Journal of Deep Learning, Computer Vision and Digital Image Processing
E-ISSN: 2986-8939 P-ISSN: 2986-8920
Vol. 4, No. 2, June 2026
DOI. https://doi.org/10.61255/decoding.v4i2.1629

Comparative Analysis of a Simple CNN and Fine-Tuned ResNet50 for Deepfake Image Detection: Performance and Computational Efficiency Evaluation

Rifda Triani Mutmainah✉
(Universitas Adhirajasa Reswara Sanjaya, Bandung, Indonesia)

Asti Herliana
(Universitas Adhirajasa Reswara Sanjaya, Bandung, Indonesia)

OPEN ACCESS

ARTICLE HISTORY
Received: May 10, 2026
Revised: June 22, 2026
Accepted: June 30, 202

KEYWORDS
CNN;
Computational efficiency;
Deepfake;
Fine-tuning

ABSTRACT

Purpose – Deepfake-related cybercrime is an increasingly troubling cybersecurity threat, since such content can now be produced from a single facial photo using freely available face-swapping techniques, while human ability to distinguish real from fake images remains limited (48.2–59%). This study compares a simple CNN and a fine-tuned ResNet50 within an identical, controlled framework.

Methods – A controlled experiment compared a simple CNN (trained from scratch) and ResNet50 (two-phase fine-tuning) on 20,000 images from the Kaggle Deepfake and Real Images dataset (80:10:10 split, seed = 42). Candidate images were deduplicated before the data split; zero cross-split duplicates were confirmed. ResNet50 used ResNet-specific preprocessing; the CNN used inputs normalized to [0,1]. Evaluation used accuracy, precision, recall, F1-score, AUC-ROC, and specificity, with a paired McNemar's test as the primary significance measure.

Findings – The CNN outperformed ResNet50 on six of seven metrics (accuracy 90.90% vs. 89.80%; AUC-ROC 97.08% vs. 96.52%), while ResNet50 achieved higher recall (93.10% vs. 91.80%). The accuracy difference was not statistically significant (p = 0.200). The CNN was 69.5 times smaller and trained faster (12.2 vs. 16.5 minutes). Neither model showed clear overfitting.

Research implications – The findings rest on a single subset, a single split, and a single training run per model, limited to 20 epochs; the potential of ResNet50 with longer training has not been explored.

Originality – This study combines accuracy, generalization-related diagnostics, and efficiency in a single controlled comparison, applies a paired test appropriate for a shared test set, and filters duplicates before the data split.

Correspondence Author: ✉rifdatrianimutmainah21@gmail.com
To cite this article : Mutmainah, R. T., & Herliana, A. (2026). Comparative analysis of a simple CNN and fine-tuned ResNet50 for deepfake image detection: Performance and computational efficiency evaluation. Journal of Deep Learning, Computer Vision and Digital Image Processing, 4(2), 297-308. https://doi.org/10.61255/decoding.v4i2.1629

INTRODUCTION

The development of artificial intelligence (AI) continues to advance and brings many benefits, but it has also produced negative consequences, including excessive dependence [1] and serious cybersecurity concerns [2]. One increasingly troubling form of cybersecurity threat today is the misuse of deepfake technology [3], a threat that has grown rapidly enough to prompt a systematic literature review documenting more than a hundred distinct detection methodologies in recent years [4]. This threat is very real and continues to grow, given that deepfake images can now be produced from a single facial photo using freely available open-source one-shot face-swapping techniques [5]. Compounding this situation, human ability to distinguish real images from fake ones remains fairly limited, ranging from 48.2% to 59% depending on the experiment [6], making manual detection a genuine challenge. This situation is what drives the need for an automated, accurate, and efficient deepfake detection system.

To address this problem, various deep learning approaches have been developed, one of the most widely used being the Convolutional Neural Network (CNN), which has proven effective at extracting and recognizing visual patterns in images [7]. CNNs themselves come in various levels of complexity, ranging from computationally cheap lightweight architectures to much deeper networks such as ResNet50, which can capture more complex visual features through its residual learning mechanism [8], consistent with a broader review of CNN architecture evolution that documents the trade-off between network depth and computational cost [9]. This difference in complexity directly affects both model performance and computational requirements, an important consideration when models are deployed under real-world conditions that often involve limited infrastructure.

Research on deepfake image detection using CNN and ResNet50 generally falls into three streams. The first stream focuses on comparisons among pretrained architectures. In [10], VGG16, VGG19, and ResNet50 were compared on 1,200 deepfake images, with VGG19 performing best at 98.5% accuracy, while [11] showed that pretrained ResNet50 features combined with XGBoost can achieve an AUC of 0.99 without full fine-tuning. Generalization remains a persistent challenge in these pretrained-architecture comparisons, as shown by a large-scale, multi-team deepfake detection and reconstruction challenge, in which most submitted models struggled to maintain performance outside their training data distribution [12]. The second stream focuses on computational efficiency; outside the deepfake domain, [13] examined ResNet50 architecture adjustments to reduce overfitting in skin disease classification, indicating that efficiency- and generalization-oriented refinements to this backbone are a recurring theme in adjacent imaging domains. The third stream examines the effectiveness of fine-tuning itself, as in [14], which achieved a 98.11% AUC on the Celeb-DF dataset after fine-tuning ResNet50.

In this context, fine-tuning is typically implemented as a two-phase transfer learning strategy: the first phase freezes the entire pretrained backbone and trains only the newly added classification layer, while the second phase unfreezes the upper portion of the backbone — for example, the last 30 layers of ResNet50 — and retrains it together with the head using a very small learning rate to avoid destroying previously learned representations [15], [16]. In at least one medical imaging application, this staged strategy was shown to improve classification performance compared with single-phase fine-tuning [17] whether this benefit generalizes across other architectures and imaging domains is not established by that single study and is not assumed here.

Architectural complexity does not always translate into better final performance, as concluded in [18] and reinforced by a study that distilled a complex deepfake detector into a model with 10,000 times fewer parameters while incurring only a small accuracy drop [19]. This is consistent with a basic principle in deep learning model development: adding network depth is only effective if matched with domain relevance and an appropriate training strategy [20]. This principle motivates a direct, controlled test between a simple model and a complex model on the same domain, rather than simply comparing accuracy scores across studies with differing experimental conditions, given that differences in dataset, sample size, and evaluation protocol across studies make direct cross-paper comparisons methodologically difficult to justify.

Studies comparing various pretrained architectures [10] generally do not include a simple CNN baseline and do not measure computational efficiency, leaving it unclear whether the reported accuracy advantage is commensurate with the computational cost incurred. The efficiency study [13] was conducted outside the deepfake domain, so its findings may not necessarily hold for the visual artifact characteristics typical of deepfake images. The study focusing on ResNet50 fine-tuning [14] evaluated only a single architecture without a comparison point, so it cannot show whether the high performance achieved is genuinely due to architectural complexity or is also attainable by a much simpler, easier-to-train model. It therefore remains empirically unclear whether fine-tuned ResNet50 truly outperforms a simple CNN when both are tested on exactly the same dataset, configuration, and computing environment, and, when compared on a shared test set, whether the difference is assessed using a statistical test appropriate for paired predictions rather than being compared solely on the basis of raw accuracy differences.

In practice, choosing a deep learning architecture is not only a matter of accuracy but also of implementation feasibility. System developers, social media platforms, and cybersecurity institutions generally operate with limited computing infrastructure, so training time and model size become just as important as accuracy alone. A model that is theoretically more accurate but requires much longer training time, large GPU memory, or a storage size that is impractical to update regularly may be less feasible to deploy than a lighter model with slightly lower performance. Testing a simple CNN and a fine-tuned ResNet50 within a single controlled experimental framework that uses the same dataset, model-appropriate preprocessing, and the same training environment is intended to provide a fairer and more practically useful answer than simply comparing reported accuracy figures obtained under differing experimental conditions.

This study aims to:

1. Compare the performance of a simple CNN trained from scratch against a fine-tuned two-phase ResNet50 in detecting deepfake images within a controlled and identical experimental framework.
2. Analyze the computational efficiency of both models based on training time and parameter count as indicators of implementation feasibility in resource-limited environments.
3. Empirically evaluate, within this controlled comparison, whether the more complex fine-tuned model consistently produces better detection performance than the simple CNN.
4. Determine the superior model, for the reported run, based on the balance between classification performance and computational efficiency.

This study hypothesizes that a simple CNN trained from scratch can achieve classification performance equal to or better than a fine-tuned ResNet50, with an accuracy difference that is not statistically significant, while offering a much greater computational efficiency advantage, particularly in training time and model size. A model is considered superior in this study if it achieves a minimum test accuracy of 85% with a shorter training time in the reported run.

METHOD

Research Design

This study uses a quantitative, controlled comparative experiment to examine the difference in deepfake image detection performance between two model pipelines: a simple CNN trained from scratch and a fine-tuned two-phase ResNet50, run on the same dataset, model-appropriate preprocessing, and training environment. Because architecture, initialization, preprocessing, learning-rate schedule, and trainable-layer configuration differ simultaneously, this design cannot isolate the causal contribution of architecture alone; the observed differences are attributed to the model-training pipeline as a whole. Both models were trained and evaluated on a single random train/validation/test split (seed = 42) and a single training run per model, without replication across seeds or repeated runs a limitation discussed at the end of this article.

Object and Sample

The object of this study is a digital image dataset consisting of two classes, real images and deepfake images, obtained from the public Deepfake and Real Images dataset on Kaggle [21], which originally contained 190,335 images (95,134 Fake, 95,201 Real after merging the original Train/Validation/Test folders). Owing to the computational limitations of the Google Colaboratory Free Tier, this study used a subset. A total of 16,000 candidate images per class were drawn from the combined pool, then filtered using exact (MD5) and near-duplicate (8×8 perceptual hash, Hamming = 0) checks applied jointly across both classes. This process removed 14 near-duplicates from the Fake candidates, and 467 exact duplicates and 170 near-duplicates from the Real candidates, leaving 10,000 unique images per class (20,000 total). This subset was divided into training (15,999 images, 80.0%), validation (2,001 images, 10.0%), and test (2,000 images, 10.0%) sets via stratified random sampling (seed 42). A dedicated verification step confirmed zero exact/near-duplicate images leaking across partitions, both before and after training.

Instrument

The primary instruments of this study are two deep learning architectures built using the TensorFlow/Keras framework:

1. A simple CNN consisting of three stacked convolutional blocks (32–64–128 filters) with BatchNormalization, GlobalAveragePooling2D, and a fully connected layer, trained from scratch (355,937 parameters, 354,273 trainable, ≈1.36 MB); and
2. ResNet50 with ImageNet-pretrained weights, trained using a two-phase transfer learning strategy feature extraction (all 175 layers frozen) and fine-tuning (last 30 layers activated) with a total of 24,771,457 parameters (1,182,209 trainable in the classification head during Phase 1; 23,589,248 in the frozen backbone).

All experiments were run on Google Colaboratory with a Tesla T4 GPU, supported by libraries such as kagglehub, scikit-learn, imagehash, statsmodels, and psutil for resource monitoring.

Procedures

The procedure covered dataset acquisition, deduplication and data splitting, preprocessing, model design, training, testing-evaluation, and comparative analysis (Figure 1).

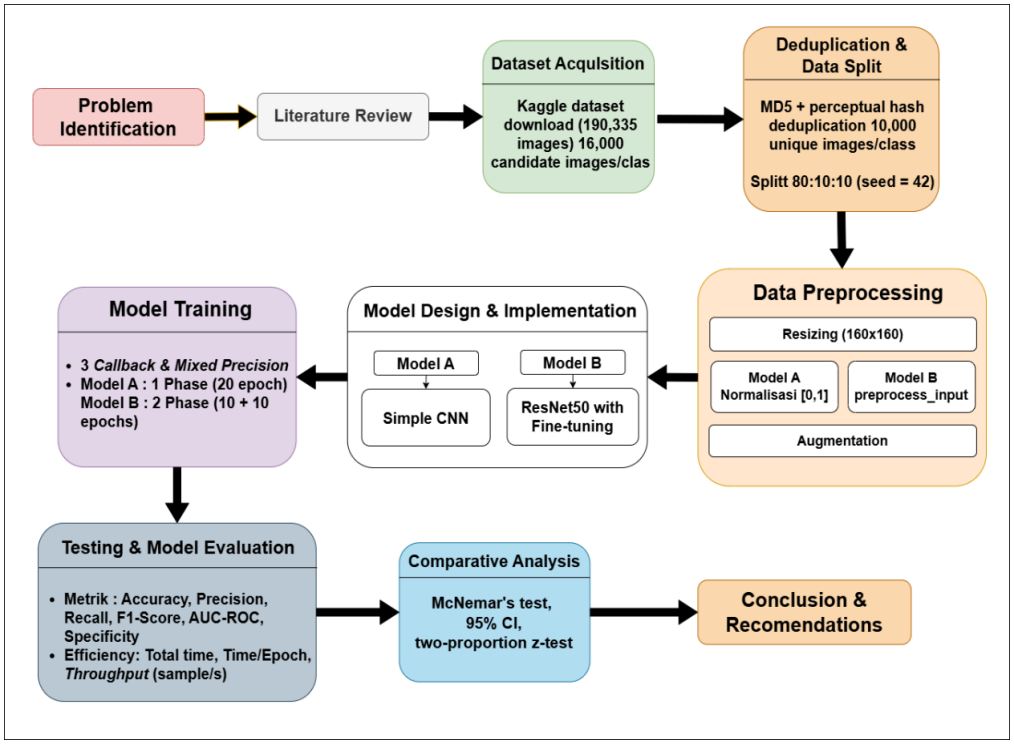

Figure 1. Research flow diagram.

After the data split, images were resized to 160×160 pixels. Preprocessing differed per model: the simple CNN received pixels normalized to [0,1], while ResNet50 received raw pixels [0,255] internally converted by keras.applications.resnet50.preprocess_input (BGR conversion + ImageNet mean centering), as required by its pretrained weights; uniform [0,1] normalization for both models as in an earlier version of this experiment would give ResNet50 an input distribution inconsistent with its pretrained features, so this model-specific preprocessing was used for all reported results. Augmentation (horizontal/vertical flip, brightness ±0.15, contrast [0.85–1.15], saturation [0.8–1.2], hue ±0.05) was applied identically to both pipelines, with clipping to each model's valid range after augmentation.

Both models were trained with an identical configuration: batch size 8 (chosen due to Tesla T4 VRAM limitations, particularly to accommodate ResNet50's memory footprint at 160×160 resolution with mixed precision), mixed precision (float16), Adam optimizer, binary cross-entropy loss, a 0.5 classification threshold, and three callbacks: EarlyStopping (validation accuracy, patience 7, min delta 1×10^{-4}, best weights restored), ReduceLROnPlateau (validation loss, factor 0.3, patience 4, min delta 1×10^{-5}, minimum LR 1×10^{-8}), and ModelCheckpoint (best weights based on validation accuracy). The simple CNN was trained for a maximum of 20 epochs (LR 1×10^{-4}); ResNet50 was trained in two phases — up to 10 epochs of feature extraction (LR 1×10^{-4}) and 10 epochs of fine-tuning (LR 1×10^{-5}). All relevant seeds (data split, weight initialization, augmentation/shuffling) were set to 42 before the dataset split and model construction.

Both models were evaluated on the same 2,000-image test set, which had not been seen during training/validation. The entire process, from dataset acquisition to final evaluation, was carried out in a single Google Colaboratory runtime session.

Analysis plan

Performance was analyzed using standard classification metrics (accuracy, precision, recall, F1-score, AUC-ROC, specificity) derived from the confusion matrix on the test set. Computational efficiency was analyzed through total training time, time per epoch, throughput, parameter count, and model size. Training-validation and validation-test consistency were assessed via the accuracy gap at the best epoch; since all subsets originate from the same source collection, these diagnostics indicate within-dataset consistency rather than generalization to a different dataset, manipulation method, or acquisition condition.

Because both models were evaluated on the same 2,000 test images, their predictions are paired, so the primary significance test is McNemar's test on the 2×2 contingency table of discordant–concordant predictions. As a descriptive complement, 95% confidence intervals for accuracy were computed per model (normal approximation for a binomial proportion), and a two-proportion z-test was used as additional context — not as the primary inferential evidence, since this test assumes independence that does not actually hold here. All comparisons are based on a single data split (seed 42) and a single run per model, not an average across multiple seeds.

Limitation

This study has several limitations. First, the 20,000-image subset drawn from 190,335 images may affect generalization, although its scale exceeds what has been shown adequate in a prior ResNet50 fine-tuning study [22]. Second, training was capped at 20 epochs due to the Colab Free Tier runtime limit (~12 hours), so ResNet50's performance with longer training remains unexplored. Third, this study did not compare other architectures beyond the simple CNN and ResNet50, nor hardware beyond Colab, so the computational efficiency results are infrastructure-specific. Fourth, all results are based on a single split and a single run per model (although data, weight, and augmentation seeds were controlled), so run-to-run variance from stochastic initialization cannot be assessed. Fifth, duplicate filtering targeted only exact matches and near-duplicates at Hamming distance = 0, and did not cover non-zero-distance near-duplicates (e.g., slight crops or rotations) that would require an all-pairs comparison; residual cases below this threshold cannot be fully ruled out even though cross-split checks found no matches. Sixth, training/validation/test consistency reflects performance

within a single source collection and does not indicate generalization to a different dataset, manipulation method, or acquisition condition.

RESULTS AND DISCUSSION

Results

The experiment was run using hyperparameter configurations tailored to each model but aligned with one another, including a 160×160 pixel input size, batch size 8, a maximum of 20 epochs, and mixed precision (float16) to optimize VRAM usage on the Tesla T4 GPU. The simple CNN used a fixed learning rate of 1×10^{-4}, while ResNet50 used a learning rate of 1×10^{-4} during the feature extraction phase and 1×10^{-5} during fine-tuning.

The simple CNN architecture, consisting of three stacked convolutional blocks with a gradually increasing number of filters (32→64→128), GlobalAveragePooling2D, and a fully connected layer, is shown in Figure 2.

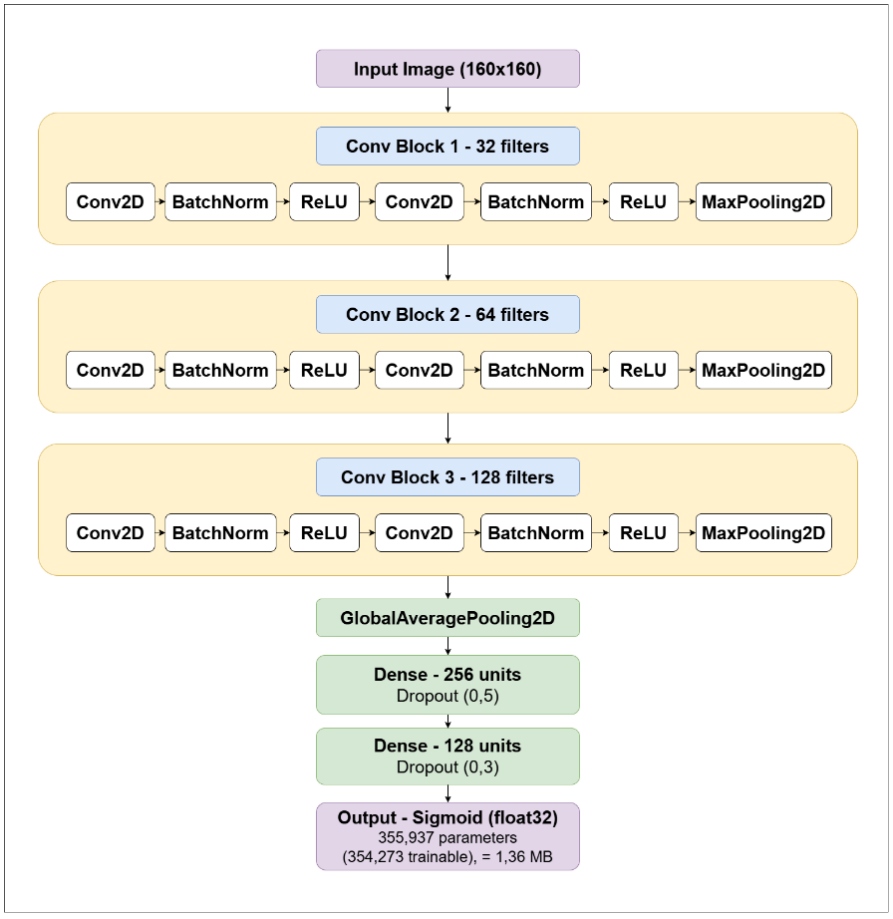

Figure 2. Simple CNN architecture.

The ResNet50 architecture, trained using a two-phase transfer learning strategy (feature extraction followed by fine-tuning of the last 30 layers) with ResNet-specific input preprocessing applied inside the model, is shown in Figure 3.

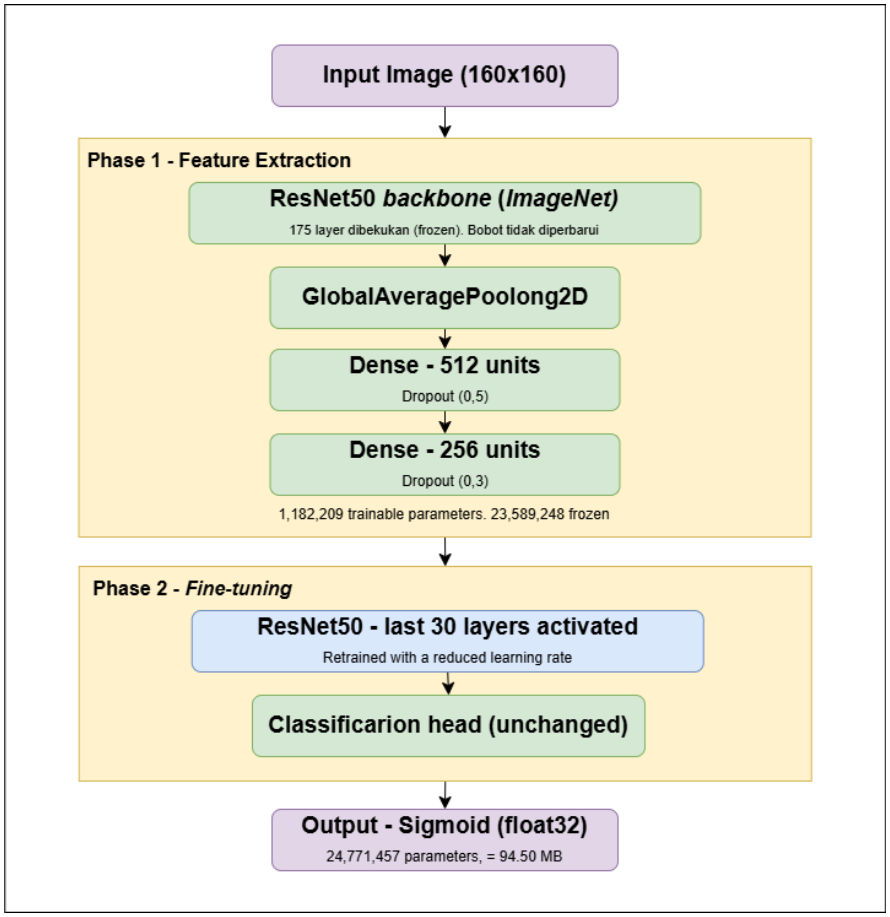

Figure 3. ResNet50 architecture with two-phase transfer learning.

The simple CNN was trained for the full 20 epochs, with the best weights restored from epoch 19, achieving a validation accuracy of 90.90% and a validation loss of 0.2333. ResNet50 was trained in two phases: feature extraction completed in 422.5 seconds (7.0 minutes), followed by fine-tuning, which completed in 567.3 seconds (9.5 minutes); the best combined validation accuracy across both phases reached 91.00% at epoch 17, with a validation loss of 0.2270. On the 2,000-image test set, the simple CNN outperformed ResNet50 on six of seven evaluation metrics, while ResNet50 achieved higher recall, as summarized in Table 1.

Table 1. Comparison of evaluation results on the test set

Metric	Simple CNN	ResNet50 FT	Best
Loss	0.2370	0.2422	CNN
Accuracy	0.9090 (90.90%)	0.8980 (89.80%)	CNN
AUC-ROC	0.9708	0.9652	CNN
Precision	0.9018	0.8734	CNN
Recall	0.9180	0.9310	ResNet50
F1-Score	0.9098	0.9013	CNN
Specificity	0.9000	0.8650	CNN

Using the 95% normal-approximation confidence interval for a binomial proportion (n = 2,000 for each model), the simple CNN's test accuracy of 90.90% corresponds to a 95% CI of [89.64%, 92.16%], while ResNet50's test accuracy of 89.80% corresponds to a 95% CI of [88.47%, 91.13%]; the two intervals overlap. A two-proportion z-test comparing the two accuracies produced z = 1.18 (p = 0.239), reported here only as descriptive context, since this test treats the two sets of 2,000 predictions as independent even though both were evaluated on the same test images.

Because both models' predictions on the shared test set are paired, McNemar's test was used as the primary significance test. Of the 2,000 test images, both models were correct on 1,673 images and both incorrect on 59 images; the CNN was correct while ResNet50 was wrong on 145 images, and

ResNet50 was correct while the CNN was wrong on 123 images. McNemar's chi-square statistic with continuity correction was 1.65 (p = 0.200), indicating that the accuracy difference between the two models is not statistically significant at = 0.05 consistent with the overlapping descriptive confidence intervals above. Confusion matrix analysis (Tables 2 and 3) shows that the simple CNN correctly detected 918 of 1,000 deepfake images (82 False Negatives) and misclassified 100 real images as deepfake (False Positives). For comparison, ResNet50 correctly detected 931 deepfake images (69 False Negatives) and misclassified 135 real images as deepfake.

Table 2. Confusion matrix of the simple CNN (N = 2,000)

	Predicted: Fake	**Predicted: Real**
Actual: Fake	918 (TP)	82 (FN)
Actual: Real	100 (FP)	900 (TN)

Training-validation consistency was fairly good for both models. The CNN's training accuracy at the best epoch (86.71%) was below its validation accuracy (90.90%, a −4.19 percentage-point gap), and its validation-to-test accuracy gap was 0.00%. ResNet50's training accuracy at the best epoch (91.33%) was close to its validation accuracy (91.00%, a +0.33 percentage-point gap), and its validation-to-test accuracy gap was 1.20%. No pattern indicating clear overfitting was observed in the reported run; since all three subsets originate from the same source collection, these gaps indicate within-dataset consistency, not generalization to unseen data.

Table 3. Confusion matrix of the fine-tuned ResNet50 (N = 2,000)

	Predicted: Fake	**Predicted: Real**
Actual: Fake	931 (TP)	69 (FN)
Actual: Real	135 (FP)	865 (TN)

In terms of computational efficiency, the simple CNN completed training in 730.6 seconds (12.2 minutes), faster than ResNet50, which required 989.8 seconds (16.5 minutes) in this run, while also being about 69.5 times smaller in parameter count, as detailed in Table 4.

Table 4. Comparison of computational efficiency

Efficiency Metric	**Simple CNN**	**ResNet50 FT**
Total training time	730.6 s (12.2 min)	989.8 s (16.5 min)
Throughput (sample/s)	438.0	323.3
Parameter count	355,937	24,771,457
Model size	1.36 MB	94.50 MB
Test accuracy	0.9090 (90.90%)	0.8980 (89.80%)

Finally, the exact- and near-duplicate filtering applied jointly across both classes before sampling removed 0 exact and 14 near-duplicate images from the Fake candidates, and 467 exact and 170 near-duplicate images from the Real candidates. Post-split verification, repeated both immediately after the data split and after training and evaluation were complete, confirmed zero exact-duplicate or identical perceptual-hash images spread across the training, validation, and test partitions.

The simple CNN's advantage on six of seven metrics is most likely related to its architectural design, although this study's controlled design cannot fully separate the contribution of architecture from other shared factors such as the sampled data subset, the shared training-duration cap, and the Google Colaboratory training environment, discussed further in the Limitations section. The three stacked convolutional blocks, with a gradually increasing number of filters (32→64→128), follow the principle of hierarchical feature learning, in which early layers capture simple patterns such as edges and textures while later layers capture patterns more specific to deepfake manipulation artifacts. The use of GlobalAveragePooling2D instead of Flatten also helps reduce the parameter count while providing an implicit regularization effect, since this layer has no parameters to optimize and thus naturally reduces the tendency toward overfitting [23].

With the correct ResNet-specific preprocessing applied, ResNet50's performance improved substantially compared with an earlier version of the experiment that used incorrect preprocessing, and its test accuracy (89.80%) is now close to, and statistically indistinguishable from, the simple CNN (90.90%; McNemar's p = 0.200). ResNet50 achieved the higher recall of the two models (93.10%

vs. 91.80%), meaning it proportionally missed fewer actual deepfakes, at the cost of a higher false-positive rate on real images (135 vs. 100). This precision–recall trade-off is a meaningful practical consideration regardless of overall accuracy: a system that prioritizes catching as many deepfakes as possible, even at the cost of more false alarms on genuine content, might prefer ResNet50's operating point, while a system that prioritizes minimal disruption to legitimate content might prefer the CNN.

The finding that a simple CNN can match a fine-tuned ResNet50 in this controlled comparison broadly aligns with [24], which concluded that architectural complexity does not always translate into better final performance in deepfake detection, and also aligns with this study's own hypothesis, which anticipated a statistically non-significant accuracy difference rather than a clear advantage for either model. This finding should be read as evidence of comparable performance under the current controlled conditions, not as proof that a simple CNN generally surpasses a fine-tuned ResNet50 under other conditions.

These findings carry practical implications for architecture selection in deepfake detection systems, particularly in infrastructure-constrained scenarios. The simple CNN's compact model size (1.36 MB) and lower parameter count make it a promising candidate for deployment on edge or mobile devices; however, this study evaluated computational efficiency only from the training side (training time, throughput, parameter count) and did not benchmark inference latency, memory footprint, energy consumption, or performance on representative mobile/edge hardware, so this deployment potential should be treated as a direction requiring dedicated evaluation rather than an established conclusion. ResNet50, at 94.50 MB, remains better suited to server infrastructure with dedicated GPUs. In this run, the simple CNN's shorter training time also indicates a potential computational cost advantage if the model needs to be retrained periodically to keep up with evolving deepfake techniques, although this advantage was observed in a single run and has not been confirmed across repeated trials. The false-negative rate of both models (8.2% for the CNN, 6.9% for ResNet50) is not yet low enough to support deployment as the sole security mechanism in critical systems such as identity verification; both are more appropriately used as an initial detection layer combined with further review for ambiguous cases.

This study contributes by combining classification accuracy, training–validation–test consistency diagnostics, and computational efficiency within a single controlled experimental framework specific to deepfake image detection, applying a statistical test appropriate for a shared test-set design (McNemar's test) rather than relying solely on independence-based tests. Through a corrected replication of the experiment, this study also demonstrates the practical importance of applying architecture-specific preprocessing to pretrained models and of filtering cross-partition duplicates before not only after the train/validation/test split is finalized, both of which significantly changed the comparison results relative to an earlier, uncorrected version of the experiment. This study provides empirical evidence that directly tests the assumption that complex, fine-tuned architectures always outperform simpler models trained from scratch, an assumption rarely tested under controlled conditions, particularly in the deepfake domain.

This study has several limitations worth noting. The 20-epoch cap, imposed by Google Colaboratory Free Tier constraints, means ResNet50's performance with much longer training remains unexplored. This study also did not test other deep learning architectures beyond the simple CNN and ResNet50, nor was it evaluated on hardware outside the Google Colab environment, so the computational efficiency results are specific to that infrastructure. All findings are based on a single random data split and a single training run per model, without repeated trials across multiple seeds; the reported performance differences should therefore be read as a single-trial comparison rather than an average with an estimate of run-to-run variance. McNemar's test, reported as the primary significance test, is appropriate for the paired prediction structure of this comparison and shows no significant difference in accuracy between the two models on this test set; this result pertains to sampling uncertainty within the single training run underlying this fixed test set, and does not by itself establish that the model ranking would remain stable across different weight initializations or resampled splits, which would require multi-seed replication. Finally, duplicate filtering in this study covered exact matches and identical perceptual-hash near-duplicates; it did not remove near-

duplicates at non-zero Hamming distances, which could be further addressed through a thorough all-pairs comparison.

Based on these limitations, future research is encouraged to run experiments using the full dataset (190,335 images) on a more capable computing platform, and to extend ResNet50 training beyond 20 epochs to more thoroughly explore its potential with longer training. Replicating this comparison across multiple random seeds and data splits would help establish whether the current findings are stable rather than specific to a single run. Testing pretrained architectures more relevant to the facial domain, such as EfficientNet-B0, which has shown strong performance in combination with ResNet50 for image classification tasks [25], is also recommended. For system developers, the simple CNN can be considered as an initial detection layer in edge- or mobile-based applications, pending dedicated inference hardware benchmarks, combined with additional verification mechanisms for cases that fall into a gray area.

CONCLUSION

This study was designed to address the gap outlined in the introduction namely, the absence of a controlled comparison between a simple CNN and a fine-tuned ResNet50 within an identical experimental framework that also accounts for computational efficiency and applies a statistically appropriate paired significance test. The experimental results support the study's hypothesis: the simple CNN trained from scratch achieved a test accuracy of 90.90%, equivalent to and statistically indistinguishable from the fine-tuned ResNet50's 89.80% (McNemar's $p = 0.200$), while outperforming ResNet50 on six of seven descriptive metrics and offering a far greater efficiency advantage — approximately 69.5 times smaller in model size and faster to train in this run. Because this comparison is based on a single random data split and a single training run per model, and both models shared the same data subset, training-duration cap, and Colab environment, these findings should be interpreted as evidence from a single controlled run, not as a precision estimate fully attributable to architecture or stable across runs. The efficiency advantage held consistently across all reported metrics (training time, throughput, parameter count, model size), while the accuracy comparison, though descriptively favoring the CNN, did not reach statistical significance — a pattern consistent with, rather than contradicting, the study's hypothesis of an equivalent, statistically non-significant accuracy difference accompanied by a substantial efficiency advantage.

These findings open opportunities for further development, particularly on aspects not yet fully explored in this study. Going forward, this research could be developed into a deepfake detection system evaluated for deployment on mobile or edge applications, pending dedicated inference hardware benchmarks, given the simple CNN's very compact model size. The same approach could also be extended to test other pretrained architectures more relevant to the facial domain, replicated across multiple seeds, and combined with frequency- or temporal-based analysis to improve the system's robustness against future video-based deepfakes.

ACKNOWLEDGMENT

The authors would like to thank Ms. Asti Herliana, S.Kom., M.Kom., as the supervising lecturer, for her guidance, direction, and valuable input throughout the research and manuscript preparation process. The authors also extend their appreciation to the Informatics Engineering Study Program of Universitas Adhirajasa Reswara Sanjaya for the academic facilities provided during this research. This study did not receive any specific funding from any party.

AUTHOR CONTRIBUTION STATEMENT

RTM was responsible for conceptualization, methodology design, model implementation, experimentation, data analysis, and manuscript preparation. AH supervised the research process and provided direction on the research design, and reviewed and edited the manuscript for critical intellectual content. Both authors have read and approved the final manuscript**.**

AI DISCLOSURE STATEMENT

The authors used Claude.ai in preparing this manuscript to refine the language, check grammar, restructure sentences, correct the deepfake detection preprocessing pipeline and data-splitting procedure in line with peer-review notes, and improve academic clarity. After using this tool/service, the authors thoroughly reviewed and edited the content as needed and take full responsibility for the content of this publication.

REFERENCES

[1] C. Zhai, S. Wibowo, and L. D. Li, "The effects of over-reliance on AI dialogue systems on students' cognitive abilities: a systematic review," *Smart Learning Environments*, vol. 11, no. 1, Jun. 2024, doi: 10.1186/S40561-024-00316-7.

[2] J. M. Camacho, A. Couce-Vieira, D. Arroyo, and D. Rios Insua, "A Cybersecurity Risk Analysis Framework for Systems with Artificial Intelligence Components," 2024, Accessed: Jul. 16, 2026. [Online]. Available: https://doi.org/10.48550/arXiv.2401.01630

[3] V. E. Pattiradjawane *et al.*, "Deepfakes as a New Cybersecurity Threat: A Literature Review of Risks to Digital Identity, Social Engineering, and Information Trust" *ALGORITHM: Journal of Computer Science and Computational Intelligence*, vol. 2, no. 1, pp. 19–30, May 2026, doi: 10.30598/ALGORITHM.V2I1.19-30.

[4] M. S. Rana, M. N. Nobi, B. Murali, and A. H. Sung, "Deepfake Detection: A Systematic Literature Review," *IEEE Access*, vol. 10, pp. 25494–25513, 2022, doi: 10.1109/ACCESS.2022.3154404.

[5] V. Pirogov, "Visual Language Models as Zero-Shot Deepfake Detectors," Jul. 2025, Accessed: Jul. 16, 2026. [Online]. Available: https://arxiv.org/pdf/2507.22469

[6] S. D. Bray, S. D. Johnson, and B. Kleinberg, "Testing human ability to detect 'deepfake' images of human faces," *J. Cybersecur.*, vol. 9, no. 1, Jan. 2023, doi: 10.1093/CYBSEC/TYAD011.

[7] S. N. Amartama, A. N. Hidayah, P. K. Sari, and R. A. Ramadhani, "Implementation of Convolutional Neural Networks (CNN) in Handwritten Pattern Recognition" *Seminar Nasional Teknologi & Sains*, vol. 3, no. 1, pp. 133–138, Jan. 2024, doi: 10.29407/STAINS.V3I1.4155.

[8] A. Puspitasari, D. Sava Salsabila, and D. Roliawati, "Application of ResNet-50 CNN for Classification Optimization on Fashion Data" *INDONESIAN JOURNAL ON DATA SCIENCE*, vol. 3, no. 1, pp. 1–12, Jun. 2025, doi: 10.30989/IJDS.V3I1.1533.

[9] L. Alzubaidi *et al.*, "Review of deep learning: concepts, CNN architectures, challenges, applications, future directions," *J. Big Data*, vol. 8, no. 1, Mar. 2021, doi: 10.1186/S40537-021-00444-8.

[10] Z. N. Ashani *et al.*, "Comparative Analysis of Deepfake Image Detection Method Using VGG16, VGG19 and ResNet50," *Journal of Advanced Research in Applied Sciences and Engineering Technology Journal homepage: Journal of Advanced Research in Applied Sciences and Engineering Technology*, vol. 47, no. 1, pp. 16–28, 2025, doi: 10.37934/araset.47.1.1628.

[11] Kusnaeni, I. R. Adriani, M. S. Hafid, A. Budi Andy B, and M. E. Rizal, "Deepfake Image Classification Using ResNet50 Feature Extraction and XGBoost Learning Model," *Journal of Mathematics, Computations and Statistics*, vol. 8, no. 2, pp. 258–268, Sep. 2025, doi: 10.35580/JMATHCOS.V8I2.8387.

[12] L. Guarnera *et al.*, "The Face Deepfake Detection Challenge," *J. Imaging*, vol. 8, no. 10, Oct. 2022, doi: 10.3390/JIMAGING8100263.

[13] H. A. Pangestu and Kusrini, "Improving ResNet50 Architecture Performance to Address Overfitting in Skin Disease Classification" *TEMATIK*, vol. 11, no. 1, pp. 65–71, Jun. 2024, doi: 10.38204/TEMATIK.V11I1.1876.

[14] Y. A. Chouhan, C. Chudasama, and D. K. Verma, "Deepfake Detection Using ResNet50: Performance Analysis on Celeb-DF Dataset," *Communications in Computer and Information Science*, vol. 2426 CCIS, pp. 397–404, 2025, doi: 10.1007/978-3-031-86296-0_28.

[15] E. H. I. Eliwa, "Enhancing Skin Cancer Diagnosis Through Fine-Tuning of Pretrained Models: A Two-Phase Transfer Learning Approach," *Int. J. Breast Cancer*, vol. 2025, no. 1, 2025, doi: 10.1155/IJBC/4362941.

[16] L. Feng *et al.*, "Adaptive multi-source domain collaborative fine-tuning for transfer learning," *PeerJ Comput. Sci.*, vol. 10, p. e2107, Jun. 2024, doi: 10.7717/PEERJ-CS.2107/SUPP-1.

[17] D. R. Rochmawati and L. Maryani, "Deep Learning-Based ResNet-50 Transfer Learning Approaches for Pneumonia Detection from Chest X-Ray Images: With and Without Fine-Tuning," *Indonesian Journal of Health Research and Development*, vol. 3, no. 3, pp. 146–153, Nov. 2025, doi: 10.58723/IJHRD.V3I3.507.

[18] J. Mallet, L. Pryor, R. Dave, and M. Vanamala, "Deepfake Detection Analyzing Hybrid Dataset Utilizing CNN and SVM," *ACM International Conference Proceeding Series*, pp. 7–11, Jan. 2023, doi: 10.1145/3596947.3596954.

[19] B. Cavia, E. Horwitz, T. Reiss, and Y. Hoshen, "Real-Time Deepfake Detection in the Real-World," Jun. 2024, Accessed: Jul. 16, 2026. [Online]. Available: https://arxiv.org/pdf/2406.09398

[20] S. T. Erukude, V. C. Marella, and S. R. Veluru, "Fourier-Based GAN Fingerprint Detection using ResNet50," *Proceedings of the 9th International Conference on Inventive Systems and Control, ICISC 2025*, pp. 971–976, Oct. 2025, doi: 10.1109/ICISC65841.2025.11187724.

[21] M. Karki, "deepfake and real images." Accessed: May 10, 2026. [Online]. Available: https://www.kaggle.com/datasets/manjilkarki/deepfake-and-real-images?resource=download

[22] A. Pakala, M. M. Rahman, S. Memon, and T. Ahmed, "Comparative Evaluation of Deep Learning Models for Fake Image Detection," May 2026, Accessed: Jul. 15, 2026. [Online]. Available: https://arxiv.org/pdf/2605.20971

[23] Y. Dogan, "A New Global Pooling Method for Deep Neural Networks: Global Average of Top-K Max-Pooling," *Traitement Du Signal*, vol. 40, no. 2, pp. 577–587, Apr. 2023, doi: 10.18280/TS.400216.

[24] A. Angeline and H. Kusniyati, "Performance Comparison of VGG19, ResNet50, DenseNet121, and MobileNetV2 in Detecting Deepfake Images" *CESS (Journal of Computer Engineering, System and Science)*, vol. 9, no. 2, pp. 397–411, Jul. 2024, doi: 10.24114/CESS.V9I2.58671.

[25] D. Vaishnavi, S. Jegan, J. Ganesh, and S. Sundaram, "Hybrid Deep Learning Approach for Deepfake Detection Using ResNet50 and EfficientNetB0," *Journal of Innovative Image Processing*, vol. 7, no. 3, pp. 622–638, Aug. 2025, doi: 10.36548/JIIP.2025.3.003.