SINERGI Vol. 29, No. 1, February 2025: 33-42
http://publikasi.mercubuana.ac.id/index.php/sinergi
http://doi.org/10.22441/sinergi.2025.1.004

Development of face image recognition algorithm using CNN in airport security checkpoints for terrorist early detection

Eca Indah Anggraini[1]*, Fachdy Nurdin[1], Muhammad Obie Restianto[1], Sudarti Dahsan[1], Andini Aprilia Ardhana[1], Asep Adang Supriyadi[1], Yahya Darmawan[2], Syachrul Arief[3], Agus Haryanto Ikhsanudin[1]
[1]Sensing Technology, Republic Indonesia Defense University, Indonesia
[2]Climatology Department, State College of Meteorology Climatology and Geophysics (STMKG), Indonesia
[3]Geospatial Information Agency, Indonesia

Abstract
Ensuring airport security is of paramount importance to safeguard the lives of passengers and prevent acts of terrorism. In this context, developing advanced technology for early terrorist detection is crucial. This paper presents a novel approach to enhancing security measures at airport checkpoints by applying Convolutional Neural Network (CNN) and Artificial Neural Network (ANN) algorithms in face image recognition. Our system utilizes state-of-the-art artificial intelligence techniques to analyze facial features. Our research uses VGG architecture and pre-trained with face data as a CNN model. This model is used to extract face embedding features from the dataset. These embedding features are then compressed with Principal Component Analysis (PCA) to obtain the meaningful feature as training data for the ANN algorithm. We trained our system using data from 500 identities data with 60 data for each identity. This training enables our system to recognize known terrorists and individuals on watchlists by comparing the facial features of individuals passing through security checkpoints with those in the database. The proposed CNN-ANN-based face recognition system not only enhances airport security but also significantly reduces the processing time for security checks. It can quickly identify potential threats, allowing security personnel to take appropriate actions in real time ensuring a rapid response to security concerns. We present the architecture, training methodology, and evaluation of the CNN-ANN model, achieving a high accuracy of 91.16% and precision of 91.36%. Through this research, we aim to increase airport security and strengthen efforts to combat terrorism, making air travel safer and more secure for all passengers.

Keywords:
Airport security;
ANN;
CNN;
Face recognition;
Terrorist detection;

Article History:
Received: December 13, 2023
Revised: June 18, 2024
Accepted: July 8, 2024
Published: January 1, 2025

Corresponding Author:
Eca Indah Anggraini,
Sensing Technology, Republic Indonesia Defense University, Jakarta, Indonesia
Email:
eca.anggraini@tp.idu.ac.id

INTRODUCTION

Airports are critical hubs of global transportation, serving millions of passengers daily. Ensuring the safety and security of these travelers is a top priority for governments and airport authorities worldwide [1]. The ever-present threat of terrorism necessitates constant innovation in security measures [2][3]. Traditional security screening procedures at airports, which often rely on manual document verification and surveillance, are resource-intensive, time-consuming, and susceptible to human error [4]. Emerging technologies, particularly in the field of artificial intelligence and deep learning, offer promising innovations for improving the efficiency and accuracy of security checks [5]. In response, we introduce an approach for enhancing airport security by applying Convolutional Neural Network (CNN) algorithms in face image recognition [6].

Convolutional Neural Network (CNN) is a class of deep learning algorithms specifically designed for processing and analyzing visual data, such as images and videos [7]. CNNs are inspired by the human visual system and are adept at automatically learning and recognizing intricate patterns, features, and structures within images [8][9]. They employ convolutional layers to scan and filter input data, enabling them to capture complex hierarchies of features and identify objects and patterns with remarkable accuracy [10][11]. In airport security, CNNs are harnessed for precise face image recognition, making them a powerful tool for enhancing security measures and early detection of potential threats [12]. This paper explores the application of CNN algorithms to improve airport security at checkpoints, leveraging their proficiency in facial feature analysis to strengthen safety protocols while expediting the screening process [13].

Several studies have explored the integration of face recognition technologies, including the use of CNN algorithms, in airport security settings over the years [14][15]. These initiatives aim to enhance security protocols and streamline passenger screening processes. Notably, early face recognition implementations at airports primarily focused on biometric authentication for expedited check-in and boarding procedures [16][17]. However, in the wake of heightened security concerns, the scope of these applications expanded to include the early detection of potential threats. Past research efforts have demonstrated the feasibility of deploying face recognition systems powered by CNN and other machine learning algorithms to identify individuals on watchlists or with known affiliations to terrorism [18, 19, 20]. These endeavors aid the foundation for more recent and comprehensive projects seeking to strengthen airport security, minimize false positives, and provide an efficient means of detecting people of interest. The evolution of these studies reflects the growing importance of advanced technologies in safeguarding airports and the critical role that CNN and ANN-based face recognition play in meeting these security challenges.

METHOD

The methods or stages in this research are shown in the flow diagram in Figure 1. According to the flowchart of the research stages, the first stage in this research is the preparation of tools and materials, as shown in Table 1.

The next step is collecting facial image data. We used the CASIA Web Face dataset, which originally contained 494,414 face images of 10,575 real identities. We selected identities with at least 60 images, resulting in 500 identities. Several samples of facial data are shown in Figure 2.

It is crucial to acknowledge that the successful implementation of CNNs often relies on the availability of substantial and diverse datasets. The more extensive and varied the dataset, the better the CNN can capture the complexities and nuances of the target domain [21][22].

Table 1. Research Tools and Materials

Tools	Material
1. Python programming language 2. Streamlit python library 3. Web camera 4. Computer	Face image data

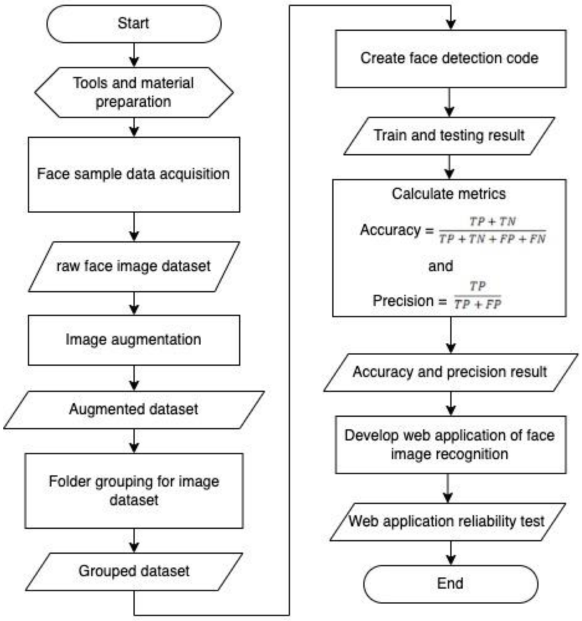


Figure 1. Flowchart of the Research Stages

Figure 2. Sample of Face Image Data

This research used several augmentation methods including translational, color, and rotational augmentation. All augmentation-type formulas that are used in this research are shown in Table 2. The augmented dataset is then standardized in pixel size to 80x80 pixels.

We implemented two translation augmentation types, right and left, for 10 pixels. The translated images were then processed for color augmentation, including Gaussian blur, gamma correction to make the image lighter or darker, histogram equalizer, Gaussian noise, and sharpening. The translation results were also used as input for rotation augmentation. For rotation augmentation, rotating the image to -6 and 6 degrees. This augmentation process increased the diversity of the face image dataset, improving the performance of the recognition model and preventing overfitting during the model-building process.

After the image dataset was successfully formed, the next step was to create a face detection program using the tools and materials mentioned in Table 1. The program processed the image data and implemented the CNN algorithm structure. Next, after the data had been processed and the CNN algorithm was ready, the detection system training and testing process was carried out according to the desired parameter variations. To test the system, we used a confusion matrix as shown in Figure 3.

From the confusion matrix, we can obtain four variables: TP (True Positive), FP (False Positive), FN (False Negative), and TN (True Negative). Those variables evaluate the model's prediction result compared with actual data [27].

Table 2. List of Augmentation Type Formula

Augmentation Type	Formula
Translation [23]	$x = x' \pm h$ $y = y' \pm k$
Gaussian Blur [24]	$G(x,y) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2+y^3}{2\sigma^3}}$
Gamma Correction [25]	$O(x.y) = \frac{I(x,y)}{255}^{\frac{1}{\gamma}} .255$
Histogram Equalizer [25]	$H'(i) = \sum_{0 \le j < i} H(j)$ $equalized(x,y) = H'(src(x,y))$
Gaussian Noise [24]	$N(z) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(z-\mu)^2}{2\sigma^3}}$ $I(x,y) = I'(x,y) + N(x,y)$
Sharpen [26]	$I(x,y) = I'(x,y) * \begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$
Rotation [23]	$x = x' \cos\theta - y' \sin\theta$ $y = x' \sin\theta - y' \cos\theta$

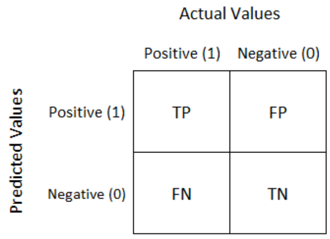


Figure 3. Confusion Matrix

Based on these variables, we can calculate the accuracy and precision values according to the formulas in (1) and (2). Accuracy value in the classification system is the percentage of data classified correctly in the process of measuring classification results. Meanwhile, precision is the proportion of cases classified as positive that are true positives in the actual data [28].

$$accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

$$precision = \frac{TP}{TP+FP} \quad (2)$$

The evaluated model is then used inside the web application to recognize the face image inputted to the website. The web application is designed simply with a single main menu and a left sidebar as shown in Figure 4. The main menu displays uploaded images and the face image recognition results. The sidebar allows users to upload or choose a face image.

A use case diagram is a visual representation in UML (Unified Modeling Language) that illustrates how a system interacts with external entities (actors) and the specific actions or functions (use cases) that the system performs in response to those interactions [29]. Users or security staff in this web application can see the results of face image recognition that have been uploaded or selected using the previously built CNN model. The user experience with this web application is shown in the use case diagram in Figure 5.

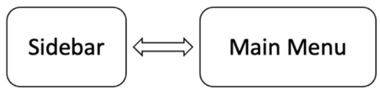


Figure 4. User Interface Navigation

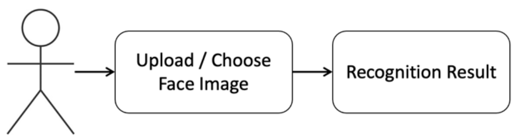


Figure 5. Use Case Diagram

Figure 6 shows a real use case of this web application to mitigate the threat of terrorism act at the airport gate. At the gate, airport security is shown the face results of possible terrorist face identities that have been trained based on a web or small sample face image of the terrorist. Based on the result, airport security staff can conduct a secondary inspection of the suspected passenger.

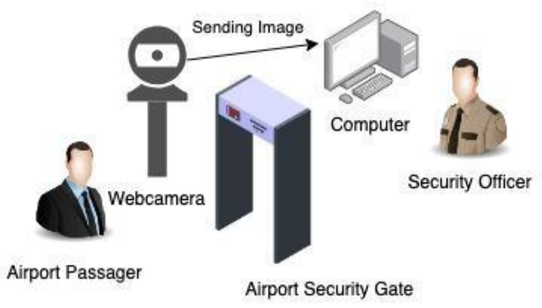


Figure 6. Face Image Recognition at Security Checkpoint Illustration

RESULTS AND DISCUSSION

Dataset Acquisition and Augmentation

From the 10,575 real identities in the CASIA Web Face Dataset, we selected 500 identities with at least 60 face images [30]. Using the augmentation type formula in Table 2, we performed an augmentation process for each image in the dataset. An illustration of the augmentation process for a single image is shown in Figure 7.

After the augmentation process, the initial 60 images per identity increased to a total of 600 images per identity, as shown in Figure 8. This resulted in a total of 300,000 images. The dataset was then divided into training, validation, and testing groups with a ratio of 75% for training data (225,000 images), 15% for validation data (45,000 images), and 10% for test data (30,000 images). This augmented dataset was then used to extract face embedding data using CNN-based VGG FaceNet architecture with pre-trained weights from another face dataset. The original feature shape of the VGG FaceNet model (2622 features) was reduced to 128 features using Principal Component Analysis (PCA). The training and validation data for the face embeddings were used in the ANN algorithm training process. Meanwhile, the test data was not included in the ANN training process and was used to calculate accuracy and precision values for model performance and evaluation. The step-by-step process is shown in Figure 9.

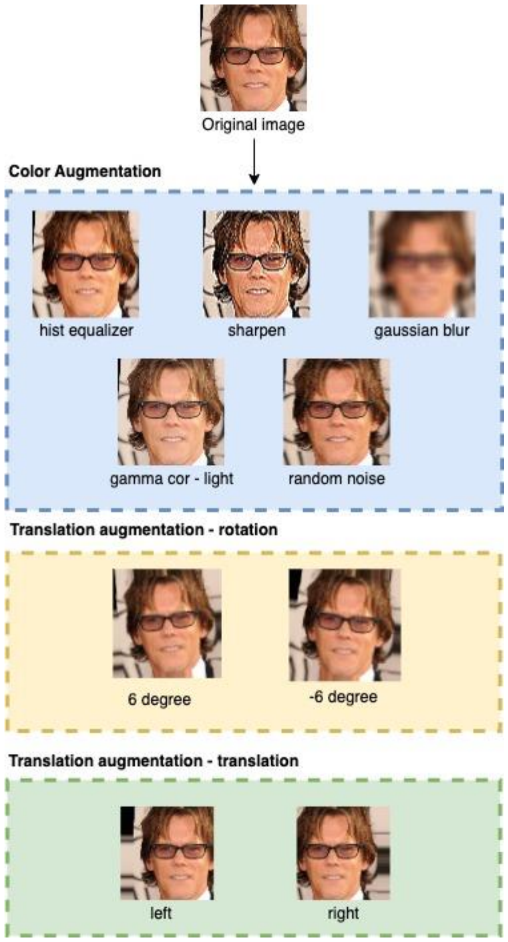


Figure 7. Image Augmentation Process

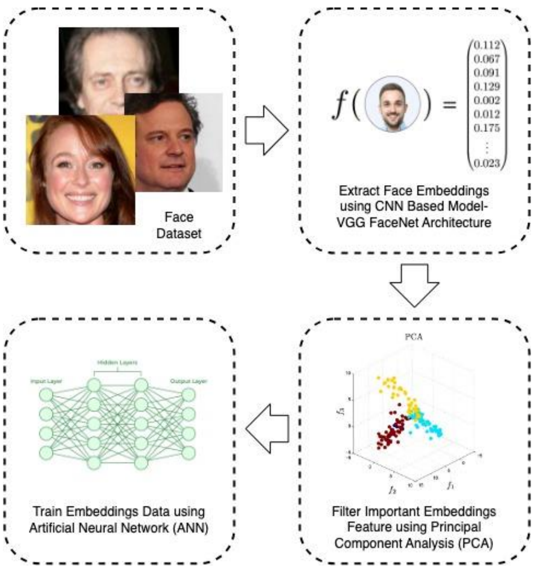


Figure 8. CNN Architecture Diagram

CNN-ANN Algorithm Layer Structure

The CNN-based VGG FaceNet architecture model shown in Figure 9 consists of multiple repetitions of the convolution layer with a

ReLU activation function, a maximum pooling layer, and a Fully Connected (FC) layer [31].

The formula for each layer in this CNN-ANN is summarized in Table 3. The convolution layer operation involves sliding filters (W) over the input data (h), resulting in a two-dimensional activation map or feature map, which indicates the presence of learned features at different spatial locations in the input. ReLU is preferred over other activation functions (like sigmoid or hyperbolic tangent) in deep learning because it helps mitigate the vanishing gradient problem and accelerates the convergence of the training process [32]. Max pooling helps make the network more robust to small translations and variations in the input data while reducing the computational load by reducing the number of parameters. Max pooling operates over a local receptive field or window within the input feature map. For an input feature map X with dimensions $H \times W \times D$, where H denotes height, W denodes width, and D denotes depth (number of channels), max pooling with a window size of $k \times k$ and stride s produces an output feature map Y of reduced dimensions. Each of these layer's studies the characteristics of the image from simple to complex characteristics [33]. The process then continues with a fully connected layer until the 2622 face embedding feature output is obtained.

For the ANN model, we construct five fully connected layers: the first layer as input, three hidden layers, and the last layer as output, as shown in Figure 10.

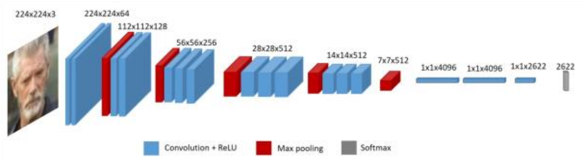


Figure 9. CNN-VGG FaceNet Architecture Diagram

Table 3. List of CNN-ANN Layer or Operations Formula

Layer/Operation	Formula
Convolution	$z^l = h^{l-1} * W^l$
ReLU [30]	$f(x) = \max(0, x)$
Max Pooling [31]	$h^l{}_{xy} = max_{i=0..s, j=0..s}\, h^{l-1}(x+i)(y+j)$
Fully Connected (FC) Layer	$z_l = W_l * h_{l-1}$
Softmax [34]	$softmax(z_i) = e^{zi} \,/\, \sum_j e^{zj}$

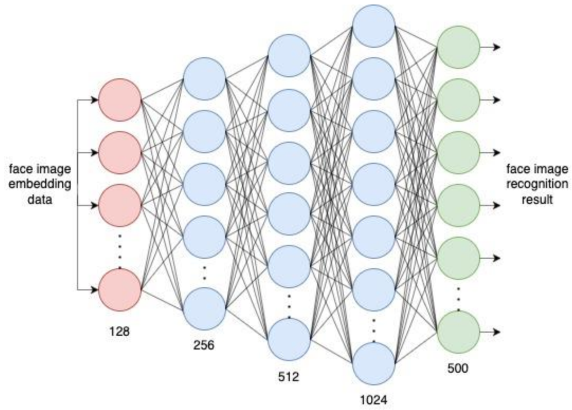


Figure 10. ANN Layer Structure

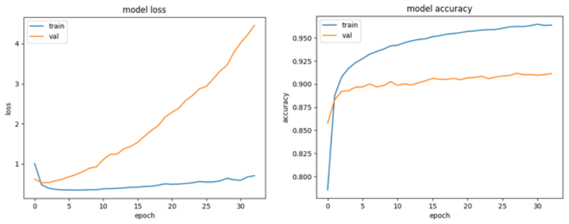
The first FC layer has a size of 128 nodes, followed by the hidden layers with sizes of 256, 512, and 1024 nodes to learn the embedding features and classify 500 outputs in the last layer. We also add the ReLu activation function for each node in the input and hidden layers, and the SoftMax activation function for the last output layer. The primary function of SoftMax is to convert raw output scores from the network's final layer into a probability distribution over a predefined set of classes, facilitating the interpretation of these scores as probabilities.

CNN-VGG FaceNet Extract Embeddings

All the face image data points are inputted into the CNN-VGG FaceNet architecture model with pre-trained weights to extract each face image embedding feature data. This embedding data is unique for each face image data, hence its representation of the face image. Before entering PCA to filter important features, each embedding data is normalized using a standard scaler to remove the mean and scale each feature/variable to unit variance.

ANN Embedding Train Process

The ANN algorithm that has been defined is then used for the face image embedding data training process. We set the training process to run for 50 epochs, with a callback function to monitor the validation accuracy. If the validation accuracy remains at its maximum for 5 epochs, the training process will stop. During the training process, the model with the best validation accuracy will be used in the testing and detection processes on face images. During the training process, a graph of model error values (model loss) and model accuracy values (model accuracy) is produced, as shown in Figure 11.



(a) (b)

Figure 11. Graph of Model (a) Loss (b) Accuracy During Training

Based on the graphs above, the model loss graph in Figure 11(a) shows that as the training process progresses, the error value decreases in the first three epochs, and then starts to increase. The model accuracy graph in Figure 11(b) shows that as the training process progresses, the accuracy increases during the first three epochs and then stabilizes around 91%. This interpretation indicates that the model training process appears to be progressing successfully in the first three epochs, as evidenced by the decreasing error value and increasing accuracy value. However, for the rest of the epochs, validation accuracy remains stagnant, and loss values increase. The training process stopped at 32 epochs after the callback function of maximum validation accuracy for five epochs was triggered.

ANN Embedding Testing Process

The total test data used comprised 30,000 images. The testing process was carried out individually for each image, starting with extracting 2,622 embeddings in the CNN-VGG FaceNet model, filtering important features with PCA, and then predicting the 128 features using the ANN model to determine the identities of the face images. Each predicted result was recorded and compared with the target label of the test data.

Based on the formulas for calculating accuracy in (1) and precision in (2). So, the following results were obtained.

$$accuracy = 91.16\%$$
$$precision = 91.31\%$$

From the calculations described above, the facial classification model evaluation achieved an accuracy of 91.16% and a precision value of 91.31%. These results indicate that the model can identify nine types of faces with high accuracy and precision. Figure 12 shows a snapshot of the prediction results on the test data. Most of the facial type of prediction results align with the actual data, although there are still some errors in the prediction results. The emergence of errors in the prediction results is due to the fact that the two individuals have similar physical facial characteristics. It can be shown from each actual facial images in Figure 13.

Comparison With Other Algorithms

In this subsection, we evaluate the face image recognition task from our study with other similar research or study. As shown in Table 4, our proposed algorithms CNN Embeddings combine with ANN as classifier to recognize face image achieves the accuracy result of 91.2% above the other algorithms result. This result was achieved due to our algorithms using CNN facial embedding extraction as input for the ANN.

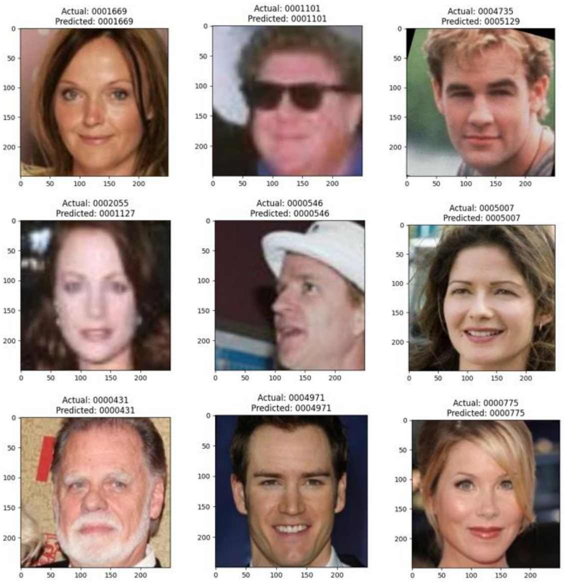


Figure 12. Sample of Image Prediction Result

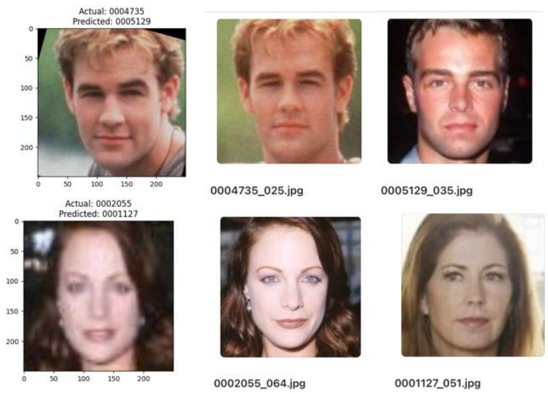


Figure 13. Analysis of False Prediction Result

Table 4. Comparison With Other Algorithms

Algorithm	Accuracy
Haar Cascade + LBP Histogram [35]	77.0%
Artificial Neural Network [36]	80.0%
Principal Component Analysis [37]	82.0%
PCA + ANN [37][38]	91.0%
CNN Embeddings + ANN	**91.2%**

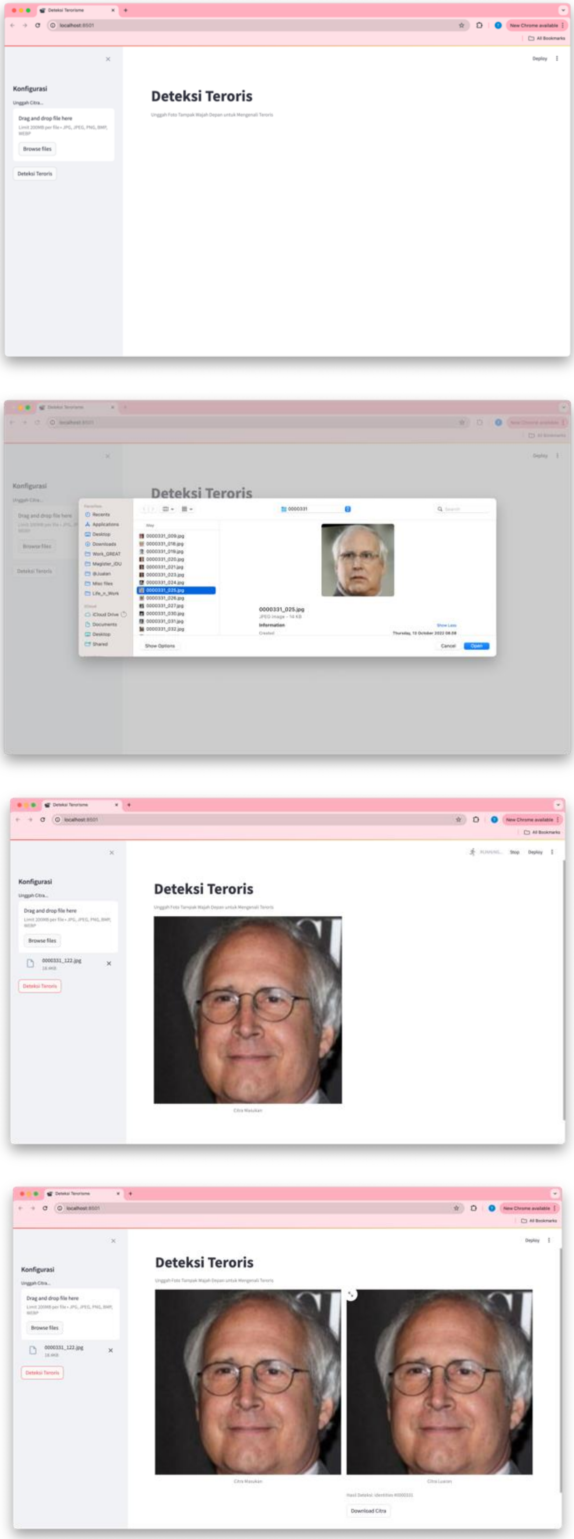


Figure 14. Web Application Result with Flow Process

It gives more meaningful features to identify characteristic differences between classes' face images. Besides that, our proposed model also can work with larger face image classes of 500 individual data. So, it can be used in airport security for terrorists' early detection considering that the number of people at the airport is large.

Web Application Development

Simple web applications are built using Python programming language, as it was used for building our CNN-ANN model, and the Streamlit library for building the web interface based on our web navigation and use case diagram in Figure 5 and Figure 6. Initially, our web application only displayed a blank main menu and a sidebar with buttons for uploading and detecting face images. After the user selects the desired face image to detect, the left column in the main menu shows the original image that the user has already uploaded. To detect who is in the images, the user clicks the detection button, and the right column in the main menu shows the original image with a caption containing the detection result of the image. The web application interface result and its flow process are shown in Figure 14.

To test the reliability of our deployed web application, we conducted another test using outside data, other than the train-validation-test dataset. We prepared 1000 face images, and the prediction results showed 873 correct identifications and 127 incorrect ones, resulting in an 87.3% accuracy score for the web application. Based on the result, we conclude that the application is reliable and accurate enough to recognize the identities of the trained face image dataset.

CONCLUSION

Through this research, we developed an artificial intelligence-based early detection system for terrorist face recognition. This system is integrated with a web application designed for airport security checkpoints and utilizes the CNN-ANN method to identify faces. Our results show that the system can accurately identify 500 terrorist faces. The accuracy of the system during testing was 91.16%, with a slightly higher precision at 91.31%. When tested as part of the web application, the system achieved an accuracy of 87.3%. This indicates that while the system performs well in a controlled environment, there is a slight decrease in performance when deployed in a real-world application. In conclusion, our AI-based system demonstrates strong potential for enhancing airport security. It effectively identifies a wide range of terrorist faces with high accuracy and precision, though there is room for improvement in its web application performance.

ACKNOWLEDGMENT

We would like to thank the students and staff of the Sensing Technology Department Republic Indonesia Defense University, Indonesia, for their support of this research.

REFERENCES

[1] A. S. Satish, A. Mangal, and P. Churi, "A systematic review of passenger profiling in airport security systems: Taking a potential case study of CAPPS II." *Journal of Transportation Security,* vol. 16, no. 1, pp. 8, 2023, doi: 10.1007/s12198-023-00260-6

[2] J. L. Austin, and A. Leander. "The State of the Sublime: Aesthetic Protocols and Global Security." *Millennium*, vol. 50, no. 3, pp. 730-759, 2022, doi: 10.1177/0305829822111728

[3] M. R. Torres-Soriano, and M. Toboso-Buezo, "Five Terrorist Dystopias", *The International Journal of Intelligence, Security, and Public Affairs*, vol. 21, no. 1, pp. 49–65, 2019, doi: 10.1080/23800992.2019.1598094.

[4] K. Man, and J. Chahl, "A Review of Synthetic Image Data and Its Use in Computer Vision." *Journal of Imaging*, vol. 8, no. 11, pp. 310, 2022, doi: 10.3390/jimaging8110310

[5] Y. Li, Y. Zuo, H. Song and Z. Lv, "Deep Learning in Security of Internet of Things," in IEEE Internet of Things Journal, vol. 9, no. 22, pp. 22133-22146, 15 Nov.15, 2022, doi: 10.1109/JIOT.2021.3106898M.

[6] S. A. Singh, and K. A. Desai. "Automated surface defect detection framework using machine vision and convolutional neural networks" *Journal of Intelligent Manufacturing,* vol. 34, no. 4, pp. 1995-2011, 2023, doi: 10.1007/s10845-021-01878-w

[7] X. Yang, J. Yan, W. Wang, S. Li, B. Hu, and J. Lin. "Brain-inspired models for visual object recognition: an overview" *Artificial Intelligence Review*, vol. 55, no. 7, pp. 5263-5311, 2022, doi: 10.1007/s10462-021-10130-z

[8] E. Özbay, "An active deep learning method for diabetic retinopathy detection in segmented fundus images using artificial bee colony algorithm" *Artificial Intelligence Review,* vol. 56, no. 4, pp. 3291-3318, 2023, doi: 10.1007/s10462-022-10231-3

[9] L. Alzubaidi, J. Zhang, A. J. Humaidi, A. Al-Dujaili, Y. Duan, O. Al-Shamma, J. Santamaría, M. A. Fadhel, M. Al-Amidie, and L. Farhan, "Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions." *Journal of big Data*, vol. 8, no. 1, pp. 53, 2021, doi: 10.1186/s40537-021-00444-8

[10] G. W. Lindsay, "Convolutional neural networks as a model of the visual system: Past, present, and future." *Journal of cognitive neuroscience*, vol. 33, no. 10, pp. 2017-2031, 2021, doi: 10.48550/arXiv.2001.07092
[11] T. Zhang, W. Aftab, L. Mihaylova, C. Langran-Wheeler, S. Rigby, D. Fletcher, S. Maddock, and G. Bosworth. "Recent advances in video analytics for rail network surveillance for security, trespass and suicide prevention—A survey." *Sensors,* vol. 22, no. 12, pp. 4323, 2022, doi: 10.3390/s22124324
[12] K. N. Babu, and S. Manne. "An Automatic Student Attendance Monitoring System Using an Integrated HAAR Cascade with CNN for Face Recognition with Mask." *Traitement du Signal*, vol. 40, no. 2, pp. 743, 2023, doi: 10.18280/ts.400234
[13] W. Liu, J. Qian, Z. Yao, X. Jiao, J. Pan, “Convolutional Two-Stream Network Using Multi-Facial Feature Fusion for Driver Fatigue Detection”, *Future Internet*, vol. 11, no. 5, pp. 115, 2019, doi: 10.3390/fi11050115
[14] E. Ouassam, Y. Dabachine, N. Hmina, & B. Bouikhalene, “Improving the efficiency and security of passport control processes at airports by using the R-CNN object detection model”. *Baghdad Science Journal*, vol. 21, no. 2, pp. 0524-0524, 2024, doi: 10.21123/bsj.2023.8546
[15] I. Karamitsos, M. Papadaki, K. Al-Hussaeni, and A. Kanavos. "Transforming Airport Security: Enhancing Efficiency through Blockchain Smart Contracts." *Electronics*, vol. 12, no. 21, pp. 4492, 2023, doi: 10.3390/electronics12214492
[16] J. H. Kim, W. Song, and H. C. Lee. "Exploring the Determinants of Travelers’ Intention to Use the Airport Biometric System: A Korean Case Study." *Sustainability*, vol. 15, no. 19, pp. 14129, 2023, doi: 10.3390/su151914129
[17] Z. Guo, and L. Kennedy. "Policing based on automatic facial recognition." *Artificial Intelligence and Law,* vol. 31, no. 2, pp. 397-443, 2023, doi: 10.1007/s10506-022-09330-x
[18] A. Rahim, Y. Zhong, T. Ahmad, S. Ahmad, P. Pławiak, M. Hammad, “Enhancing Smart Home Security: Anomaly Detection and Face Recognition in Smart Home IoT Devices Using Logit-Boosted CNN Models”, *Sensors*, vol. 23, no. 15, pp. 6979, 2023, doi: 10.3390/s23156979
[19] R. Weiskopf, and H. K. Hansen. "Algorithmic governmentality and the space of ethics: Examples from ‘People Analytics’." *human relations,* vol. 76, no. 3, pp. 483-506, 2023, doi: 10.1177/00187267221075346
[20] J. Xu, “A deep learning approach to building an intelligent video surveillance system”. *Multimedia Tools and Applications*, vol. 80, no. 4, pp. 5495-5515, 2021, doi: 10.1007/s11042-020-09964-6
[21] Y. Xie, F. Zaccagna, L. Rundo, C. Testa, R. Agati, R. Lodi, D. N. Manners, and C. Tonon. "Convolutional neural network techniques for brain tumor classification (from 2015 to 2022): Review, challenges, and future perspectives." *Diagnostics*, vol. 12, no. 8, pp. 1850, 2022, doi: 10.3390/diagnostics12081850
[22] R. Gulakala, B. Markert, & M. Stoffel, “Generative adversarial network based data augmentation for CNN based detection of Covid-19”. *Scientific Reports*, vol. 12, no. 1, pp. 19186, 2022, doi: 10.1038/s41598-022-23692-x
[23] P. Oza, P. Sharma, S. Patel, F. Adedoyin, and A. Bruno. "Image augmentation techniques for mammogram analysis." *Journal of Imaging*, vol. 8, no. 5, pp. 141, 2022, doi: 10.3390/jimaging8050141
[24] M. Khairudin, R. Mahaputra, W. Wiharto, Y. Mufidah, L. A. M. Huda, R. A. Reswara, & G. J. Hartanti, “Early detection of diabetes potential using cataract image processing approach”. *SINERGI*, vol. 28, no. 1, pp. 55-62, 2024, doi: 10.22441/sinergi.2024.1.006.
[25] J. Lu, L. Tan, and H. Jiang, "Review on convolutional neural network (CNN) applied to plant leaf disease classification." *Agriculture*, vol. 11, no. 8, pp. 707, 2021, doi: 10.3390/agriculture11080707
[26] Z. Vujovic. "Classification model evaluation metrics." *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 6, pp. 599-606, 2021, doi: 10.14569/IJACSA.2021.0120670PDF
[27] S. Sharma, and S. Kumar. "The Xception model: A potential feature extractor in breast cancer histology images classification." *ICT Express* 8, no. 1, pp. 101-108, 2022, doi: 10.1016/j.icte.2021.11.010
[28] B. Wang, L. Dai, and B. Liao. "System architecture design of a multimedia platform to increase awareness of cultural heritage: A case study of sustainable cultural heritage." *Sustainability*, vol. 15, no. 3, pp. 2504, 2023, doi: 10.3390/su15032504

[29] F. Sonata, “Pemanfaatan UML (Unified Modeling Language) dalam perancangan sistem informasi e-commerce jenis customer-to-customer”. *Jurnal Komunika: Jurnal Komunikasi, Media Dan Informatika*, vol. 8, no. 1, pp. 22-31, 2019, doi: 10.31504/komunika.v8i1.1832

[30] A. Zafar, M. Aamir, N. M. Nawi, A. Arshad, S. Riaz, A. Alruban, A. K. Dutta, and S. Almotairi. "A comparison of pooling methods for convolutional neural networks." *Applied Sciences*, vol. 12, no. 17, pp. 8643, 2022, doi: 10.3390/app12178643

[31] Z. Wang, Y. Yin, and R. Yin. "Multi-tasking atrous convolutional neural network for machinery fault identification." *The International Journal of Advanced Manufacturing Technology*, vol. 124, no. 11-12, pp. 4183-4191, 2023, doi: 10.1007/s00170-022-09367-x

[32] C. F. G. D. Santos, Claudio Filipi Gonçalves Dos, and J. P. Papa. "Avoiding overfitting: A survey on regularization methods for convolutional neural networks." *ACM Computing Surveys (CSUR)*, vol. 54, no. 10s, pp. 1-25. 2022, doi: 10.48550/arXiv.2201.03299

[33] K. Aziz, I. K. Trisiawan, K. D. Suyasmini, Z. Iklima, & M. Yunita, “Multilabel image analysis on Polyethylene Terephthalate bottle images using PETNet Convolution Architecture”, SINERGI, vol. 27, no. 2, pp. 163-170, 2023, doi: 10.22441/sinergi.2023.2.003

[34] S. Khosa, A. Mehmood, and M. Rizwan. "Unifying Sentence Transformer Embedding and Softmax Voting Ensemble for Accurate News Category Prediction." *Computers*, vol. 12, no. 7, pp. 137, 2023, doi: 10.3390/computers12070137

[35] B. Tej Chinimilli, A. T., A. Kotturi, V. Reddy Kaipu and J. Varma Mandapati, "Face Recognition based Attendance System using Haar Cascade and Local Binary Pattern Histogram Algorithm," *2020 4th International Conference on Trends in Electronics and Informatics (ICOEI)(48184),* Tirunelveli, India, 2020, pp. 701-704, doi: 10.1109/ICOEI48184.2020.9143046.

[36] R. M. Prakash, N. Thenmoezhi and M. Gayathri, "Face Recognition with Convolutional Neural Network and Transfer Learning," *2019 International Conference on Smart Systems and Inventive Technology (ICSSIT)*, Tirunelveli, India, 2019, pp. 861-864, doi: 10.1109/ICSSIT46314.2019.8987899.

[37] P. Lu, B. Song, and L. Xu, ‘Human face recognition based on convolutional neural network and augmented dataset’, *Systems Science & Control Engineering*, vol. 9, no. sup2, pp. 29–37, 2021. doi: 10.1080/21642583.2020.1836526

[38] R. Muwardi, J. M. R. Permana, H. Gao, M. Yunita,” Human Object Detection for Real-Time Camera using Mobilenet-SSD,” *Journal of Integrated and Advanced Engineering (JIAE)*, vol. 3, no. 2, pp. 141-150. 2023, doi: 10.5162/jiae.v3i2.108