e-*ISSN 2548-6438*
p-ISSN 2614-7173

A CNN-BASED APPROACH FOR DEVELOPING AN AUTOMATED FACE DETECTION SYSTEM

Muhammad Ridha Albaar [1*], Seh Turuy [2], M. Arhan Darmawan [3], Muhammad Fikri Ramadhan [4], Gamal Hadi [5]
[1,2,3,4]Engineering, Informatics Study Program, Khairun University Ternate
[5]Disaster Management Agency of North Maluku Province
Email: ridha.albaar@unkhair.ac.id

Abstract

Facial detection technology has become increasingly significant across various domains, including security, surveillance, biometrics, and human–computer interaction. This study aims to develop a facial detection system using a Convolutional Neural Network (CNN) approach to recognize and classify facial features from image data. A quantitative experimental method was employed, integrating deep learning modeling with software engineering principles. The system was implemented in MATLAB using a graphical user interface (GUI), and the model was trained on a dataset of 800 facial expression images. Data preprocessing involved image resizing, pixel normalization, and data augmentation to enhance model robustness. The dataset was divided into training, validation, and testing sets to ensure reliable model evaluation and generalization. The CNN model was trained and empirically tested to assess its performance under various conditions. The results indicate that the proposed system is capable of accurately detecting facial regions, even under natural lighting and complex backgrounds. The model achieved high accuracy in face identification, as evidenced by correct bounding box localization in the output interface. Furthermore, the system demonstrated stable performance across different facial orientations. However, several limitations remain, including the need for improved multi-face detection, real-time processing optimization, and broader evaluation across diverse expressions and demographic variations. In conclusion, the CNN-based facial detection system demonstrates strong effectiveness and accuracy, indicating its potential applicability in real-world visual recognition systems.

Keywords: CNN, Face Detection, Deep Learning, Influence. Image Recognitiion, Matlab

INTRODUCTION

The face is a fundamental component in the individual identification process. Facial expressions can represent a person's emotional state and feelings, especially in the context of social interactions [1]. As a form of nonverbal communication, facial expressions play a strategic role in building understanding between individuals.

Through facial expressions, various psychological aspects such as emotions, feelings, and even a person's cognitive processes can be implicitly communicated.

Rapid developments in digital image processing and artificial intelligence have led to the development of increasingly accurate and efficient systems for detecting and classifying facial expressions. One approach that has proven effective is the use of Convolutional Neural Networks (CNN), which are known for their superior feature extraction capabilities and high accuracy in facial expression recognition tasks [2].

Convolutional Neural Network (CNN) is an artificial neural network architecture specifically designed for image-based data processing. CNN consists of several layers, such as convolutional layers, pooling layers, and fully connected layers, which hierarchically function to extract important features from images and produce output in the form of classification [3]. CNN-based approaches in facial expression detection have shown significant performance and have the potential for wide application in various fields [4].

Face detection is a rapidly growing topic in the fields of pattern recognition and computer vision. This technology has a variety of applications, including security systems, surveillance, biometrics, and facial recognition on mobile devices. In recent years, the use of artificial neural networks, especially CNNs, has demonstrated excellent performance in various image recognition tasks. In addition, MATLAB, equipped with various deep learning toolboxes, has become a reliable and widely used platform in developing face detection models [5].

Based on this background, this study aims to implement the Convolutional Neural Network (CNN) algorithm in the process of identifying certain patterns or objects. The main focus of this study is to analyze the effectiveness of implementing the CNN algorithm in recognizing visual patterns accurately and efficiently [6].

In line with the background formulation that has been described, this study focuses on the main question regarding the accuracy level of the CNN algorithm in recognizing and identifying human facial expressions based on digital images. This research question forms the basis for designing and implementing experiments, with the aim of evaluating the performance of the CNN model in the context of automatic facial image processing [6,7].

LITERATURE REVIEW

TensorFlow is used as a visualization tool that allows the design of the process flow of a human facial expression detection system to be presented in a structured and systematic manner. Through graphical representation, TensorFlow illustrates each stage in the CNN algorithm, from the image input process to the expression classification stage. The use of TensorFlow helps improve the understanding of the system workflow and facilitates the logical analysis of the developed model [9,11].

RESEARCH METHODS

This study applies a quantitative experimental approach combined with software engineering methods and deep learning algorithm modeling to detect facial expressions using a Convolutional Neural Network (CNN). This approach was chosen because it can provide measurable analysis of the system's performance in recognizing facial expressions from digital images [8]. In research on human facial expressions, the diversity of data collection locations is an important aspect that needs to be considered. Each environment has different characteristics and can influence how individuals express their emotions. Public spaces, such as parks, streets, or shopping malls, tend to generate more spontaneous expressions due to extensive social interaction. Conversely, private spaces such as homes allow for more natural, personal, and less stressful expressions. By considering these various contexts, this study seeks to obtain a more comprehensive and realistic representation of facial expressions [10].

In the development of a CNN-based facial expression detection system, empirical testing was carried out to evaluate the effectiveness of the model built. This testing involved a series of structured stages designed to assess the model's ability to accurately recognize and classify facial expressions. The evaluation is carried out based on the results of training and testing using facial image data, so that it can reflect the actual performance of the system. Each stage was designed to ensure that the system not only functioned conceptually but also had reliability in practical implementation [12].

The initial stage in the facial expression recognition process is the selection of a representative dataset. The dataset used must be able to reflect the diversity of human expressions in various conditions, including variations in lighting, background, shooting angle, and differences in individual characteristics. This diversity is important to improve the generalization ability of the model to real-world conditions [13,15].

Before being used in the training process, facial image data needs to go through a preprocessing stage. This stage includes image resizing, pixel value normalization to maintain data consistency, and data augmentation implementation to enrich data variation and improve model performance. This process aims to ensure that the data matches the needs of the CNN architecture used [14].

Next, the dataset is divided into three main parts, namely training data, validation, and testing. Training data is used to build a model to recognize facial expression patterns, while validation data serves to optimize model parameters and prevent overfitting. After the training process is complete, the test data is used to objectively evaluate the model's performance against previously unused data. This division is important to ensure good generalization ability of the model [1,15].

The next stage is designing a CNN architecture that suits the research needs. The choice of architecture, such as ResNet, VGG, or other architectures, is an important factor that influences the level of accuracy and quality of the system in recognizing facial expressions. Selecting the right architecture allows the model to optimally extract features from facial images [16].

Once the model architecture is determined, the training process is carried out using the prepared training data. At this stage, the model parameters are updated iteratively to minimize the prediction error. This process allows the model to gradually learn important patterns contained in facial image data [7,10].

After the model reaches optimal performance in the validation stage, testing is carried out using separate test data. This test aims to evaluate the model's ability to recognize facial expressions under new conditions that were not involved in the training process, thus providing an objective picture of the model's performance.

The final stage in this research is the analysis of the test results to identify the strengths and limitations of the model. This analysis includes evaluation of prediction errors, visualization of classification results, and comparison of model performance with previous research or other relevant architectures. This stage is important to provide a deeper understanding of the effectiveness of the model and its future development opportunities [3,7].

Based on the evaluation results, conclusions can be drawn regarding the level of effectiveness of the CNN-based facial expression

detection system in recognizing human emotions. Furthermore, the results of this study were compared with previous studies to assess the contributions and advantages offered, thus demonstrating the added value and relevance of this research in the development of image-based emotion recognition technology.

RESULTS AND DISCUSSION

The image dataset used in this study was obtained using an image acquisition system using a mobile phone camera and consisted of 800 facial expression images.

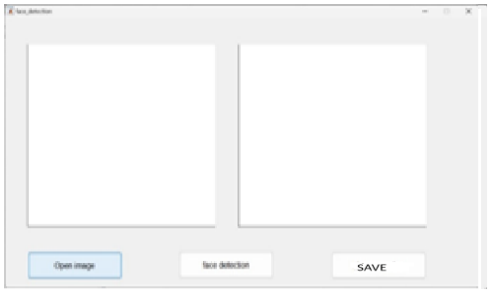

Figure 1. Initial input before inserting a photo

The illustration represents the graphical user interface (GUI) of a face detection application at based on Convolutional Neural Network (CNN). If you need to install MATLAB, you can't install App Designer and use GUIDE, if you shouldn't use your environment on your own and what you need to do. The Interface application is divided into two parts of the home for the display of images (image display axes/panels). This field is also added to the list of original images created by the "Open Image" keyword. Similarly, this area is also useful for target acquisition detection, the inheritance system is a small part of the detection system of the identity set being developed. This application is equipped with several key components that support the operation of the system. The "Open Image" button is used to load an image from the user's device into the system, which is then displayed in the left panel. The "Face Detection" button runs the face detection process using the pre-trained CNN model. After this operation is executed, the system analyzes the image and displays the detection results, typically marked with a calibration box, in the right panel. Then, click on the Save Image button to save the image and then click on the Detect button to save the image. The feature will be displayed in the document and the image will be processed by the system.

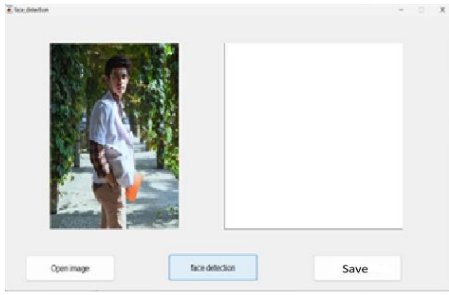

Figure 2. Face Detection System Interface

The image shows the interface of a convolutional neural network (CNN)-based face detection

application before the detection process begins. The application was developed using MATLAB, likely via App Designer or a similar visual programming environment, designed to support intuitive user interaction. At this stage, the input image has been successfully loaded into the system and displayed in a designated area, ready for further processing. This represents a transitional phase in the system flow, where the input data is available and the system is ready to perform the face detection process by activating the "Face Detection" button. This phase is a crucial part of the system workflow, as it serves as the starting point before the face identification and labeling process begins using the trained CNN model.

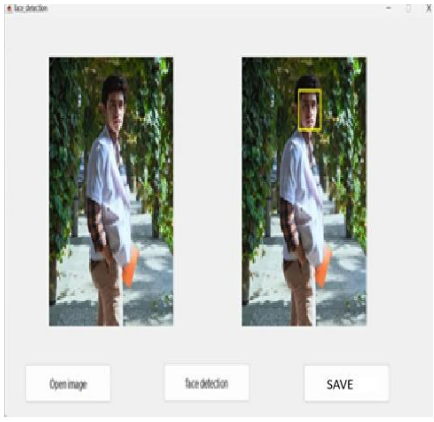

Figure 3. Face Detection

The image shown is a visual documentation of the implementation of a Convolutional Neural Network (CNN) based face detection system developed through a simple and interactive graphical interface. This system is designed to automatically recognize human faces from user-inputted images. The application interface consists of two main panels displayed side by side. The left panel is used to display the original image loaded via the "Open Image" button. The image shows an individual standing in an outdoor area with bright natural lighting conditions and a relatively complex background. The position of the face in the image is not completely facing the camera, thus representing the real conditions in the facial data acquisition process.

Once the image is loaded, the user can activate the detection process by pressing the "Face Detection" button. The detection results are then displayed in the right panel, where the system successfully identifies the facial area by marking it with a bounding box. This successful detection demonstrates that the CNN model is capable of effectively recognizing faces, despite variations in facial orientation and lighting complexity. Furthermore, a "Save Image" button allows users to save the detection results for documentation or further development. Overall, the system interface reflects a systematic and efficient workflow, and is easy to use without requiring in-depth technical expertise.

Empirical testing in the CNN-based facial expression detection

system research was conducted through a series of structured steps to directly evaluate the model's performance. This step begins with selecting a representative dataset, which should encompass a wide variety of human facial expressions, such as happiness, sadness, anger, surprise, fear, and neutrality. Furthermore, the dataset should consider variations in lighting, background, shooting angle, and ethnic and age diversity to improve the model's generalizability.

The image data used then goes through a preprocessing stage to suit the CNN training needs. This process includes normalizing pixel values, adjusting the image size to specific dimensions, and applying data augmentation techniques such as rotation, image flipping, and contrast adjustment to increase data diversity and reduce the risk of overfitting. Next, the dataset is divided into three subsets, namely training data (training set), validation data (validation set), and testing data (testing set), each of which has a different role in the model development and evaluation process.

The next step is determining the appropriate CNN architecture, whether using popular architectures like ResNet, VGGNet, or DenseNet, or a custom-designed architecture tailored to the research needs. Key parameters such as the number of layers, filter size, activation function, and regularization technique are considered to achieve optimal performance. The model is then trained using the training data through an iterative process with optimization algorithms like Adam or Stochastic Gradient Descent (SGD), and using a loss function like categorical cross-entropy to measure prediction error.

During the training process, evaluations are performed periodically using validation data to monitor model performance and adjust hyperparameters. Techniques such as early stopping or learning rate scheduling can be applied to prevent overfitting. Once the model reaches optimal performance, final testing is performed using completely separate test data to obtain an objective evaluation of the model's ability to recognize new data.

The test results are then analyzed to identify the system's strengths and limitations. This analysis may include the use of a confusion matrix to view the distribution of predictions, as well as measuring metrics such as precision, recall, and F1-score on each expression class. In addition, visualization of prediction results and comparison with previous research are also carried out to provide a comprehensive picture of model performance.

Facial recognition is a method in the biomedical field that is used to identify or verify an individual's identity through facial pattern analysis. This method utilizes the unique biological characteristics of the human face, such as the shape of the eyes, nose, mouth, and the distance between facial features. Humans naturally have a good ability to recognize faces, but this ability has limitations, especially in situations involving large amounts of data or long-term observation. Therefore, technology-based facial recognition systems have been developed to improve efficiency and accuracy in various applications, such as security, surveillance, and forensics.

The process of face recognition in a computing system begins with image acquisition, followed by the extraction of important features from the face. The feature data is then compared with the available database to determine identity match. The success of this process is greatly influenced by the quality of the input image and the algorithm used. Unlike humans, computers lack natural visual perception, so factors such as lighting, shooting angle, facial expression, and image resolution significantly impact system accuracy. Therefore, selecting the right algorithm is a crucial factor in improving the performance of image-based facial recognition systems.



Figure 4. Graphical User Interface Development Environment

The code shown in the image is the initial part of the MATLAB program used to build a graphical user interface (GUI) for the face detection system. This program was developed using GUIDE (Graphical User Interface Development Environment) available in MATLAB. The main function in the program is face_detection, which plays a role in the initialization process and operational management of the face detection GUI. In general, this function is designed with a flexible approach through the use of dynamic input and output parameters, namely varargin and varargout. This approach allows functions to handle a variety of calling scenarios, such as opening a GUI interface, executing certain callbacks, or setting system display and behavior properties according to user requirements.

This program is implemented in singleton mode, which ensures that only one face_detection GUI window is active at a time. This approach aims to avoid interface

duplication, which could potentially lead to user confusion or data conflicts during processing. If a user attempts to open the GUI more than once, the system automatically redirects focus to the active window without creating a new instance. Furthermore, this GUI supports the application of property-value pairs at runtime, allowing customization of the system's appearance and behavior before the interface is displayed. All of these initial configurations are processed through the face_detection_OpeningFcn function, which is the first function to be executed and plays a role in initializing the system as a whole. Thus, this code structure represents a crucial foundation for ensuring the GUI runs stably, is structured, and adapts to various user needs.

The implementation of the developed Convolutional Neural Network (CNN)-based face detection system demonstrates strong ability to detect faces from input images. The system interface provides three main features that support the interactive face detection process. The first feature loads images from the user's device into the system. In the example shown, the system successfully displays an image of an individual in an outdoor environment with natural lighting, demonstrating the successful performance of the data input mechanism.

After the image is loaded, the next feature is used to run the face detection process by utilizing the trained CNN model. The results displayed in the output panel demonstrate that the system is capable of identifying and marking facial areas using bounding boxes. This success indicates that the CNN model has good generalization capabilities to test data outside the training set. Detection remains accurate despite variations in facial orientation, non-uniform lighting conditions, and complex backgrounds. This indicates that the model has been trained using sufficiently representative data or is supported by a CNN architecture that has strong and deep feature extraction capabilities.

The third feature allows users to save the face detection results to the device. This functionality not only supports visualization of results but also enables further utilization, such as creating new datasets, conducting additional analysis, and documenting research findings. Overall, the system is able to display the face detection process visually and relatively in real-time with an adequate level of accuracy, even on images with varying object

complexity and environmental conditions.

However, there are several aspects that require attention for further development. First, the multi-face detection capability has not been comprehensively tested on images containing more than one individual. Second, further testing of variations in facial expression, age, and pose is needed to evaluate the model's robustness to changes in data characteristics. Third, optimization of system performance in real-time scenarios, such as the use of live cameras, needs to be carried out to assess the speed and efficiency of detection under actual operational conditions. In the initial stage before the labeling process is carried out, all images that have been collected are first uploaded to the Roboflow platform. This stage aims to facilitate a systematic data annotation process and prepare a structured dataset before use in CNN model training.



Figure 5. Upload Dataset

In the labeling stage using the Roboflow platform, each object in the image is given an annotation in the form of a bounding box or marker framework to identify the relevant area. After the annotation process is complete, the system automatically classifies and displays the detected emotion categories, such as happy, sad, angry, and afraid.

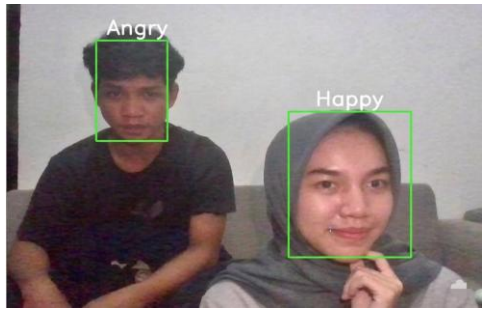

Figure 6. Dataset labelling

The image processing stage using the Roboflow platform includes adjusting aspects of color, shape, and size of the image, as well as applying preprocessing and data augmentation techniques. The process of resizing the image to dimensions of 320 × 320 pixels was carried out as an effort to optimize model performance while reducing the computational load, especially on GPU usage during the training process.

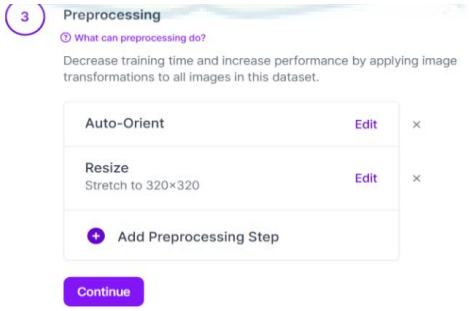

Figure 7. Image Preprocessing

The model training process was carried out using the Google

Colab platform with a dataset division of 80% for training data and 20% for testing data. This research utilizes a MobileNet-based Convolutional Neural Network (CNN) architecture with 150 epochs, which produces an average accuracy rate of 78%. The trained model is then saved in the model_checkpoint2.h5 file format for use in the facial expression detection stage.



Figure 8. Training Process

A more in-depth evaluation of model performance was performed using precision, recall, and F1-score metrics. The test results show that the model achieved the best performance in the happy emotion category, with an F1-score value of 0.93. This indicates that the model has excellent ability to detect and classify happy emotions, with a high balance between precision and recall.

Relatively good performance was also shown in the fear category with an F1-score of 0.76, as well as the anger category with an F1-score of 0.70, which reflects a still quite adequate level of detection success. However, the model showed significant weakness in the sad category, where the recall value only reached 0.49. This value indicates that the model still often fails to accurately identify sad emotions. This condition is likely due to the similarity in visual characteristics between sad expressions and other emotions, the limited variety of training data in this category, and the influence of varying lighting conditions on the images. Overall, the model performance can be categorized as quite good with an average accuracy value of 0.78, an average precision (macro average) of 0.78, a recall of 0.76, and an F1-score of 0.75. While these results demonstrate adequate performance across several emotion categories, there is still room for improvement, particularly in categories with low detection rates. Improvement efforts that can be made include increasing the variety of training datasets, applying more specific data augmentation techniques, and adjusting or developing the CNN architecture to improve overall model performance.

5/5 [==========] – 05 69ms/step

Confussion Matrix
[17 1 1 8]
[9 17 1 8]
[2 1 46 1]
[2 1 1 34]
Classfication Report :

	Precesion	Recall	f1-score	support
Anggry	0,68	0,73	0,70	57
Sad	0,85	0,49	0,63	35
Like	0,94	0,92	0,92	50
Afraid	0,67	0,89	0,77	38
Accuracy			0,78	160
Macro avg	0,78	0,76	0,75	160
Weighted avg	0,79	0,78	0,77	160

Figure 9. Training Data Results

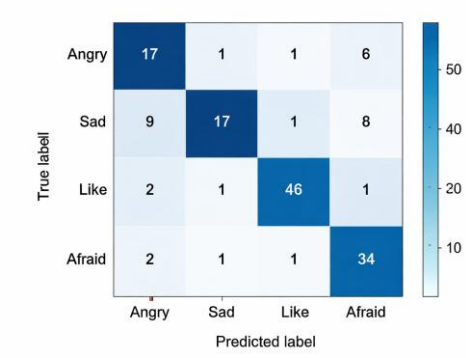

Figure 10. Confusion Matrix

The developed classification model showed relatively good performance with an accuracy rate of 78%, which indicates that most of the test data was successfully predicted correctly. However, further analysis through confusion matrix and classification report showed that there were variations in performance in each emotion class. The Like class shows the best performance, as indicated by a precision value of 0.94, a recall of 0.92, and an F1-score of 0.92. These results indicate that the model is not only able to recognize data in the Like class accurately, but also has a low classification error rate against other classes. In other words, the feature characteristics of this class were relatively clear and easy for the model to learn, resulting in stable and consistent predictions.

Conversely, the Sad class proved the most challenging for the model. Despite its relatively high precision of 0.85, the recall was only 0.49. This indicates that while predictions for the Sad class tended to be accurate, the model failed to identify a significant portion of data that actually belonged to that class. Much of the data labeled Sad was misclassified into other classes, such as Angry or Afraid. This condition indicates that the model is having difficulty recognizing the distinctive pattern of sad expressions, possibly due to the similarity of visual features to other emotion classes or the lack of representativeness of the training data in that category.

Furthermore, the model also tends to experience confusion (misclassification) between several emotion classes, particularly between Sad, Angry, and Afraid. This is evident from the high number of misclassifications between these three classes in the confusion matrix. Conceptually, these three emotions do have relatively similar characteristics, both in terms of facial expressions and the representation of features captured

by the model. As a result, the model has difficulty distinguishing clear boundaries between these classes.

This phenomenon shows that the feature space used is not yet able to optimally separate emotion classes. Models tend to only capture general patterns of negative emotions, but are not yet sensitive enough to identify specific differences between emotions such as sadness (Sad), anger (Angry), and fear (Afraid). Therefore, further development is needed, both through improving the quality and variety of the dataset and refining the model architecture, to enhance the system's discriminatory ability in facial emotion classification.

CONCLUSION

Based on the design, implementation, and testing results, the developed Convolutional Neural Network (CNN)-based face detection system has proven capable of recognizing human faces with a fairly good level of accuracy. Detection success is demonstrated by the system's ability to identify facial areas using bounding boxes, despite images containing varying lighting, complex backgrounds, and non-frontal shooting angles.The model used, a MobileNet architecture with 150 epochs, achieved an average accuracy of 78%, indicating that the CNN approach is quite effective in the task of face detection and expression recognition. Further evaluation results showed that the model performed optimally on the Happy emotion, and performed quite well on the Fear and Angry emotions. However, the model still showed limitations in recognizing the Sad emotion, as indicated by a relatively low recall value.Overall, this research demonstrates that the CNN approach has strong potential in the development of digital image-based face detection and emotion recognition systems, and is capable of producing fairly stable results across a variety of input conditions

REFERENCES

[1].S. Widodo, D. Setiawan, T. Ridwan, and R. Ambari, "*Designing a human emotion detection system based on facial expressions using the VGG16 algorithm*," Syntax: Journal of Informatics, 2022, doi: *10.35706/SYJI.V11I01.6594.*

[2] A. K. Muttaqiin, H. Yuana, and M. T. Chulkamdi, "*Implementation of a convolutional neural network algorithm for facial expression recognition*," Journal of Information Systems and Informatics Engineering Research (JURASIK), vol. 8, no. 2, pp.

772–792, 2023.

[3] P. A. Nugroho, I. Fenriana, and R. Arijanto, "*Implementation of deep learning using convolutional neural networks (CNN) on human expression*," Algor, vol. 2, no. 1, pp. 12–21, 2020.

[4] R. D. Ramadhani, N. A. Thohari, C. Kartiko, A. Junaidi, T. G. Laksana, and N. A. Setya Nugraha, "*Accuracy optimization of the convolutional neural network method for waste type identification*," RESTI Journal (Systems Engineering and Information Technology), vol. 5, no. 2, pp. 312–318, 2021, doi: *10.29207/RESTI.V5I2.2754.*.

[5] I. R. Pramutighna and A. Hermawan, "*Recognizing potential toxins and improving food safety in mushrooms using convolutional neural networks,*" Budidarma Informatics Media Journal, vol. 7, pp. 1716–1726, 2023, doi:*10.30865/MIB.V7I4.6372*

[6] A. Rohim, Y. A. Sari, and Tibyani, "*Convolutional neural network (CNN) for classifying traditional food images*," Journal of Information Technology and Computer Science Development, vol. 3, no. 7, pp. 7038–7042, 2019.

[7] A. Tiara Sari and E. Haryatmi, "*Application of convolutional neural network deep learning in detecting dry corn seed images*," RESTI Journal (System Engineering and Information Technology), vol. 5, no. 2, pp. 265–271, 2021.

[8] Syahromi, "*Facial expression recognition using the MTCNN (Multi-Task Cascaded Convolutional Neural Network) algorithm and VGG16*," Applications and Literature Analysis, Fasilkom UI, pp. 4–25, 2022.

[9] M. R. D. Ulhaq, M. A. Zaidan, and D. Firdaus, "*Real-time facial expression recognition using the Android-based SSD MobileNet method*," Journal of Technology and Informatics (JOTI), vol. 5, no. 1, pp. 48–52, 2023, doi: *10.37802/JOTI.V5i1.387.*

[10] M. Fasounaki, E. B. Yüce, S. Öncül, and G. Ince, "*CNN-based text-independent automatic speaker identification using short utterances,*" in Proc. 6th Int. Conf. Computer Science and Engineering (UBMK), 2021, pp. 413–418, doi: *10.1109/UBMK52708.2021.9559031.*

[11] S. N. H. Duli, "*Analysis of machine learning and deep* learning on *dataset size*: Case study of text sentiment analysis," 2024..

[12] A. Gupta, Narwaria, and D. R. Pratap, "*Review on deep learning handwritten digit recognition using convolutional neural network,*" International Journal of Recent Technology and Engineering (IJRTE), vol. 9, no. 5, pp. 245–247, 2021, doi:*10.35940/IJRTE.E5287.019 521.*

[13] F. Amaludin, M. I. Zulfa, and H. Siswantoro, "*The effect of hyperparameter tuning on the performance of MobileNetV2 with transfer learning for skin disease detection*," SINTA Journal of Information Systems and Computational Technology, vol. 2, no. 2, pp. 84–94, Apr. 2025, doi: *10.61124/sinta.v2i2.43.*

[14]Yenila F, Marfalino H, Defit S. *Model analisis machine learning dengan pendekatan deep learning dalam penentuan kolektabilitas*. Jurnal Sains dan Teknologi. 2023;12(2):403–414. doi:*10.23887/jstundiksha.v12i2. 54035.*

[15]Prasetyo S, Dewayanto T. *Application of machine learning, deep learning, and data mining in detecting financial statement fraud: A systematic literature review*. Diponegoro Journal of Accounting. 2024;13(3):1–12.

[16] R. G. Wardhana, G. Wang, and F. Sibuea, "*Application of machine learning in predicting disease case rates in Indonesia*," Journal of Information System Management (JOISM), vol. 5, no. 1, 2023.