hierarchically extract essential features from images, rectified linear unit (ReLU) activation layers to introduce non-linearity, pooling layers that reduce data dimensions, and fully connected layers that play a role in decision making. CNN is trained using machine learning algorithms like backpropagation to optimize performance in tasks such as image classification. With its ability to automatically extract features from image data, CNN has dominated many image processing applications. It is a cornerstone in developing technologies like object recognition, autonomous vehicles, and medical image analysis.

The LeNet model, also known as LeNet-5, was employed in this research. It represents one of the early milestones in developing CNNs [48], [49]. Designed by LeCun *et al.* [50] in 1998, LeNet was initially created for handwritten character recognition tasks. This model consists of convolutional layers that utilize filters to extract features from input images, followed by pooling layers that reduce data dimensions. Subsequently, two fully connected layers process these features and generate predictions. LeNet introduced the concept of convolutional layers, which has now become the core of modern CNN architectures. Although there are now more extensive and complex CNN architectures, LeNet remains a significant landmark in deep learning and image processing history, paving the way for further innovations in this field. In Figure 5, you can observe the visual representation of the 'model CNN LeNet'. This diagram illustrates the architecture of LeNet, showcasing the arrangement of convolutional layers, pooling layers, and fully connected layers.

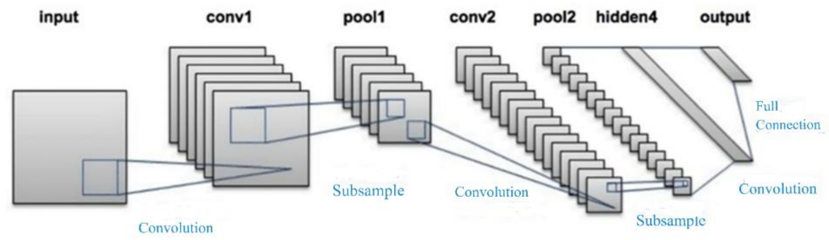


Figure 5. Model CNN LeNet [51]

2.4.3. Testing and evaluation

Testing and evaluating this facial recognition model uses several key metrics to measure model performance [52]. First, the accuracy and loss during training and testing will be calculated. Accuracy shows how far the model recognizes faces correctly [53], while loss measures how well the model minimizes errors [54]. Additionally, evaluations were performed using classification reports and confusion matrices to assess the model's face recognition performance [55], including accuracy, loss, recall, precision, and F1 score. Next, to measure the level of determination of the model, the correlation coefficient is used, which measures the closeness of the relationship between the independent variable (feature extraction data) and the dependent variable (face recognition accuracy level) to provide an understanding of the extent to which the model can differentiate between different and similar faces in the dataset [56].

3. RESULTS AND DISCUSSION

In this section, experimental analysis is carried out using the ORL dataset consisting of 410 images from 41 people. So, each face has ten images with different facial expressions and angles. All images were grayscaled before the DCT transformation. Extracting low-frequency data from DCT is carried out by taking the 8×8 image at the top left. After extracting the low-frequency DCT data, GMM is applied to each data, producing a GMM matrix for each data.

3.1. Implementation of feature extraction and backpropagation

At this stage, the backpropagation method is applied to train a facial recognition model using data extracted through DCT and GMM. This method plays a critical role in adjusting the model's weights to minimize error and improve performance over time. As shown in Table 1, the results indicate that the training accuracy achieved with backpropagation and DCT-GMM feature extraction remains relatively low, suggesting that further optimization or alternative approaches may be required.

Table 1. Result of data training trial with DCT and GMM feature extraction

Learning rate	Hidden node	Accuracy (%)	Epoch	Training time (s)
	50	4.88	402	1.79
0.01	150	1.22	320	3.89
	350	3.66	290	8.29
	50	3.66	1643	16.7
0.001	150	2.44	1270	12.56
	350	1.22	1096	13.73
	50	4.88	616	8.73
0.005	150	1.22	473	8.8
	350	2.44	424	7.95
	50	4.88	134	2.11
0.2	150	0.00	126	1.65
	350	3.66	123	1.26
	50	4.88	124	1.25
0.8	150	1.22	121	2.02
	350	4.88	120	3.02

From the results in Table 1, the highest accuracy value is found in the learning rate parameter, namely 0.8, and hidden nodes, namely 50, with an accuracy value of 4.88%. In this experiment, GMM extraction uses three n-component parameters and a random state 1. Based on these results, the smaller the learning rate parameter value, the faster the training data training time because, based on the trials carried out, the learning rate value of 0.001 has the longest training time was 16.7 seconds on 50 hidden nodes and with the same number of hidden nodes with a learning rate value of 0.8 the fastest training time was 1.25 seconds.

3.2. Implementation of feature extraction and convolutional neural network

Several studies have shown that a learning rate of 0.0001 yielded the best performance, and testing was conducted once the dataset was prepared. The testing process was carried out to evaluate the model's accuracy and robustness across different datasets. The results are presented in Table 2 for models using only CNN, and in Table 3 for models utilizing feature extraction methods with DCT, GMM, and CNN.

Table 2. Test results using CNN without feature extraction

Dataset	Accuracy (%)	Training time (s)
ORL	97.2	372.59
Yale	97.9	1330.48
JAFFE	99.2	2017.49

Table 3. Test results using DCT, GMM, and CNN

Dataset	Accuracy (%)	Training time (s)
ORL	98.2	360.59
Yale	98.9	1210.8
JAFFE	100	1749.49

From the results of the experiments that have been carried out, it can be seen in Table 3 that the addition of the DCT and GMM methods with a learning rate of 0.0001 and GMM parameters (n_component 10, random_state 300) produces the best accuracy compared to just using classification from CNN as shown in Table 2. This experiment produces an accuracy of 98.2% with a training time of 360 seconds on the ORL dataset, 98.9% accuracy with a training time of 1210 seconds on the Yale dataset, and 100% accuracy with a training time of 1749 seconds on the JAFFE dataset. Before looking for the coefficient of determination value, look for the correlation coefficient value or r with a result of 0.989 in the model created. Then, the value of the coefficient of determination is calculated by increasing the correlation coefficient to the power of two or r^2, which is then multiplied by 100% to obtain the percentage. The coefficient of determination value obtained was 97.9%, which means the model created has a strong correlation and can show that the efficiency of the method used influences the facial recognition value by 97.9%, and the remainder is influenced by other factors by 2.3%.

This research's increase in accuracy and testing time was caused by adding feature extraction methods, namely DCT and different classifiers (CNN). Applying low-frequency DCT helps eliminate noise at medium and high frequencies such as background, skin, and hair. Because this research focuses on low-frequency features such as the nose, mouth, and eyes. The role of CNN as a classification method also contributes to increasing accuracy because the convolution method used by CNN helps the model learn images so that the resulting model can classify images much more accurately. This research shows significant success compared to previous research with ORL data using the GLCM and backpropagation methods, which obtained accuracy results of 89%. This research combines three methods, namely DCT, GMM, and CNN, which achieved an accuracy level of 98.2% with a significant increase in accuracy due to the combination of feature extraction, especially DCT, which uses low frequencies. In terms of training time, this research reached 360.59 seconds, lower than previous research, which took 3.53 seconds.

3.3. Discussion

This research shows significant improvements compared to previous research [49], which used a combined GLCM and neural networks method with an accuracy of 89%, training time of 3.53 seconds, precision value of 0.85, recall value of 0.86, and f1 score of 85%. This research aims to fill the gaps in previous research by applying various feature extraction techniques, such as DCT and GMM, and utilizing CNN to improve facial recognition accuracy. The results of this study show variations in accuracy depending on parameters such as learning rate and number of hidden nodes. The highest accuracy value, although relatively low, is 4.88%, achieved with a combination of a learning rate of 0.8 and 50 hidden nodes using the backpropagation method. Despite the low accuracy, the relatively fast training time is a trade-off. Then, the implementation of DCT and GMM feature extraction, which was processed using a CNN, showed significant accuracy results, namely an accuracy of 98.2% and a training time of 360 seconds on the ORL dataset, an accuracy of 98.9% and a training time of 1210 seconds on the dataset. Yale, 100% accuracy and 1749 seconds training time on the JAFFE dataset. The training time is longer than the backpropagation method, especially on the JAFFE dataset, but the results show that the combination of DCT, GMM, and CNN provides superior performance. The training time for the backpropagation method is relatively fast, even though the accuracy is low. At the same time, the combination of DCT, GMM, and CNN requires longer training time, especially on the JAFFE dataset, but produces high accuracy. So, the results obtained from the combination of DCT, GMM, and CNN show that the benefits received from this method are higher than the increase in training time.

In conclusion, this research overcame the limitations of previous research by applying various feature extraction techniques and classifiers. Although training time can be a limiting factor, the results obtained from the combination of DCT, GMM, and CNN show a significant increase in accuracy. With a coefficient of determination of 97.9%, this research significantly contributes to understanding the factors influencing facial recognition results. It is hoped that this research can become a reference in the development of facial recognition technology in the future and can overcome several obstacles faced in previous research.

4. CONCLUSION

This research explores DCT and GMM feature extraction to improve facial recognition accuracy, combined with backpropagation and CNN training methods. The test results show that the backpropagation method with DCT and GMM feature extraction provides a limited accuracy of 4.88% but with the advantage of a relatively fast training time of 1.25 seconds. On the other hand, combining DCT, GMM, and CNN significantly improves the accuracy rate, reaching 98.2, 98.9, and 100% for the ORL, Yale, and JAFFE datasets, respectively. Although it requires more extended training, this combination provides superior results and shows excellent potential for developing facial recognition technology. Analysis of the coefficient of determination of 97.9% confirms that the efficiency of the method used greatly influences the facial recognition results, with other factors contributing around 2.3%. This conclusion highlights the strength of the developed model in handling variations in facial position and orientation and improves overall accuracy. Comparison with previous research shows a positive evolution in this technology, and the development of new methods, especially the combination of DCT, GMM, and CNN, opens the door to further advances in facial recognition. Therefore, this research makes a valuable contribution to the development of facial recognition technology, with wide application potential in various sectors, especially in improving the security and reliability of individual identification. Thus, this innovative combination opens up a new direction in improving facial recognition accuracy and positively impacts personal identification technology's security development.

ACKNOWLEDGEMENTS

Our appreciation goes to Kemendikbud-Ristek Republic of Indonesia with number of contract 073/E5/PG.02.00.PL/2023, for their financial support, and to Universitas Multimedia Nusantara for providing necessary resources.

REFERENCES

[1] S. M. Bah and F. Ming, “An improved face recognition algorithm and its application in attendance management system,” *Array*, vol. 5, 2020, doi: 10.1016/j.array.2019.100014.

[2] R. V. Petrescu, “Face recognition as a biometric application,” *SSRN Electronic Journal*, vol. 3, pp. 237–257, 2019, doi: 10.2139/ssrn.3417325.

[3] M. Taskiran, N. Kahraman, and C. E. Erdem, “Face recognition: Past, present and future (a review),” *Digital Signal Processing*, vol. 106, 2020, doi: 10.1016/j.dsp.2020.102809.

[4] S. Karnila, S. Irianto, and R. Kurniawan, “Face recognition using content based image retrieval for intelligent security,” *International Journal of Advanced Engineering Research and Science*, vol. 6, no. 1, pp. 91–98, 2019, doi: 10.22161/ijaers.6.1.13.

[5] Z. Wang, X. Zhang, P. Yu, W. Duan, D. Zhu, and N. Cao, “A new face recognition method for intelligent security,” *Applied Sciences*, vol. 10, no. 3, 2020, doi: 10.3390/app10030852.

[6] T. Saarikko, U. H. Westergren, and T. Blomquist, “Digital transformation: Five recommendations for the digitally conscious firm,” *Business Horizons*, vol. 63, no. 6, pp. 825–839, 2020, doi: 10.1016/j.bushor.2020.07.005.

[7] A. Anwar and A. Raychowdhury, “Masked face recognition for secure authentication,” *arXiv-Computer Science*, pp. 1–8, 2020.

[8] M. Smith and S. Miller, “The ethical application of biometric facial recognition technology,” *AI & SOCIETY*, vol. 37, no. 1, pp. 167–175, 2022, doi: 10.1007/s00146-021-01199-9.

[9] V. Wati, K. Kusrini, H. Al Fatta, and N. Kapoor, “Security of facial biometric authentication for attendance system,” *Multimedia Tools and Applications*, vol. 80, no. 15, pp. 23625–23646, 2021, doi: 10.1007/s11042-020-10246-4.

[10] K. H. Teoh, R. C. Ismail, S. Z. M. Naziri, R. Hussin, M. N. M. Isa, and M. Basir, “Face recognition and identification using deep learning approach,” *Journal of Physics: Conference Series*, vol. 1755, no. 1, 2021, doi: 10.1088/1742-6596/1755/1/012006.

[11] Y. Kortli, M. Jridi, A. Al Falou, and M. Atri, “Face recognition systems: a survey,” *Sensors*, vol. 20, no. 2, 2020, doi: 10.3390/s20020342.

[12] C. M. Cook, J. J. Howard, Y. B. Sirotin, J. L. Tipton, and A. R. Vemury, “Demographic effects in facial recognition and their dependence on image acquisition: an evaluation of eleven commercial systems,” *IEEE Transactions on Biometrics, Behavior, and Identity Science*, vol. 1, no. 1, pp. 32–41, 2019, doi: 10.1109/TBIOM.2019.2897801.

[13] K. Raju, B. C. Rao, K. Saikumar, and N. L. Pratap, “An optimal hybrid solution to local and global facial recognition through machine learning,” in *Intelligent Systems Reference Library*, Springer, Cham, 2022, pp. 203–226, doi: 10.1007/978-3-030-76653-5_11.

[14] L. C. Ngugi, M. Abelwahab, and M. Abo-Zahhad, “Recent advances in image processing techniques for automated leaf pest and disease recognition – a review,” *Information Processing in Agriculture*, vol. 8, no. 1, pp. 27–51, 2021.

[15] K. Adnan and R. Akbar, “An analytical study of information extraction from unstructured and multidimensional big data,” *Journal*

of Big Data, vol. 6, no. 1, 2019, doi: 10.1186/s40537-019-0254-8.
[16] V. P. Vishwakarma and T. Goel, "An efficient hybrid DWT-fuzzy filter in DCT domain based illumination normalization for face recognition," *Multimedia Tools and Applications*, vol. 78, no. 11, pp. 15213–15233, 2019, doi: 10.1007/s11042-018-6837-0.
[17] M. M. Oghaz, M. A. Maarof, M. F. Rohani, A. Zainal, and S. Z. M. Shaid, "An optimized skin texture model using gray-level co-occurrence matrix," *Neural Computing and Applications*, vol. 31, no. 6, pp. 1835–1853, 2019, doi: 10.1007/s00521-017-3164-8.
[18] J. Xie *et al.*, "DS-UI: Dual-supervised mixture of gaussian mixture models for uncertainty inference in image recognition," *IEEE Transactions on Image Processing*, vol. 30, pp. 9208–9219, 2021, doi: 10.1109/TIP.2021.3123555.
[19] Vera, A. Kusnadi, I. Z. Pane, M. V. Overbeek, and S. G. Prasetya, "Face recognition accuracy improving using gray level co-occurrence matrix selection feature algorithm," in *2023 International Conference on Smart Computing and Application (ICSCA)*, 2023, pp. 1–6, doi: 10.1109/ICSCA57840.2023.10087414.
[20] S. Khan, M. H. Javed, E. Ahmed, S. A. A. Shah, and S. U. Ali, "Facial recognition using convolutional neural networks and implementation on smart glasses," in *2019 International Conference on Information Science and Communication Technology (ICISCT)*, 2019, pp. 1–6, doi: 10.1109/CISCT.2019.8777442.
[21] S. Hsia, S. Wang, and C. Chen, "Fast search real-time face recognition based on DCT coefficients distribution," *IET Image Processing*, vol. 14, no. 3, pp. 570–575, 2020, doi: 10.1049/iet-ipr.2018.6175.
[22] C. Scribano, G. Franchini, M. Prato, and M. Bertogna, "DCT-former: efficient self-attention with discrete cosine transform," *Journal of Scientific Computing*, vol. 94, no. 3, 2023, doi: 10.1007/s10915-023-02125-5.
[23] A. Singhal, P. Singh, B. Lall, and S. D. Joshi, "Modeling and prediction of COVID-19 pandemic using Gaussian mixture model," *Chaos, Solitons & Fractals*, vol. 138, 2020, doi: 10.1016/j.chaos.2020.110023.
[24] L. Chen, S. Li, Q. Bai, J. Yang, S. Jiang, and Y. Miao, "Review of image classification algorithms based on convolutional neural networks," *Remote Sensing*, vol. 13, no. 22, 2021, doi: 10.3390/rs13224712.
[25] S. P. Jaiprakash, M. B. Desai, C. S. Prakash, V. H. Mistry, and K. L. Radadiya, "Low dimensional DCT and DWT feature based model for detection of image splicing and copy-move forgery," *Multimedia Tools and Applications*, vol. 79, no. 39–40, pp. 29977–30005, 2020, doi: 10.1007/s11042-020-09415-2.
[26] S. Misra and R. H. Laskar, "Integrated features and GMM based hand detector applied to character recognition system under practical conditions," *Multimedia Tools and Applications*, vol. 78, no. 24, pp. 34927–34961, 2019, doi: 10.1007/s11042-019-08105-y.
[27] T. P. Lillicrap, A. Santoro, L. Marris, C. J. Akerman, and G. Hinton, "Backpropagation and the brain," *Nature Reviews Neuroscience*, vol. 21, no. 6, pp. 335–346, 2020, doi: 10.1038/s41583-020-0277-3.
[28] M. A. Hossain and M. S. A. Sajib, "Classification of image using convolutional neural network (CNN)," *Global Journal of Computer Science and Technology*, vol. 19, no. 2, pp. 13–18, 2019, doi: 10.34257/gjcstdvol19is2pg13.
[29] M. Tavares, "The ORL database for training and testing," *Kaggle*. 2020. Accessed: Nov. 04, 2024. [Online]. Available: https://www.kaggle.com/datasets/tavarez/the-orl-database-for-training-and-testing
[30] W. Wang, X. Wu, X. Yuan, and Z. Gao, "An experiment-based review of low-light image enhancement methods," *IEEE Access*, vol. 8, pp. 87884–87917, 2020, doi: 10.1109/ACCESS.2020.2992749.
[31] W. A. Mustafa *et al.*, "Image enhancement based on discrete cosine transforms (DCT) and discrete wavelet transform (DWT): a review," *IOP Conference Series: Materials Science and Engineering*, vol. 557, no. 1, 2019, doi: 10.1088/1757-899X/557/1/012027.
[32] M. D. Deepak, P. Karthik, S. S. Kumar, and N. A. Deepak, "Comparative study of feature extraction using different transform techniques in frequency domain," in *International Conference on Automation, Signal Processing, Instrumentation and Control*, 2021, pp. 2835–2846, doi: 10.1007/978-981-15-8221-9_265.
[33] W. M. Alaluosi, "Recognition of human facial expressions using DCT-DWT and artificial neural network," *Iraqi Journal of Science*, vol. 62, no. 6, pp. 2090–2098, 2021, doi: 10.24996/ijs.2021.62.6.34.
[34] E. G. Wahyuni, L. M. F. Fauzan, F. Abriyani, N. F. Muchlis, and M. Ulfa, "Rainfall prediction with backpropagation method," *Journal of Physics: Conference Series*, vol. 983, 2018, doi: 10.1088/1742-6596/983/1/012059.
[35] A. Kusnadi, L. Nathania, I. Z. Pane, and M. V. Overbeek, "Face detection keypoints using Dct And Clahe," *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, vol. 12, no. 11, pp. 4365–4372, 2021, doi: 10.17762/turcomat.v12i11.6568.
[36] H. Wan, H. Wang, B. Scotney, and J. Liu, "A novel gaussian mixture model for classification," in *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*, 2019, pp. 3298–3303, doi: 10.1109/SMC.2019.8914215.
[37] L. Jiao, T. Denœux, Z. Liu, and Q. Pan, "EGMM: An evidential version of the gaussian mixture model for clustering," *Applied Soft Computing*, vol. 129, 2022, doi: 10.1016/j.asoc.2022.109619.
[38] E. Patel and D. S. Kushwaha, "Clustering cloud workloads: k-means vs gaussian mixture model," *Procedia Computer Science*, vol. 171, pp. 158–167, 2020, doi: 10.1016/j.procs.2020.04.017.
[39] S. Cao, Z. Hu, X. Luo, and H. Wang, "Research on fault diagnosis technology of centrifugal pump blade crack based on PCA and GMM," *Measurement*, vol. 173, 2021, doi: 10.1016/j.measurement.2020.108558.
[40] J. Qiao *et al.*, "Data on MRI brain lesion segmentation using k-means and gaussian mixture model-expectation maximization," *Data in Brief*, vol. 27, 2019, doi: 10.1016/j.dib.2019.104628.
[41] W. Jannah and D. R. S. Saputro, "Parameter estimation of gaussian mixture models (GMM) with expectation maximization (EM) algorithm," in *International Conference of Mathematics and Mathematics Education (I-CMME) 2021*, 2022, doi: 10.1063/5.0117119.
[42] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. Cambridge, Massachusetts: MIT Press, 2016.
[43] M. Kuhn and K. Johnson, *Applied predictive modeling*. New York: Springer, 2013, doi: 10.1007/978-1-4614-6849-3.
[44] T. P. Lillicrap and A. Santoro, "Backpropagation through time and the brain," *Current Opinion in Neurobiology*, vol. 55, pp. 82–89, 2019, doi: 10.1016/j.conb.2019.01.011.
[45] S. Anam, "Rainfall prediction using backpropagation algorithm optimized by Broyden-Fletcher-Goldfarb-Shanno algorithm," *IOP Conference Series: Materials Science and Engineering*, vol. 567, no. 1, 2019, doi: 10.1088/1757-899X/567/1/012008.
[46] P. R. Vlachas *et al.*, "Backpropagation algorithms and reservoir computing in recurrent neural networks for the forecasting of complex spatiotemporal dynamics," *Neural Networks*, vol. 126, pp. 191–217, 2020, doi: 10.1016/j.neunet.2020.02.016.
[47] L. Alzubaidi *et al.*, "Review of deep learning: concepts, CNN architectures, challenges, applications, future directions," *Journal of Big Data*, vol. 8, no. 1, 2021, doi: 10.1186/s40537-021-00444-8.
[48] A. Khan, A. Sohail, U. Zahoora, and A. S. Qureshi, "A survey of the recent architectures of deep convolutional neural networks," *Artificial Intelligence Review*, vol. 53, no. 8, pp. 5455–5516, 2020, doi: 10.1007/s10462-020-09825-6.
[49] M. Krichen, "Convolutional neural networks: A survey," *Computers*, vol. 12, no. 8, 2023, doi: 10.3390/computers12080151.
[50] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998, doi: 10.1109/5.726791.
[51] T. Saeed, M. Sufian, M. Ali, and A. U. Rehman, "Convolutional neural network based career recommender system for Pakistani engineering students," in *2021 International Conference on Innovative Computing (ICIC)*, 2021, pp. 1–10, doi:

10.1109/ICIC53490.2021.9715788.
[52] Z. Wang, B. Huang, G. Wang, P. Yi, and K. Jiang, "Masked face recognition dataset and application," *IEEE Transactions on Biometrics, Behavior, and Identity Science*, vol. 5, no. 2, pp. 298–304, 2023, doi: 10.1109/TBIOM.2023.3242085.
[53] S. Almabdy and L. Elrefaei, "Deep convolutional neural network-based approaches for face recognition," *Applied Sciences*, vol. 9, no. 20, 2019, doi: 10.3390/app9204397.
[54] A. M. Ayalew, A. O. Salau, B. T. Abeje, and B. Enyew, "Detection and classification of COVID-19 disease from X-ray images using convolutional neural networks and histogram of oriented gradients," *Biomedical Signal Processing and Control*, vol. 74, 2022, doi: 10.1016/j.bspc.2022.103530.
[55] M. Billah, X. Wang, J. Yu, and Y. Jiang, "Real-time goat face recognition using convolutional neural network," *Computers and Electronics in Agriculture*, vol. 194, 2022, doi: 10.1016/j.compag.2022.106730.
[56] F. M. Talaat and S. A. Gamel, "Predicting the impact of no. of authors on no. of citations of research publications based on neural networks," *Journal of Ambient Intelligence and Humanized Computing*, vol. 14, no. 7, pp. 8499–8508, 2023, doi: 10.1007/s12652-022-03882-1.

BIOGRAPHIES OF AUTHORS

Cand. Dr. Adhi Kusnadi, S.T., M.Si. completed his undergraduate degree (S1) at Sriwijaya University in Palembang in 1996. He continued his education at IPB Bogor, majoring in computer science, and graduated with a master's degree (S2) in 2008. Currently, he is pursuing his doctoral studies (S3) in computer science at IPB. He works as a permanent lecturer at Universitas Multimedia Nusantara in the Department of Informatics. He is actively involved in various professional associations, including the Indonesian Association of Lecturers and the National Education Commission. He can be contacted at email: adhi.kusnadi@umn.ac.id.

Dr. Ivransa Zuhdi Pane, B.Eng., M.Eng., completed his undergraduate (S1) and master's (S2) degrees at Kyushu Institute of Technology, Japan, in the field of computer science and systems engineering in 1992 and 1994, respectively. He obtained his doctoral degree (S3) from Kyushu University, Japan, in the field of electronics in 2010. Currently, he works as a senior expert engineer at the National Research and Innovation Agency and as a lecturer in the Department of Informatics at Universitas Multimedia Nusantara. He is actively engaged in research and development activities in the fields of information system engineering and expert systems. He can be contacted at email: ivransa.zuhdi@lecturer.umn.ac.id.

Fenina Adline Twince Tobing, S.Kom., M.Kom., graduated with a master's degree from the University of North Sumatra (USU) in 2015. Currently, she is an active informatics lecturer at Universitas Multimedia Nusantara (UMN) in Jakarta, where she also serves as the research coordinator for the Faculty of Engineering and Informatics. She is a member of several professional associations such as Institute of Electrical and Electronics Engineers (IEEE), Indonesian Association of Higher Education in Computer Science (APTIKOM), and Association for Synergy of Service and Empowerment in Indonesia (ASPPI). In addition to her involvement in professional organizations, she is also active in social and community organizations, including the Dharma Wanita Persatuan (DWP) at the Jakarta Class II Correctional Institution. She can be contacted at email: fenina.tobing@umn.ac.id.