Building of Informatics, Technology and Science (BITS)
Volume 8, No 1, June 2026 Page: 677−686
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v8i1.9555

Comparative Evaluation of CNN, Random Forest, and SVM for Deepfake Image Classification

Dyo Arya Purnama*, Idris Sonny, Ongki Juliandra, Dhafin Rizky Trisandy, Ken Ditha Tania, Alsella Meiriza

Faculty of Computer Science, Information Systems Study Program, Universitas Sriwijaya, Palembang, Indonesia
Email: [1,*]dyoaryapurnama12@email.com, [2] idrissonny1502@email.com, [3]ongkijuliandra2@gmail.com,
[4]dhafinrizkyt@gmail.com, [5]kenya_tania@gmail.com, [6]allsela_meiriza@yahoo.co.id
Corresponding Author Email: dyoaryapurnama12@email.com
Submitted: **25/03/2026**; Accepted: **30/06/2026**; Published**: 30/06/2026**

Abstract−Deepfake image detection remains a challenging image classification problem because the effectiveness of a detection model can vary according to the classification approach and image representation used. This study comparatively evaluates three classification algorithms, namely Convolutional Neural Network (CNN), Random Forest, and Support Vector Machine (SVM), for distinguishing real facial images from deepfake images. The dataset consists of two classes, real and deepfake images, which undergo preprocessing through resizing to 32 × 32 pixels and pixel normalization from 0–255 to 0–1. The preprocessed dataset is divided into 80% training data and 20% testing data. CNN performs feature learning through convolutional layers, whereas Random Forest and SVM perform classification based on the prepared image feature representation. Model performance is evaluated using accuracy, precision, recall, and F1-score on the testing data. The experimental results show that CNN achieves the highest performance, with 91% accuracy, 93% precision, 88% recall, and 91% F1-score, followed by SVM with 77% accuracy and Random Forest with 75% accuracy. Thus, under the experimental conditions of this study, CNN provides better classification performance than the two conventional machine-learning algorithms. However, the result is limited to the dataset, 32 × 32 pixel input representation, and experimental configuration used in this study; therefore, the 91% accuracy should not be interpreted as evidence of general superiority across other datasets or deepfake generation techniques. These findings provide an empirical comparison of the three approaches and highlight the need for cross-dataset and higher-resolution evaluation in subsequent research.

Keywords: Deepfakes; Machine Learning; Convolutional Neural Networks; Random Forest; Support Vector Machines; Image Classification

1. INTRODUCTION

The rapid development of Artificial Intelligence (AI), particularly deep learning and generative AI, has substantially transformed image processing and digital multimedia. One of the most prominent applications is deepfake technology, which uses deep learning models to generate or manipulate visual and audio content with characteristics that closely resemble authentic media. Deepfake techniques can produce manipulated images, videos, and audio that are increasingly difficult to distinguish from genuine content through human observation alone. Although this technology has legitimate applications in entertainment, media production, and digital content creation, its misuse creates significant risks, including misinformation, public opinion manipulation, identity theft, and social engineering attacks [1], [2], [3]. The increasing accessibility of AI-based generative technologies has further reduced the technical barriers to producing manipulated content, expanding the potential forms and scale of misuse in information security contexts [4].

The increasing realism and complexity of deepfake manipulation make reliable automatic detection technically necessary. Modern generative models can introduce subtle modifications to facial appearance, texture, expression, lighting, and other visual characteristics that may not be consistently identifiable through manual inspection. Consequently, distinguishing authentic from manipulated media has become a challenging problem in cybersecurity and digital forensics [5], [6]. The challenge is further complicated by the diversity of manipulated media, including differences in identities, facial characteristics, video quality, compression, and manipulation techniques. Therefore, deepfake detection systems need to learn discriminative features that remain reliable across variations in the input data rather than relying only on obvious visual artifacts. The availability of datasets such as Celeb-DF, FaceForensics, and other deepfake datasets has supported the development of detection methods; however, differences in dataset characteristics and manipulation distributions also create challenges in evaluating whether a model can generalize beyond the specific data used during training [6]. Furthermore, the continuous development of AI-generated content introduces new manipulation patterns that require detection approaches to remain adaptable [7].

Deepfake detection research has consequently shifted from conventional machine-learning approaches toward deep learning methods that can automatically extract complex representations from multimedia data. Early approaches commonly relied on manually designed features followed by conventional machine-learning classifiers, whereas more recent studies have increasingly employed architectures such as Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and other deep neural networks [8], [9]. CNN-based approaches are particularly relevant for spatial analysis because they can learn visual patterns associated with manipulation, while recurrent architectures can model temporal information when detecting manipulated video sequences. Previous studies indicate that deep learning can identify manipulation patterns that are difficult to capture using conventional feature-based approaches [5]. However, the increasing complexity of these architectures does not necessarily guarantee consistent performance across datasets. Differences in training data, preprocessing procedures, manipulation types, and evaluation settings can produce substantial variations in detection performance [5], [8], [9]. Thus, comparisons between algorithms under

Building of Informatics, Technology and Science (BITS)
Volume 8, No 1, June 2026 Page: 677−686
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v8i1.9555

a consistent experimental setting remain important for determining whether a more complex model actually provides a meaningful performance advantage.

Several studies have investigated advanced architectures for deepfake detection. Al-Dulaimi and Kurnaz proposed a hybrid CNN-LSTM model that combines spatial and temporal feature extraction to improve deepfake image detection [10]. Al-Adwan developed a CNN-RNN approach optimized using Particle Swarm Optimization (PSO) for manipulative content classification [11]. Other studies have employed architectures such as XceptionNet within digital image forensics frameworks to improve the detection of manipulated images [12]. Dharma et al. investigated EfficientNet and demonstrated the potential of transfer learning for improving deepfake detection performance in digital images [13]. These studies demonstrate the effectiveness of increasingly sophisticated architectures; however, their primary emphasis is on developing or optimizing advanced models rather than establishing a controlled comparison with conventional machine-learning approaches. Consequently, the relative performance advantage of a basic CNN over conventional classifiers under the same experimental conditions remains insufficiently addressed in the studies reviewed.

Research combining different neural network architectures has also been conducted. Afandi and Purnama employed a CNN-GRU model for deepfake video detection using the Celeb-DF(v2) dataset, demonstrating the potential benefit of integrating spatial and temporal representations [14]. Kavitha et al. proposed an ensemble learning approach that combines multiple models to improve robustness against variations in visual manipulation [15]. Meanwhile, Varun Kumar et al. investigated machine-learning-based approaches for detecting manipulation in multimedia data [16]. Although these studies provide valuable evidence regarding the performance of individual approaches, they also reveal an important methodological issue: comparisons across studies are difficult because the datasets, preprocessing procedures, feature representations, and evaluation settings are not necessarily consistent. In addition, studies focusing on sophisticated architectures such as CNN-LSTM, CNN-RNN, XceptionNet, EfficientNet, and ensemble models do not directly establish how much performance improvement is obtained when these more complex approaches are compared with simpler machine-learning baselines under equivalent conditions [10]–[16].

This gap is particularly relevant because deepfake detection involves a trade-off between predictive performance and computational complexity. Advanced hybrid and ensemble architectures may provide richer feature representations, but they generally require more parameters, training resources, and computational time than conventional machine-learning methods. Conversely, conventional machine-learning algorithms may be computationally simpler but depend more strongly on the quality of the extracted features. Therefore, a direct algorithmic comparison using the same dataset, preprocessing pipeline, feature representation, and evaluation metrics can provide a more controlled assessment of the relative strengths and limitations of different approaches. Rather than attempting to propose another state-of-the-art architecture, such a comparison can establish an empirical baseline for determining whether the additional complexity of deep learning provides a meaningful improvement over conventional machine-learning methods for the specific deepfake detection setting investigated in this study.

Based on this research gap, this study focuses on comparing the performance of a CNN-based deep learning approach with two conventional machine-learning algorithms for deepfake detection. The comparison is conducted under a consistent experimental setting to evaluate differences in classification performance and to examine the practical trade-off between detection capability and model complexity. This approach complements previous studies that predominantly emphasize hybrid, optimized, or state-of-the-art deep learning architectures [10]–[15] by providing a controlled comparison between a basic deep learning model and conventional machine-learning methods. The findings are expected to provide an empirical basis for understanding the relative suitability of the compared algorithms for deepfake detection and to clarify whether the use of a more complex deep learning approach offers a measurable advantage within the dataset and experimental conditions employed in this study [17], [18].

2. RESEARCH METHODOLOGY

2.1 Research Stages

This study aims to conduct a comparative analysis of the performance of several machine learning algorithms in classifying deepfake images in the context of information security. Deepfake technology is a form of Artificial Intelligence -based media manipulation that is capable of producing synthetic images with a high degree of similarity to the original image, making it difficult for humans to visually distinguish them [5], [6].

The development of this technology has given rise to various threats in the field of information security, such as the spread of disinformation, digital identity manipulation, and social engineering attacks that utilize fake visual media [1], [2]. In recent years, improvements in the quality of deep learning-based generative models have made deepfake technology easier to use and potentially misused in various sectors, including politics, social media, and cybersecurity [7], [9]

Various previous studies have shown that machine learning and deep learning- based methods are quite effective in detecting digital image manipulation. Deepfake detection techniques typically utilize visual pattern analysis in images to identify characteristic differences between original and manipulated images [8], [11].

Some algorithms frequently used in image classification are Random Forest, Support Vector Machine (SVM), and Convolutional Neural Network (CNN) because they are able to recognize complex feature patterns in visual data

Building of Informatics, Technology and Science (BITS)
Volume 8, No 1, June 2026 Page: 677−686
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v8i1.9555

[16], [14], [19]. Deep learning- based methods such as CNN are even capable of automatically extracting features from images, resulting in more accurate classification performance in many deepfake detection cases, [12], [13].

Based on this, this study conducted a comparative analysis of three classification algorithms: Random Forest, Support Vector Machine, and Convolutional Neural Network to determine which method performed best in detecting deepfake images. The research stages in this study are shown in Figure 1.

Figure 1 presents the systematic workflow of this study, beginning with dataset collection, followed by image preprocessing, dataset distribution, feature extraction, classification using Random Forest, SVM, and CNN, and finally model performance evaluation. Each stage is conducted sequentially to ensure that the three classification algorithms are evaluated using the same experimental data and preprocessing conditions.

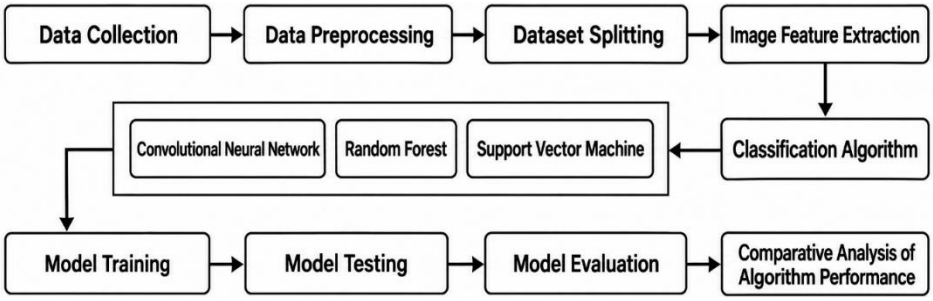


Figure 1. Research Stages Flow

2.2 Dataset Collection

The dataset used in this study consists of facial images categorized into two classes: real images and manipulated images (deepfake images). This dataset serves as the primary data source for training and testing the classification model. The dataset used in this study was obtained from a public deepfake image dataset repository. The dataset consists of facial images divided into two classes, namely real images and deepfake images. The dataset contains a total of 2,000 images, consisting of 1,000 real images and 1,000 deepfake images, resulting in a balanced class distribution of 50% real images and 50% deepfake images.

The dataset contains facial images with variations in facial appearance, pose, expression, background, and image quality. These variations provide different visual characteristics that can be learned by the classification models. The use of both real and manipulated facial images also enables the models to learn differences in visual patterns associated with deepfake manipulation. In the training stage, the model will learn the visual characteristics of both image classes so that it can automatically distinguish between original and manipulated images. Deepfake datasets are generally obtained from a collection of images or videos that have been manipulated using deep learning-based generative techniques such as Generative Adversarial Network (GAN) [6], [7].

Dataset quality plays a very important role in improving the model's ability to recognize visual manipulation patterns in digital images[5], [8]. A good dataset generally has a large amount of data, a variety of faces, and a balanced class distribution between original and deepfake images. With a representative dataset, the model can learn feature patterns more accurately, thus optimizing classification performance when tested on new data. Furthermore, dataset diversity can also improve the model's generalization ability when faced with various types of image manipulation [9], [20].

2.3 Data Preprocessing

Preprocessing stage is carried out to improve the quality of the data before it is used in the model training process. This process aims to ensure that the dataset has a uniform format and is ready to be used by the machine learning algorithm [8], [11]. This process includes several steps such as:

a. Image Resizing
   Image resizing is the process of changing the size of an image to a specific dimension so that all data has the same size and can be processed by the model. This process is especially important in deep learning methods like CNNs, which require consistent image size input. In this study, all facial images are resized to 32 × 32 pixels. The same image dimension is applied to all classification algorithms to maintain consistent input data and ensure that the comparison is conducted under the same image representation. The 32 × 32 pixel dimension also reduces the number of input features and computational requirements during the training process.
b. Pixel Value
   Normalization Pixel value normalization is the process of changing the range of image pixel values from the initial scale, for example 0–255, to a smaller range such as 0–1. Normalization aims to increase the stability of the model training process and accelerate the convergence process during training [16]. In this study, pixel values are normalized from the original range of 0–255 to a range of 0–1 by dividing each pixel value by 255. This normalization method produces a consistent numerical range for all input images and is particularly useful for stabilizing the numerical processing and training of the CNN model.

Building of Informatics, Technology and Science (BITS)
Volume 8, No 1, June 2026 Page: 677−686
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v8i1.9555

c. Cleaning Irrelevant Data
Data cleaning is carried out to remove corrupted, duplicate, or inappropriate data for research purposes. This process is important to ensure that the dataset used is of high quality and free from noise that could affect model performance [7]. The cleaning process includes identifying corrupted images, duplicate images, images that cannot be read properly, and images that are not relevant to the two classification categories. Images identified as unsuitable are removed before the dataset is divided into training and testing data. This step is intended to prevent irrelevant or defective samples from introducing noise into the classification process.

Preprocessing process aims to improve the quality of the dataset so that the model can learn image patterns more effectively and produce better classification performance [11], [14]. Figure 2 shows the preprocessing sequence applied to the image data, beginning with the original facial image, followed by resizing to 32 × 32 pixels, pixel-value normalization from 0–255 to 0–1, and cleaning of unsuitable data. This process produces standardized image data that can subsequently be used by the three classification algorithms.

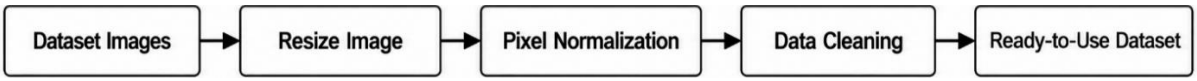


Figure 2. Image Preprocessing Process

2.4 Dataset Distribution

The dataset that has gone through the preprocessing process is then divided into two parts, namely:

a. Training Data:
Training data is the data used to train a machine learning model to learn patterns from the dataset. At this stage, the algorithm analyzes the data's features and labels (e.g., real and fake) so the model can develop classification rules or patterns. The better the quality and quantity of training data, the better the model's ability to recognize patterns. In this study, 80% of the dataset is used as training data, while the remaining 20% is used as testing data. The training data is used exclusively for learning the classification patterns, while the testing data is retained for evaluating the model after training.

b. Testing Data:
Testing data is data used to test model performance after the training process is complete. This data is not used in the training process, so it provides an objective picture of how well the model predicts new data. The results of testing data are typically used to calculate the accuracy, precision, recall, and F1-score of the model. The 20% testing data is not included in the model training process. This separation ensures that the final evaluation is performed on data that was not directly used to learn the classification patterns. Dataset partitioning is an important step in the process of developing a machine learning model because it can be used to measure the model's generalization ability to data that has never been studied before [19], [20].

The same dataset distribution is applied to Random Forest, SVM, and CNN so that the performance comparison is conducted using identical training and testing conditions.

2.5 Feature Extraction

Feature extraction is the process of obtaining important information from an image that can be used by a classification algorithm. The extracted features can be texture, edge patterns, facial structures, or other visual characteristics that can distinguish between the original image and the manipulated image, [13], [14]. Deep learning methods like Convolutional Neural Networks, the feature extraction process is performed automatically through convolution layers. These layers are capable of detecting various visual patterns in images, such as lines, textures, and object structures, in a hierarchical manner. With this capability, the CNN model can learn complex visual characteristics so that it is able to distinguish between images belonging to different classes with a high level of accuracy, [12], [17]. For Random Forest and SVM, the image data is converted into numerical feature vectors before classification. Since the images have been resized to 32 × 32 pixels, the pixel values are flattened into a one-dimensional representation before being provided to the conventional machine-learning algorithms. Thus, Random Forest and SVM use the same standardized pixel representation as input, while CNN performs feature extraction automatically through its convolutional layers. This distinction means that Random Forest and SVM rely on the predefined pixel representation, whereas CNN simultaneously learns the relevant visual features and performs classification. Therefore, the comparison examines the performance of conventional classifiers using standardized image features against a deep learning classifier capable of automatically learning hierarchical visual representations.

2.6 Random Forest Classification

Random Forest is an ensemble learning- based classification algorithm that combines a number of decision trees to improve prediction performance. Each decision tree is built using a different subset of data through a bootstrap sampling technique [16]. The final classification results are obtained through a majority voting mechanism from all the decision trees formed. This method has the advantage of reducing the risk of overfitting and is able to handle dataset with a large number of features [4]. In this study, Random Forest is configured using multiple decision trees to obtain the final classification through majority voting. The number of trees is selected to provide a sufficiently

Building of Informatics, Technology and Science (BITS)
Volume 8, No 1, June 2026 Page: 677−686
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v8i1.9555

diverse ensemble while maintaining reasonable computational requirements. The same training and testing data are used as those applied to the SVM and CNN models. Due to its capabilities, Random Forest is often used in various data classification studies including digital image analysis and AI-based media manipulation detection.

2.7 Support Vector Machine (SVM)

Support Vector Machine (SVM) is a classification algorithm that works by finding the optimal hyperplane to separate data into two different classes. This algorithm attempts to determine the dividing boundary that has the maximum margin between two groups of data to produce a stable classification model [3]. SVM has good capabilities in handling datasets with high feature dimensions so it is widely used in various machine learning studies. In addition, the use of kernel functions in SVM allows data to be mapped into a higher dimensional space so that separation between classes can be carried out more optimally, especially when the data cannot be separated linearly [16].

In this study, the SVM classifier uses a radial basis function (RBF) kernel. The RBF kernel is used because it enables the SVM to construct a nonlinear decision boundary by mapping the input features into a higher-dimensional feature space. This characteristic is relevant to image classification because the relationship between pixel-level visual features and the real/deepfake classes may not be linearly separable. The use of the RBF kernel is kept consistent throughout the experiment so that the SVM performance can be evaluated under a clearly defined kernel configuration.

2.8 Convolutional Neural Networks (CNN)

Convolutional Neural Network (CNN) is a deep learning method widely used in digital image analysis. CNN is designed to automatically extract visual features through several layers of data processing [12]. The CNN architecture consists of several main layers, namely:

a. Convolution layer is the main layer in a CNN, responsible for extracting features from an image. This layer works by using filters or kernels that move across the image to detect important patterns such as edges, texture, and shape.
b. Pooling Layers: Pooling layers reduce the dimensionality of the features generated by the convolution layer. This process helps reduce computational complexity and preserves important image information, typically using methods such as max pooling or average pooling.
c. Fully Connected Layer: The fully connected layer is the final layer in CNN which functions to carry out the classification process. At this stage, all previously extracted features will be processed to determine the output class, whether the image belongs to the real image or deepfake image category, [17], [18].

In this study, the CNN architecture consists of two convolutional blocks followed by a fully connected classification layer. Each convolutional block consists of a convolution layer, an activation function, and a max-pooling layer. The convolutional layers use 3 × 3 filters to extract local visual patterns, while the pooling layers reduce the spatial dimensions of the resulting feature maps. The first convolutional layer uses 32 filters, while the second convolutional layer uses 64 filters. The feature maps generated by the second convolutional block are flattened before being passed to the fully connected layer. The final output layer uses a sigmoid activation function to produce the probability of the two-class classification task, namely real and deepfake. The CNN model uses the ReLU activation function in the convolutional layers because it introduces nonlinearity while maintaining computational simplicity. The model is trained using the Adam optimizer with a binary cross-entropy loss function, which is appropriate for binary classification. The training process is performed using the same training dataset used for the Random Forest and SVM models, while the testing dataset is retained for final performance evaluation.

Figure 3 represents the CNN architecture specifically implemented in this study rather than a general conceptual representation of CNN. The architecture begins with a 32 × 32 pixel input image, followed by two convolutional layers with 32 and 64 filters, respectively, max-pooling operations, flattening, a fully connected classification stage, and a final sigmoid output for real and deepfake classification.

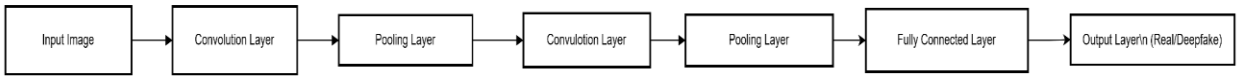


Figure 3. CNN Architecture for Image Classification

2.9 Model Performance Evaluation

To evaluate the performance of the classification model, this study uses several evaluation metrics commonly used in machine learning and deepfake detection research, [5], [11].

a. Accuracy
Accuracy is a measure of the percentage of correct predictions compared to the total data set. The accuracy value provides a general overview of how well a model classifies deepfake images and authentic images.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

b. Precision
Precision is used to measure the model's accuracy in predicting detected images as deepfakes. This metric indicates how many deepfake predictions are actually deepfakes.

Building of Informatics, Technology and Science (BITS)
Volume 8, No 1, June 2026 Page: 677−686
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v8i1.9555

$$Precision = \frac{TP}{TP+FP} \quad (2)$$

c. Recall
Recall indicates the model's ability to detect all deepfake images in the dataset. This metric is crucial in information security systems because it relates to the system's ability to optimally detect threats [2].

$$Recall = \frac{TP}{TP+FN} \quad (3)$$

d. F1-Score
The F1-score is the harmonic mean of precision and recall. This metric is used to strike a balance between a model's accuracy and detection ability in deepfake image classification.

$$F1\ Score = 2 \times \frac{Precision \times Recall}{Precision+Recall} \quad (4)$$

e. Confusion Matrix
Model evaluation is also carried out using a confusion matrix to see the distribution of classification results between real and deepfake classes. The confusion matrix provides information on the number of correct and incorrect predictions for each class so that it can be used to calculate accuracy, precision, recall, and F1-score values in more detail, [10], [20].

Table 1. Confusion Matrix

	Deepfake Prediction	Real Prediction
Deepfake	TP	FN
Real	FP	TN

Table 1 presents the four possible classification outcomes used to evaluate the models. The rows represent the actual class, while the columns represent the predicted class. The matrix therefore provides a direct representation of the correct and incorrect predictions produced by each classification algorithm. As shown in Table 1, TP represents deepfake images correctly predicted as deepfake, TN represents real images correctly predicted as real, FP represents real images incorrectly predicted as deepfake, and FN represents deepfake images incorrectly predicted as real. These four outcomes form the basis for calculating accuracy, precision, recall, and F1-score.
TP (True Positive) : the number of deepfake images that were correctly classified as deepfakes.
TN (True Negative) : the number of original images that were correctly classified as real.
FP (False Positive) : the number of original images that are incorrectly classified as deepfakes
FN (False Negative) : the number of deepfake images that are incorrectly classified as real.

The evaluation values of each algorithm are then compared to determine which method has the best performance in detecting deepfake images. This analysis is expected to contribute to the development of information security systems based on digital image manipulation detection [4], [7], [13]. The comparison is performed using the same evaluation metrics and testing data for Random Forest, SVM, and CNN. The results are interpreted with reference to Table 1 to identify both the overall classification performance and the specific types of errors generated by each algorithm.

3. RESULTS AND DISCUSSION

This section presents the implementation results of the methods used in the research and a discussion of the experimental results obtained. The results presented in this section focus on the performance obtained by the three classification algorithms under the experimental conditions described in the methodology. This study aims to compare the performance of several machine learning algorithms in detecting deepfake images in the context of information security. The comparison focuses on the classification performance of Convolutional Neural Network (CNN), Random Forest, and Support Vector Machine (SVM) using the same dataset and evaluation procedure.

This study used three different classification algorithms: Convolutional Neural Network (CNN), Random Forest, and Support Vector Machine (SVM). These three algorithms were chosen because they have different characteristics in the data learning process and have been widely used in image classification research. CNN performs feature learning and classification through its convolutional architecture, whereas Random Forest and SVM perform classification based on the feature representation provided as input.

This difference provides the basis for comparing a deep learning approach with conventional machine-learning approaches in the same deepfake image classification task. The research process is carried out through several main stages, namely dataset collection, data pre-processing, implementation of classification algorithms, and evaluation of model performance using several evaluation metrics such as accuracy, precision, recall, and F1-score. In addition, this study also uses a confusion matrix to provide a more detailed analysis of the classification results produced by each model.

Building of Informatics, Technology and Science (BITS)
Volume 8, No 1, June 2026 Page: 677−686
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v8i1.9555

3.1 Experimental Results

The test results for the three algorithms are shown in Table 2.

Table 2. Comparison of Algorithm Performance

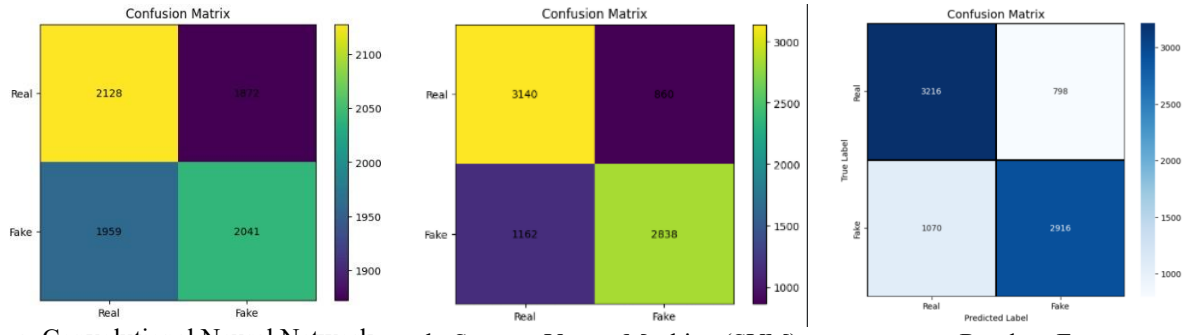
Algorithm	Accuracy	Precision	Recall	F1-Score
CNN	91%	93%	88%	91%
Random Forest	75%	77%	71%	74%
SVM	77%	79%	73%	76%

Based on experimental results, the CNN algorithm demonstrated superior classification performance compared to Random Forest and SVM. As shown in Table 2, CNN achieved the highest accuracy of 91%, followed by SVM at 77% and Random Forest at 75%. CNN also achieved the highest precision (93%), recall (88%), and F1-score (91%). The difference in accuracy between CNN and SVM is 14 percentage points, while the difference between CNN and Random Forest is 16 percentage points. The F1-score shows a similar pattern, with CNN achieving 91%, compared with 76% for SVM and 74% for Random Forest. These results indicate that CNN produced the strongest overall classification performance among the three approaches under the experimental conditions of this study. Random Forest demonstrated fairly stable performance in image classification, although it still fell short of CNN. Meanwhile, SVM also performed well, but its performance was significantly affected by the feature representation used. The results in Table 2 should be interpreted as a single experimental evaluation. Because repeated-run results, standard deviations, or confidence intervals are not reported in the present experiment, statistical uncertainty cannot be quantified from the available results. Therefore, the performance differences are reported descriptively rather than interpreted as statistically significant differences.

In addition, the current results do not provide sufficient information to establish a quantitative comparison of computation time between the three models. Computation time would require recording training and/or inference time under the same hardware and software conditions.

3.2 Visualization of results

To provide more in-depth analysis results on the classification results, this study also provides an evaluation using a confusion matrix. The confusion matrix is intended to show the number of correct and incorrect predictions generated by a classification model. The results are as follows. To avoid ambiguity with the CNN architecture figure in the methodology section, the confusion-matrix visualization is assigned a separate figure number from the CNN architecture. The three visualizations are labeled (a), (b), and (c), corresponding to CNN, SVM, and Random Forest, respectively.



a. Convolutional Neural Network (CNN)

b. Support Vector Machine (SVM)

c. Random Forest

Figure 3. Visualization of Confusion Matrix Results (left)

Figure 3 presents the confusion-matrix visualization for the three classification algorithms. Panel (a) represents the CNN results, panel (b) represents the SVM results, and panel (c) represents the Random Forest results. Each panel should display the actual numerical values of TP, TN, FP, and FN so that the classification errors of the three models can be directly examined. The confusion matrix provides additional information that cannot be observed from accuracy, precision, recall, and F1-score alone because it distinguishes between correctly classified and incorrectly classified samples for each class. In particular, the false-negative value is important because it represents deepfake images that are incorrectly classified as real images.

Through confusion matrix analysis, the distribution of classification errors can be examined for each model. However, based on the current visualization, a specific qualitative explanation of the visual characteristics of the misclassified images cannot be established without examining the corresponding misclassified image samples. Therefore, the misclassification results are not attributed to specific visual characteristics such as facial distortion, blending artifacts, or lighting inconsistencies without supporting image-level evidence.

Building of Informatics, Technology and Science (BITS)
Volume 8, No 1, June 2026 Page: 677−686
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v8i1.9555

3.3 Discussion

Based on the experimental results, each classification algorithm demonstrates different levels of effectiveness in addressing the problem of deepfake image detection. The Convolutional Neural Network (CNN) achieved the highest accuracy of 91%, outperforming Support Vector Machine (77%) and Random Forest (75%). The results in Table 2 therefore indicate that CNN provided the strongest classification performance among the three algorithms under the experimental conditions used in this study.

This finding is consistent with the general direction of previous deepfake detection research that has increasingly employed deep learning architectures to learn complex visual representations. For example, Al-Dulaimi and Kurnaz proposed a CNN-LSTM approach that combines spatial and temporal feature representations [10], while Al-Adwan employed a CNN-RNN model optimized using Particle Swarm Optimization [11]. XceptionNet has also been investigated for digital image manipulation detection [12], and Dharma et al. reported the use of EfficientNet with transfer learning for deepfake image detection [13]. These studies and the present results point in a similar direction regarding the potential of deep learning for extracting discriminative representations from manipulated visual content.

However, the present study differs from these approaches in its comparative scope. Rather than proposing a hybrid or optimized deep learning architecture, this study compares a CNN with Random Forest and SVM under the same experimental setting. The results in Table 2 show that the CNN obtained a 14-percentage-point accuracy advantage over SVM and a 16-percentage-point advantage over Random Forest. This provides an empirical indication that the deep learning approach used in this experiment performed better than the two conventional classifiers for the selected dataset and input representation.

The result is also relevant to previous work using combinations of CNN and GRU for deepfake detection on Celeb-DF(v2) [14] and ensemble learning approaches for improving robustness to variations in visual manipulation [15]. While those studies investigate more sophisticated combinations of models, the present study demonstrates that even a comparatively direct CNN-based approach can achieve higher classification performance than the conventional machine-learning approaches evaluated here. Nevertheless, the difference in experimental datasets and model configurations means that the numerical results should not be directly interpreted as evidence that the CNN used in this study is superior to all architectures reported in previous research.

From a problem-solving perspective, the CNN result can be attributed to its ability to learn image representations directly through convolutional operations, whereas Random Forest and SVM rely on the feature representation supplied to them. The performance difference therefore reflects not only differences between classifiers but also differences in how the three approaches represent and process visual information.

The SVM achieved an accuracy of 77% and an F1-score of 76%, while Random Forest achieved an accuracy of 75% and an F1-score of 74%, as reported in Table 2. The relatively close performance of these two conventional approaches indicates that neither achieved the same level of classification performance as CNN under the experimental conditions. However, the results do not by themselves establish that SVM or Random Forest are generally unsuitable for deepfake detection because their performance may vary with feature representation, hyperparameter configuration, and dataset characteristics.

In contrast, the Support Vector Machine (SVM) approach reflects a more traditional problem-solving strategy that relies heavily on the quality of input features. SVM attempts to solve the classification problem by finding an optimal hyperplane that separates data into distinct classes. The 77% accuracy obtained by SVM indicates that the selected feature representation contains sufficient information for distinguishing part of the real and deepfake samples, although the resulting performance remains below CNN.

Meanwhile, the Random Forest algorithm employs an ensemble-based problem-solving approach by combining multiple decision trees to improve classification stability. Its accuracy of 75% and F1-score of 74% indicate that the feature representation used in this experiment allowed Random Forest to distinguish between the two classes, but with lower overall performance than both CNN and SVM.

Further analysis using the confusion matrix reveals important insights into the nature of classification errors across all models. However, the available results do not provide sufficient image-level evidence to determine whether the errors were specifically caused by facial distortions, blending artifacts, lighting inconsistencies, or other visual characteristics. Such an interpretation would require qualitative examination of representative false-positive and false-negative images.

Another important limitation concerns cross-dataset generalization. The models in this study were evaluated using the same dataset for training and testing, so the results primarily describe performance within the distribution represented by that dataset. The reported 91% CNN accuracy should therefore not be interpreted as evidence of robustness against unseen datasets with different identities, image quality, manipulation techniques, compression levels, or demographic characteristics. Cross-dataset evaluation using an independent deepfake dataset would be necessary to determine whether the learned representations generalize beyond the dataset used in this study.

This limitation is particularly important because previous studies have used different datasets and architectures, including Celeb-DF(v2) in CNN-GRU-based detection [14]. Differences in dataset composition and manipulation characteristics can influence model performance, making direct comparison across studies difficult. Therefore, the

Building of Informatics, Technology and Science (BITS)
Volume 8, No 1, June 2026 Page: 677−686
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v8i1.9555

results of this study provide evidence of comparative performance within the selected experimental dataset rather than a universal ranking of deepfake detection algorithms.

Overall, the findings of this study demonstrate that within the experimental conditions used in this research, CNN achieved higher classification performance than Random Forest and SVM. CNN's ability to integrate feature extraction and classification into a single, end-to-end process provides a potential advantage in handling the visual patterns represented in the dataset.

Nevertheless, the findings should be interpreted together with the limitations of the experimental design, particularly the use of 32 × 32 pixel input images, the absence of cross-dataset evaluation, and the lack of repeated experimental runs or statistical uncertainty measures. These limitations restrict the extent to which the reported performance can be generalized to other datasets and real-world deepfake content.

Therefore, future research should evaluate the models using higher-resolution images, independent datasets, repeated experimental runs, and cross-dataset testing. Such evaluation would provide stronger evidence regarding the robustness and generalizability of the detection models. In addition, reporting computation time and model complexity would provide a more complete comparison between CNN, Random Forest, and SVM, particularly when considering practical deployment in information security systems.

4. CONCLUSION

Based on the results of the research that has been conducted, the comparison of Convolutional Neural Network (CNN), Random Forest, and Support Vector Machine (SVM) shows that the choice of classification approach has a substantial effect on deepfake image detection performance under the experimental conditions of this study. The research process consisted of dataset collection, data preprocessing, dataset division, implementation of the three classification algorithms, and model performance evaluation using accuracy, precision, recall, and F1-score. Experimental results show that the CNN algorithm achieved the highest performance, with an accuracy of 91%, precision of 93%, recall of 88%, and F1-score of 91%, compared with Random Forest, which achieved 75% accuracy, and SVM, which achieved 77% accuracy. The 14-percentage-point accuracy difference between CNN and SVM and the 16-percentage-point difference between CNN and Random Forest indicate that the performance advantage of CNN was not limited to a single evaluation metric. CNN also obtained the highest F1-score, suggesting a more balanced classification performance between precision and recall within the evaluated dataset. These findings indicate that the advantage of CNN in this experiment is associated with its ability to learn visual representations directly from image data, whereas Random Forest and SVM depend on the feature representation provided to the classifier. However, the result should not be interpreted as evidence that CNN is universally superior for deepfake detection. Rather, it demonstrates that CNN performed better than the two conventional classifiers evaluated in this study under the specific dataset, preprocessing procedure, image resolution, and experimental configuration employed. An important limitation of the findings is the restricted basis for generalization. The use of 32 × 32 pixel images may reduce computational requirements but can also discard fine-grained facial information that may be relevant to detecting sophisticated manipulation. In addition, the evaluation was conducted within the selected dataset and did not include cross-dataset testing. Consequently, the reported 91% CNN accuracy does not establish how well the model would perform on images originating from different datasets, identities, image qualities, compression levels, or deepfake generation techniques. Therefore, future research should focus not only on increasing the amount of data but also on testing model generalization through cross-dataset evaluation and using higher-resolution facial images that preserve manipulation-related details. Future experiments should also compare the existing CNN approach with more advanced architectures or transfer-learning approaches and report computational time alongside classification performance. These evaluations would provide stronger evidence regarding whether the performance advantage observed in this study remains consistent when the model encounters previously unseen forms of deepfake manipulation.

REFERENCES

[1] R. Ratnawita, "Cybersecurity in the AI Era Measures Deepfake Threats and Artificial Intelligence-Based Attacks," *Journal of the American Institute*, vol. 2, no. 2, pp. 180–189, Feb. 2025, doi: 10.71364/s3emxx77.

[2] K. T. Pedersen, L. Pepke, T. Stærmose, M. Papaioannou, G. Choudhary, and N. Dragoni, "Deepfake-Driven Social Engineering: Threats, Detection Techniques, and Defensive Strategies in Corporate Environments," *Journal of Cybersecurity and Privacy*, vol. 5, no. 2, p. 18, Apr. 2025, doi: 10.3390/jcp5020018.

[3] L. Gaur, *DeepFakes*. New York: CRC Press, 2022. doi: 10.1201/9781003231493.

[4] F. Hu and X. Hei, "AI, Machine Learning and Deep Learning: A Security Perspective," *AI, Machine Learning and Deep Learning: a Security Perspective*, pp. 1–334, Jan. 2023, doi: 10.1201/9781003187158/AI-MACHINE-LEARNING-DEEP-LEARNING-FEI-HU-XIALI-HEI/RIGHTS-AND-PERMISSIONS.

[5] A. A. Khan, A. A. Laghari, S. A. Inam, S. Ullah, M. Shahzad, and D. Syed, "A survey on multimedia-enabled deepfake detection: state-of-the-art tools and techniques, emerging trends, current challenges & limitations, and future directions," *Discover Computing*, vol. 28, no. 1, p. 48, Apr. 2025, doi: 10.1007/s10791-025-09550-0.

[6] L. Y. Gong and X. J. Li, "A Contemporary Survey on Deepfake Detection: Datasets, Algorithms, and Challenges," *Electronics (Basel).*, vol. 13, no. 3, p. 585, Jan. 2024, doi: 10.3390/electronics13030585.

Building of Informatics, Technology and Science (BITS)
Volume 8, No 1, June 2026 Page: 677−686
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v8i1.9555

[7] Y. Zhang, Z. Pang, S. Huang, C. Wang, and X. Zhou, "Unmasking AI-created visual content: a review of generated images and deepfake detection technologies," *Journal of King Saud University Computer and Information Sciences*, vol. 37, no. 6, p. 148, Aug. 2025, doi: 10.1007/s44443-025-00154-8.

[8] T. Luan, "A Survey on Deepfake Detection Technologies," *International Journal of Emerging Technologies and Advanced Applications*, vol. 2, no. 1, pp. 1–9, Feb. 2025, doi: 10.62677/IJETAA.2501131.

[9] A. Raza *et al.*, "A Comprehensive Review of Deepfake Detection Techniques: From Traditional Machine Learning to Advanced Deep Learning Architectures," *AI*, vol. 7, no. 2, p. 68, Feb. 2026, doi: 10.3390/ai7020068.

[10] O. A. H. H. Al-Dulaimi and S. Kurnaz, "A Hybrid CNN-LSTM Approach for Precision Deepfake Image Detection Based on Transfer Learning," *Electronics (Basel).*, vol. 13, no. 9, p. 1662, Apr. 2024, doi: 10.3390/electronics13091662.

[11] 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*, vol. 13, no. 4, p. 99, Apr. 2024, doi: 10.3390/computers13040099.

[12] Muh. H. A. Akbar, J. Jimsan, Y. Yahya, I. Ilcham, and N. Nasrullah, "XceptionNet-based Digital Image Forensics with DFRWS Framework for Deepfake Detection," *Insect (Informatics and Security): Jurnal Teknik Informatika*, vol. 11, no. 2, pp. 221–228, Oct. 2025, doi: 10.33506/insect.v11i2.4996.

[13] E. M. Dharma, N. M. S. Iswari, and I. P. R. A. Arimbawa, "DeepFake Image Detection Using Convolutional Neural Network with EfficientNet Architecture," *G-Tech: Jurnal Teknologi Terapan*, vol. 9, no. 4, pp. 1829–1838, Oct. 2025, doi: 10.70609/g-tech.v9i4.7727.

[14] R. Afandi and B. Purnama, "Detecting Deepfake Videos Using CNN and GRU Methods: Evaluating Performance on the Celeb-DF(v2) Dataset," *Jurnal Sistem Komputer dan Informatika (JSON)*, vol. 7, no. 2, pp. 448–458, Dec. 2025, doi: 10.30865/json.v7i2.9372.

[15] D. V. Kavitha, A. Dhanalakshmi, A. Sweena, and V. S. R. Harshini, "AI-Powered Deepfake Detection Using Ensemble Learning," *International Journal of Engineering and Computer Science*, vol. 14, no. 12, pp. 28070–28077, 2025, doi: 10.18535/ijecs/v14i12.5365.

[16] Hemanth Chandra N, Nikhitha S Naik, and Sameeksha A B, "Deepfake Creation and Detection of Multimedia Data," *International Journal of Advanced Research in Science, Communication and Technology*, vol. 12, no. 1, pp. 36–41, Feb. 2024, doi: 10.48175/IJARSCT-15308.

[17] Nimrita. Koul, "Ultimate Deepfake Detection Using Python Master Deep Learning Techniques Like CNNs, GANs, and Transformers to Detect Deepfakes in Images, Audio, and Videos Using Python (English Edition)," 2024, Accessed: Apr. 03, 2026. [Online]. Available: https://books.google.com/books/about/Ultimate_Deepfake_Detection_Using_Python.html?id=AHkqEQAAQBAJ

[18] T. Moulahi, M. N. Halgamuge, and P. Lorenz, *Mitigating the Risks of AI Deepfakes*. Boca Raton: CRC Press, 2026. doi: 10.1201/9781003539032.

[19] J. Mu, M. Adrezo, and A. N. Haikal, "Identifikasi Wajah Asli dan Buatan Deepfake Menggunakan Metode Convolutional Neural Network," *Teknika*, vol. 13, no. 1, pp. 45–50, Jan. 2024, doi: 10.34148/teknika.v13i1.705.

[20] A. F. Akbar, P. D. W. Ayu, and D. P. Hostiadi, "Performance Analysis of Deep Learning Architectures in Classifying Fake and Real Images," *JUITA: Jurnal Informatika*, vol. 13, pp. 167–176, Aug. 2025, doi: 10.30595/juita.v13i2.25790.