Hybrid Deep Learning for Image Authenticity: Distinguishing Between Real and AI-Generated Images

Wella[1,*], Suryasari[2], Ririn Ikana Desanti[3]

[1,2,3]*Universitas Multimedia Nusantara, Jl. Scientia Boulevard Gading Serpong, Tangerang, 15810, Indonesia*

(Received: May 18, 2025; Revised: July 12, 2025; Accepted: November 1, 2025; Available online: December 1, 2025)

Abstract

The increasing use of artificially generated images raises significant concerns about the authenticity of digital content. This study introduces a hybrid deep learning model for binary classification of real and generated images by combining spatial and relational features. The central idea is to integrate a convolutional backbone adapted from ResNet18 for visual feature extraction with a graph representation based on nearest-neighbor relations to capture inter-image similarities. The objective is to evaluate whether this dual-feature approach improves classification performance compared to single-feature baselines. Using a balanced dataset of 1,256 images (744 real and 512 generated), the model was trained on 70% of the data and tested on the remaining 30%. Experimental findings demonstrate that the model achieved an overall accuracy of 88%, with precision of 0.91 and recall of 0.89 for real images, and precision of 0.85 and recall of 0.87 for generated images. The corresponding F1 scores were 0.90 and 0.86, yielding a macro average F1 of 0.88. Confusion matrix analysis shows balanced misclassification across both classes, while stable performance across epochs indicates reliable learning behavior. Results confirm that the hybrid model achieves stronger classification effectiveness than convolution-only or graph-only baselines. The novelty of this work lies in demonstrating that the integration of spatial and relational learning provides a more robust framework for detecting synthetic images than single-modality approaches. The contribution of this research is both methodological, in proposing a hybrid architecture that unifies convolutional and graph-based learning, and practical, in providing empirical evidence that such integration enhances the reliability of image authenticity verification. While the absence of a validation set limited hyperparameter optimization and early stopping, the findings indicate that this hybrid design offers a promising direction for improving the robustness and generalizability of synthetic image detection.

Keywords: Deepfake Detection, Hybrid CNN-GNN, Graph Neural Networks, Image Classification, k-NN Graph Construction

1. Introduction

The rapid advancement of artificial intelligence-driven synthetic media, such as deep-fakes, has emerged as a serious threat to information integrity and individual privacy. During 2023 and 2024, deep-fake media have been widely used in political disinformation, non-consensual pornography, and catfishing scams, resulting in significant social and legal consequences [1], [2]. A notable case in South Korea in late 2024 revealed that deepfake materials were used to harass students and educators through the Telegram platform, leading to over 800 recorded criminal cases within a few months [3], [4]. This alarming trend prompted legislative responses, including the criminalization of both the possession and distribution of explicit deep-fake content.

A significant rise in deepfake content on social media has been documented in 2024, with public concern particularly centered on the prevalence of nonconsensual deepfake pornography [5], [6], [7]. In 2023, independent analysis reported over 244,000 such videos across major platforms, a 54 percent increase from the previous year, which led experts to call for more effective content moderation and legislative action to protect victims, primarily women, from psychological harm and privacy violation [8]. Further evidence of public concern emerged in early 2025 in a sentiment analysis study of over 17,700 deepfake related social media posts. The results revealed that 47 percent of public sentiment toward the abuse of deepfakes in adult content was overwhelmingly negative, reflecting widespread social anxiety about the ethical, legal, and emotional implications of non-consensual use [9]. Together, these findings reveal both the scale of abuse and the urgency of reliable detection solutions.

***Corresponding author: Wella (wella@umn.ac.id)**
DOI: https://doi.org/10.47738/jads.v7i1.991

ISSN 2723-6471

A comprehensive review published in late 2024 emphasized that existing deep-fake detection systems frequently lack adversarial robustness, real-time capability, and standardized evaluation metrics. The study highlighted the urgent need for resilient and adaptive detection frameworks as new manipulation methods emerge rapidly [10]. Additionally, an empirical analysis published in mid-2023 investigated how well deep-fake image detectors generalize unseen datasets. The authors concluded that most detectors fail in zero-shot scenarios, often overfitting to synthesis methods present in the training data. They identified the presence of a small set of neurons that consistently contribute to detection across different forgery types, suggesting a possible path toward improving generalizability [11]. These limitations indicate that improving technical detection strategies is a necessary step to complement legal and policy interventions. In other words, the social and ethical consequences of deepfake proliferation directly motivate the development of more accurate and resilient computational methods for image authenticity verification. These phenomena underscore the urgent need for detection systems that are capable not only of recognizing manipulated media, but also of doing so in a manner that is robust, explainable, and sensitive to individual privacy concerns.

Traditional detection methods based on pure Convolutional Neural Networks (CNNs) have proven effective in capturing spatial features within images or video frames [12], [13], [14]. However, these methods often fall short in identifying subtle structural manipulations and contextual relationships either between frames or across image regions. Therefore, there is a growing need for approaches that can explicitly model spatial representations along with topological dependencies in visual data [15]. To address these gaps, researchers have begun exploring hybrid models that incorporate graph-based learning. For instance, El-Gayar et al. introduced the FuNet model, where image patches are transformed into graph nodes and CNN–GNN fusion is applied for video-based deepfake detection. This architecture demonstrated enhanced accuracy and robustness against adversarial interference [16]. Similarly, Khalid et al. proposed DFGNN, an interpretable and generalizable graph-based model for facial deep-fake detection, utilizing facial patches as graph nodes and a multi-scale pyramid structure to capture fine-grained features through graph-based reasoning [11]. To date, most hybrid CNN–GNN studies have focused predominantly on video rather than still images. While models such as FuNet and DFGNN have shown the effectiveness of graph-based architectures in detecting video deepfakes, the application of similar hybrid approaches to single images remains underexplored, with no systematically designed models specifically targeting static deepfake detection.

This study addresses that gap by developing and evaluating a hybrid CNN–GNN model tailored for detecting manipulated static facial images. The model is designed to integrate the strengths of CNNs in capturing fine-grained visual details with the ability of GNNs to represent relational dependencies among image patches or facial regions. Through this integration, the proposed framework seeks to improve detection accuracy while providing a foundation for more interpretable analysis of manipulated content. By situating the technical contribution within the broader societal problem, this research responds directly to the growing demand for effective, generalizable, and ethically relevant solutions to the deepfake challenge.

Building on the pressing need for more robust deepfake image detection methods, particularly for still images, it is important to examine how recent studies have leveraged hybrid CNN–GNN architectures across various domains. A review of these prior works provides insights into how the integration of spatial feature extraction and relational modeling has been approached in different contexts and highlights both the potential and current limitations of such methods in detecting manipulated visual content.

2. Literature Review

Building on the pressing need for more robust deepfake image detection methods, particularly for still images, it is important to examine how recent studies have leveraged hybrid CNN–GNN architectures across various domains. A review of these prior works provides insights into how the integration of spatial feature extraction and relational modeling has been approached in different contexts and highlights both the potential and current limitations of such methods in detecting manipulated visual content.

The reviewed studies have demonstrated that hybrid models integrating CNN, and Graph Neural Networks (GNN) can effectively address visual classification problems that require both spatial feature extraction and relational reasoning.

 ISSN 2723-6471

In these studies, convolutional features were typically extracted first, followed by graph-based modeling to capture structural relationships within the data.

The hybrid CNN–GNN architecture was employed to detect deepfake videos [16]. Image patches were converted into graph nodes, and spatial relationships among them were modeled using a graph neural network. A similar approach was implemented, where a CNN–GNN model was developed for the detection of soybean leaf diseases [17]. Spatial patterns of symptoms were represented as graph structures derived from CNN-extracted features to improve interpretability and detection accuracy. In the other research, object-level features were extracted from scenes and used to build graph representations, which were then processed by a lightweight graph convolutional network for scene classification [16].

A focus on architectural integration was presented in previous research. A multimodal graph attention network was constructed to capture spatial, frequency-based, and landmark-based representations across video frames [18]. Temporal and cross-modal dependencies were modeled through graph attention mechanisms applied to frame-level embeddings. In the scene classification, a generic framework known as CNN2GNN was proposed, where a transition module was designed to convert feature maps produced by a CNN into graph structures suitable for GNN processing [19].

Across these studies, improved performance was consistently reported when CNN–GNN architectures were used in place of traditional single-model designs, particularly in cases where spatial dependencies or topological patterns were central to the problem. For deep-fake detection tasks, the hybrid approaches introduced in El-Gayar et al. [16] and Khormali et al. [18] were shown to enhance model robustness and accuracy by combining local feature representations with graph-based relational reasoning.

3. Methodology

The CRISP-ML(Q) framework, as depicted in figure 1, offers a systematic approach for managing machine learning workflows. It emphasizes iterative refinement and incorporates quality control across the entire pipeline [20]. Building on the foundation of CRISP-DM, which was originally developed for data mining applications, this framework has been adapted to meet the demands of modern machine learning systems. It organizes the project lifecycle into three essential phases: understanding the problem and the data, constructing the predictive model, and maintaining the model in operational settings. Each phase is crucial for ensuring that the resulting model performs reliably and adapts well to real-world challenges, particularly in complex tasks such as detecting deepfake imagery.

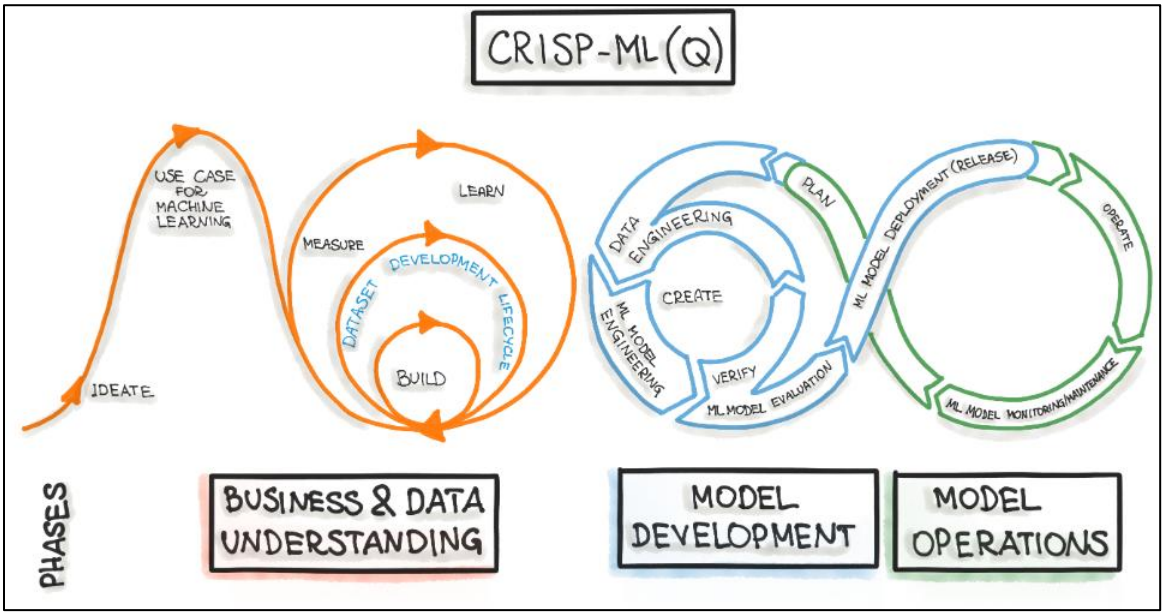

Figure 1. CRISP-ML(Q) framework

 ISSN 2723-6471

The initial stage focuses on defining the objectives of the machine learning task, understanding its alignment with broader concerns related to digital security and misinformation, and assessing data availability and characteristics. In the context of deep-fake detection, this phase involves identifying the threat posed by synthetic media to individual privacy and public trust and determining how a predictive model can assist in mitigating such risks. The analysis includes investigating image-level manipulations, evaluating class distributions, and selecting datasets that represent realistic deepfake scenarios. A thorough understanding of visual and spatial attributes is essential to inform model design and feature representation strategies.

This stage encompasses the design and implementation of predictive architecture. For this study, a hybrid model integrating a CNN for visual feature extraction and a GNN for modeling spatial relationships is proposed. Key tasks include image preprocessing, facial patch segmentation, graph construction, and model fusion. Training and validation are performed using standard benchmarks, and model performance is evaluated through metrics such as accuracy, F1-score, and AUC. Special attention is given to cross-dataset generalization and robustness against perturbations, which are critical in real-world applications of deepfake detection.

The final stage addresses the practical deployment of the model and its long-term maintenance. Although this study focuses on model development and evaluation, the proposed architecture is designed with deployment feasibility in mind. Considerations include model compression for deployment on edge devices, inference speed, and compatibility with forensic or content moderation systems. Post-deployment, ongoing performance monitoring is necessary to detect concept drift as new types of deepfake content emerge. Continuous refinement and retraining using updated datasets ensure that the model remains effective in operational environments and responsive to evolving threat landscapes.

4. Results and Discussion

4.1. Business and Data Understanding

In the initial stage of this study, a comprehensive understanding of both the business context and the dataset was established. The core objective was to develop a reliable detection model for identifying manipulated images, particularly in the context of deep-fake forensics. The dataset selected for this purpose was CASIA v2, which contains a total of 12,560 images comprising 7,437 tampered and 5,123 authentic images. These images were obtained from a publicly available source on Kaggle [21]. To highlight inconsistencies indicative of digital manipulation, figure 1 depicts the application of Error Level Analysis (ELA). This technique involved comparing each original image with its recompressed version, followed by enhancing the brightness of the resulting ELA image to accentuate compression artifacts. These processed images were used to facilitate the model's learning process. The dataset was organized through the creation of a custom class named CASIA2_ELA, which stored image file paths and returned two outputs for each item: the ELA-transformed image and its corresponding label, determined by the presence of the substring "Tp" in the filename. All images were resized to 128 by 128 pixels and converted into tensor format to ensure compatibility with the training pipeline. This preparatory phase ensured that the dataset was both semantically meaningful and technically suitable for subsequent modeling.

The ela_image function is used to show the difference between an original image and its lower resolution counterpart; in this case the one saved with 90% of its original quality. The ELA image's brightness is then enhanced by a certain value. The result of this operation can be found in figure 2.

Figure 2. ELA image's brightness results

 ISSN 2723-6471

On the left side is the original image of a zebra in a natural setting, loaded using the Image.open(real_img) function. This image represents an unaltered, real image sample from the CASIA v2 dataset. On the right side is the result of the ela_image(real_img) function, which highlights the compression artifacts introduced by saving the image at a slightly lower quality (e.g., 90%). In this ELA-transformed image, areas of high compression are brighter, while regions with minimal change remain darker. Since the input image is authentic, the resulting ELA output appears mostly uniform and low in intensity, with the zebra's silhouette faintly visible. This subtle representation is expected for unmanipulated images, where compression inconsistencies are minimal.

This transformation is critical for training the detection model, as manipulated images typically exhibit more pronounced ELA artifacts due to localized editing. By including these ELA outputs as input features, the model is better equipped to distinguish between authentic and tampered images.

The second step is CASIA v2 data preprocessing to identify visual manipulation. Integration with PyTorch enables high efficiency in batch training and compatibility with the CNN–GNN model architecture used in this study. The dataset is defined as CASIA2_ELA, which stores an array of image_paths and accepts a transform parameter that can be used to define image preprocessing methods (if needed). It also returns 2 additional items: an ELA image (ela_img) of each respective image stored in image_paths; along with its label, which is determined by whether its file name includes "Tp" (1 if true).

After finishing data preprocessing, the next step is data splitting. First, the labels from the dataset are extracted by iterating through each data point and retrieving the corresponding target value. These labels are then used to perform a stratified split, which helps maintain the original class distribution between data subsets. This step is important in binary classification tasks, such as identifying tampered versus authentic images, to avoid bias caused by imbalanced class representation.

The dataset is divided into training, validation, and testing sets, with 80% of the data used for training, and 10% each for validation and testing. Subset objects are then created for each split, with a fixed random state to ensure the reproducibility of the split. After the split, the training dataset consists of 5,950 real images and 4,098 generated images, providing a diverse and balanced set of examples for the model to learn distinguishing features between authentic and tampered content. The validation dataset contains 743 real images and 513 generated images, which helps fine-tune the model and prevent overfitting. Meanwhile, the testing dataset comprises 744 real images and 512 generated images, enabling a comprehensive evaluation of the model's ability to generalize its detection performance to unseen data.

Finally, DataLoaders are configured for all subsets with a batch size of 32, meaning each batch in a data loader will have 32 images. The training loaders are shuffled to promote robust learning, and the remaining loader maintaining a fixed order to ensure consistent evaluation. The result is a reproducible and balanced data pipeline that supports the reliable development and validation of the deepfake detection model.

4.2. Model Development

After the data was completed, the CNN model was used as model initialization. The CNN employed in this study is a pretrained ResNet18 model obtained from the PyTorch model library [22], [23], [24], [25]. Several modifications were made to adapt it for feature extraction purposes. The final fully connected layer was removed, enabling the model to function solely as a feature extractor. A custom forward method was implemented, in which the feature extractor produces an output feature map with dimensions of (batch_size, 512, 1, 1), which is subsequently flattened into a vector of shape (batch_size, 512). ResNet18 was selected because it provides a practical balance between representational capacity and computational cost for transfer-learning feature extraction in moderate-size datasets. Moreover, recent applied studies have shown that ResNet18 variants often match or closely approach the performance of larger architectures on domain-specific tasks when paired with appropriate fine-tuning or lightweight attention modules, making it a pragmatic backbone for hybrid CNN–GNN pipelines where the graph component adds additional model complexity.

The GNN model in this study was developed using the torch_geometric library in PyTorch. It consists of two GCNConv layers. The first layer transforms the node features from the input dimension to a specified hidden dimension, while

 ISSN 2723-6471

the second layer projects the features to a single output value suitable for binary classification. To improve generalization, a dropout layer is included after the first convolution.

These components are integrated within a custom forward method. The process begins by passing the input through the first convolutional layer, followed by a ReLU activation function and the dropout layer. The result is then processed through the second convolutional layer. Finally, a global mean pooling operation is applied to aggregate node-level representations into a graph-level output, which serves as the final prediction of the model.

The previously constructed models are integrated into a single architecture referred to as the Hybrid CNN_GNN model. In this design, the CNN component is responsible for extracting feature representations from the input images. These feature maps are flattened when necessary to ensure compatibility with the subsequent processing steps. The extracted features are then used to construct a k-NN graph, with k equals to 2 and the use of Euclidean distance to find the nearest neighbor, where each node represents an image feature vector and edges are defined based on feature similarity using the build_edge_index function. This graph structure is subsequently processed by the GNN, which operates on both the node features and the graph topology to produce the final classification output.

The training phase of the model employed the Adam optimizer with a learning rate set at 1e-4. The binary cross-entropy loss function with logits (BCEWithLogitsLoss) was selected as the learning criterion to guide the optimization process. The model was trained in over a total of 20 epochs consisting of both training and validation loop. During each training iteration, the input images and their corresponding labels were first transferred to the computational device. The hybrid model then generated predictions, which were compared against the ground truth to compute the loss using the specified criterion. Backpropagation was performed to calculate gradients, and these gradients were subsequently used by the optimizer to update the model parameters. After completing all batches in an epoch, the average loss was calculated and recorded to track training progress over time.

As for the evaluation loop, the model was set into evaluation mode and iterated over the validation data loader. Similar with the training iteration, the images and labels were loaded into the device as the first step, and predictions were generated before computing the loss. The difference lies in the next step, as gradients weren't recalculated, skipping the subsequent steps. The validation losses are also saved at the end, providing a reliable measure of the model's generalization performance and enabling comparison with the training loss.

4.3. Model Operations

The result of hybrid CNN-GNN can be seen in figure 3.

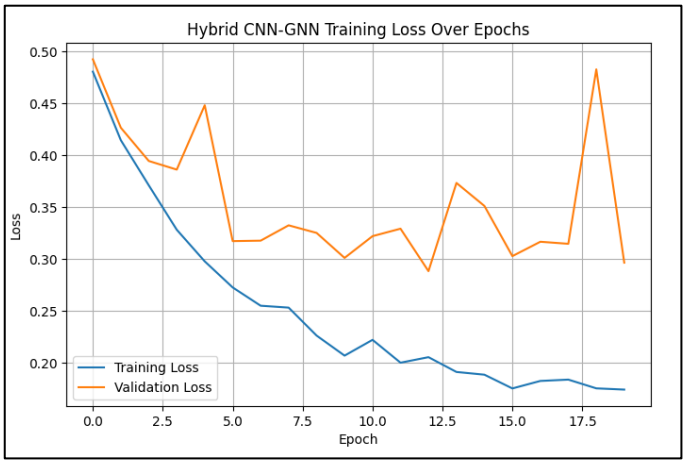

Figure 3. Hybrid CNN-GNN

The line plot illustrates the learning curve for the hybrid CNN-GNN model across 20 training epochs. At the beginning of training, both losses were relatively high, scoring nearly 0.50, which is expected due to the model's randomly initialized parameters. As training advances, the training loss steadily declines, indicating that the model is successfully learning from the data. In contrast, the unstable validation loss suggested that the model was having difficulties in generalizing the data, which can be seen from epoch 4 onwards, where the loss spiked multiple times. By the final

ISSN 2723-6471

epoch, the training loss stabilizes at approximately 0.17, with 0.30 validation loss. The gap between losses indicates overfitting, meaning that the model can learn from the training data well but fails to generalize on the validation dataset and may struggle on predicting unseen samples.

The classification report (table 1) indicates a very good level of performance from the model, with most of the precision, recall, and f1-score scoring within the range of 0.85 to 0.91.

Table 1. Classification Report

	Precision	**Recall**	**F1-score**	**Support**
0	0.91	0.89	0.90	744
1	0.85	0.87	0.86	512
Accuracy			0.88	1256
Macro avg	0.88	0.88	0.88	1256
Weighted avg	0.88	0.88	0.88	1256

The overall accuracy achieved is 88%, which indicates that the model correctly classified a large majority of the samples. When examined per class, class 0 (real images) achieved a precision of 0.91 and a recall of 0.89, resulting in an F1-score of 0.90. For class 1 (generated images), the model achieved a precision of 0.85, a recall of 0.87, and an F1-score of 0.86. These values suggest that the model performs slightly better in class 0, possibly due to a larger sample size in the training data.

The macro average F1-score, which gives equal weight to each class, is 0.88, while the weighted average F1-score, which accounts for class imbalance, is 0.88. These metrics confirm that the model maintains balanced performance across both classes, even though class 0 has more examples than class 1. This result demonstrates that the hybrid model generalizes well to unseen data, successfully capturing both image-based and graph-structured information for binary classification. The confusion matrix shown in the figure 4, provides further insight into the classification performance of the hybrid CNN-GNN model. The matrix indicates that out of 744 actual instances of class 0, the model correctly predicted 664 cases, while misclassifying 80 instances as class 1. Conversely, for the 512 actual instances of class 1, the model accurately predicted 443 samples, with 69 misclassified as class 0.

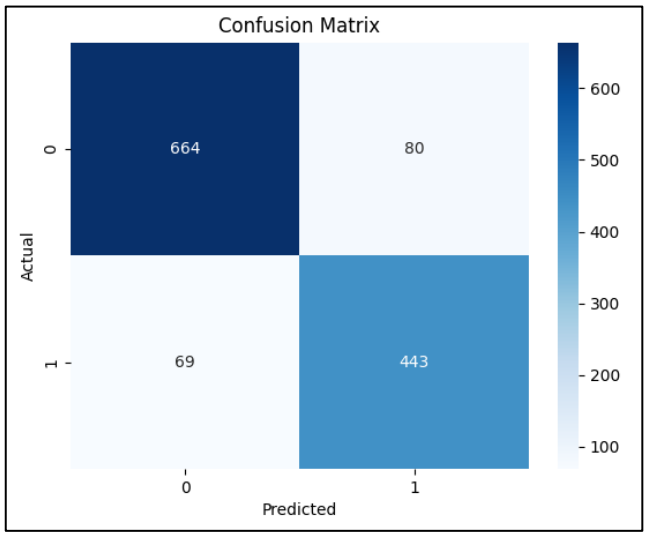

Figure 4. Confusion matrix.

This distribution suggests that the model demonstrates strong predictive ability for both classes, with slightly higher accuracy for class 0. The relatively low number of false positives (80) and false negatives (69) indicates that the model maintains a balanced trade-off between precision and recall. These results align well with the previous classification report and reaffirm the model's reliability in distinguishing between the two classes in a binary classification setting.

 ISSN 2723-6471

4.4. Discussion

The learning curve indicates the overfitting of the model, as shown by the widening gap between its training losses and validation losses over epochs. The divergence started from epoch 4, with the most significant ones in epoch 5, 13, and 19. The gap was being corrected in some epochs, but the final result still shows a huge gap, with validation loss being nearly twice as high as the training loss. This suggests that the model can learn the data well but struggles with unseen data.

Further evidence of the model's strong performance can be seen in the classification report (Figure 10). The model achieves an overall accuracy of 91 percent, with precision, recall, and F1 scores all exceeding 0.88 for both classes. These high scores indicate that the model is consistently able to distinguish between real and generated images with a high degree of reliability. The macro average and weighted average values also reinforce the balance in performance across classes, suggesting that the model does not disproportionately favor one class over the other.

The confusion matrix (figure 4) provides a more granular view of this performance. Most predictions align with the true labels, with 1,359 out of 1,487 real images and 916 out of 1,025 generated images correctly classified. Only a small proportion of instances are misclassified, 128 false positives and 109 false negatives, amounting to roughly 9 percent of the total predictions. This relatively low error rate is consistent with the high overall accuracy reported in the classification metrics.

This study demonstrates strong classification performance, achieving an overall accuracy of 91 percent, with precision, recall, and F1-score values averaging between 0.90 and 0.91. The proposed hybrid CNN-GNN model effectively distinguishes between real and generated images with consistent performance across evaluation metrics. However, despite these positive results, the model overfitting constitutes a critical methodological drawback. Without hyperparameter tuning or early stopping, the possibility of the generated model being overfit stays high. This limitation raises the possibility that the model may struggle to predict unseen data. To mitigate this, future experiments should adopt a proper hyperparameter tuning strategy such as grid search or random search, employ early stopping based on validation loss. In addition, the CASIA v2 dataset itself presents constraints: it includes limited image types and relatively low-resolution samples, and the manipulations it contains reflect outdated generation methods rather than the most recent deepfake techniques. These factors may restrict the ecological validity of the results, as modern synthetic image generators produce higher-quality artifacts that differ from those in CASIA v2. Future work should therefore test the model on more diverse and up-to-date datasets to evaluate robustness under realistic conditions.

Several additional analyses would strengthen the claims and clarify the model’s contribution. An ablation study that compares the convolutional backbone alone, the graph component alone, and the integrated model would quantify the marginal benefit of relational learning. An interpretability analysis using techniques such as gradient based saliency for the convolutional branch and GNN explainers for the graph branch would reveal which features and relations drive decisions and whether the model relies on semantically meaningful cues or spurious artifacts. Robustness assessment is also necessary; experiments that apply common image perturbations, compression, resizing or adversarial perturbations would evaluate practical reliability. Finally, benchmarking on external and standardized datasets and reporting computational metrics such as parameter count and inference latency would support claims about applicability and deployment.

Compared to the study by El-Gayar et al. [16], which applies to a GNN-based framework for detecting deep-fake videos and achieves 87.6 percent accuracy on the DFDC dataset, this research offers higher classification accuracy, albeit within a simpler image-based setting. While El-Gayar et al.'s [16], approach benefits from modeling spatial-temporal relationships inherent in videos, it also introduces significant computational complexity. In contrast, the current study focuses solely on static images, enabling a lighter and more accessible model but without capturing temporal dynamics that may be critical in real-world deepfake scenarios.

The work by Khormali et al. [18] explores multimodal graph learning by integrating visual and audio signals, reporting an F1-score of 93.4 percent. Their multimodal approach results in higher predictive performance, but at the cost of requiring multi-source data and a complex fusion strategy. This study, despite relying exclusively on visual input, still achieves competitive performance. However, a notable limitation is the model’s inability to process non-visual cues, which could be relevant in applications involving multimodal deepfake detection.

Pintelas [17] propose a hybrid CNN-GNN model for soybean disease classification, reaching 95.7 percent accurately. Their task domain likely exhibits more localized and structured visual patterns, making classification more direct. In contrast, distinguishing between generated and real images typically involves subtle and unstructured differences, making the classification task more challenging. While this study reports slightly lower accuracy, the complexity of the classification task may justify the performance gap. Additionally, unlike Pintelas, this work does not yet incorporate interpretability mechanisms, which would be useful for understanding model decisions.

The research by Soudy et al. [19] presents CNN2GNN, a framework that bridges CNN and GNN components and reports an F1-score improvement of 1.2 to 3.6 percent over standard CNNs on datasets such as CIFAR-10. This aligns with the current findings, where the hybrid model also produces an F1-score of 0.91, suggesting that the CNN-GNN integration contributes positively to classification performance. Nevertheless, Soudy et al. evaluate their model on multiple standardized datasets, while the present study is limited to a single image domain, which may constrain the generalizability of the reported results.

Lastly, El-Gayar et al. [16] introduce a lightweight hybrid CNN-GNN architecture for scene classification, achieving 90.3 percent accuracy. Their work emphasizes computational efficiency and fast inference, which is not addressed in the current research. While this study surpasses their reported accuracy, it does not include analysis of computational performance or deployment considerations, which represent another important direction for future research.

5. Conclusion

This study demonstrates that the proposed hybrid CNN-GNN model provides a reliable framework for distinguishing between real and artificially generated images. The experimental results, with accuracy, precision, and recall consistently around 0.88, confirm that integrating convolutional feature extraction with graph-based relational learning enhances classification performance compared to single-modality approaches. These findings contribute to the growing body of work on synthetic image detection by showing that spatial and relational features can be jointly leveraged to improve model robustness and predictive reliability. Nevertheless, the evaluation also revealed signs of overfitting, indicating limited generalization when applied to unseen data. This limitation underscores the need for further optimization through hyperparameter tuning, the incorporation of validation sets, and the application of early stopping strategies. Future research should also test the model on larger and more diverse datasets to ensure scalability and broader applicability. Overall, this work provides both methodological and practical contributions, offering evidence that hybrid deep learning architectures represent a promising direction for advancing the authenticity verification of digital images.

6. Declarations

6.1. Author Contributions

Conceptualization: W., S., R.I.D.; Methodology: S.; Software: W.; Validation: W., S., and R.I.D.; Formal Analysis: W., S., and R.I.D.; Investigation: W.; Resources: W.; Data Curation: W.; Writing Original Draft Preparation: W.; Writing Review and Editing: S., W., and R.I.D.; Visualization: W.; All authors have read and agreed to the published version of the manuscript.

6.2. Data Availability Statement

The data presented in this study are available on request from the corresponding author.

6.3. Funding

The authors received no financial support for the research, authorship, and/or publication of this article.

6.4. Institutional Review Board Statement

Not applicable.

6.5. Informed Consent Statement

Not applicable.

6.6. Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] A. Al-Adwan, H. Alazzam, N. Al-Anbaki, and E. Alduweib, "Detection of Deepfake Media Using a Hybrid CNN–RNN Model and Particle Swarm Optimization (PSO) Algorithm," *Computers* 2024, Vol. 13, Page 99, vol. 13, no. 4, pp. 99-113, Apr. 2024, doi: 10.3390/COMPUTERS13040099.

[2] R. Raman et al., "Fake news research trends, linkages to generative artificial intelligence and sustainable development goals," *Heliyo*n, vol. 10, no. 3, pp. 1-12, Feb. 2024, doi: 10.1016/j.heliyon.2024.e24727.

[3] S. G. Ji, "#MeToo in an AI-generated deepfake sexual violence era in South Korea," *Womens Stud Int Forum*, vol. 112, no. pp. 1-12, Sep. 2025, doi: 10.1016/j.wsif.2025.103146.

[4] A. Heidari, N. Jafari Navimipour, H. Dag, and M. Unal, "Deepfake detection using deep learning methods: A systematic and comprehensive review," *Wiley Interdiscip Rev Data Min Knowl Discov*, vol. 14, no. 2, pp. 1-20, Mar. 2024, doi: 10.1002/WIDM.1520.

[5] T. Saheb, M. Sidaoui, and B. Schmarzo, "Convergence of artificial intelligence with social media: A bibliometric and qualitative analysis," *Telematics and Informatics Reports,* vol. 14, no. 6. Pp. 1-12, 2024, doi: 10.1016/j.teler.2024.100146.

[6] C. Yavuz, "Adverse human rights impacts of dissemination of nonconsensual sexual deepfakes in the framework of European Convention on Human Rights: A victim-centered perspective," *Computer Law and Security Review*, vol. 56, no. 4, pp. 1-12, 2025, doi: 10.1016/j.clsr.2025.106108.

[7] A. Diel, T. Lalgi, I. C. Schröter, K. F. MacDorman, M. Teufel, and A. Bäuerle, "Human performance in detecting deepfakes: A systematic review and meta-analysis of 56 papers," *Computers in Human Behavior Reports*, vol. 16, no. 12, pp. 1-12, 2024, doi: 10.1016/j.chbr.2024.100538.

[8] G. Chen, C. Du, Y. Yu, H. Hu, H. Duan, and H. Zhu, "A Deepfake Image Detection Method Based on a Multi-Graph Attention Network," *Electronics,* 2025, vol. 14, no. 3, pp. 482-457, Jan. 2025, doi: 10.3390/ELECTRONICS14030482.

[9] Z. Xu, X. Wen, G. Zhong, and Q. Fang, "Public perception towards deepfake through topic modelling and sentiment analysis of social media data," *Soc Netw Anal Min,* vol. 15, no. 1, pp. 1–20, Dec. 2025, doi: 10.1007/S13278-025-01445-8/FIGURES/10.

[10] N. Saeed, G. Mumtaz, M. Yaqub, and M. H. Ahmad, "Improving DeepFake Detection: A Comprehensive Review of Adversarial Robustness, Real-Time Processing and Evaluation Metrics | Journal of Computing and Biomedical Informatics," *Journal of Computing and Biomedical Informatics*, vol. 7, no. 2, p.1-12, 2024.

[11] F. Khalid, A. Javed, Q. ul ain, H. Ilyas, and A. Irtaza, "DFGNN: An interpretable and generalized graph neural network for deepfakes detection," *Expert Syst Appl*, vol. 222, no. 7, pp. 1-23, 2023, doi: 10.1016/J.ESWA.2023.119843.

[12] F. A. Twince Tobing, A. Kusnadi, I. Z. Pane, and R. Winantyo, "Deepfake detection using convolutional neural networks: a deep learning approach for digital security," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 39, no. 2, pp. 1092-1108, Aug. 2025, doi: 10.11591/ijeecs.v39.i2.pp1092-1099.

[13] S. Pomalingo, Irmawati, Suryasari, and M. D. Kusnadi, "Optimizing CNN Hyperparameters for Copy-Move Tampered Images Detection," *Proceedings: ICMERALDA 2023 - International Conference on Modeling and E-Information Research, Artificial Learning and Digital Applications*, vol 1, no. 1, pp. 236–241, 2023, doi: 10.1109/ICMERALDA60125.2023.10458150.

[14] N. I. R. Prasetya and Irmawati, "Detection of Image Splicing and Copy-Move Forgery Using the Prewitt Operator and CNN Approach," I*CSINTESA 2024 - 2024 4th International Conference of Science and Information Technology in Smart Administration: The Collaboration of Smart Technology and Good Governance for Sustainable Development Goals,* vol. 4, no. 1, pp. 475–480, 2024, doi: 10.1109/ICSINTESA62455.2024.10747914.

[15] A. Kusnadi, I. Z. Pane, and F. A. T. Tobing, "Enhancing facial recognition accuracy through feature extractions and artificial neural networks," *IAES International Journal of Artificial Intelligence (IJ-AI),* vol. 14, no. 2, pp. 1056-1067, Apr. 2025, doi: 10.11591/IJAI.V14.I2.PP1056-1066.

[16] M. M. El-Gayar, M. Abouhawwash, S. S. Askar, and S. Sweidan, "A novel approach for detecting deep fake videos using graph neural network," *J Big Data,* vol. 11, no. 1, Dec. pp.1-12, 2024, doi: 10.1186/s40537-024-00884-y.

[17] E. Pintelas and I. E. Livieris, "Convolutional neural network framework for deepfake detection: A diffusion-based approach," *Computer Vision and Image Understanding*, vol. 257, no. 6, pp. 1-15, 2025, doi: 10.1016/J.CVIU.2025.104375.

[18] A. Khormali and J. S. Yuan, "Self-Supervised Graph Transformer for Deepfake Detection," *IEEE Access*, vol. 12, no. 7, pp. 58114–58127, 2023, doi: 10.1109/ACCESS.2024.3392512.

[19] A. H. Soudy, "Deepfake detection using convolutional vision transformers and convolutional neural networks," *Neural Comput Appl,* vol. 36, no. 31, pp. 19759–19775, Nov. 2024, doi: 10.1007/S00521-024-10181-7/TABLES/6.

[20] N. A. Chandra, R. Murtfeldt, L. Qiu, A. Karmakar, H. Lee, E. Tanumihardja, K. Farhat, B. Caffee, S. Paik, C. Lee, J. Choi, A. Kim, and O. Etzioni, "Deepfake-Eval-2024: A Multi-Modal In-the-Wild Benchmark of Deepfakes Circulated in 2024," arXiv preprint arXiv:2503.02857, vol. 2025, no. Mar., pp. 1-19, 2025.

[21] dk_9892, "CASIA_v2." Accessed: Jul. 23, 2025. [Online]. Available: https://www.kaggle.com/datasets/dk9892/casia-v

[22] S. Alanazi and S. Asif, "Exploring deepfake technology: creation, consequences and countermeasures," *Human-Intelligent Systems Integration* 2024 6:1, vol. 6, no. 1, pp. 49–60, Sep. 2024, doi: 10.1007/S42454-024-00054-8.

[23] M. Alruwaili and M. Mohamed, "An Integrated Deep Learning Model with EfficientNet and ResNet for Accurate Multi-Class Skin Disease Classification," *Diagnostics*, vol. 15, no. 5, pp. 551-564, Mar. 2025, doi: 10.3390/DIAGNOSTICS15050551

[24] C. L. Lin and K. C. Wu, "Development of revised ResNet-50 for diabetic retinopathy detection," *BMC Bioinformatics*, vol. 24, no. 1, pp. 1–18, Dec. 2023, doi: 10.1186/S12859-023-05293-1/FIGURES/13.

[25] S. B. Francis and J. Prakash Verma, "Deep CNN ResNet-18 based model with attention and transfer learning for Alzheimer's disease detection," *Front Neuroinform*, vol. 18, no. 1, pp. 1-17, 2025, doi: 10.3389/FNINF.2024.1507217.