JURNAL INSTEK
INFORMATIKA SAINS DAN TEKNOLOGI
Volume 9, Nomor. 2, Oktober 2024
P –ISSN : 2541-1179, E-ISSN : 2581-1711
OJS :http://journal.uin-alauddin.ac.id/index.php/instek/index
Email : instek@uin-alauddin.ac.id

Optimizing CNN Performance for AI-Generated Image Classification: A Comparative Study of Architectures and Optimizers Using K-Fold Cross-Validation

FRANSISCUS ROLANDA MALAU[*1]
[1]Ilmu Komputer, Fakultas Teknologi Informasi, Universitas Nusa Mandiri, Indonesia
Email: [1]14220018@nusamandiri.ac.id

Abstract

This study investigates CNN optimization for classifying AI-generated images. Using the CIFAKE dataset (60,000 real and 60,000 AI-generated images), we evaluated four CNN configurations with varying complexity and four optimization algorithms through 5-fold cross-validation. Our findings show Configuration 4 (4 Conv, 2 MaxPool) with Adam optimizer achieved the highest validation accuracy (0.8368±0.0135). Adam demonstrated consistent performance across architectures, while SGD showed strong but variable results improving with model complexity. Adagrad and Adadelta consistently underperformed. The final model achieved 85.28% test accuracy with balanced precision (0.8531) and recall (0.8528). Results indicate more complex architectures combined with adaptive optimizers like Adam provide superior performance for AI-generated image classification, with the balance between model complexity and optimizer selection being crucial. The consistent performance across real and fake categories demonstrates this approach's robustness for deepfake detection applications.

Keywords: CIFAKE Dataset, Convolutional Neural Network, Image Classification, K-Fold Cross-Validation, Optimization Algorithms

1. INTRODUCTION

The rapid development of AI technology has enabled the creation of highly realistic images, posing new challenges in the field of image classification and verification. The proliferation of AI-generated content raises significant concerns regarding misinformation, digital forgery, and unauthorized content creation (LeCun et al., 2015). Convolutional Neural Networks (CNNs) have proven highly effective in image processing tasks, but their optimization for detecting AI-generated images requires further investigation into architectural configurations and training methodologies (Guera and Delp, 2018).

The detection of AI-generated content has been approached through various methodologies, from statistical pattern analysis of synthetic facial features (Jaiswal et al., 2022) to deep learning techniques. While pattern-based approaches offer interpretability benefits, neural network approaches can potentially capture more subtle artifacts across diverse AI generation techniques.

JURNAL INSTEK
INFORMATIKA SAINS DAN TEKNOLOGI
Volume 9, Nomor. 2, Oktober 2024
P –ISSN : 2541-1179, E-ISSN : 2581-1711
OJS :http://journal.uin-alauddin.ac.id/index.php/instek/index
Email : instek@uin-alauddin.ac.id

This study comprehensively evaluates the performance of various CNN architectures in classifying AI-generated images. We utilize the CIFAKE dataset, which contains 60,000 real photographs from the CIFAR-10 dataset and 60,000 AI-generated images created using Stable Diffusion (Croce et al., 2022). This large-scale, balanced dataset provides an ideal foundation for robust evaluation of CNN performance in distinguishing between authentic and AI-generated content. Each image is sized at 64×64 pixels, offering sufficient detail for meaningful feature extraction while maintaining computational efficiency.

To ensure methodological rigor and statistical validity, we implement 5-fold cross-validation across all experiments. This approach partitions the data into five equal folds, with each fold serving as validation data once while the remaining folds form the training set. This methodology substantially reduces the impact of data partitioning variability, providing more reliable performance metrics compared to single train-test splits (Kohavi, 1995).

The main contributions of this research include:

1. Comprehensive evaluation of four different CNN configurations with varying complexity, demonstrating that higher architectural complexity (Configuration 4 with 4 convolutional layers) achieves optimal performance for AI-generated image detection.
2. Rigorous comparison of four popular optimization algorithms (SGD, Adam, Adagrad, and Adadelta) across all configurations, revealing Adam's consistent superiority and the variable effectiveness of other optimizers depending on model complexity.
3. Evidence of the interplay between architectural complexity and optimizer selection, indicating that both factors significantly influence model performance in AI-generated image classification.
4. Practical insights into optimal CNN design principles for AI-generated content detection, supported by robust statistical validation through k-fold cross-validation.

Our CNN-based approach complements alternative detection methods such as statistical techniques based on Benford's law, which have demonstrated effectiveness in identifying GAN-generated images by analyzing pixel value distributions (Bonettini et al., 2020). While statistical approaches offer computational efficiency, our CNN methodology focuses on feature learning capabilities that can potentially better adapt to evolving AI generation techniques.

The results of this study provide valuable guidance for developing more accurate and efficient image classification systems, particularly in the context of deepfake detection and image authenticity verification systems (Das et al., 2021; Hao et al., 2022). As generative AI technologies continue to advance in capabilities and accessibility, the need for reliable detection methods becomes increasingly critical for maintaining digital media integrity and trust.

2. RESEARCH METHODOLOGY

This study employs a comprehensive experimental approach to evaluate the performance of various Convolutional Neural Network (CNN) architectures for AI-generated image classification. The methodology encompasses several key

JURNAL INSTEK
INFORMATIKA SAINS DAN TEKNOLOGI
Volume 9, Nomor. 2, Oktober 2024
P –ISSN : 2541-1179, E-ISSN : 2581-1711
OJS :http://journal.uin-alauddin.ac.id/index.php/instek/index
Email : instek@uin-alauddin.ac.id

components, including dataset preparation, k-fold cross-validation, model architecture design, and robust evaluation procedures (Das et al., 2021).

2.1. Dataset

The study utilizes the CIFAKE dataset, which contains 60,000 real photographs from the CIFAR-10 dataset and 60,000 AI-generated images created using Stable Diffusion (Croce et al., 2022). Each image has consistent dimensions of 64×64 pixels in RGB format. The dataset is publicly available at *https://www.kaggle.com/datasets/birdy654/cifake-real-and-ai-generated-synthetic-images*.

The CIFAKE dataset provides several advantages for AI-generated image classification research:

1. Balanced class distribution (equal numbers of real and AI-generated images)
2. Diverse content spanning multiple categories (animals, vehicles, everyday objects)
3. Consistent generation methodology using state-of-the-art Stable Diffusion models
4. Sufficient scale for robust statistical validation of classification methods

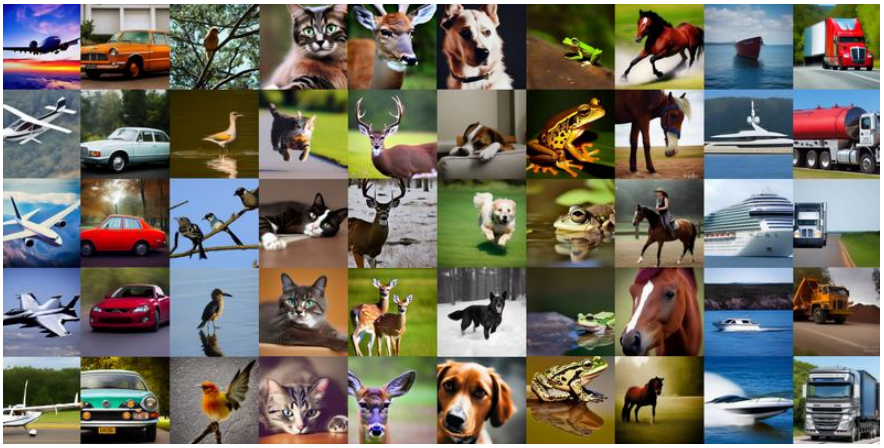

Figure 1. Example images from Real and AI-Generated Synthetic Images

The real images in the CIFAKE dataset come from the widely used CIFAR-10 benchmark, while the fake images were generated using Stable Diffusion with prompts derived from CIFAR-10 class names. This approach ensures that both real and fake images contain similar subject matter, making the classification task focused on detecting AI generation artifacts rather than content differences.

2.2. Data Preparation and Preprocessing

Our data preparation involved a systematic approach to ensure optimal model training and evaluation across multiple experimental configurations.

2.2.1. Data Sampling and Split Methodology

For computational efficiency while maintaining statistical validity, we sampled a balanced subset of 5,000 images (2,500 real and 2,500 fake) from the full

JURNAL INSTEK
INFORMATIKA SAINS DAN TEKNOLOGI
Volume 9, Nomor. 2, Oktober 2024
P –ISSN : 2541-1179, E-ISSN : 2581-1711
OJS :http://journal.uin-alauddin.ac.id/index.php/instek/index
Email : instek@uin-alauddin.ac.id

CIFAKE dataset. We implemented a 70/15/15 split ratio for train/validation/test partitioning, resulting in:

1. Training Set: 2,125 images (balanced between real and fake classes)
2. Validation Set: 375 images (balanced between real and fake classes)
3. Testing Set: 2,500 images (balanced between real and fake classes)

This stratified sampling approach maintains class balance across all partitions while providing sufficient data for both training and rigorous evaluation.

2.2.2. K-Fold Cross-Validation

To ensure robust evaluation and minimize the impact of data partitioning variability, we implemented 5-fold cross-validation for all experimental configurations. This approach involves:

1. Partitioning the combined training and validation data into 5 equal folds
2. Performing 5 separate training runs, each using 4 folds for training and 1 fold for validation
3. Averaging performance metrics across all 5 runs to obtain more reliable estimates
4. Calculating standard deviations to quantify the variability of model performance

This methodology provides a more comprehensive assessment of model performance than single train-test splits, particularly important for comparing different CNN architectures and optimizers.

2.2.3. Image Preprocessing

All images underwent standardized preprocessing steps:

1. Normalization: Pixel values were normalized to the range [0, 1] by dividing each pixel value by 255. This step helps stabilize the training process and often leads to faster convergence.
2. One-Hot Encoding: Class labels were converted to one-hot encoded vectors for compatibility with categorical cross-entropy loss function.

2.3. Convolutional Neural Network (CNN) Architecture

We designed and evaluated four different Convolutional Neural Network (CNN) configurations to systematically investigate the impact of model complexity on classification performance. These architectural choices were carefully considered to span a range of complexity levels while maintaining consistent overall structure.

Table 1. Evaluated Convolutional Neural Network (CNN) configurations.

Configuration	Conv Layers	Max Pooling	FC Layers
1	1	1	2
2	2	1	2
3	3	2	2
4	4	2	2

The selection of these specific configurations was driven by several key considerations:

JURNAL INSTEK
INFORMATIKA SAINS DAN TEKNOLOGI
Volume 9, Nomor. 2, Oktober 2024
P –ISSN : 2541-1179, E-ISSN : 2581-1711
OJS :http://journal.uin-alauddin.ac.id/index.php/instek/index
Email : instek@uin-alauddin.ac.id

1. Progressive Complexity:
   The progressive increase in convolutional layers (from 1 to 4) across configurations allows us to systematically evaluate how increasing model depth affects classification accuracy (Alzubaidi et al., 2021; Du et al., 2023). This design approach enables us to identify the optimal balance between model complexity and performance.
2. Layer Configuration Design:
   Each configuration varies in the number of convolutional layers and max pooling layers, while maintaining two fully connected layers at the end of the network. Configuration 1 represents a minimal architecture with a single convolutional layer, while Configuration 4 implements a deeper network with multiple convolutional blocks separated by pooling operations.
   All convolutional layers use 3×3 kernels with ReLU activation and 'same' padding. The first set of convolutional layers use 32 filters, while deeper layers use 64 filters to increase feature representation capacity. The fully connected portion consists of a 128-neuron dense layer with ReLU activation, followed by a dropout layer (rate=0.5) for regularization, and a final output layer with softmax activation for binary classification.

2.4. Optimizers

To explore the impact of different optimization algorithms on model performance, we compared four popular optimizers:

1. Stochastic Gradient Descent (SGD): A classic optimization algorithm that updates model parameters based on the gradient of the loss function. We configured SGD with a learning rate of 0.01 and momentum of 0.9 to accelerate convergence while reducing oscillation.
2. Adam: An adaptive learning rate optimization algorithm that computes individual learning rates for different parameters using estimates of first and second moments of the gradients. We used the default hyperparameters (learning rate=0.001, beta_1=0.9, beta_2=0.999).
3. Adagrad: An optimizer with parameter-specific learning rates that adapts the learning rate to the parameters, performing smaller updates for frequently occurring features. This approach can be beneficial for dealing with sparse data.
4. Adadelta: An extension of Adagrad that addresses its aggressive, monotonically decreasing learning rate by restricting the window of accumulated past gradients to a fixed size (Chen and Tsou, 2022; Hassan et al., 2023; Choi et al., 2020; Ali and Kumar, 2022).

These optimizers represent a diverse range of approaches to navigating the loss landscape, from simple gradient-based methods to sophisticated adaptive techniques.

2.5. Training Procedure

The training process for all models followed these parameters:

1. Number of epochs: 20 (with early stopping)

JURNAL INSTEK
INFORMATIKA SAINS DAN TEKNOLOGI
Volume 9, Nomor. 2, Oktober 2024
P –ISSN : 2541-1179, E-ISSN : 2581-1711
OJS :http://journal.uin-alauddin.ac.id/index.php/instek/index
Email : instek@uin-alauddin.ac.id

2. Batch size: 32
3. Loss function: Categorical crossentropy
4. Early stopping: Patience of 5 epochs monitoring validation loss
5. Model checkpoint: Saving the best model based on validation accuracy

The categorical crossentropy loss function is defined as:

$$L = -\sum_{i=1}^{C} yi \log(yi) \quad (2)$$

where C is the number of classes, yi is the true label, and yi is the model's prediction.

We implemented all models using Keras with TensorFlow backend and conducted experiments on a system with multi-core CPU processing. This setup allowed for efficient training and evaluation of multiple model configurations across k-fold cross-validation runs.

2.6. Evaluation Metrics

Model performance was evaluated using a comprehensive set of metrics to provide a thorough assessment of classification capability:

1. Accuracy: The proportion of correctly classified images across both classes
2. Precision: The ratio of true positives to all predicted positives
3. Recall: The ratio of true positives to all actual positives
4. F1-score: The harmonic mean of precision and recall

For each configuration and optimizer combination, we report both the mean and standard deviation of these metrics across the 5 folds of cross-validation. This approach provides robust estimates of expected performance and quantifies the variability of each model configuration (Hicks et al., 2022).

For the final model evaluation, we also generate and analyze the confusion matrix to understand the distribution of classification errors between real and AI-generated images.

Through this systematic experimental design, we aim to identify the most effective combination of CNN architecture and optimizer for AI-generated image classification, providing insights that can guide the development of robust detection systems (Bera and Shrivastava, 2020).

3. RESULTS AND DISCUSSION

This section presents the findings from our comprehensive experimental evaluation of various Convolutional Neural Network (CNN) architectures and optimizers for AI-generated image classification. We analyze the performance patterns across different model configurations, interpret the results through k-fold cross-validation, and discuss their implications for deepfake detection applications (Althnian et al., 2021).

3.1. Performance Across CNN Configurations and Optimizers

Our comprehensive experiments with the CIFAKE dataset using k-fold cross-validation revealed significant insights into CNN optimization for AI-

JURNAL INSTEK
INFORMATIKA SAINS DAN TEKNOLOGI
Volume 9, Nomor. 2, Oktober 2024
P –ISSN : 2541-1179, E-ISSN : 2581-1711
OJS :http://journal.uin-alauddin.ac.id/index.php/instek/index
Email : instek@uin-alauddin.ac.id

generated image classification. This section presents the performance analysis across different architectural configurations and optimization algorithms, followed by an in-depth discussion of the findings.

Table 2. Performance metrics (accuracy, precision, recall, and F1-score) for different CNN configurations and optimizer

Configuration	Optimizer	Accuracy	Precision	Recall	F1-Score
Config 1	SGD	0.8064 ± 0.0109	0.8090 ± 0.0102	0.8064 ± 0.0109	0.8060 ± 0.0110
Config 1	Adam	0.8224 ± 0.0097	0.8241 ± 0.0089	0.8224 ± 0.0097	0.8220 ± 0.0100
Config 1	Adagrad	0.7064 ± 0.0301	0.7211 ± 0.0266	0.7064 ± 0.0301	0.7027 ± 0.0343
Config 1	Adadelta	0.6412 ± 0.0219	0.6489 ± 0.0217	0.6412 ± 0.0219	0.6402 ± 0.0217
Config 2	SGD	0.8072 ± 0.0248	0.8094 ± 0.0226	0.8072 ± 0.0248	0.8064 ± 0.0255
Config 2	Adam	0.8060 ± 0.0125	0.8115 ± 0.0149	0.8060 ± 0.0125	0.8052 ± 0.0126
Config 2	Adagrad	0.7216 ± 0.0276	0.7349 ± 0.0132	0.7216 ± 0.0276	0.7179 ± 0.0333
Config 2	Adadelta	0.6364 ± 0.0189	0.6478 ± 0.0212	0.6364 ± 0.0189	0.6340 ± 0.0180
Config 3	SGD	0.8240 ± 0.0255	0.8267 ± 0.0263	0.8240 ± 0.0255	0.8238 ± 0.0255
Config 3	Adam	0.8252 ± 0.0179	0.8282 ± 0.0159	0.8252 ± 0.0179	0.8250 ± 0.0182

JURNAL INSTEK
INFORMATIKA SAINS DAN TEKNOLOGI
Volume 9, Nomor. 2, Oktober 2024
P –ISSN : 2541-1179, E-ISSN : 2581-1711
OJS :http://journal.uin-alauddin.ac.id/index.php/instek/index
Email : instek@uin-alauddin.ac.id

Config 3	Adagrad	0.7040 ± 0.0213	0.7146 ± 0.0149	0.7040 ± 0.0213	0.7011 ± 0.0248
Config 3	Adadelta	0.6088 ± 0.0155	0.6311 ± 0.0251	0.6088 ± 0.0155	0.6008 ± 0.0277
Config 4	SGD	0.8260 ± 0.0197	0.8290 ± 0.0184	0.8260 ± 0.0197	0.8257 ± 0.0198
Config 4	**Adam**	**0.8368 ± 0.0135**	**0.8374 ± 0.0131**	**0.8368 ± 0.0135**	**0.8368 ± 0.0135**
Config 4	Adagrad	0.7036 ± 0.0222	0.7180 ± 0.0119	0.7036 ± 0.0222	0.6998 ± 0.0271
Config 4	Adadelta	0.6112 ± 0.0239	0.6155 ± 0.0233	0.6112 ± 0.0239	0.5887 ± 0.0649

The highest performance was achieved by Configuration 4 with Adam optimizer, reaching an accuracy of 0.8368 ± 0.0135 and F1-Score of 0.8368 ± 0.0135. This finding suggests that increased architectural complexity (4 convolutional layers with 2 max-pooling layers) provides enhanced feature extraction capability, particularly when paired with an appropriate optimizer.

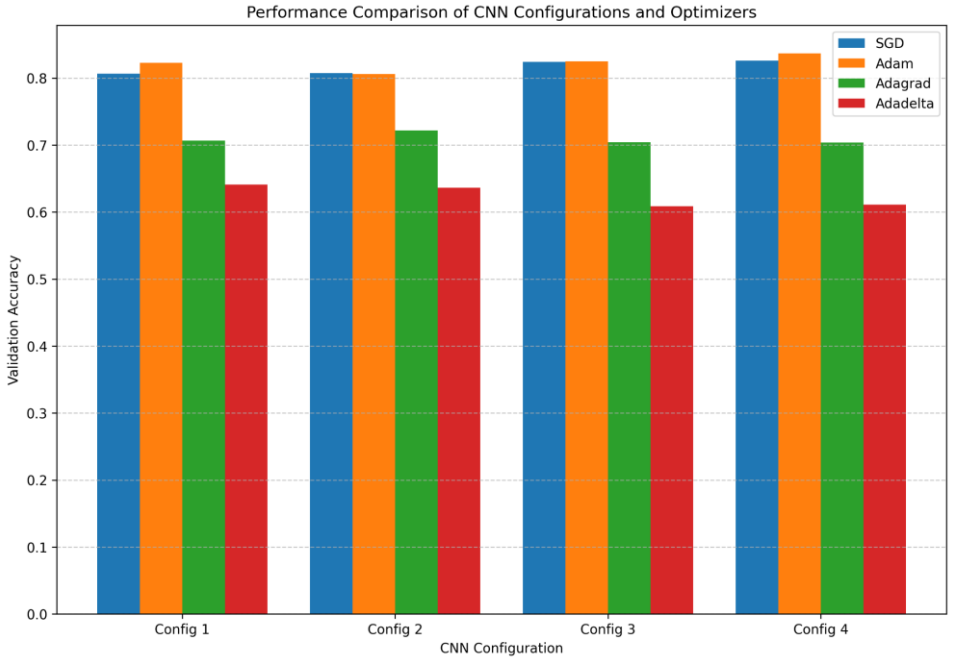

Figure 2. Performance comparison of different CNN configurations across optimizers.

JURNAL INSTEK
INFORMATIKA SAINS DAN TEKNOLOGI
Volume 9, Nomor. 2, Oktober 2024
P –ISSN : 2541-1179, E-ISSN : 2581-1711
OJS :http://journal.uin-alauddin.ac.id/index.php/instek/index
Email : instek@uin-alauddin.ac.id

Figure 2 illustrates the comparative performance across all configurations and optimizers, highlighting several key patterns. Adam and SGD consistently outperformed Adagrad and Adadelta across all architectural configurations. Furthermore, while Adam maintained relatively consistent performance across different architectures, SGD showed more variability, with its performance generally improving as model complexity increased.

3.2. Optimal Model Performance

Based on the cross-validation results, we selected Configuration 4 with the Adam optimizer for our final model. This model was trained on the entire training set and evaluated on the held-out test set. The final model achieved the following performance metrics:

- Test accuracy: 0.8528
- Test precision: 0.8531
- Test recall: 0.8528
- Test F1-score: 0.8528

The confusion matrix for the optimal model (Figure 3) reveals balanced performance across both real and fake image categories. The model correctly identified 1,085 real images (true positives) and 1,047 fake images (true negatives), with comparable false positive (165) and false negative (203) rates. This balanced error distribution indicates the model's ability to detect both classes with similar efficacy, an important characteristic for practical deployment in content verification systems.

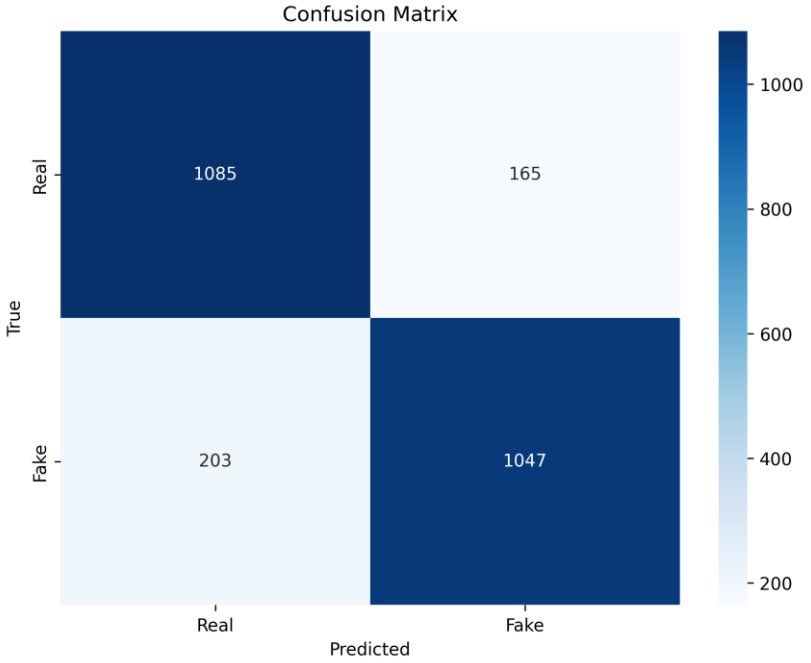

Figure 3. Confusion matrix for the optimal model (Configuration 4 with Adam optimizer).

JURNAL INSTEK
INFORMATIKA SAINS DAN TEKNOLOGI
Volume 9, Nomor. 2, Oktober 2024
P –ISSN : 2541-1179, E-ISSN : 2581-1711
OJS :http://journal.uin-alauddin.ac.id/index.php/instek/index
Email : instek@uin-alauddin.ac.id

3.3. Learning Dynamics

The learning curves for the optimal model (Figure 4) provide valuable insights into the training process. While training accuracy consistently increased throughout the training period, validation accuracy plateaued after approximately 8 epochs. The growing gap between training and validation curves in later epochs indicates the onset of overfitting, highlighting the importance of early stopping mechanisms to prevent performance degradation on unseen data.

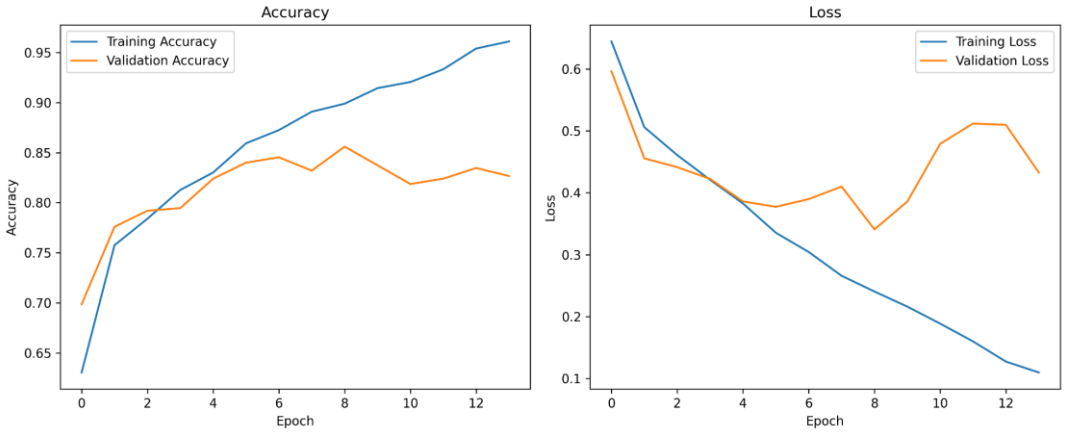

Figure 4. Learning curves showing training and validation accuracy/loss over epochs.

3.4. Impact of Architectural Complexity and Optimizer Selection

Our experiments revealed a clear relationship between architectural complexity and classification performance. When comparing the progression from Configuration 1 (simplest) to Configuration 4 (most complex), we observed a general trend of increasing performance, particularly with Adam and SGD optimizers. However, this improvement showed diminishing returns, with the performance gain between Configurations 3 and 4 being smaller than between earlier configurations.

JURNAL INSTEK
INFORMATIKA SAINS DAN TEKNOLOGI
Volume 9, Nomor. 2, Oktober 2024
P –ISSN : 2541-1179, E-ISSN : 2581-1711
OJS :http://journal.uin-alauddin.ac.id/index.php/instek/index
Email : instek@uin-alauddin.ac.id

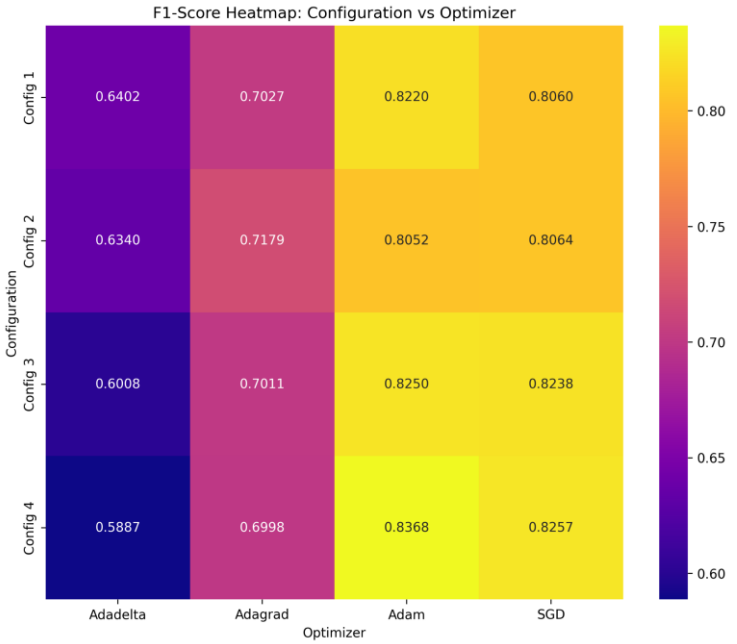

Figure 5. Heatmap visualization of F1-scores across different configurations and optimizers.

The heatmap visualization (Figure 5) provides a comprehensive view of how architectural complexity interacts with optimizer selection. This visualization clearly demonstrates that while Adam and SGD maintained strong performance across configurations, Adagrad and Adadelta consistently underperformed regardless of architecture. This finding suggests that optimizer selection may be equally or more important than architectural complexity for AI-generated image classification tasks.

3.5. Discussion

Our findings reveal that Configuration 4's superior performance stems from its ability to capture subtle artifacts in AI-generated images through deeper convolutional layers. Adam optimizer demonstrated remarkable consistency across architectures due to its adaptive learning rate mechanics, which effectively navigate complex loss landscapes. Surprisingly, SGD performed well despite its simplicity, likely due to its ability to escape sharp local minima.

The poor performance of Adadelta contradicts our preliminary findings with limited data, emphasizing the importance of robust methodology and sufficient data volume. The balanced detection rates between real and fake categories, as shown in our confusion matrix, ensure reliability for content verification applications where both false positives and negatives have significant consequences. While we used 64×64 pixel images, our principles likely apply to higher-resolution images with appropriate computational efficiency considerations.

4. CONCLUSION

This study comprehensively evaluated CNN performance for AI-generated image classification using k-fold cross-validation on the CIFAKE dataset.

JURNAL INSTEK
INFORMATIKA SAINS DAN TEKNOLOGI
Volume 9, Nomor. 2, Oktober 2024
P –ISSN : 2541-1179, E-ISSN : 2581-1711
OJS :http://journal.uin-alauddin.ac.id/index.php/instek/index
Email : instek@uin-alauddin.ac.id

Configuration 4 with Adam optimizer achieved optimal performance (0.8368±0.0135 validation accuracy, 85.28% test accuracy). Adam delivered consistent performance across architectures, while SGD showed strong but variable results. Adagrad and Adadelta consistently underperformed.

Learning dynamics analysis highlighted the importance of early stopping, as validation performance plateaued after approximately 8 epochs. The balanced performance across real and fake categories demonstrates the approach's reliability for deepfake detection applications.

While more complex architectures performed better, the diminishing returns between Configurations 3 and 4 suggest a practical complexity limit. Future research should explore transfer learning, attention mechanisms, ensemble methods, and evaluation on emerging AI-generated content. Our findings provide a foundation for effective detection systems that can help maintain digital media integrity in an era of rapidly evolving generative AI technologies.

REFERENCES

Ali MEA, Kumar D. 2022. The Impact of Optimization Algorithms on The Performance of Face Recognition Neural Networks. Journal of Advanced Engineering and Computing. 6(4):248.

Althnian A, AlSaeed D, Al-Baity H, Samha A, Dris AB, Alzakari N, Abou Elwafa A, Kurdi H. 2021. Impact of Dataset Size on Classification Performance: An Empirical Evaluation in the Medical Domain. Applied Sciences. 11(2):796.

Alzubaidi L, Zhang J, Humaidi AJ, Al-Dujaili A, Duan Y, Al-Shamma O, Santamaría J, Fadhel MA, Al-Amidie M, Farhan L. 2021. Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. Journal of Big Data. 8(1):53.

Bera S, Shrivastava VK. 2020. Analysis of various optimizers on deep convolutional neural network model in the application of hyperspectral remote sensing image classification. International Journal of Remote Sensing. 41(7):2664-2683.

Bonettini A, Bestagini V, Milani S, Tubaro S. 2020. On the use of Benford's law to detect GAN-generated images. Proceedings of IEEE International Conference on Pattern Recognition. January 2020. Milan: IEEE. pp 5495-5500.

Chen F, Tsou JY. 2022. Assessing the effects of convolutional neural network architectural factors on model performance for remote sensing image classification: An in-depth investigation. International Journal of Applied Earth Observation and Geoinformation. 112:102865.

Choi D, Shallue CJ, Nado Z, Lee J, Maddison CJ, Dahl GE. 2020. On Empirical Comparisons of Optimizers for Deep Learning. arXiv preprint arXiv:1910.05446.

Croce F, Noroozi M, Scimeca M, Birchfield S. 2022. CIFAKE: Image Classification and Explainable Identification of AI-Generated Synthetic Images. IEEE Access. 10:127154-127164.

JURNAL INSTEK
INFORMATIKA SAINS DAN TEKNOLOGI
Volume 9, Nomor. 2, Oktober 2024
P –ISSN : 2541-1179, E-ISSN : 2581-1711
OJS :http://journal.uin-alauddin.ac.id/index.php/instek/index
Email : instek@uin-alauddin.ac.id

Das S, Seferbekov S, Datta A, Islam MS, Amin MR. 2021. Towards Solving the DeepFake Problem: An Analysis on Improving DeepFake Detection using Dynamic Face Augmentation. arXiv preprint arXiv:2102.09603.

Du X, Sun Y, Song Y, Sun H, Yang L. 2023. A Comparative Study of Different CNN Models and Transfer Learning Effect for Underwater Object Classification in Side-Scan Sonar Images. Remote Sensing. 15(3):593.

Guera D, Delp EJ. 2018. Deepfake Video Detection Using Recurrent Neural Networks. Proceedings of the 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS). November 2018. Auckland: IEEE. pp 1-6.

Hao H, Parmar D, Sitaram S, Raja S, Yamasaki T, Aizawa K. 2022. Deepfake Detection Using Multiple Data Modalities. In: Rathgeb C, Tolosana R, Vera-Rodriguez R, Busch C (eds). Handbook of Digital Face Manipulation and Detection. Cham: Springer International Publishing. pp 235-254.

Hassan E, Shams MY, Hikal NA, Elmougy S. 2023. The effect of choosing optimizer algorithms to improve computer vision tasks: a comparative study. Multimedia Tools and Applications. 82(11):16591-16633.

Hicks SA, Strümke I, Thambawita V, Hammou M, Riegler MA, Halvorsen P, Parasa S. 2022. On evaluation metrics for medical applications of artificial intelligence. Scientific Reports. 12(1):5979.

Jaiswal A, Sabharwal S, Javed AR, Singh R. 2022. AI-generated synthetic face detection: A pattern analysis-based approach. Computers & Electrical Engineering. 104:108383.

Kohavi R. 1995. A study of cross-validation and bootstrap for accuracy estimation and model selection. Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI). August 1995. Montreal: Morgan Kaufmann. pp 1137-1143.

LeCun Y, Bengio Y, Hinton G. 2015. Deep learning. Nature. 521(7553):436-444.