IAES International Journal of Artificial Intelligence (IJ-AI)
Vol. 14, No. 5, October 2025, pp. 3576~3587
ISSN: 2252-8938, DOI: 10.11591/ijai.v14.i5.pp3576-3587

Pre-trained convolutional neural network-based algorithms: application for recognizing the age category

Yuni Yamasari[1], Lusiana Anggraini[1], Anita Qoiriah[1], Ricky Eka Putra[1], Hapsari Peni Agustin Tjahyaningtijas[2], Tohari Ahmad[3]
[1]Department of Informatics, Faculty of Engineering, Universitas Negeri Surabaya, Surabaya, Indonesia
[2]Department of Electrical Engineering, Faculty of Engineering, Universitas Negeri Surabaya, Surabaya, Indonesia
[3]Department of Informatics, Faculty of Electrical Engineering and Intelligent Informatics, Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia

Article Info

Article history:

Received Aug 28, 2024
Revised Jul 14, 2025
Accepted Aug 06, 2025

Keywords:

Age category
Artificial neural network
Computational learning
Convolutional neural network
K-nearest neighbor
VGG-16

ABSTRACT

Cybercrime is a major issue in the current digital era, with one of its branches-cyber pornography-notably affecting Indonesia. Various efforts have been made to suppress or prevent this problem. One alternative solution involves using technological advances to recognize age ranges based on facial recognition. This age range recognition can be implemented to prevent users from accessing content that is not appropriate for their age. An optimal age-range recognition system is essential for this purpose. However, limited research has focused on this domain. Therefore, our research aimed to develop the best possible system. The proposed method applies a trained convolutional neural network (CNN) as a feature extractor to the artificial neural network (ANN) and k-nearest neighbor (K-NN) methods for age recognition based on facial images. By incorporating computational learning techniques, the system's performance is significantly enhanced, leveraging advanced algorithms to improve accuracy. The test results show that the performance of the pre-trained CNN-based ANN model is superior. This is indicated by the model's accuracy and F1-score, which were 11% and 0.11 higher, than the pre-trained CNN-based K-NN model. The error rate of the pre-trained CNN-based ANN model was also reduced by 0.11.

Corresponding Author:

Yuni Yamasari
Department of Informatics, Faculty of Engineering, Universitas Negeri Surabaya
Surabaya, Indonesia
Email: yuniyamasari@unesa.ac.id

1. INTRODUCTION

Currently, social media is an activity that we cannot separate from our daily lives [1]. However, security in social media is often ignored by the public. In fact, this security can prevent crimes on social media (cyber security) and can protect computer systems from potential threats or illegal access [2]. Cyber security can also reduce the possibility of threats entering a computer system [3]. Moreover, security tools, policies, and concepts related to security in social media can be used to protect organizational and user assets [4]. Therefore, cyber security must always be improved to prevent computer system crimes (cybercrime).

One of the cyber-crimes that often occurs in Indonesia is pornography crime. In recent years, there have been many cases of underage children accessing adult content, as reported by the Indonesian news site jawapos.com, Indonesia is the country that accesses the most pornographic content after India, especially in

2015 and 2016, 74% of the users are young people. To minimize this, an application is needed to filter adult content users such as pornographic sites, gambling sites, and black market sites.

On the other hand, technology is also experiencing very rapid development along with the times. Almost all aspects of life today use sophisticated and newest technology. It is hoped that the use of technology will be able to help overcome various problems currently being faced. As we know, one form of technological development used to solve a problem is artificial intelligence. Artificial intelligence is a field of science that makes computers imitate human habits [5]. In other words, it can also be understood as a part of computer science that focuses on machines with intelligent capabilities that can interact or act like humans [6]. Humans increasingly develop based on the learning and experiences they have gone through. Likewise, artificial intelligence can also learn like humans, and the more it learns, the better its abilities will be. In contrast to humans, artificial intelligence can learn, find patterns, and record them much more quickly and efficiently [7]. Learning in artificial intelligence is also referred to as a learning process. Within the field of artificial intelligence, there is a more specific and sophisticated approach, namely deep learning. Deep learning is a method that utilizes algorithms based on mathematical principles designed to mimic the workings of the human brain [8], [9]. Deep learning is used for various tasks [10] such as probability or event prediction [11], object recognition [12]–[14], and disease diagnosis [15]–[17].

One of the uses of deep learning is in image processing. Image processing systems are developed to assist humans in recognizing or categorizing objects more effectively and efficiently [18]. This field of image processing is also widely applied in the cyber security domain, including cyber security uses sensing with intelligent systems [19], network intrusion detection systems [20], and intelligent malware detection [21]. In our research, the technology of image processing is applied to solve cybercrime in Indonesia. Image processing is focused on facial images to recognize age categories.

The rapid growth of image processing research has encouraged the exploration of methods to generate a quick-, accurate, and capable model of processing large amounts of data [22]. There are several popular algorithms that can be applied to solve various problems, including k-nearest neighbor (K-NN) [23]–[27], artificial neural network (ANN) [28]–[31], and support vector machine (SVM) [32]–[34]. Several studies have explored K-NN for image processing in solving various problems because of the advantages possessed by K-NN, for examples: the parameters in K-NN are flexible because the input data can be categorical and continuous [35], and this algorithm is simple and fast [36].

Therefore, this research carries out an innovative integration of a trained convolutional neural network (CNN) model (VGG16) with ANN and K-NN algorithms to improve the accuracy of age recognition aimed at cyber security. This method specifically aims to prevent minors from accessing inappropriate content on social media and addresses an important problem in Indonesia, where cyber pornography is common. The application of artificial intelligence techniques in this context offers tailored and impactful solutions to these contemporary problems. In addition, this study also provides a comparative analysis of the performance of ANN and K-NN together with VGG16 in age recognition. This research not only improves understanding of algorithms in real-time cybersecurity applications but also strives for high accuracy, thereby increasing the reliability of the overall age recognition system. By incorporating various computational learning techniques, including traditional machine learning and deep learning approaches, the system can adapt and enhance its predictive capabilities more effectively.

2. MATERIAL AND METHOD

This chapter describes the materials required for age category detection. Then, the proposed architecture is discussed in this chapter as well. The materials include datasets used, and preprocessing techniques essential for accurate age prediction. Additionally, the architecture section outlines the model design, layers, and the rationale behind the chosen methods.

2.1. Data preparation

The dataset used in this research is a public dataset accessible on Kaggle. Before applying the method to this dataset, preprocessing is necessary. The initial step involves image enhancement to remove noise. Then, normalization is performed by scaling the image pixels to a smaller range, such as 0 to 1, to help increase the convergence speed during model training [37]. The aim of image normalization is not to reduce the image resolution but rather to standardize the pixel values, which aids in improving recognition accuracy. This normalization process involves image scaling.

Next, resizing or resampling is conducted to ensure all images have the same dimensions, as they often come in various sizes. Following this, data augmentation techniques are applied. These techniques create data variations by manipulating the original images through rotating, translating, flipping, zooming, and other transformations. Data augmentation helps increase the amount of training data, thereby reducing overfitting and improving model generalization. Subsequently, the data is split into training, validation, and testing sets. Training data is used to train the model, validation data is used for tuning, and testing data is used to measure the model's final performance.

Additionally, the research applies the under-sampling technique during the preprocessing stage. This technique is used to reduce the number of examples from the majority class to balance the dataset, particularly useful for imbalanced datasets in classification tasks with sparse minority classes. By randomly under-sampling the majority class data, the dataset becomes more balanced, leading to a model that better recognizes minority classes. The data distribution is found to be imbalanced, so a random under-sample is conducted to achieve a balanced dataset.

2.2. The proposed architecture

After preparing the dataset, this research implements techniques arranged in the proposed architecture, as shown in Figure 1. The two main methods used for age recognition are ANN and K-NN. Before implementing these methods, a pre-trained CNN model with the VGG-16 architecture is applied which is the most popular architecture [38] and an effective deep CNN model. VGG-16 consists of a total of 16 weight layers, including 13 convolutional layers and 3 fully connected (FC) layers. The primary advantage of this model is its homogeneous architecture, consistently using 3×3 convolutional filters and 2×2 max-pooling layers throughout the network. This consistency ensures systematic and detailed feature extraction from the input images.

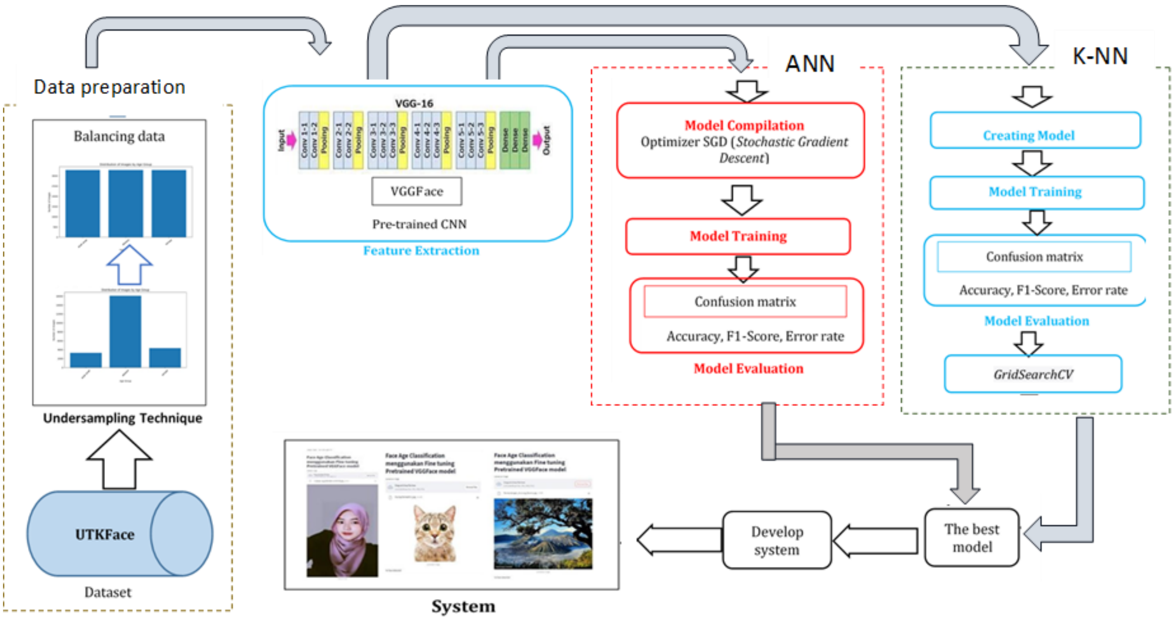


Figure 1. The proposed architecture of the identifying system of the age categories

Initially, VGG-16 is trained on the VGGFace dataset, which contains a large collection of facial images. The knowledge gained from this training is then transferred to the pre-trained VGG-16 model, which is applied to the University of Tennessee, Knoxville Face (UTKFace) dataset. This pre-trained CNN leverages the convolutional layers to extract high-dimensional feature representations from the facial images, capturing essential characteristics such as edges, textures, and complex facial patterns. This feature extraction process is crucial for the subsequent classification tasks performed by the ANN and K-NN methods in the research.

For the first approach, ANN, the extracted features are fed into the ANN for classification. The model is compiled using the stochastic gradient descent (SGD) optimizer, which is instrumental in training the network by adjusting the weights to improve performance. SGD is chosen for its efficiency and effectiveness in handling large datasets. The loss function used is categorical cross-entropy, suitable for multi-class classification tasks. To train the model, the image data generator from Keras is employed, performing real-time data augmentation by applying random transformations to the images. This augmentation helps make the model more general and prevents overfitting.

Training the ANN involves using training data as input to find the optimal model configuration. The architecture of the ANN is determined experimentally to identify the best composition of layers and neurons. The model's training involves processing the input data through the network, where each neuron's output is determined by an activation function. This function monitors if the accumulated signal reinforcement has exceeded a certain threshold, causing the neuron to "fire" and produce an output signal. The ANN's weights are iteratively adjusted based on the error between the predicted and actual labels, guided by the gradients computed through the loss function. This process is repeated for multiple epochs until the model achieves satisfactory performance. The equation used is as follows:

$$y = f(\sum_{i=1}^{n} x_i w_i) \tag{1}$$

The correlation between the three components in (1) is as follows:
- A signal $x$in the form of a n --dimensional vector $(x_1, x_2, \dots, x_n)y$will be strengthened by synapse $w(w_1, w_2, \dots, w_n)y$.
- The accumulation of these reinforcements will undergo transformation by the activation function f. This f function will monitor if the accumulated signal reinforcement has exceeded a certain limit, then the neuron cells which were originally in the "0" condition, will issue a "1" signal.
- Based on the output value (y), a neuron can be in two states: "0" or "1". A neuron is said to be in firing condition when it produces an output is "1".

In the second approach, the model is built using the K-NN method with a specified number of neighbors, denoted by the parameter k. The K-NN model is trained using features extracted from the training images. This training involves using the training data to classify objects based on the proximity of the learning data to the new objects. The objective of K-NN is to classify new objects based on their attributes and the training data samples closest to them [39]. During the learning stage, K-NN stores feature vectors and their corresponding classifications from the training data [40]. The classification process in K-NN relies on the number of nearest neighbors to determine the class, with distance calculations using the Euclidean distance.

To evaluate the models in this research, a confusion matrix is used to measure accuracy and F1-score. To find the best-performing model, GridSearchCV is applied. This method performs a systematic search on combinations of various hyperparameters to optimize the K-NN model based on accuracy values. The implementation of GridSearchCV involves creating grid parameters to optimize, with a list of values for each parameter. A GridSearchCV object is created, with the K-NN model as the first parameter and the parameter grid as the second parameter. Cross-validation (CV) is set to a specific fold, and the scoring parameter is set to accuracy for model assessment. The grid is trained using the feature data and target labels to generate simulation results for accuracy and F1-score. After determining the best model, it is implemented in the age category recognition system.

3. RESULTS AND DISCUSSION

The chapter begins by detailing the dataset utilized in this research, followed by the application of techniques aimed at optimizing the modeling process. Subsequently, simulations are conducted to evaluate the model, and experimental results are obtained and analyzed. Finally, the discussion concludes with the implementation of the model into the system.

3.1. Data description

This research utilizes a public dataset sourced from the website address: https://susanqq.github.io/UTKFace. Known as the UTKFace dataset, it comprises over 20,000 facial images spanning a wide age range from 0 to 116 years. To facilitate the application of machine learning methods, these images are categorized into three age groups: *anak-anak* or children (0-11 years), *remaja* or teenagers (12-25 years), and *dewasa* or adults (25 years and over). Ensuring accurate labeling of each image is crucial before proceeding to subsequent processes. However, the original dataset initially suffered from an imbalanced distribution among these age groups. To address this issue, the research employs an undersampling technique, equalizing the number of samples in each category. This ensures that each age group is equally represented in the dataset. Additionally, Figure 2 illustrates sample images from each age category: Figure 2(a) *dewasa*, Figure 2(b) *remaja*, and Figure 2(c) *anak-anak.*

3.2. Pre-processing stage

After preparing the dataset, Google Colab is integrated with Google Drive to facilitate access to files within the Colab environment. The dataset is labeled, and global variables are defined to store important information throughout the program. Specifically, the "age_group" variable contains strings representing three classes: "*anak-anak*", "*remaja*", and "*dewasa*", which categorize the age groups used in the study. A CSV file location string is set as a global variable, pointing to a modified file that balances image labels across different classes. This CSV file contains 23,708 image data entries, structured to simplify labeling and conserve storage resources.

Next, data inspection is conducted to analyze the dataset's characteristics, format, structure, and distribution. To address the initial imbalance in data distribution, under sampling is employed to reduce the majority class and equalize the number of examples across all classes. Under sampling is a vital technique in handling imbalanced datasets, particularly in classification tasks with sparse minority classes. The dataset achieves better balance by randomly reducing the majority class, enhancing the model's ability to recognize minority classes. Specifically, the dataset is balanced to include 3,283 images for each class.

Finally, TensorFlow is utilized to load and preprocess the data using a data generator. The CSV-formatted data is loaded into memory as images, enabling efficient training of machine learning models based on image data. The data loading process emphasizes data augmentation and data generator settings for the training and validation phases. Augmentation techniques such as rotation angle adjustment, horizontal and vertical shifting, and image stretching enrich the training data by introducing variations. Additionally, the method for filling shifted pixels closest to the original image is regulated. Image sizes are standardized to 200×200 pixels, and a batch size 64 is set. Rescaling ensures that image data ranges from 0 to 1, simplifying computational processes during training.

(a) (b) (c)

Figure 2. Example facial images from each age category in the UTKFace dataset: (a) *dewasa*, (b) *remaja*, and (c) *anak-anak*

3.3. Modeling of age categories using VGG-16 ANN

Modeling begins with determining the architecture. This research uses the VGGFace model, a pre-trained VGG16 variant tuned for facial recognition tasks, as the base model. Additional layers are added to form a classifier, consisting of three Dense layers with 1024, 512, and 128 units, each followed by a dropout layer with a 0.5 dropout rate to prevent overfitting. The final dense layer has three units, each representing one of the three age categories, using a SoftMax activation function to produce interpretable class probabilities.

The model is created by combining the base model and the classifier using TensorFlow's model class. If a saved model weights configuration is available, it is loaded using model.load_weights, enabling transfer learning or fine-tuning. The resulting variables—base model, classifier, and complete model—are configured for further training or evaluation. Input images are resized to 200×200 pixels, and the model is trained for 10 epochs. Next, the model is compiled using the SGD optimizer with a learning rate of 0.01 and momentum of 0.9, chosen for its efficiency with large datasets. The loss function is categorical cross-entropy, suitable for multi-class classification tasks. Accuracy and F1-scores are used as metrics to monitor performance, especially useful for imbalanced classes.

To train the model, Keras' image data generator is used for real-time data augmentation, applying transformations such as 30-degree rotation, 0.1 widths and height shift, 0.2 shear, 0.2 zooms, and horizontal flip, helping the model generalize better and prevent overfitting. The results of the model's training and tuning process are summarized in Figure 3, which presents both qualitative and quantitative outputs. Figure 3(a) displays sample predictions comparing true and predicted labels across different age categories, while Figure 3(b) shows the variation in loss levels during the parameter tuning iterations. The model is trained for 10 epochs with a batch size of 64, as shown in Figure 3(a). A tuning test is conducted 70 times, with varying parameters for optimal results. The loss levels during tuning are shown in Figure 3(b). The results of 10 experiments are presented in Table 1. For model evaluation, a confusion matrix is used to measure accuracy and F1-score. Based on Table 1, the lowest accuracy of 53.03% occurs in the 5th tuning, while fine-tuning at iterations 68, 69, and 70 yields consistent results, stopping at 70 with an accuracy of 83%. The 70th parameter setting is used for data testing, with results shown in the confusion matrix in

Table 2. Based on Table 2 regarding the confusion matrix, the error rate is found for 95 image data whose prediction results are wrong, so the error rate can be calculated using the following formula:

The total false negative (TFN) value
- TFN *anak-anak* =3+13=16
- TFN *remaja* =0+42=42
- TFN *dewasa* =10+27=37

The total false positive (TFP) value
- TFP *anak-anak* =0+10=10
- TFP *remaja* =3+27=30
- TFP *dewasa* =13+42=55

The total true negatif (TTN) value
- TTN *anak-anak*=3+13+0+213+42+10+27+211=519
- TTN *remaja*=219+13+51+10+27+211=523
- TTN *dewasa*=220+3+0+213=436

The total true positif (TTP) value:
- TTP *anak-anak* =220
- TTP *remaja* =213
- TTP *dewasa* =211

$$Error\ rate = The\ wrong\ data\ total/\ Data\ Total = 95/739 = 0.12$$

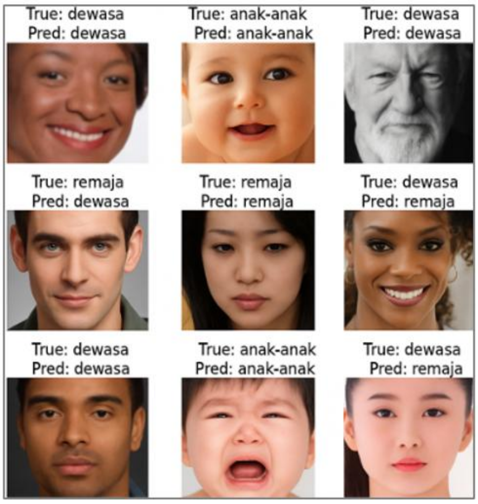
$$\Sigma TTP = TTPanak - anak + TTPremaja + TTPdewasa = 220 + 213 + 211 = 664$$

$$Accuracy = (\Sigma \mathrm{TTP})/Data\ Total = 664/739 = 0.87$$

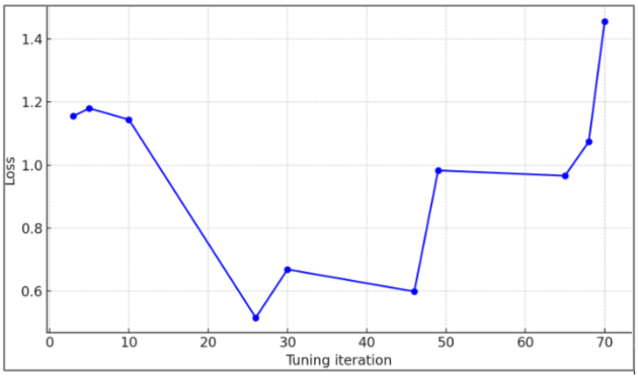
$$Precision = \frac{TTP}{TTP+TFP} = \frac{P_{anak-anak}+P_{remaja}+P_{dewasa}}{3} = \frac{0.96+0.88+0.79}{3} = 0.88$$

$$Recall = \frac{TTP}{TTP+TFN} = \frac{R_{anak-anak}+R_{remaja}+\ R_{dewasa}}{3} = \frac{0.93+0.8+0.85}{3} = 0.86$$

$$F1 - score = 2 \times (P \times R)/(P + R) = 2 \times ((0.88 \times 0.86)/(0.88 + 0.86)) = 0.87$$



(a)



(b)

Figure 3. Example predictions and loss graph during ANN model training and tuning: (a) sample prediction results and (b) loss levels across tuning iterations

Table 1. Tuning results 10 out of 70 times on the ANN model

Tuning	Batch size	Dropout rates	Dense layer	Optimizer learning rate	Regularization rate	Loss	F1-score	Accuracy
3	-	-	-	-	-	1.155	0.4948	54.71
5	-	-	-	-	-	1.18	0.4576	53.03
10	-	-	-	-	-	1.144	0.4431	54.9
26	32	0.2	256	-	0.001	0.516	0.7686	76.64
30	32	0,3	1024	-	0.001	0,669	0.7181	72.62
46	16	0,6	1024	-	0.001	0,599	0.8172	82.44
49	16	0,5	1024	0.25	0.0005	0.983	0.8186	82.29
65	64	Null, 0.5	1,024,512	0.01	0.0005	0.966	0.8132	81.78
68	64	Null, 0.5	1,024,512	0.01	0.0005	1,074	0.827	82.75
70	64	null, 0.5, 0.4	1,024,512,128	0.01	0.0005	1,455	0.8297	82.87

Table 2. Confusion matrix of ANN

		Predict		
		Anak-anak	*Remaja*	*Dewasa*
Actual	*Anak-anak*	220	3	13
	Remaja	0	213	42
	Dewasa	10	27	211

3.4. Modeling of age categories using VGG-16 K-NN

The VGGFace model is used as a feature extractor in the second approach. Training images are fed through this model, and the resulting features are saved. These features represent important attributes of the image that the K-NN model uses for classification. This means manual checking of features such as wrinkles, facial expressions, or birthmarks is unnecessary. The pre-trained VGGFace model, without its classification layer, is used for this purpose.

The feature extractor is created using VGG16. Image data dimensions are (200, 200, 3), indicating the image size is 200×200 pixels with 3 color channels red, green, and blue (RGB). Average pooling is applied to produce a concise final feature by taking each channel's average feature values. This is suitable for achieving a compact representation of facial features before feeding them to the classification layer. The adjusted VGGFace model predicts features from batches of images using the predict function. These predicted features represent each image and are used as input for the K-NN classification model. The extracted features are saved into a CSV file named 'knn_features.csv', which will be used to train the K-NN algorithm.

The K-NN model is created with a parameter k (number of neighbors) of n. This model classifies new images based on the majority labels of the k most similar training images. For example, if k=3, the model uses the three nearest neighbors for classification. The K-NN model is trained using features extracted from training images. Unlike ANN models, K-NN models do not "learn" traditionally but "remember" training examples to classify new ones. The training output of the K-NN model with k=3 is shown in Figure 4. For evaluation, K-NN models are assessed similarly to ANN models. Features from test images are extracted using the VGGFace model and used as input for the K-NN model to make predictions. The evaluation metrics are the confusion matrix and the F1-score, with some sample predictions displayed for visual inspection.

Figure 4. The sample output of K-NN model training with k=3 using UTKFace dataset

Overall, this study presents two approaches for age group classification from facial images: the first approach uses an ANN model with a pre-trained VGGFace model as the base, and the second approach uses a K-NN model with the VGGFace model as a feature extractor. The confusion matrix for the K-NN model with k=3 is shown in Table 3. Based on the confusion matrix in this table, the total test data is 739. To better understand performance measurement, the calculation of measurement results can be explained as follows:

The TFN value
TFN *anak-anak* =5+8=13
TFN *remaja* =7+74=81
TFN *dewasa* =36+45=81
The TTN value
TTN *anak-anak* =5+8+7+174+74+36+45+167=516
TTN *tremaja* =223+8+74+36+45+167=553
TTN *dewasa* =220+3+0+213=436

The TFP value
TFP *anak-anak* =7+36=43
TFP *remaja* =5+45=50
TFP *dewasa* =8+74=82
The TTP value:
TTP *anak-anak* =223
TTP *remaja* =174
TTP *dewasa* =167

After calculating the confusion matrix in Table 3, the error rate found 175 images that were predicted incorrectly, so the error rate can be calculated as follows:

$$Error\ rate = The\ wrong\ data\ total / \ Data\ Total = 175/739 = 0.23$$

$$\Sigma TTP = TTPanak - anak + TTPremaja + TTPdewasa = 220 + 174 + 167 = 564$$

$$Accuracy = (\Sigma \mathrm{TTP})/Data\ Total = 564/739 = 0.76$$

$$Precision = \frac{TTP}{TTP+TFP} = \frac{P_{anak-anak}+P_{remaja}+P_{dewasa}}{3} = \frac{0.94+0.78+0.67}{3} = 0.76$$

$$Recall = \frac{TTP}{TTP+TFN} = \frac{R_{anak-anak} + R_{remaja} + \ R_{dewasa}}{3} = \frac{0.94+0.68+0.67}{3} = 0.76$$

$$F1 - score = 2 \times (P \times R)/(P+R) \ = 2 \times ((0.76 \times 0.76)/(0.76+0.76)) = 0.76$$

Table 3. Confusion Matrix of K-NN

		Predict		
		Anak-anak	*Remaja*	*Dewasa*
Actual	*Anak-anak*	223	5	8
	Remaja	7	174	74
	Dewasa	36	45	167

After completing the modeling, the K-NN model is simulated to determine the best level of accuracy. In this research, GridSearchCV is applied to find the optimal parameters for the K-NN model, particularly the value of k. GridSearchCV systematically searches combinations of various hyperparameters in machine learning models. For K-NN, the key hyperparameters are k (the number of neighbors) and weights. Both parameters are optimized, with k values ranging from 1 to 15. A CV technique with 10 folds is applied, and the model is assessed using accuracy and F1-score. The simulation results of the K-NN model are presented in the graph in Figure 5. Both accuracy and F1-score tend to increase gradually from k=1 to k=15, reaching their highest values at k=14, with an accuracy of 0.80108 (80%) and an F1-score of 0.79694. Based on the test results, the ANN model demonstrates more optimal performance compared to the K-NN model. So, in the next step, the ANN model is applied to generate the identification system. For further research, performance in facial recognition needs to be improved by exploring deep learning methods such as CNN, as has been done in previous research [41].

3.5. The identifying system of age categories

This identification system is implemented using ANN with pre-trained VGG-16 with the last tuning carried out, namely the 70th tuning as shown in Figure 6. Therefore, the facial age classification interface starts with an initial display created using the results of this previous setup process. Users are guided to upload images via the "Drag and drop files" menu which accepts images up to 200 MB in JPG, JPEG, and PNG formats. Once the image is uploaded, the system uses multi-task convolutional neural network (MTCNN) to detect and extract faces and their features from the uploaded image, ensuring accurate facial region focus for better feature extraction.

After facial detection and cropping, users can press the "Predict" button, prompting the system to classify the age of the detected face into one of three categories: children, teenagers, and adults. This classification is achieved by passing the cropped face through the pre-trained and tuned model, which utilizes a SoftMax activation function to output probabilities for each age category, selecting the highest probability as the prediction.

The system also robustly handles non-facial images. If the uploaded image does not contain a face, the MTCNN algorithm will fail to detect facial landmarks, resulting in the system outputting "no face detected." This feature ensures the integrity and reliability of the system by processing only valid facial images. Overall, the interface offers a seamless user experience from image upload to accurate age prediction while maintaining robustness against invalid inputs. The findings of this research can be applied to address cybercrime issues in Indonesia. Facial recognition technology can be used to filter users, allowing them to access content appropriate for their age category. This can help tackle cyber pornography, a significant problem in Indonesia.

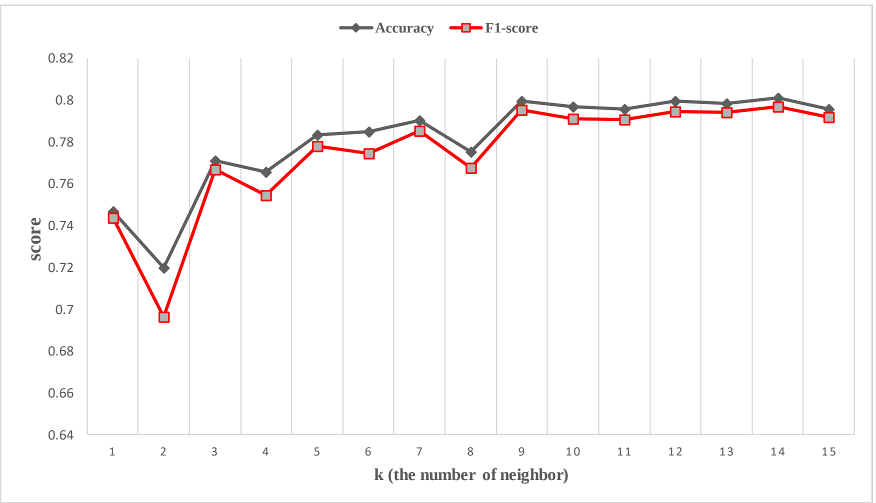


Figure 5. The accuracy and F1-score level of K-NN model simulation from k=1 to k=15

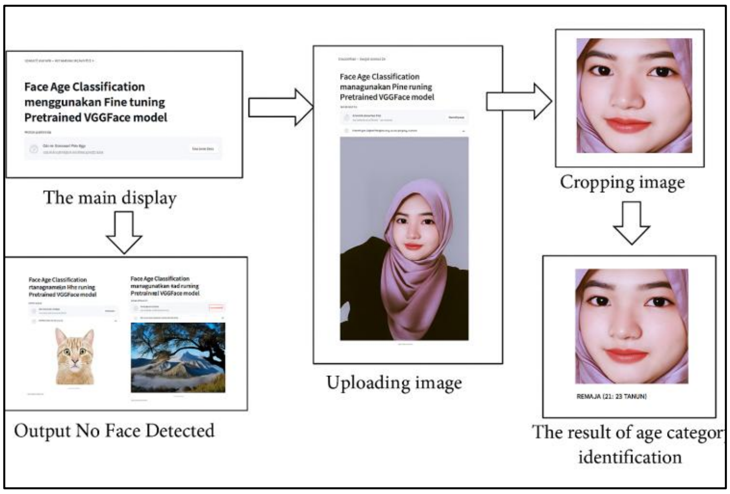


Figure 6. The system for detecting the age categories based on facial

4. CONCLUSION

Both algorithms based on pre-trained CNNs are good enough to recognize age ranges based on facial images. The model performance of the pre-trained CNN-based ANN is more optimal than that of the pre-trained CNN-based K-NN. This is indicated by the model built by pre-trained CNN-based ANN, which achieved an accuracy of 87%, an F1-score of 0.87, and an error rate of 0.12. Meanwhile, the pre-trained CNN-based K-NN achieved an accuracy of around 76%, an F1-score of 0.76, and an error rate of 0.23.

Future research could explore larger and more diverse datasets or other classifiers to improve generalization. The findings imply that CNN-based ANN can effectively support age-related applications, though limited dataset diversity may affect robustness.

FUNDING INFORMATION

We extend our gratitude for the financial support provided by the Ministry of Education, Culture, Research, and Technology through its fundamental research grant program. This work is also supported by LPPM Universitas Negeri Surabaya.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Yuni Yamasari	✓	✓			✓	✓	✓	✓	✓	✓		✓	✓	✓
Lusiana Anggraini		✓	✓			✓		✓	✓	✓	✓			
Anita Qoiriah	✓			✓			✓			✓	✓		✓	
Ricky Eka Putra			✓	✓		✓		✓		✓	✓			
Hapsari Peni Agustin Tjahyaningtijas					✓		✓			✓			✓	
Tohari Ahmad	✓					✓				✓		✓	✓	✓

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

The authors declare no conflict of interest.

DATA AVAILABILITY

The data that support the findings of this study are openly available in the UTKFace dataset at https://susanqq.github.io/UTKFace.

REFERENCES

[1] B. Martínez-Núñez *et al.*, "Facebooking suicide: Evaluation of pro-suicide websites in most used Spanish social networks by adolescents," *European Child and Adolescent Psychiatry*, vol. 24, no. 1, pp. S197–S198, 2015.

[2] Ö. Aslan, S. S. Aktuğ, M. Ozkan-Okay, A. A. Yilmaz, and E. Akin, "A comprehensive review of cyber security vulnerabilities, threats, attacks, and solutions," *Electronics*, vol. 12, no. 6, 2023, doi: 10.3390/electronics12061333.

[3] V. C. Sharmila, H. M. Aslam, and M. M. Riswan, "Analysing and identifying harm propagation of cyber threats in autonomous vehicles and mitigation through ANN," in *Smart Trends in Computing and Communications*, 2022, pp. 405–417, doi: 10.1007/978-981-16-4016-2_38.

[4] S. M. Pedapudi and N. Vadlamani, "A comprehensive network security management in virtual private network environment," in *2022 International Conference on Applied Artificial Intelligence and Computing (ICAAIC)*, 2022, pp. 1362–1367, doi: 10.1109/ICAAIC53929.2022.9793196.

[5] H. Sheikh, C. Prins, and E. Schrijvers, "Artificial intelligence: definition and background," in *Mission AI*, Cham, Switzerland: Springer, 2023, pp. 15–41, doi: 10.1007/978-3-031-21448-6_2.

[6] Y. Xu *et al.*, "Artificial intelligence: A powerful paradigm for scientific research," *Innovation*, vol. 2, no. 4, 2021, doi: 10.1016/j.xinn.2021.100179.

[7] S. Mascarenhas and M. Agarwal, "A comparison between VGG16, VGG19 and ResNet50 architecture frameworks for image classification," in *2021 International Conference on Disruptive Technologies for Multi-Disciplinary Research and Applications (CENTCON)*, 2021, pp. 96–99, doi: 10.1109/CENTCON52345.2021.9687944.

[8] M. M. Taye, "Understanding of machine learning with deep learning: architectures, workflow, applications and future directions," *Computers*, vol. 12, no. 5, 2023, doi: 10.3390/computers12050091.

[9] H. Reese, "Understanding the differences between AI, machine learning, and deep learning," *TechRepublic*. 2017. [Online]. Available: https://www.techrepublic.com/article/understanding-the-differences-between-ai-machine-learning-and-deep-learning/

[10] I. H. Sarker, "Deep learning: a comprehensive overview on techniques, taxonomy, applications and research directions," *SN Computer Science*, vol. 2, no. 6, 2021, doi: 10.1007/s42979-021-00815-1.

[11] W. Fang, Q. Xue, L. Shen, and V. S. Sheng, "Survey on the application of deep learning in extreme weather prediction," *Atmosphere*, vol. 12, no. 6, 2021, doi: 10.3390/atmos12060661.

[12] A. Voulodimos, N. Doulamis, A. Doulamis, and E. Protopapadakis, "Deep learning for computer vision: a brief review," *Computational Intelligence and Neuroscience*, vol. 2018, pp. 1–13, 2018, doi: 10.1155/2018/7068349.

[13] S. S. A. Zaidi, M. S. Ansari, A. Aslam, N. Kanwal, M. Asghar, and B. Lee, "A survey of modern deep learning based object detection models," *Digital Signal Processing: A Review Journal*, vol. 126, 2022, doi: 10.1016/j.dsp.2022.103514.
[14] H. Fujiyoshi, T. Hirakawa, and T. Yamashita, "Deep learning-based image recognition for autonomous driving," *IATSS Research*, vol. 43, no. 4, pp. 244–252, 2019, doi: 10.1016/j.iatssr.2019.11.008.
[15] H. Li, Y. Pan, J. Zhao, and L. Zhang, "Skin disease diagnosis with deep learning: A review," *Neurocomputing*, vol. 464, pp. 364–393, 2021, doi: 10.1016/j.neucom.2021.08.096.
[16] M. Khojaste-Sarakhsi, S. S. Haghighi, S. M. T. F. Ghomi, and E. Marchiori, "Deep learning for Alzheimer's disease diagnosis: a survey," *Artificial Intelligence in Medicine*, vol. 130, 2022, doi: 10.1016/j.artmed.2022.102332.
[17] V. Thanikachalam, S. Shanthi, K. Kalirajan, S. Abdel-Khalek, M. Omri, and L. M. Ladhar, "Intelligent deep learning based disease diagnosis using biomedical tongue images," *Computers, Materials and Continua*, vol. 70, no. 3, pp. 5667–5681, 2022, doi: 10.32604/cmc.2022.020965.
[18] V. Abhyankar and R. Kene, "Generation of RBC, WBC subtype, and platelet count report using YOLO," *International Journal for Research in Applied Science and Engineering Technology*, vol. 11, no. 4, pp. 1623–1630, 2023, doi: 10.22214/ijraset.2023.50459.
[19] S. Petikam, F. D. C. D. Sales, S. Suma, J. L. A. Gonzáles, K. Joshi, and B. Pant, "Image processing with intelligence system using sensing in cyber security," in *2023 International Conference on Artificial Intelligence and Smart Communication (AISC)*, 2023, pp. 570–574, doi: 10.1109/AISC56616.2023.10085025.
[20] M. A. Siddiqi and W. Pak, "Tier-based optimization for synthesized network intrusion detection system," *IEEE Access*, vol. 10, pp. 108530–108544, 2022, doi: 10.1109/ACCESS.2022.3213937.
[21] R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, and S. Venkatraman, "Robust intelligent malware detection using deep learning," *IEEE Access*, vol. 7, pp. 46717–46738, 2019, doi: 10.1109/ACCESS.2019.2906934.
[22] T. Hidayat, D. U. E. Saputri, and F. Aziz, "Meat image classification using deep learning with Resnet152V2 architecture," *Jurnal Techno Nusa Mandiri*, vol. 19, no. 2, pp. 131–140, 2022, doi: 10.33480/techno.v19i2.3932.
[23] S. S. Harakannanavar, J. M. Rudagi, V. I. Puranikmath, A. Siddiqua, and R. Pramodhini, "Plant leaf disease detection using computer vision and machine learning algorithms," *Global Transitions Proceedings*, vol. 3, no. 1, pp. 305–310, 2022, doi: 10.1016/j.gltp.2022.03.016.
[24] T. R. Chenthil, B. A. Vijayakumar, S. Ranjith, and G. Balachandran, "Labeled image segmentation and retrieval for fast images processing using K-NN algorithm," in *2023 Eighth International Conference on Science Technology Engineering and Mathematics (ICONSTEM)*, Apr. 2023, pp. 1–3, doi: 10.1109/ICONSTEM56934.2023.10142880.
[25] P. Harsan, A. Qurania, and K. Damayanti, "Maize plant desease identification (Zea Mays L. Saccharata) using image processing and k-nearest neighbor (K-NN)," *International Journal of Engineering & Technology*, vol. 7, no. 3.20, pp. 402–405, 2018, doi: 10.14419/ijet.v7i3.20.20581.
[26] A. Putri, A. Virgono, and C. Setianingsih, "Analysis of image processing in barcode using the k-nearest neighbor (K-NN) classification," *International Journal of Advanced Trends in Computer Science and Engineering*, vol. 9, no. 1.5, pp. 185–190, 2020, doi: 10.30534/ijatcse/2020/2691.52020.
[27] S. A. AlDera and M. T. Ben Othman, "A model for classification and diagnosis of skin disease using machine learning and image processing techniques," *International Journal of Advanced Computer Science and Applications*, vol. 13, no. 5, pp. 252–259, 2022, doi: 10.14569/IJACSA.2022.0130531.
[28] K. A. Vakilian and J. Massah, "An artificial neural network approach to identify fungal diseases of cucumber (Cucumis sativus L.) plants using digital image processing," *Archives Of Phytopathology And Plant Protection*, vol. 46, no. 13, pp. 1580–1588, 2013, doi: 10.1080/03235408.2013.772321.
[29] D. K. Shetty, U. D. Acharya, V. G. Narendra, and P. J. Prajual, "Intelligent system to evaluate the quality of DRC using image processing and then categorize using artificial neural network (ANN)," *Indian Journal of Agricultural Research*, vol. 54, no. 6, pp. 716–723, 2020, doi: 10.18805/IJARe.A-5374.
[30] A.-E. Andon and G. Covatariu, "A study on image processing using artificial neural networks in civil engineering," *Bulletin of the Polytechnic Institute of Iași. Construction. Architecture Section*, vol. 67, no. 3, pp. 85–94, 2022, doi: 10.2478/bipca-2021-0027.
[31] D. G. -López, A. G. -Rodríguez, S. G. -Rodríguez, A. Suárez-García, M. Díez-Mediavilla, and C. Alonso-Tristán, "Pixel-based image processing for CIE standard sky classification through ANN," *Complexity*, vol. 2021, 2021, doi: 10.1155/2021/2636157.
[32] Z. M. Mosa, N. H. Ghae, and A. H. Ali, "Detecting keratoconus by using SVM and decision tree classifiers with the aid of image processing," *Baghdad Science Journal*, vol. 16, no. 4, pp. 1022–1029, 2019, doi: 10.21123/bsj.2019.16.4(Suppl.).1022.
[33] L. Wang, L. Zhao, and D. Liu, "A review on the application of SVM in hyperspectral image processing," *Harbin Gongcheng Daxue Xuebao/Journal of Harbin Engineering University*, vol. 39, no. 6, pp. 973–983, 2018, doi: 10.11990/jheu.201704074.
[34] A. S. Ansari *et al.*, "Detection of pancreatic cancer in CT scan images using PSO SVM and image processing," *BioMed Research International*, vol. 2022, 2022, doi: 10.1155/2022/8544337.
[35] E. O. Tomppo, C. Gagliano, F. De Natale, M. Katila, and R. E. McRoberts, "Predicting categorical forest variables using an improved k-nearest neighbour estimator and landsat imagery," *Remote Sensing of Environment*, vol. 113, no. 3, pp. 500–517, 2009, doi: 10.1016/j.rse.2008.05.021.
[36] I. M. Hanmastiana, B. Warsito, R. Rahmawati, H. Yasin, and P. Kartikasari, "Classification of public opinion on social media Twitter concerning the education in Indonesia using the k-nearest neighbors (K-NN) algorithm and k-fold cross validation," *STATISTIKA Journal of Theoretical Statistics and Its Applications*, vol. 21, no. 2, pp. 99–106, 2022, doi: 10.29313/statistika.v21i2.297.
[37] Y. Wang, F. Sun, W. Huang, F. He, and D. Tao, "Channel exchanging networks for multimodal and multitask dense image prediction," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 5, pp. 5481–5496, 2023, doi: 10.1109/TPAMI.2022.3211086.
[38] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 2015.
[39] P. H. Putra, M. S. Novelan, and M. Rizki, "Analysis k-nearest neighbor method in classification of vegetable quality based on color," *Journal of Applied Engineering and Technological Science*, vol. 3, no. 2, pp. 126–132, 2022, doi: 10.37385/jaets.v3i2.763.
[40] A. M. J. Raj, F. S. Francis, and P. J. Benadit, "Particle swarm and cuckoo search (PSCS) optimization based feature selection method to improve the web page classification," *Journal of Advanced Research in Dynamical and Control Systems*, vol. 9, no. 6, pp. 1125–1140, 2017.
[41] P. A. J. Orlando, J. M. Robinson, and J. E. M. Baquero, "Comparison of convolutional neural network models for user's facial recognition," *International Journal of Electrical and Computer Engineering*, vol. 14, no. 1, pp. 192–198, 2024, doi: 10.11591/ijece.v14i1.pp192-198.

BIOGRAPHIES OF AUTHORS

Yuni Yamasari is Associate Professor at Department of Informatics, Universitas Negeri Surabaya, Surabaya, Indonesia. She received the bachelor degree, the master's degree in computer science and the Dr degree in electrical engineering from Institut Teknologi Sepuluh Nopember (ITS), Indonesia. She received scholarships, including *Peningkatan Kualitas Publikasi Internasiona*l (PKPI)/the Sandwich Like to conduct research in Japan. Her current research interests include artificial intelligence, machine learning, and data mining. She can be contacted at email: yuniyamasari@unesa.ac.id.

Lusiana Anggraini is an alumna of the 2019 class of Surabaya State University, Bachelor of Information Engineering study program. She is interested in design, software testing, and artificial intelligence. She can be contacted at email: lusiana.19069@mhs.unesa.ac.id.

Anita Qoiriah received the bachelor degree and the master's degree in computer science, Institut Teknologi Sepuluh Nopember (ITS), Indonesia. She is Associate Professor at Department of Informatics, Universitas Negeri Surabaya, Surabaya, Indonesia. Her research interest areas are artificial intelligence, natural language processing, and machine learning. She can be contacted at email: anitaqoiriah@unesa.ac.id.

Ricky Eka Putra is an Assistant Professor at Department of Informatics, Universitas Negeri Surabaya, Surabaya, Indonesia. He received the bachelor's to doctoral degree in Computer Science from Institut Teknologi Sepuluh Nopember (ITS), Surabaya, Indonesia. His research interests include artificial intelligence, image processing, and computer vision. He can be contacted at email: rickyeka@unesa.ac.id.

Hapsari Peni Agustin Tjahyaningtijas is presently a lecturer in the Department of Electrical Engineering Universitas Negeri Surabaya. She holds a doctorate degree from the Institut Teknologi Sepuluh Nopember. Her research interests include deep learning, computer vision, and machine vision. She can be contacted at email: hapsaripeni@unesa.ac.id.

Tohari Ahmad received the bachelor degree in computer science from Institut Teknologi Sepuluh Nopember (ITS), Indonesia, the master's degree in information technology from Monash University, Australia, and the Ph.D. degree in computer science from RMIT University, Australia. He was a consultant for some international companies. In 2003, he moved to ITS, where he is currently a professor. His current research interests include network security, information security, data hiding, and computer networks. His awards and honors include the Hitachi Research Fellowship and JICA Research Program to conduct research in Japan. He is a reviewer of several journals. He can be contacted at email: tohari@if.its.ac.id.