Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 10, Issue 1, Januari 2026
DOI : https://doi.org/10.33395/sinkron.v10i1.15682
e-ISSN : 2541-2019
p-ISSN : 2541-044X

A Comparative Study of MobileNetV2 and ResNet50 for Multi-Class AI-Generated and Real Image Classification

I Gusti Ngurah Agus Ega Patria Pramudita[1)], I Gede Iwan Sudipa[2)*], Yuri Prima Fittryani[3)], Ida Bagus Ary Indra Iswara[4)], I Gusti Ayu Agung Mas Aristamy[5)]

[1,2*),3,4,5)] Informatika, Fakultas Teknologi dan Informatika, Institut Bisnis dan Teknologi Indonesia, Denpasar, Bali, Indonesia

[1)]egapatria162@gmail.com, [2)*]iwansudipa@instiki.ac.id, [3)]yuri.prima@instiki.ac.id , [4)]indraiswara@instiki.ac.id, [5)]agungmas.aristamy@instiki.ac.id

Submitted :Dec 8, 2025 | **Accepted** : Dec 29, 2025 | **Published** : Jan 06, 2026

Abstract: This study aims to classify AI-generated and real images using Convolutional Neural Network (CNN) architecture by comparing the performance of MobileNetV2 and ResNet50. Previous studies on AI-generated image detection have primarily focused on binary classification without explicitly analyzing object-level context in multi-class scenarios, leaving a gap in understanding model performance across diverse visual categories. The dataset consists of 23,941 images divided into two main classes of real and fake and five subclasses of human, animal, art, view, and vehicle. The training process employs data augmentation and a K-Fold Cross Validation strategy on the training and validation set to maintain balanced class proportions, while a separate unseen test set is used exclusively for final performance evaluation. Model evaluation is performed based on accuracy, precision, recall, and F1-score metrics on test data. The results showed that MobileNetV2 achieved the best accuracy of 89% at the 10th epoch, but experienced a decline in performance at the 30th and 50th epochs, indicating overfitting. In contrast, ResNet50 showed the most stable performance with the highest accuracy of 93% at the 30th epoch and consistently high precision, recall, and F1-score values. Thus, ResNet50 was found to be the most effective architecture for classification of AI-generated and real images on multi-class datasets, while MobileNetV2 remains relevant for implementation on devices with computational limitations.

Keywords: AI-Generated Images, Real Images, Deep Learning, Convolutional Neural Network (CNN), Image Classification.

INTRODUCTION

The involvement of Artificial Intelligence (AI) in human life is becoming increasingly inevitable in the era of the Industrial Revolution 4.0 (Słapczyński, 2022). AI is not just a branch of technology, but an effort to make machines capable of reasoning, learning, and acting in a human-like manner, with the aim of not only creating computer vision capabilities but also building systems that are able to understand the environment, adapt to change, and respond naturally so that technology is increasingly relevant to humans (Lin et al., 2023). The development of AI began with Alan Turing's question about the possibility of machines to think which gave birth to the Turing Test, then progressed from mathematical logic to machine learning and deep learning, and is now applied to natural language processing, facial recognition, and image processing (Hang Rai, 2024).

The surge in AI utilization is becoming more apparent with its integration into various human activities, especially with the emergence of AI Image Generators that are capable of automatic image generation through text commands and previous image manipulation using deep learning models, including the use of Diffusion Models to generate realistic images (Peng, 2024). However, this technology has also led to abuse, with many AI technologies being used for exploitation and harassment on the internet (Döring et al., 2024), as well as the creation of politically manipulated videos and fake news that spread quickly and are difficult to verify (Sophia LI, 2025). Visual manipulation is also used for disaster-related fake news that triggers mass panic (Komendantova & Erokhin, 2025), and raises issues of copyright infringement due to AI's ability to imitate visual works without permission (Ghiurău & Popescu, 2025).

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 10, Issue 1, Januari 2026
DOI : https://doi.org/10.33395/sinkron.v10i1.15682
e-ISSN : 2541-2019
p-ISSN : 2541-044X

These problems show the urgency of AI-based manipulated image detection technology because AI-made images are now so realistic that they are difficult to distinguish with the naked eye, so deep learning approaches, especially Convolutional Neural Network (CNN), are a promising solution. This research builds a classification model of AI artificial images and real images through the comparison of MobileNetV2 and ResNet50 architectures, where MobileNetV2 offers a lightweight structure through depthwise separable convolutions and inverted residual blocks making it more efficient for transfer learning (Hajar et al., 2025), while ResNet50 has a deeper network with residual blocks for richer feature extraction with higher computational requirements (Daviana et al., 2025).

The novelty of this research lies in its focus on testing the performance of both CNN architectures on a multi-class dataset that includes images of humans, animals, art, landscapes, and vehicles, both original and AI-generated versions. This approach allows for a more comprehensive evaluation of model performance than its predecessors. Scientifically, this research contributes by providing a comparison between MobileNetV2 and ResNet50 across various visual categories, analyzing model performance consistency, and assessing the effect of epoch number variation on accuracy stability.

This research is expected to contribute to the development of AI image detection technology, become a reference in choosing the optimal CNN architecture, and become a foundation for further research towards the implementation of AI image detection systems on websites or applications in the future.

LITERATURE REVIEW

Previous research shows various approaches to distinguish between real and artificial images. (Fatoni et al., 2025)developed deep learning models using ResNet, VGG16, and CNN with Error Level Analysis (ELA) pre-processing on the CASIA dataset, and found that ResNet provided the best accuracy. (Hakim et al., 2024)performed AI generation image classification through fine tuning the ResNet architecture, with a dataset of 120,000 images, and obtained the best performance on ResNet152 with F1-Score 0.963. (Suharyanto et al., 2024)compared ResNet50, VGG19, and Xception combined with Capsule Network for deepfake detection, and showed that ResNet50 achieved the highest test accuracy. (Mu et al., 2024)proposed a CNN model resembling VGG architecture to detect deepfake faces with 91% accuracy. Meanwhile, (Bahrul Subkhi et al., 2023)used VGG19 to distinguish between human and AI paintings, and achieved very high accuracy despite indications of overfitting.

Although various studies have successfully classified AI-generated and real images using various CNN architectures, most studies still have limitations, especially in terms of the datasets used and the classification approaches applied. Most previous studies only focused on binary classification, which is distinguishing between AI-generated images and real images, without touching on the recognition of sub-categories or image types in more detail (Li et al., 2022). This indicates that there is a lack of exploration into the diversity of distinguishable image types, which could potentially limit the application potential of the developed algorithms. One gap in the literature is the lack of research evaluating MobileNetV2 specifically in the context of AI image classification. For example, research by (Aljohani & Turki, 2022),shows that more complex CNN architectures such as DenseNet and ResNet have been explored for several applications, but no research highlights MobileNetV2 for AI-generated image classification. Other research shows how MobileNet is used in the context of skin disease recognition, emphasizing its efficiency in processing large images (Pradnya Dhuhita et al., 2023). However, this research does not specifically address the broader categorization of AI-generated images compared to real images.

Deeper convolutional neural network architectures, such as ResNet, tend to achieve better performance on complex image classification tasks because increased network depth enables richer hierarchical feature learning, capturing subtle texture inconsistencies and structural artifacts commonly found in AI-generated images. Recent survey studies report that residual-based deep CNNs maintain stable training behavior and stronger representation capacity on visually diverse datasets, whereas lightweight architectures such as MobileNet prioritize computational efficiency through parameter reduction, which may limit performance in multi-class scenarios with high visual variability (Khan et al., 2020).

This research gap will be filled by this study, which compares two CNN architectures, ResNet50 and MobileNetV2, in the classification of AI and real images on a multi-class dataset covering five subclasses: humans, animals, landscapes, artwork, and vehicles. In addition, a comprehensive evaluation is conducted using accuracy, precision, recall, and F1-score metrics to assess the model's ability to recognize various types of objects in more detail.

METHOD

This methodology section outlines the research stages in the development of AI artificial image classification models and real images using Convolutional Neural Network (CNN). This research begins with the collection of real and fake datasets. The dataset then goes through a preprocessing process, such as cropping, relabeling, renaming and dividing the data into train & validation and test to maintain evaluation consistency. The test set is separated at the beginning of the experiment and is not involved in the K-Fold Cross Validation process. K-Fold

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 10, Issue 1, Januari 2026
DOI : https://doi.org/10.33395/sinkron.v10i1.15682
e-ISSN : 2541-2019
p-ISSN : 2541-044X

Cross Validation is then performed on the train & validation data to ensure training and validation runs evenly across all data subsets. K-Fold Cross Validation is used solely for model selection and hyperparameter stability analysis, not for final performance evaluation. This process is complemented by image augmentation on the train data, to increase data diversity. This research utilizes transfer learning with two CNN architectures, namely MobileNetV2 and ResNet50, which are then trained using Adam's optimizer. Model performance is evaluated using accuracy, precision, recall, and F1-score metrics to comprehensively determine the model's capabilities. Final evaluation is conducted once on the unseen test set using the best selected model from the K-Fold process. The complete flow of research from dataset preparation to model evaluation is presented in the following Figure.

The CNN MobileNetV2 and ResNet50 architectures were selected due to their combination of efficiency and model depth. MobileNetV2 was chosen for its training speed and computational lightness, making it suitable for large datasets, while ResNet50 was chosen for its ability to handle vanishing gradients in very deep networks. K-Fold Cross Validation was used to reduce bias in limited datasets and ensure that model evaluation was evenly distributed across all data. Potential dataset bias was addressed by stratified splitting and data augmentation to keep the class distribution balanced, as well as by combining primary and secondary data so that the model did not learn from only one data source. The advantage of this method lies in its ability to distinguish between AI-generated and real images while recognizing the context of objects in the human, animal, view, art, and vehicle subclasses, allowing the model to learn more diverse features and improve generalization. By incorporating multi-class object categories, this method enables contextual object recognition in addition to binary AI–real discrimination, allowing the model to learn more diverse features and improve generalization.

All experiments were conducted on a personal computer equipped with an Intel Core i5-12500H CPU, without GPU acceleration. The implementation was carried out using Python version 3.12.10, TensorFlow version 2.17.0, with the Keras API 3.5.0**.** To ensure consistency in data partitioning, 42 random states were applied during dataset shuffling and Stratified K-Fold Cross Validation. Due to the stochastic nature of deep learning optimization, a global random seed for model initialization and training was not explicitly fixed. However, all experiments were performed using identical data splits, hyperparameter settings, and training configurations to ensure fair and reliable comparison between models.

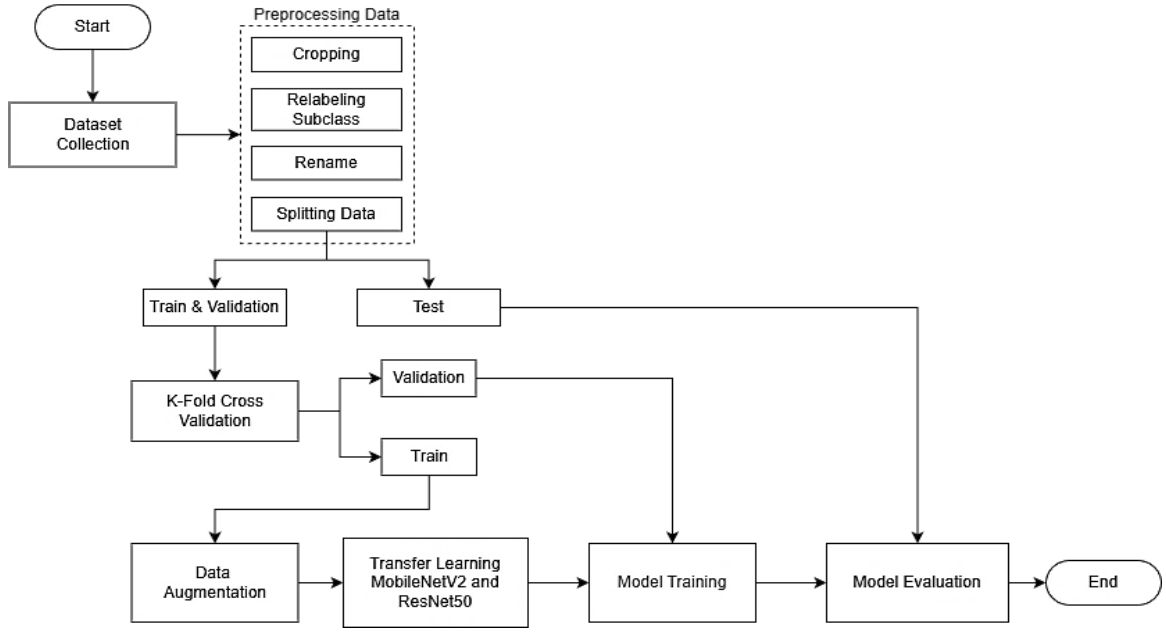

Fig. 1 Research Flow Stages

Dataset Collection

The dataset in this research consists of primary data and secondary data used to build AI artificial image classification models and real images. Primary data was collected directly by the researcher through photographing real objects using a smartphone camera with a variety of angles and backgrounds, and creating AI artificial images using the Meta AI and Stable Diffusion platforms with various prompts, resulting in a total of 660 images (304 real and 356 fake). To expand the scope and diversity of the data, this study also utilized a secondary dataset from the Kaggle AI Detection Dataset platform that totaled 23,281 images, consisting of 11,635 real images and 11,646 AI-generated images. Thus, the total dataset used in this study reached 23,941 images, including 11,939 real classes and 12,002 fake classes.

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 10, Issue 1, Januari 2026
DOI : https://doi.org/10.33395/sinkron.v10i1.15682
e-ISSN : 2541-2019
p-ISSN : 2541-044X

Fig. 2 Sample Dataset

Data Preprocessing

Data preprocessing is an important stage in the model training process because it determines the quality of the data to be used. In this research, preprocessing is done to prepare the data to be more structured, clean, and ready to be processed by the CNN architecture. This process includes several main stages, namely cropping, relabeling subclasses, renaming, and splitting data.

A. Data Cropping

Cropping is the process of cutting a certain part of the image to obtain the most relevant area while adjusting the image size according to processing needs (Sun et al., 2023). At this stage, image cropping is performed to adjust the size and shape of the various images, so that all images have a 1:1 (square) ratio before being used in the model training process. The dataset used has images with non-uniform aspect ratios, so it needs to be uniformed to fit the CNN model input. This is important because most CNN architectures such as MobileNetV2 and ResNet50 require the same sized input image.

B. Subclass Relabeling

Subclass relabeling is the process of relabeling data so that each image is not only categorized based on its main class, but also based on the characteristics or objects contained in it (Yun et al., 2021). The research dataset initially consists of two main classes, namely real and fake. To enrich the analysis and support the multiclass classification process, relabeling is performed by grouping each image based on the main object contained in the image so that five subclasses are formed, namely human, animal, view, art, and vehicle. Relabeling is done manually through visual review and adjustment of the folder structure of the dataset so that the model not only learns to distinguish the authenticity of real and fake images, but also understands the context of the objects contained in them.

Table 1. Relabeling Subclass Result

Class	Subclass	Images	Total
Real	Human	2.300	11.939
	Animal	2.300	
	View	2.242	
	Art	2.974	
	Vehicle	2.123	
Fake	Human	2.300	12.002
	Animal	2.300	
	View	2.300	
	Art	2.802	
	Vehicle	2.300	

C. Splitting Data

Splitting data is the process of separating data into several parts according to their intended use, such as training, validation, and testing, so that model performance can be evaluated objectively (Bichri et al., 2024). In the initial stage of this research, the dataset consisting of 23,941 images was divided into two main parts, namely 90% train & validation data and 10% test data. This division aims to separate a portion of the data that will be used

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 10, Issue 1, Januari 2026
DOI : https://doi.org/10.33395/sinkron.v10i1.15682
e-ISSN : 2541-2019
p-ISSN : 2541-044X

specifically for final testing of the model's performance, so that the evaluation results can reflect the model's ability on data that has never really been used during the training and validation process. The division process is carried out in a stratified manner, namely by maintaining the proportion of the main classes, namely real and fake, along with all subclasses, namely human, animal, view, art, vehicle, to remain balanced in each part, so that the distribution of data in train & validation and test remains representative of the entire dataset.

D. Rename

Rename is the process of renaming files to have a uniform and informative naming pattern so that they are easier to recognize and manage in the data processing stage (Kanza & Knight, 2022). In this research, renaming is done automatically using Python with a naming pattern that includes the main class real or fake and subclasses human, animal, view, art, and vehicle. This helps to keep the dataset organized and makes it easier to track and call data during the model training stage.

K-Fold Cross Validation

K-Fold Cross Validation is a statistical evaluation method that divides the training data into *K* subsets or *folds*. In each iteration, one *fold* is used as validation data, while the rest as training data. This process is repeated K times so that each data is used as validation once. The average performance of all iterations is used as a measure of model accuracy. This technique is commonly used to avoid bias in evaluation, especially on limited-sized datasets (de Oro et al., 2022). In this study, a K-Fold of 5 folds was used. This method divides the train data into five sections with an equal proportion of labels in each fold. In each iteration, four sections are used for train data and one section is used for validation data, while the test set was completely separated and used only once for final evaluation.

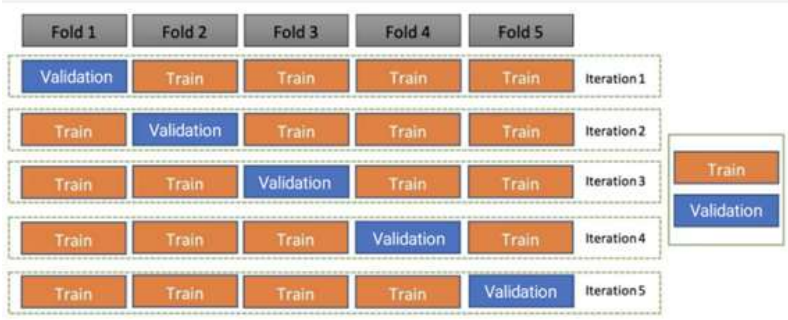

Fig. 3 K-Fold Cross Validation

Data Augmentation

Data augmentation is the process of randomly transforming images to produce new variations of existing images. Data augmentation is used to reduce the risk of overfitting the model (Kumar et al., 2023). This research applies data augmentation techniques to images, especially only in train data, the goal is that validation and test data still reflect the original data, so that the evaluation is accurate and fair. Some of the augmentation methods used include rescale, translation, flip, zoom, and rotation, utilizing features from the Tensorflow framework.

Table 2. Augmentation Parameters

Parameters	Value
rescale	preprocess_input
shear_range	0.1
width_shift_range	0.1
height_shift_range	0.1
horizontal_flip	True
zoom_range	0.2
rotation_range	20

Convolutional Neural Network (CNN) Model Development

Model development was conducted by utilizing the Convolutional Neural Network (CNN) architecture using two types of pretrained models, namely MobileNetV2, and ResNet50. MobileNetV2 was chosen due to its lightweight and efficient architecture, suitable for large datasets with fast training time. ResNet50 was chosen for its deep architecture and residual connection technique that can handle vanishing gradient.

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 10, Issue 1, Januari 2026
DOI : https://doi.org/10.33395/sinkron.v10i1.15682
e-ISSN : 2541-2019
p-ISSN : 2541-044X

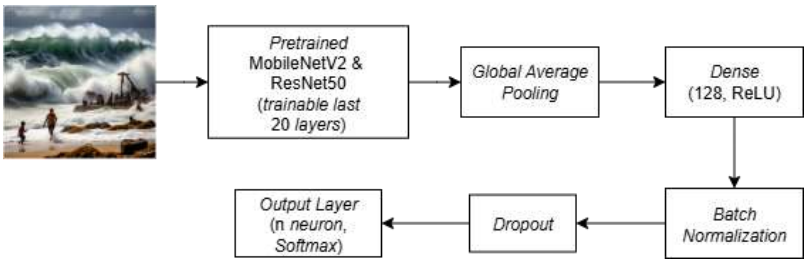

Fig. 4 Model Build

A. MobileNetV2

MobileNetV2 is one of the Convolutional Neural Network (CNN) architectures designed for high efficiency, especially on devices with limited computing resources. This architecture comes as a solution for lightweight computing needs while still maintaining good performance. MobileNetV2 uses a depthwise separable convolution approach that divides the convolution process into two stages, namely depthwise convolution and pointwise convolution, thus reducing computational complexity (Hajar et al., 2025). The MobileNetV2 architecture is used as the base model with pretrained weights from ImageNet. The model is built using transfer learning and accepts an input of 224×224 pixels with 3 color channels. Before fine tuning, the last 20 layers of MobileNetV2 were activated for training in order to adapt to the dataset in this study, while the rest were frozen. After feature extraction, the model continued with Global Average Pooling, a Dense layer of 128 neurons with ReLU activation, Batch Normalization, Dropout, and a softmax output layer, with n neurons (corresponding to the number of classes). This model is compiled using the categorical_crossentropy loss function.

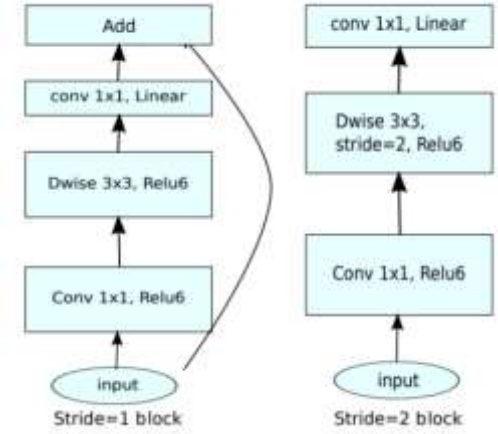

Fig. 5 MobileNetV2 Architecture

B. ResNet50

Residual Network 50 (ResNet50) is one of the CNN architectures developed to overcome the challenges of deep networks, especially the vanishing gradient problem that often occurs during the training process. This architecture uses a residual learning approach, where the network not only learns the feature transformation directly, but also learns the difference (residual) between the input and output of a layer. In this way, ResNet50 allows for very deep network training without degrading performance or accuracy. The model consists of 50 layers and is capable of extracting visual feature representations deeply and efficiently (Daviana et al., 2025). The ResNet50 architecture is also used as a base model with pretrained weights from ImageNet. Just like in MobileNetV2, this model is built with a transfer learning approach and accepts an input image of 224×224 pixels with 3 color channels. Before fine tuning, the last 20 layers of ResNet50 are activated for training, while the other layers are frozen to maintain the initial weights. After the feature extraction process by ResNet50, the architecture continued with Global Average Pooling, followed by a Dense layer of 128 neurons with activation functions ReLU, Batch Normalization, Dropout, and a softmax output layer consisting of n neurons according to the number of classes. The model is compiled with a similar configuration for multi-class classification.

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 10, Issue 1, Januari 2026
DOI : https://doi.org/10.33395/sinkron.v10i1.15682
e-ISSN : 2541-2019
p-ISSN : 2541-044X

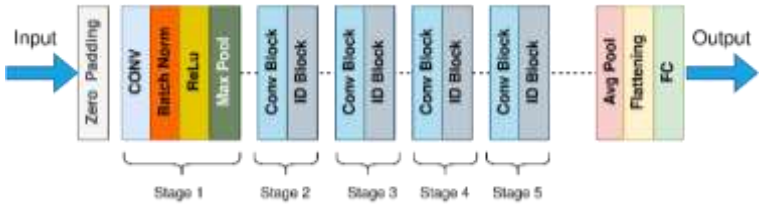

Fig. 6 ResNet50 Architecture

Model Training

Model training is performed by fine tuning the two previously developed CNN architectures, MobileNetV2 and ResNet50, to accomplish the classification task between AI-generated and real images. Fine tuning was performed by utilizing the initial weights of each architecture as the basis for further training. To obtain the best model configuration. To obtain the best model configuration, a series of scenarios were conducted with several combinations of hyperparameters, namely dropout, learning rate, optimizer, and epoch. In this study, dropout was used with a value of 0.3 to prevent overfitting by randomly deactivating a number of neurons without losing much information. A learning rate of 0.001 was chosen because it is stable and safe for fine-tuning pretrained models such as MobileNetV2 and ResNet50. All scenarios used the Adam Optimizer because of its ability to adaptively adjust the learning rate and its good performance on various deep learning tasks. The model was trained with variations of 10, 30, and 50 epochs to evaluate performance under conditions of underfitting, optimality, and potential overfitting.

Table 3. Fine Tuning Hyperparameters

Model	Architecture	*Dropout*	*Learning Rate*	*Optimizer*	*Epoch*
A	MobileNetV2	0.3	0.001	Adam	10
B	MobileNetV2	0.3	0.001	Adam	30
C	MobileNetV2	0.3	0.001	Adam	50
D	ResNet50	0.3	0.001	Adam	10
E	ResNet50	0.3	0.001	Adam	30
F	ResNet50	0.3	0.001	Adam	50

Model Evaluation

After the training process using K-Fold Cross Validation and fine-tuning, the next step is to evaluate the model performance. This evaluation aims to measure the model's ability to classify images into a combined class between the main class, namely real and fake along with its subclasses human, animal, art, view, and vehicle. In each fold, the performance of the model during training is evaluated based on the accuracy, loss, valid accuracy, and valid loss values to determine the fold with the best performance. At the end of the K-Fold process, the fold with the highest performance is selected as the best model. This best model is then further evaluated using test data that is separate from the train and validation data. This stage provides a more objective picture of the model's generalization ability to new data. The evaluation and comparison of MobileNetV2 and ResNet50 was conducted based on the confusion matrix and key metrics, namely accuracy, precision, recall, and F1-score. In addition, the confusion matrix of each model was also analyzed to see the misclassification patterns in each class, especially in distinguishing AI-generated and real images in various subclasses of human, animal, art, view, and vehicle.

RESULT

Model Training

In training the AI-generated and real image classification models, each fold is trained using the generator that has been prepared, with the number of epochs adjusting the hyperparameter configuration at the fine tuning stage. Training was conducted using MobileNetV2 and ResNet50 architectures with a combination of dropout parameter 0.3, Adam optimizer, and learning rate 0.001 at variations of 10, 30, and 50 epochs. During the training process on each fold, the accuracy and loss values on the train and validation data are stored as a basis for performance evaluation. After all folds have been trained, a comparison of the results is carried out to determine the fold with the most optimal performance, which is then used as a reference in the final testing stage on the test dataset.

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 10, Issue 1, Januari 2026
DOI : https://doi.org/10.33395/sinkron.v10i1.15682
e-ISSN : 2541-2019
p-ISSN : 2541-044X

Table 4. Training Model Result

Model	Dropout	Learning Rate	Optimizer	Epoch	Fold	Accuracy	Loss	Valid Accuracy	Valid Loss
					1	**0.96**	**0.12**	**0.90**	**0.55**
					2	0.95	0.14	0.83	1.01
A	0.3	0.001	Adam	10	3	0.95	0.14	0.69	1.98
					4	0.96	0.12	0.86	0.72
					5	0.96	0.13	0.80	1.20
					1	0.93	0.20	0.80	0.88
					2	**0.93**	**0.20**	**0.80**	**0.85**
B	0.3	0.001	Adam	30	3	0.93	0.21	0.78	1.04
					4	0.93	0.20	0.72	1.32
					5	0.93	0.20	0.79	0.96
					1	0.96	0.13	0.79	1.03
					2	0.96	0.13	0.82	0.96
C	0.3	0.001	Adam	50	3	0.95	0.14	0.80	0.98
					4	0.96	0.13	0.80	0.98
					5	**0.95**	**0.13**	**0.82**	**0.88**
					1	0.97	0.08	0.90	0.39
					2	0.98	0.08	0.91	0.37
D	0.3	0.001	Adam	10	3	0.98	0.08	0.92	0.32
					4	0.97	0.08	0.89	0.47
					5	**0.97**	**0.09**	**0.92**	**0.30**
					1	0.99	0.03	0.92	0.41
					2	0,99	0.03	0.90	0.51
E	0.3	0.001	Adam	30	**3**	**0.99**	**0.03**	**0.93**	**0.35**
					4	0.99	0.03	0.92	0.46
					5	0.99	0.02	0.93	0.40
					1	1.00	0.01	0.92	0.53
					2	0.99	0.02	0.92	0.56
F	0.3	0.001	Adam	50	**3**	**0.99**	**0.02**	**0.92**	**0.52**
					4	0.99	0.02	0.91	0.58
					5	0.99	0.01	0.91	0.54

The training results on all folds show a difference in performance between the two architectures. On MobileNetV2 with 10 epochs, the training accuracy is in the range of 0.95-0.96 with the highest validation accuracy of 0.90, but the validation loss increases dramatically on the 3rd fold to 1.98 which indicates overfitting. At 30 epochs, the training accuracy stabilizes around 0.93 but the validation loss remains high, so the additional epochs have not been able to improve generalization. At 50 epochs, the training accuracy increased again to 0.95-0.96, but the validation accuracy remained low at 0.79-0.82, and the validation loss was still high, so overfitting was increasingly visible.

In contrast to MobileNetV2, ResNet50 shows a much more stable performance. At 10 epochs, the training accuracy reaches 0.97-0.98 with validation accuracy up to 0.92 and consistently low validation loss. At 30 epochs, the performance further improved with training accuracy close to 0.99 and consistently high validation accuracy of 0.90-0.93. At 50 epochs, the training accuracy reached 0.99-1.00 and the validation accuracy remained strong around 0.91-0.92 although the validation loss increased slightly, so the overfitting tendency was still under control. Although the training accuracy of ResNet50 reaches very high values, this result should be interpreted with caution and does not directly indicate memorization. The consistently high validation accuracy across folds and epochs, combined with relatively stable validation loss, indicates that the model maintains generalization capability rather than merely fitting the training data. From a technical perspective, the residual connections in ResNet50 help preserve gradient flow and reduce degradation problems in deep networks, enabling more effective feature learning

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 10, Issue 1, Januari 2026
DOI : https://doi.org/10.33395/sinkron.v10i1.15682
e-ISSN : 2541-2019
p-ISSN : 2541-044X

without excessive dependence on memorizing training samples. In addition, the use of data augmentation and K-Fold Cross Validation further reduces the risk of memorization by exposing the model to diverse data variations and multiple validation splits.

Overall, ResNet50 is able to maintain the best balance between accuracy and stability compared to MobileNetV2.After the evaluation process of all folds in each scenario, one best fold is selected based on a combination of accuracy, loss, validation accuracy, and validation loss values. Determining the best fold not only considers the performance on training data, but mainly prioritizes the results on validation data, because validation data reflects the generalization ability of the model to new data. Thus, the fold with the highest validation accuracy and lowest validation loss is declared the best fold, as it shows the most accurate prediction rate as well as the lowest error on data that is not used during training. This best fold summary is then used as the basis for determining the model used in the final testing stage.

Table 5. Best Fold per Model

Model	Dropout	Learning Rate	Optimizer	Epoch	Fold	Accuracy	Loss	Valid Accuracy	Valid Loss
A	0.3	0.001	Adam	10	1	0.96	0.12	0.90	0.55
B	0.3	0.001	Adam	30	2	0.93	0.20	0.80	0.85
C	0.3	0.001	Adam	50	5	0.95	0.13	0.82	0.88
D	0.3	0.001	Adam	10	5	0.97	0.09	0.92	0.30
E	0.3	0.001	Adam	30	3	0.99	0.03	0.93	0.35
F	0.3	0.001	Adam	50	3	0.99	0.02	0.92	0.52

Overall, these results show that ResNet50 consistently performs better than MobileNetV2, both in terms of training and validation accuracy. The low and stable loss values also show that ResNet50 is more adaptable to the complexity of the multi-class data used in this study. In addition to being displayed in tabular form, the performance of each model is also visualized through accuracy and loss graphs at the best fold of each scenario.

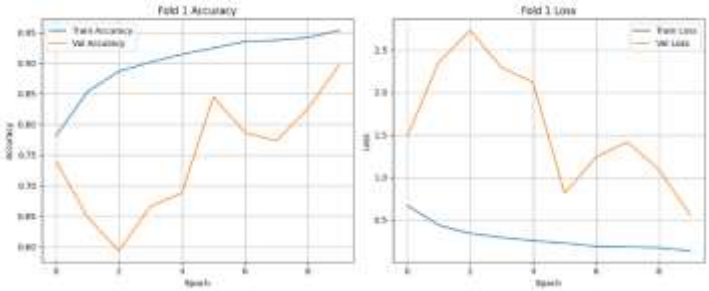

Fig. 7 Accuracy and Loss Best Fold Model A

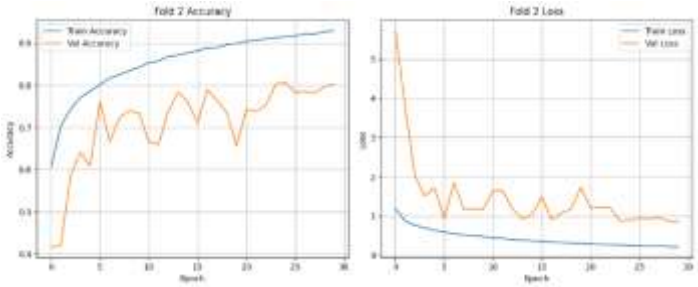

Fig. 8 Accuracy and Loss Best Fold Model B

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 10, Issue 1, Januari 2026
DOI : https://doi.org/10.33395/sinkron.v10i1.15682
e-ISSN : 2541-2019
p-ISSN : 2541-044X

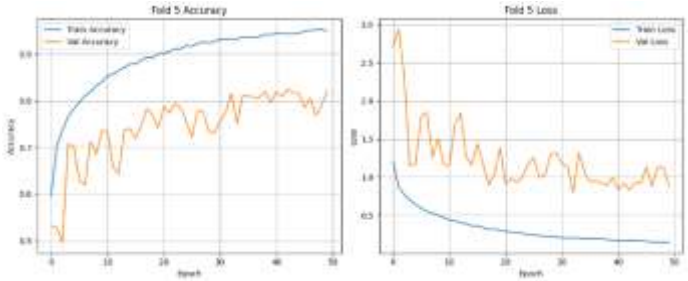

Fig. 9 Accuracy and Loss Best Fold Model C

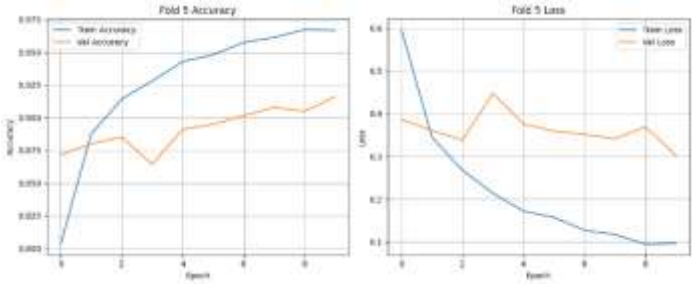

Fig. 10 Accuracy and Loss Best Fold Model D

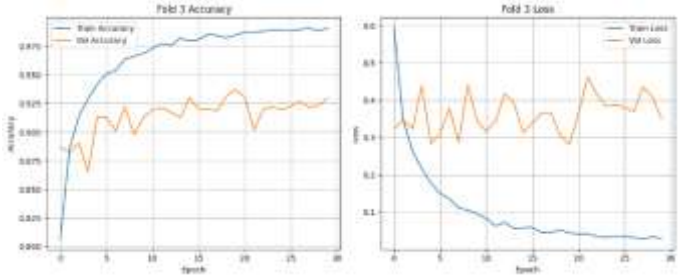

Fig. 11 Accuracy and Loss Best Fold Model E

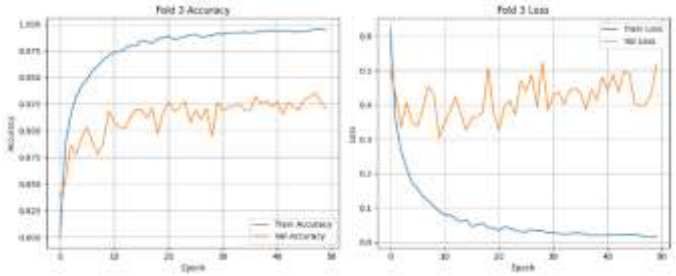

Fig. 12 Accuracy and Loss Best Fold Model F

Model Evaluation

The evaluation stage is carried out using test data to measure the extent to which the model is able to recognize new images that have never been seen during the training process. This evaluation is carried out based on the best fold from each scenario in the previous stage, so that the results taken truly represent the most optimal performance of each model. At this stage, the metrics used include comfusion matrix, as well as accuracy, precision, recall, and F1-score.

Table 6. Model Evaluation Result

Model	Dropout	Learning Rate	Optimizer	Epoch	Accuracy	Precision	Recall	F1-Score
A	0.3	0.001	Adam	10	89%	90%	89%	89%
B	0.3	0.001	Adam	30	80%	81%	81%	80%
C	0.3	0.001	Adam	50	81%	82%	81%	81%
D	0.3	0.001	Adam	10	92%	93%	92%	92%

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 10, Issue 1, Januari 2026
DOI : https://doi.org/10.33395/sinkron.v10i1.15682
e-ISSN : 2541-2019
p-ISSN : 2541-044X

E	0.3	0.001	Adam	30	93%	93%	93%	93%
F	0.3	0.001	Adam	50	92%	93%	92%	92%

The evaluation results show that MobileNetV2 achieves the best performance at 10 epochs with 89% accuracy and balanced precision, recall, and F1-score of 89-90%. However, at 30 and 50 epochs the performance decreased with an accuracy of only 80-81%. This shows that increasing epochs decreases the generalization ability of the model so that MobileNetV2 is less stable when training is extended.

In contrast, ResNet50 showed a much more consistent performance. At 10 epochs, the model achieved 92% accuracy with balanced evaluation metrics. Performance improved at 30 epochs with 93% accuracy, being the best result in this test. At 50 epochs, the accuracy remained high at 92%, so a larger number of epochs did not decrease the stability of the model. Thus, ResNet50 proved to be more effective and reliable than MobileNetV2 in the evaluation process. To clarify the evaluation results, the confusion matrix of each best model is shown. This confusion matrix illustrates the comparison between the predicted and actual labels.

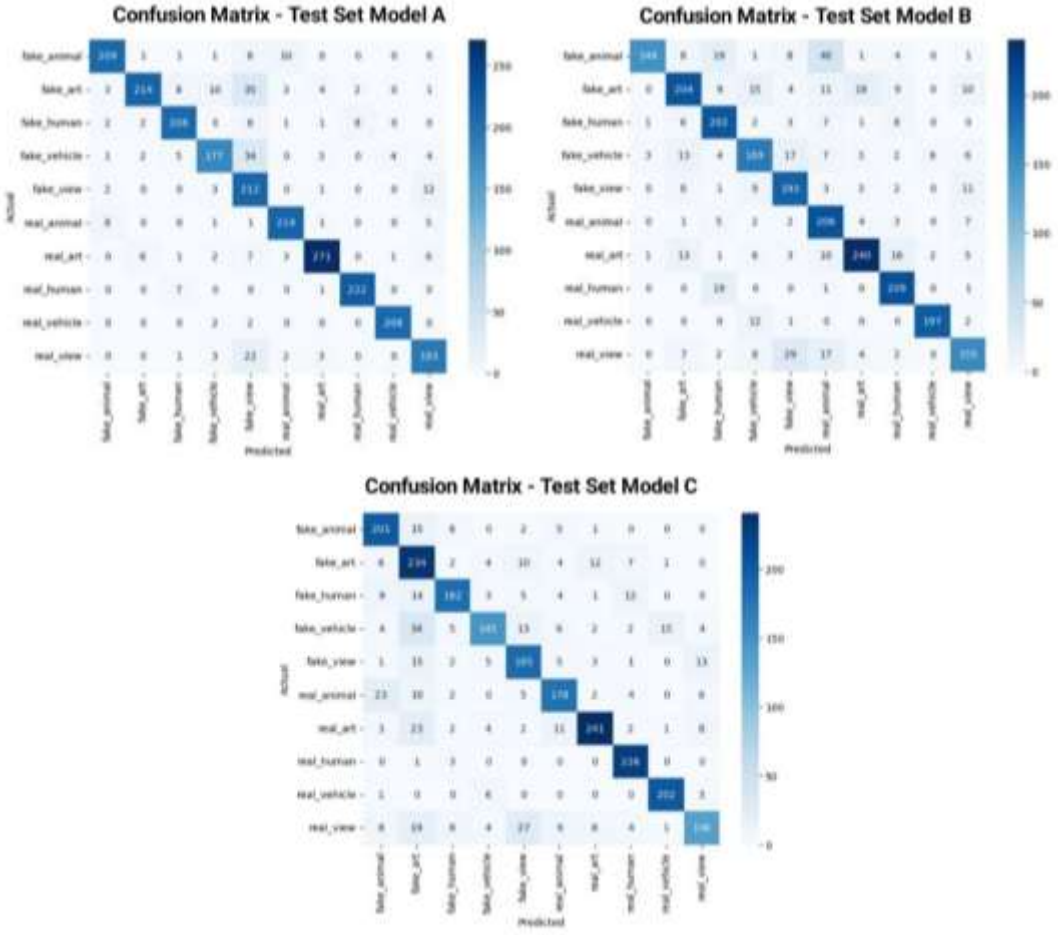

Fig. 13 Confusion Matrix Architecture MobileNetV2 (Model A, B, C)

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 10, Issue 1, Januari 2026
DOI : https://doi.org/10.33395/sinkron.v10i1.15682
e-ISSN : 2541-2019
p-ISSN : 2541-044X

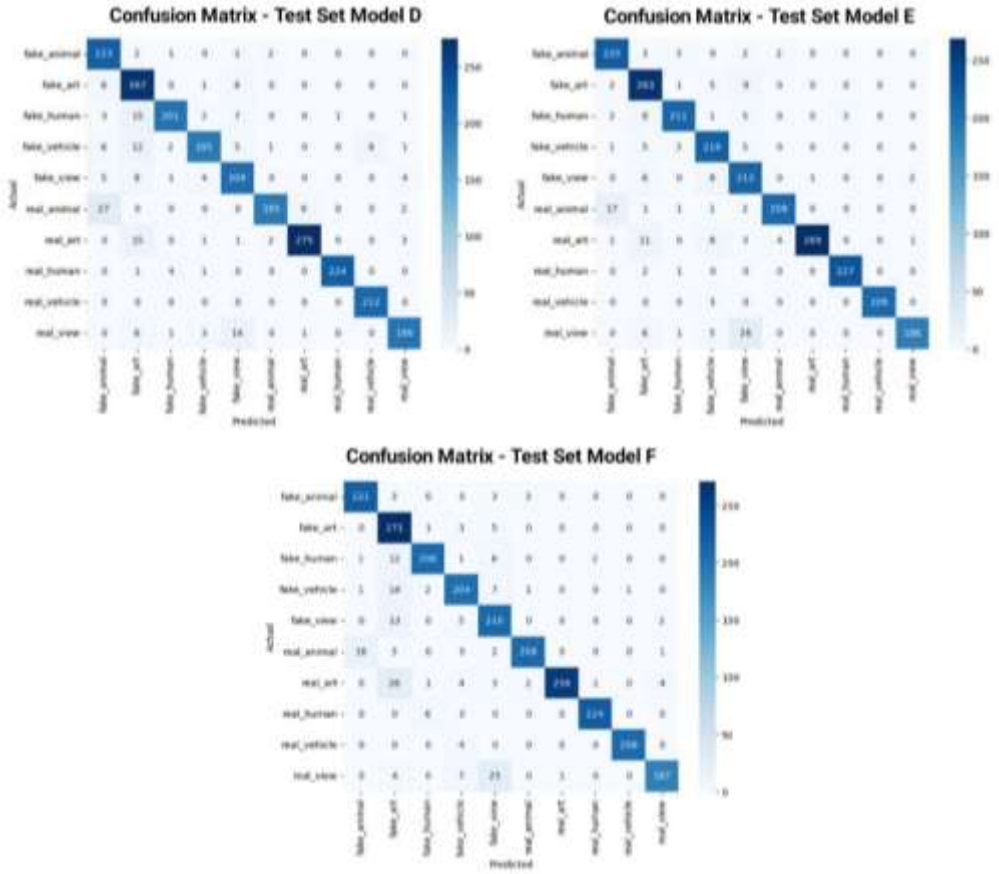

Fig. 14 Confusion Matrix Architecture ResNet50 (Model D, E, F)

DISCUSSIONS

The results show a clear performance difference between MobileNetV2 and ResNet50 in classifying AI-generated and real images. MobileNetV2 achieves the best performance at 10 epochs with 89% testing accuracy, but the performance drops significantly at 30 and 50 epochs with only 80-81% accuracy. This decline is in line with the training results, where validation loss increases at higher epochs, indicating overfitting. The decline in MobileNetV2 performance occurred because the smaller model capacity made it difficult to capture the complexity of features in the multi-class dataset, making it prone to overfitting at high epochs. This finding confirms that MobileNetV2 is less stable when the training duration is extended, especially on multi-class datasets with complex visual variations.

In contrast, ResNet50 showed a much more consistent performance across all scenarios. In the training stage, the model has high validation accuracy and low validation loss with no sharp upward trend, even when the number of epochs is increased to 50. This trend is reversed in the testing stage, where ResNet50 produces 92% accuracy at 10 epochs, increases to 93% at 30 epochs, and only slightly drops at 50 epochs without compromising the stability of precision, recall, and F1-score. This performance reliability confirms the residual block advantage of ResNet50, which is able to maintain gradient flow so that learning remains effective on complex architectures. This result is consistent with previous studies such as Hakim et al. (2024) and Fatoni et al. (2025), which reported that ResNet-based architectures achieve more stable and superior performance than lightweight CNNs in AI-generated image classification tasks.

The confusion matrix shows that the three MobileNetV2 models exhibit similar error patterns. MobileNetV2 with 10 epochs still produces many errors because the features have not yet stabilized. At 30 epochs, accuracy improves, but confusion still occurs in classes with high visual similarity, such as fake_view, real_view, fake_art, and real_art. At 50 epochs, performance is more stable, but overfitting begins to appear and errors still occur, especially in the vehicle and view classes.

In the ResNet50 architecture, all three models show more stable performance than MobileNetV2, but still have consistent error patterns in classes with high visual similarity. ResNet50 with 10 epochs is still lacking in predicting the view and art classes, while 30 epochs show an increase in accuracy but still make errors in fake_view and real_view. The 50-epoch model provides the most stable results, although there are still minor errors in real_view

*name of corresponding author

and real_vehicle. Overall, adding epochs improves the performance of ResNet50, but classes with very similar visual patterns remain the main source of error. Similar observations were also reported by Li et al. (2022), who highlighted that visually similar sub-classes remain challenging even for deep CNN architectures.

The research findings show that the difference in performance between the two models is due to the learning capabilities formed during the training process. MobileNetV2 experiences a decrease in accuracy at high epochs due to an increase in validation loss, which indicates overfitting, so that the features learned no longer represent the test data. The confusion matrix shows that errors persist in classes with high visual similarity, such as view, art, and vehicle, indicating the model's limitation in distinguishing subtle details. In contrast, ResNet50 maintains performance stability because its residual structure keeps gradients under control, allowing effective high-level feature formation at large epochs. These results are evident from the increase in accuracy at epoch 30, stability at epoch 50, and reduced errors in classes that were previously difficult to distinguish. These findings emphasize that ResNet50 is more adaptive to visual complexity and more resistant to overfitting than MobileNetV2. This outcome aligns with recent survey findings by Khan et al. (2020), which emphasize that deeper residual-based CNNs provide stronger generalization capability than lightweight architectures in visually complex multi-class scenarios.

CONCLUSION

This study concludes that ResNet50 is the best architecture for AI-generated and real image classification tasks on multi-class datasets. It provides the highest and most stable performance in both training and testing stages, with the best accuracy of 93% at the 30th epoch and consistently high precision, recall, and F1-score values. In addition, the low validation loss indicates strong generalization ability and does not exhibit overfitting tendencies MobileNetV2 remains competitive at low epochs (89% accuracy), but degrades significantly when epochs are increased, making it less suitable for long-term training on datasets with high visual complexity. Thus, ResNet50 is more recommended as the primary model for AI -generated and real image classification, especially in systems that require high accuracy. However, MobileNetV2 can still be used in real-time implementations or devices with computational limitations due to its advantage in efficiency.

However, this study is limited by the use of datasets collected from Kaggle with minor subclass distribution differences, particularly in the art category, which may introduce dataset specific bias. In addition, the rapid development of state of the art generative AI systems, including recent diffusion based models and large scale multimodal generators, may produce images with visual characteristics that differ from those represented in the dataset, potentially affecting the generalization of the proposed models.

ACKNOWLEDGMENT

This research can be carried out well thanks to the help of various parties. Therefore, the researcher would like to thank Mr. I Gede Iwan Sudipa, S.Kom., M.Cs., Mrs. Yuri Prima Fittryani, S.T, M.T. As lecturers who have guided in the implementation of this research. In addition, I would also like to thank my fellow students who always support the implementation of this research.

REFERENCES

Aljohani, K., & Turki, T. (2022). Automatic Classification of Melanoma Skin Cancer with Deep Convolutional Neural Networks. *AI (Switzerland)*, *3*(2), 512–525. https://doi.org/10.3390/ai3020029

Bahrul Subkhi, M., Bagus Setiawan, A., & Yusuf Alif Candra, M. (2023). Klasifikasi Gambar: Membedakan Lukisan Buatan Manusia dan AI dengan CNN. *Jurnal Filsafat, Sains, Teknologi, Dan Sosial Budaya*, *29*(4), 149–155. https://doi.org/10.33503/paradigma.v30i4.1284

Bichri, H., Chergui, A., & Hain, M. (2024). Investigating the Impact of Train / Test Split Ratio on the Performance of Pre-Trained Models with Custom Datasets. *IJACSA) International Journal of Advanced Computer Science and Applications*, *15*(2). https://doi.org/10.14569/IJACSA.2024.0150235

Daviana, F. P., Aryanti, A., & Anugraha, N. (2025). Performance Comparison Between ResNet50 and MobileNetV2 for Indonesian Sign Language Classification. *JURIKOM (Jurnal Riset Komputer)*, *12*(3), 319–328. https://doi.org/10.30865/jurikom.v12i3.8667

de Oro, J. E. C. G., Koch, P. J., Krois, J., Ros, A. G. C., Patel, J., Meyer-Lueckel, H., & Schwendicke, F. (2022). Hyperparameter Tuning and Automatic Image Augmentation for Deep Learning-Based Angle Classification on Intraoral Photographs—A Retrospective Study. *Diagnostics*, *12*(7). https://doi.org/10.3390/diagnostics12071526

Döring, N., Le, T. D., Vowels, L. M., Vowels, M. J., & Marcantonio, T. L. (2024). The Impact of Artificial Intelligence on Human Sexuality: A Five-Year Literature Review 2020–2024. *Current Sexual Health Reports*, *17*(1), 4. https://doi.org/10.1007/s11930-024-00397-y

*name of corresponding author

Fatoni, Kurniawan, T. B., Dewi, D. A., Zakaria, M. Z., & Muhayeddin, A. M. M. (2025). Fake vs Real Image Detection Using Deep Learning Algorithm. *Journal of Applied Data Sciences*, *6*(1), 366–376. https://doi.org/10.47738/jads.v6i1.490

Ghiurău, D., & Popescu, D. E. (2025). Distinguishing Reality from AI: Approaches for Detecting Synthetic Content. *Computers*, *14*(1), 1–33. https://doi.org/10.3390/computers14010001

Hajar, S., Murinto, M., & Yudhana, A. (2025). Comparison of Transfer Learning Strategies Using MobileNetV2 and ResNet50 for Ecoprint Leaf Classification. *Jurnal Teknik Informatika (Jutif)*, *6*(5), 3251–3264. https://doi.org/10.52436/1.jutif.2025.6.5.5266

Hakim, S. A., Ubaidillah, M., Ramadhan, A. R., Zulvia, R., Hawari, A., Rizky, A. B., Lutfi, R., Tsania, P., Hermanto, M., Yudistira, N., & Korespondensi, P. (2024). KLASIFIKASI CITRA GENERASI ARTIFICIAL INTELLIGENCE MENGGUNAKAN METODE FINE TUNING PADA RESIDUAL NETWORK. *Jurnal Teknologi Informasi Dan Ilmu Komputer (JTIIK)*, *11*(3), 655–666. https://doi.org/10.25126/jtiik.938118

Hang Rai, D. (2024). Artificial Intelligence Through Time: A Comprehensive Historical Review. *Bachelor of Science in Computer Science and Information Technology*. https://doi.org/10.13140/RG.2.2.22835.03364

Kanza, S., & Knight, N. J. (2022). Behind every great research project is great data management. In *BMC Research Notes* (Vol. 15, Issue 1). BioMed Central Ltd. https://doi.org/10.1186/s13104-022-05908-5

Khan, A., Sohail, A., Zahoora, U., & Qureshi, A. S. (2020). A survey of the recent architectures of deep convolutional neural networks. *Artificial Intelligence Review*, *53*(8), 5455–5516. https://doi.org/10.1007/s10462-020-09825-6

Komendantova, N., & Erokhin, D. (2025). Artificial Intelligence Tools in Misinformation Management during Natural Disasters. *Public Organization Review*, 1–25. https://doi.org/10.1007/s11115-025-00815-2

Kumar, T., Mileo, A., Brennan, R., & Bendechache, M. (2023). Image Data Augmentation Approaches: A Comprehensive Survey and Future directions. *Science Foundation Ireland*. https://doi.org/10.48550/arXiv.2301.02830

Li, Z., Liu, F., Yang, W., Peng, S., & Zhou, J. (2022). A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects. *IEEE Transactions on Neural Networks and Learning Systems*, *33*(12), 6999–7019. https://doi.org/10.1109/TNNLS.2021.3084827

Lin, C. C., Huang, A. Y. Q., & Lu, O. H. T. (2023). Artificial intelligence in intelligent tutoring systems toward sustainable education: a systematic review. In *Smart Learning Environments* (Vol. 10, Issue 1). Springer. https://doi.org/10.1186/s40561-023-00260-y

Mu, J., Adrezo, M., & Haikal, A. N. (2024). Identifikasi Wajah Asli dan Buatan Deepfake Menggunakan Metode Convolutional Neural Network. *TEKNIKA*, *13*(1), 45–50. https://doi.org/10.34148/teknika.v13i1.705

Peng, Y. (2024). A Comparative Analysis Between GAN and Diffusion Models in Image Generation. *Transactions on Computer Science and Intelligent Systems Research*, *5*, 189–195. https://doi.org/10.62051/0f1va465

Pradnya Dhuhita, W. M., Ubaid, M. Y., & Baita, A. (2023). MobileNet V2 Implementation in Skin Cancer Detection. *ILKOM Jurnal Ilmiah*, *15*(3), 498–506. https://doi.org/10.33096/ilkom.v15i3.1702.498-506

Słapczyński, T. (2022). Artificial Intelligence in Science and Everyday Life, Its Application and Development Prospects. *ASEJ - Scientific Journal of Bielsko-Biala School of Finance and Law*, *26*(4), 78–85. https://doi.org/10.19192/wsfip.sj4.2022.12

Sophia LI. (2025). The Social Harms of AI-Generated Fake News: Addressing Deepfake and AI Political Manipulation. *Digital Society & Virtual Governance*, *1*(1), 72–88. https://doi.org/10.6914/dsvg.010105

Suharyanto, D., Lubis, C., & Dharmawan, A. B. (2024). PENERAPAN CONVOLUTIONAL NEURAL NETWORK DAN CAPSULE NETWORKS DALAM MENDETEKSI DEEPFAKE. *Jurnal Ilmu Komputer Dan Sistem Informasi*, *12*(1). https://doi.org/10.24912/jiksi.v12i1.28190

Sun, Y., Miao, L., Zhao, Z., Pan, T., Wang, X., Guo, Y., Xin, D., Chen, Q., & Zhu, R. (2023). An Efficient and Automated Image Preprocessing Using Semantic Segmentation for Improving the 3D Reconstruction of Soybean Plants at the Vegetative Stage. *Agronomy*, *13*(9). https://doi.org/10.3390/agronomy13092388

Yun, S., Oh, S. J., Heo, B., Han, D., Choe, J., & Chun, S. (2021). Re-labeling ImageNet: from Single to Multi-Labels, from Global to Localized Labels. *CVPR*. https://doi.org/10.48550/arXiv.2101.05022

*name of corresponding author