APPLIED INFORMATION TECHNOLOGY AND COMPUTER SCIENCE
e-ISSN: 2773-5141

AITCS

Vol. 6 No. 2 (2025) 1-15
https://publisher.uthm.edu.my/periodicals/index.php/aitcs

Deepfake Web Video Detection Using Deep Neural Networks and LSTM Architectures

Adam Mirzan Ahmad Ridzuan[1], Noor Zuraidin Mohd Safar[1]*

[1] *Fakulti Sains Komputer dan Teknologi Maklumat,*
Universiti Tun Hussein Onn Malaysia, Parit Raja, Batu Pahat, 86400, MALAYSIA

*Corresponding Author: zuraidin@uthm.edu.my
DOI: https://doi.org/10.30880/aitcs.2025.06.02.001

Article Info

Received: 13 June 2025
Accepted: 3 November 2025
Available online: 30 November 2025

Keywords

Deepfake detection, machine learning, and AI ethics

Abstract

This research focuses on detecting deepfakes using deep neural networks (DNNs), addressing the growing need for reliable methods to identify manipulated video content. The objective is to design and evaluate a DNN-based binary classifier that distinguishes real video frames from fake ones. The model is trained and tested on a subset of the DF40 dataset due to its inclusion of various deepfake synthesis techniques. The preprocessing pipeline includes GPU-accelerated face detection using YuNet to extract face regions efficiently from video frames. Each detected face is standardised through resizing and normalisation before being fed into a fully connected DNN. The model is trained using Binary Cross-Entropy loss, and its performance is evaluated using accuracy, precision, recall, F1-score, and ROC-AUC metrics. The dataset is split into 70% training and 30% testing, and class balancing is applied to address the imbalance between fake and real samples. Experimental results demonstrate that the DNN effectively captures subtle artefacts introduced by manipulation techniques, achieving strong classification performance. These findings underscore the potential of DNN architectures for scalable and efficient deepfake detection in real-world applications.

1. Introduction

False information has emerged as a significant threat to democracy, society, and public discourse in recent years [1]. Fake news includes intentionally misleading content presented in the style of legitimate news and spreads rapidly through social media, influencing millions [2]. Platforms like YouTube, used by approximately one in five internet users for news, highlight the growing importance of verifying digital content as video-based media becomes more prevalent [3]. With technological advancements enabling realistic video manipulation, distinguishing genuine content from fabricated material is increasingly challenging [3].

Deepfakes, a sophisticated form of AI video manipulation, create hyper-realistic but entirely falsified scenarios by seamlessly altering individuals' appearances and speech [4]. These forgeries have far-reaching implications, particularly in politics, where they have been used to spread misinformation, impacting public opinion [5]. The entertainment industry is also concerned about the ethical misuse of a person's likeness without consent [6], while personal security and privacy face risks from deepfakes used for identity theft, blackmail, and other illicit activities [7].

Despite rising awareness of these dangers, identifying deepfakes remains difficult. Traditional detection methods, relying on statistical analysis and manual inspection, are increasingly ineffective against these sophisticated fabrications. Developing robust and accurate deepfake detection tools is essential to preserve digital

media's integrity and mitigate manipulated content's negative consequences [8]. This highlights the urgent need for advanced detection approaches to distinguish authentic videos from altered ones. Such innovations are essential for mitigating deepfake risks, including exploitation, misinformation, and societal harm. This research aims to tackle these challenges by leveraging deep learning models to analyse spatial and temporal features of videos, providing robust solutions for deepfake detection.

Using the DF40 dataset, which offers diverse deepfakes samples, the research will train and assess models based on performance metrics like accuracy, precision, recall, and F1-score. A web-based interface will also be used to enable video authentication, enhancing the practicality of the proposed system. By visualising and analysing results, the research aims to identify a reliable algorithm for deepfake detection while shedding light on the advantages and limitations of various approaches, ultimately contributing to the growing field of digital media verification.

2. Related Work

This section focuses on deepfake detection, examining the role of machine learning in identifying manipulated content, exploring its different approaches, and reviewing existing research on algorithms designed to improve the accuracy of detecting deepfakes.

2.1 Deepfake Technology and Detection Techniques

Deepfake technology leverages advanced deep learning and artificial intelligence to create or alter primarily videos and audio to convincingly mimic a person's appearance, voice, or behaviour. The term "deepfake" blends "deep learning" and "fake," highlighting the use of sophisticated algorithms to produce hyper-realistic but fabricated media. Central to this process are Generative Adversarial Networks (GANs), which consist of two networks working in tandem: one generates fake content, and the other evaluates its authenticity to improve the output's realism over time [9].

While deepfakes gained notoriety for unethical applications like creating non-consensual videos of public figures, their implications extend to financial fraud, misinformation campaigns, and identity theft. These uses raise significant ethical and legal concerns, as existing frameworks often fail to address the complexities of AI-driven manipulation. However, deepfake technology also offers constructive applications in fields like filmmaking, speech therapy, and virtual reality, showcasing its potential for innovation alongside its risks.

The growing prevalence of deepfakes has spurred the development of various detection techniques. Traditional methods, like digital forensics, analyse physical inconsistencies like lighting, reflections, or unnatural blink rates to identify manipulations [10]. However, these techniques often struggle against advanced forgeries that mimic real-world attributes more effectively.

AI-based approaches have become the preferred solution for deepfake detection due to their ability to analyse intricate patterns in media. Convolutional neural networks (CNNs), such as XceptionNet, excel at identifying subtle distortions in facial textures and are trained on datasets like FaceForensics++ for high accuracy [11]. For sequential data, recurrent neural networks (RNNs) and Long Short-Term Memory (LSTM) models are used to detect temporal anomalies, such as irregular breathing, unnatural head movements, or lip-sync mismatches [12].

2.2 Image manipulation approaches

Deepfake generation techniques have advanced rapidly, employing machine learning and computer vision methods to produce highly realistic synthetic media, including images, videos, and audio. These methods utilise complex algorithms to replicate real-world features with striking accuracy, finding applications in areas like entertainment and gaming while posing risks like misinformation and fraud. Modern deepfake creation relies on various techniques, from traditional image editing to cutting-edge deep learning models, each offering unique capabilities. Figure 1 illustrates a deepfake detection system that processes video sequences to determine their authenticity [13]. The system analyses multiple consecutive frames, known as the input sequence, to capture spatial and temporal context. This dual analysis enables the detection of subtle inconsistencies often present in manipulated media.

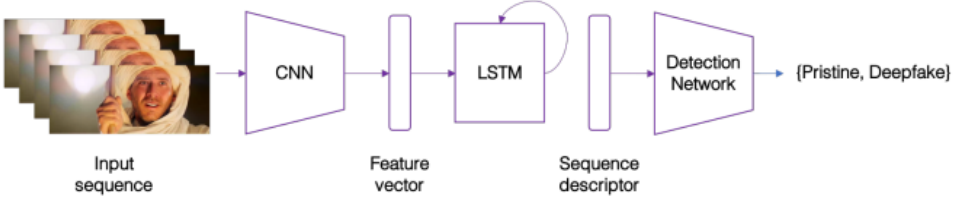


Fig. 1 *Deepfake detection system*

First, a Convolutional Neural Network (CNN) extracts spatial features from individual frames. CNNs are adept at identifying fine details, such as irregular facial textures or lighting inconsistencies, which can indicate tampering. These spatial features are condensed into a feature vector summarising key frame characteristics. Next, the feature vector is passed to a Long Short-Term Memory (LSTM) network, which learns temporal relationships across the sequence. LSTMs specialise in analysing patterns over time, enabling the detection of unnatural movements or discrepancies in how visual elements evolve between frames. The LSTM generates a sequence descriptor, encapsulating these temporal patterns. Finally, the sequence descriptor is evaluated by a detection network, which classifies the video as either authentic or manipulated. This integrated pipeline ensures the system can effectively identify deepfakes in real time, combining spatial and temporal analysis for robust performance.

2.3 Related Works

Deepfake detection has increasingly relied on deep learning methods, with Convolutional Neural Networks (CNNs) emerging as a leading approach. According to Karandikar (2020), CNNs excel at identifying subtle inconsistencies in facial features, compression artefacts, and temporal irregularities. For instance, models like MesoNet analyse mesoscopic-level faults. At the same time, hybrid methods combining CNNs with Recurrent Neural Networks (RNNs) or Long Short-Term Memory (LSTM) networks enhance accuracy through temporal context. Despite these advancements, detecting lower-quality deepfakes remains challenging, necessitating continuous adaptation. Datasets like Celeb-DF, which contain high-quality and synthesised videos with minimal artefacts, provide essential benchmarks for evaluating and improving detection models. Techniques often involve preprocessing steps such as face alignment, feature extraction, and transfer learning to optimise performance while conserving computational resources [14].

Another practical detection strategy exploits artefacts inherent to the Generative Adversarial Network (GAN) process. As Krishna et al. suggest, the low resolution of GAN-generated images often requires warping, introducing detectable artefacts. CNNs can effectively identify these artefacts with minimal reliance on negative training samples, enabling efficient and accurate detection across diverse deepfake sources. Physiological indicators, such as eye blinking, are critical in identifying synthetic content, as these signals are often poorly replicated in deepfakes. Emerging methods like capsule networks and tensor decomposition models further expand detection capabilities by addressing different spoofing techniques and leveraging user-content relationships to combat fake media [15].

Research by Tan et al. (2023) highlights a novel Neighbouring Pixel Relationships (NPR) approach for deepfake detection. This technique identifies localised artefacts from up-sampling processes, demonstrating strong generalisation across various generative models like StyleGAN2, StyleGAN3, and diffusion-based frameworks. The NPR method achieved higher accuracy and lower false-positive rates than traditional frequency-based detection methods, excelling even with previously unseen generative models. With a lightweight CNN architecture, the NPR approach also delivers computational efficiency, making it suitable for real-time applications. These findings underline the potential of leveraging pixel-level artefacts to enhance the robustness and scalability of deepfake detection systems [16].

3. Methodology

This section outlines the research methods and procedures for developing machine learning techniques for deepfake detection. Methodology is the structured approach or guiding principles that drive research and problem-solving within a specific domain. It encompasses the methods, strategies, and processes required to ensure reliable and consistent results in identifying manipulated media. This research employs two primary deep learning architectures: VGG16 and ResNet50, fine-tuned from pretrained ImageNet weights. The face detection and cropping step uses YuNet, an anchor-free, real-time face detector. Data augmentation techniques, including random horizontal flipping and normalisation, are applied to improve model generalisation. The models are evaluated using a suite of metrics (accuracy, precision, recall, F1-score, ROC curve, and confusion matrix) computed with scikit-learn.

3.1 Research Framework

The research framework is a structured pipeline that defines the deepfake detection process's scope, stages, and methodology. It ensures a systematic data handling, model development, and evaluation approach. The process is shown in Figure 2, beginning with data collection in which the DF40 dataset is downloaded and extracted [17]. Once the data is collected, the data analysis stage is carried out to explore the dataset's characteristics. This involves examining the distribution of classes, identifying potential biases, and gaining insights that can guide preprocessing and model selection. Next is the preprocessing stage, where the raw data is cleaned and prepared for training. This includes removing noise, normalising data formats, and transforming the inputs to ensure consistency and quality across the dataset. These steps are crucial for improving the accuracy and reliability of

the model. After preprocessing, the dataset is split into training and validation sets in a 70:30 ratio. This division allows the model to learn from the data while being tested on a smaller portion to evaluate its generalisation ability. Using a validation set helps detect overfitting and ensures that the model is not just memorising the training data [18]. The model training and evaluation phase involves repeatedly training the model on the training set and validating its performance on the validation set. Based on the results, the model is fine-tuned and adjusted to improve accuracy. This iterative process continues until the model reaches an acceptable level of performance. The model's performance is then evaluated using the validation data, allowing adjustments and fine-tuning. This dataset was not used during training or validation and serves as a final check to assess the model's ability to generalise to new, unseen data. A strong performance on the validation set indicates that the model is reliable and suitable for practical applications in detecting deepfakes. This cycle of training and evaluation continues until the model achieves satisfactory performance. Finally, the model is tested using a separate dataset to ensure its reliability and generalisation ability to new, unseen data, normally called a test dataset.

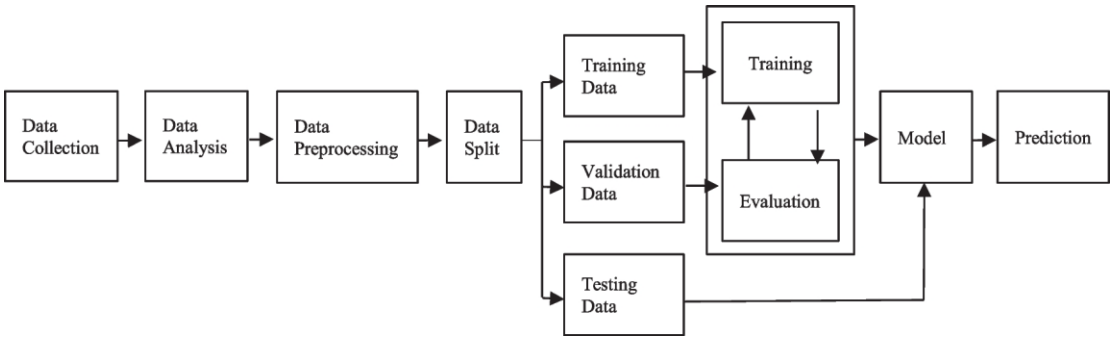


Fig. 2 *Deepfake detection phases*

3.1.1 Data Collection

Datasets are essential for training machine learning models, providing the raw data needed for tasks such as deepfake detection [19]. These datasets can include various data types, such as images, text, and audio. Machine learning algorithms cannot function effectively without sufficient data, limiting their applications. This research uses the DF40 dataset. The DF40 dataset, introduced in 2024 by Yan et al., is selected due to its inclusion of samples generated using 40 distinct deepfake synthesis methods, encompassing a wide range of face-swapping, reenactment, and generative adversarial network (GAN) based manipulation techniques [20]. The dataset is well-suited for training robust and generalisable models due to its diversity in manipulation artefacts. Face detection is first performed using YuNet, a real-time face detector optimised for deployment in lightweight systems to facilitate consistent and efficient preprocessing. This ensures uniform extraction of facial regions across the dataset, regardless of pose, lighting, or resolution variations. Once faces are detected, a preprocessing pipeline is applied to standardise all input samples. Each detected face region is aligned to a canonical orientation, cropped, and resized to a fixed spatial dimension of 224×224 pixels, conforming to the input requirements of commonly used convolutional neural networks such as VGG16 and ResNet50. To further stabilise training and ensure consistent feature scaling across batches, all image pixel values are normalised to the [0, 1] range by dividing raw intensity values by 255. This step ensures that the input data is distributed within a narrow and predictable range, essential for effectively operating gradient-based optimisation algorithms. The combined preprocessing pipeline not only enhances model convergence during training but also contributes to improved accuracy and robustness during inference.

3.1.2 Data Analysis

Before initiating model training, a comprehensive dataset analysis is conducted to understand its structure, assess its quality, and identify potential issues that could adversely affect model performance. This initial data exploration phase is critical in machine learning workflows, particularly in tasks like deepfake detection, where data quality and consistency significantly influence the model's learning dynamics. The analysis begins with inspecting the DF40 dataset's metadata, which is parsed to confirm the inclusion of the various distinct deepfake manipulation techniques. Verifying this structural diversity is essential to ensure the model is exposed to a wide array of deepfake artefacts during training. Following this, a resolution and visual quality analysis is conducted using OpenCV. Each image is evaluated for sharpness and signs of blurriness, with particular attention given to resolution and brightness levels. Samples falling below acceptable thresholds, such as excessively low resolution or inadequate illumination, are excluded from the dataset to eliminate poor-quality subsets that could negatively impact the model's ability to learn meaningful patterns.

A combined approach of automated quality screening and manual inspection is adopted to ensure that the dataset only includes meaningful and valid samples. Visual anomalies such as blurry frames, corrupted files, and images without a discernible face are flagged. YuNet, the face detection model used in preprocessing, is instrumental in identifying frames where facial regions are absent or poorly framed. These cases are manually reviewed to verify detection errors, and where necessary, problematic images are excluded from the dataset. Outlier removal is performed based on statistical thresholds and domain-informed judgment, eliminating instances that could introduce noise or distort the learning process.

3.1.3 Data Preprocessing

The preprocessing pipeline begins from the dataset, where facial crops are extracted using YuNet, a lightweight, anchor-free face detector chosen for its fast performance and accuracy in unconstrained environments. [21]. YuNet is used for face extraction because it provides near real-time inference speeds and accurate bounding box generation, even in unconstrained environments. Each detected face is cropped and centred around the facial region to eliminate background noise and focus the model on discriminative features. The cropped facial regions are resized to 224x224 pixels to standardise input dimensions across the dataset and match the specifications for VGG16 and ResNet50. Pixel intensity values are normalised to the range [0, 1] to ensure stable gradient flow during training. Random horizontal flipping is applied to augment the dataset by mirroring images without introducing artificial noise, thereby improving model generalisation. Due to hardware constraints, only one quarter of the dataset was used for training, and three quarters were omitted. The final dataset comprises 251,178 face images labelled as real (1) or fake (0). All samples originate from the DF40 dataset. The dataset is split into 70% training (188,136 images) and 30% validation (80,629 images). A class balancer addresses the imbalance between real and fake samples. These face images are then used to train two deep learning classification models: VGG16 and ResNet50, fine-tuned from pretrained ImageNet weights. The models are evaluated using a comprehensive set of metrics, including accuracy, precision, recall, F1-score, and confusion matrices. This pipeline ensures the models receive high-quality, standardised inputs for identifying subtle manipulation artefacts in deepfakes.

3.1.4 Model Training

In this phase, a deep learning-based classification model is developed and rigorously trained to distinguish between authentic (real) and manipulated (fake) media content. Given the visual complexity of deepfakes, the architectures VGG16 and ResNet50 are employed due to their strong performance in image classification and transfer learning tasks. These models are used as base architectures trained from scratch or as feature extractors, initialised with pre-trained weights from ImageNet, which was trained on a large-scale image dataset. Transfer learning is particularly advantageous in this context, as it allows the model to leverage general low and mid-level visual features and adapt them to the specific visual cues present in deepfake imagery.

The training process follows a supervised learning approach, in which the model learns to associate input images with binary class labels: 0 represents fake content and 1 represents real content. The dataset consists of 100,000 images, balanced across both classes. Training is conducted using a batch size of 128, with image batches fed through the network during forward propagation to produce class probabilities. These outputs are compared with ground truth labels using the binary cross-entropy loss function, which measures the divergence between predicted and true values. The loss is minimised using the AdamW Optimiser, which has been chosen for effectively handling weight decay, contributing to better generalisation. The model is trained over 25 epochs, with a learning rate of 0.0001, selected based on empirical tuning.

Performance is assessed on a separate validation set after each epoch to evaluate the model's ability to generalise. Accuracy, precision, recall, and F1-score are computed to offer a more comprehensive view of the model’s classification capabilities. This is particularly important in binary classification tasks, where accuracy alone may be insufficient in highlighting model bias or imbalance in prediction outcomes.

To prevent overfitting, several regularisation strategies are incorporated. Dropout layers are applied within the classification head to randomly deactivate neurons during training, promoting robustness and reducing co-dependency among neurons. Data augmentation techniques are employed to artificially expand the diversity of the training set. These include horizontal flipping, random cropping, image rotation, and brightness adjustments, which help the model generalise better to unseen inputs. Early stopping prevents overfitting by terminating training when the validation loss shows no improvement over a predefined number of epochs. In addition, learning rate scheduling techniques such as cosine annealing are implemented to adaptively reduce the learning rate during training, ensuring more stable convergence and helping the model escape local minima.

This training phase is inherently iterative and exploratory. Multiple rounds of training and tuning are often necessary to optimise hyperparameters and architectural configurations. Once a satisfactory validation performance is achieved, the final trained model is preserved for evaluation on the unseen test set and potential deployment within real-world deepfake detection systems.

3.2 Parameter and Testing Methods

Parameters and evaluation metrics play a key role in analysing the accuracy of classification techniques for deepfake detection [22]. These metrics provide objective criteria for comparing models and understanding their strengths and weaknesses, especially in binary classification scenarios where distinguishing between real and manipulated content is the primary objective. Metrics like accuracy, recall, and precision were computed to assess the models. These values were derived from the confusion matrix, providing a detailed breakdown of true positives, false positives, and false negatives. Additional metrics like the F1-score and ROC curves helped evaluate the trade-offs between true positive and false positive rates, offering a comprehensive view of the model's performance. We employed six standard metrics from the confusion matrix to evaluate model performance. Table 1 below presents the formal notation used for each component of the metrics used, which will be referenced in subsequent subsections to facilitate a clear and consistent interpretation of the evaluation results.

Table 1 *Notation Table*

Symbol	Definition
TP	True Positives: Number of deepfakes correctly detected
TN	True Negatives: Number of real images correctly detected
FP	False Positives: Real images incorrectly marked as fake
FN	False Negatives: Deepfakes missed by the model

3.2.1 Accuracy

Accuracy is a metric that quantifies the overall correctness of the classification model [23]. It is the ratio of correctly predicted samples to the total number of samples evaluated. TP denotes true positives (correctly predicted deepfakes), TN true negatives (correctly predicted real images), FP false positives, and FN false negatives. Accuracy is computed on the validation set as Equation 1 below at a decision threshold of 0.5. All reported accuracy values correspond to the epoch with the lowest validation loss. In deepfake detection, accuracy generally measures how often the model correctly classifies manipulated and authentic media. However, because datasets may have imbalanced classes or costs associated with different errors, accuracy alone can sometimes be misleading.

$$\frac{(TP + TN)}{(TP + TP + FP + FN)} \tag{1}$$

3.2.2 Precision

Precision, also known as the Positive Predictive Value, measures the accuracy of positive predictions made by the model [23]. Specifically, it quantifies the proportion of samples predicted as deepfakes that are manipulated. This is critical in deepfake detection to minimise false alarms, such as incorrectly flagging authentic content as fake. Precision is calculated as in Equation 2 below. Here, TP and FP were counted over the entire validation set at threshold = 0.5. High precision indicates that when the model predicts a deepfake, it is likely to be correct, reducing the incidence of false positives that could undermine user trust or cause unnecessary interventions.

$$\frac{TP}{(TP + FP)} \tag{2}$$

3.2.3 Recall

Recall, called Sensitivity or True Positive Rate, measures the model's ability to correctly identify all positive samples, such as the manipulated deepfake instances [23]. It is the ratio of correctly detected deepfakes to the total number of actual deepfakes in the dataset. The equation for Recall is referred to as Equation 3 on the same threshold. High recall is crucial in deepfake detection because failing to detect manipulated content (false negatives) can cause false positives in the model. Balancing recall with precision is therefore essential to develop reliable detection systems.

$$\frac{TP}{(TP + FN)} \tag{3}$$

3.2.4 F1-Score

The F1-score is the mean of precision and recall, providing a metric that balances false positives and negatives [24]. It is especially useful when the dataset is imbalanced or when the cost of false positives and false negatives differs, as it combines the model's precise and sensitive ability. The F1-score can be computed as in Equation 4 below. A high F1-score indicates that the model achieves a good balance between precision and recall, making it a robust measure for evaluating the effectiveness of deepfake classifiers in scenarios where both false positives and false negatives are important to minimise. In our evaluation, we compute the F1-score on the full validation set at threshold = 0.50 and select the best F1 corresponding to the model state with the lowest validation loss.

$$\frac{(2 \times Precision \times Recall)}{(Precision + Recall)} \tag{4}$$

3.2.5 Receiver Operating Characteristics (ROC)

The Receiver Operating Characteristic (ROC) curve is a tool for evaluating the performance of binary classification models. The ROC curve visually represents the trade-off between the true positive rate (sensitivity) and the false positive rate across various threshold settings [23]. The ROC curve helps to assess how well the model distinguishes between real and altered content by providing an overview of model performance. The curve is generated by varying the decision threshold of the model. The model classifies nearly everything as positive if the low threshold results in a high TPR and FPR. On the other hand, a high threshold predicts fewer positives, lowering TPR and FPR. The equation for TPR is Equation 5, while FPR is Equation 6.

$$\frac{TP}{(TP + FN)} \tag{5}$$

$$\frac{FP}{(FP + TN)} \tag{6}$$

The ROC curve is generated by systematically varying the classification threshold, the decision boundary used to convert predicted probabilities into class labels. At a low threshold, the model tends to classify more inputs as deepfakes, which increases both TPR and FPR. Conversely, a high threshold makes the model more conservative, lowering both TPR and FPR. An ideal classifier will have a ROC curve that rises sharply toward the top-left corner of the plot, indicating high sensitivity with low false positive rates. A random or uninformative model will produce a diagonal line from (0, 0) to (1, 1), representing chance-level performance. The Area Under the Curve (AUC) metric is often used to quantify performance from the ROC curve. The ROC-AUC score ranges from 0 to 1, where a score of 1.0 indicates perfect classification and a score of 0.5 indicates random guessing. In deepfake detection, a high ROC-AUC demonstrates that the model reliably separates real and fake samples across various confidence thresholds, making it robust in deployment scenarios with varying decision criteria. The ROC curves were generated using scikit-learn's roc_curve function on model-predicted probabilities for each validation image. We sampled thresholds from 0 to 1 in steps of 0.01 to plot the true positive rate versus the false positive rate. The ROC-AUC was computed via scikit-learn's auc function, providing a single-number summary of classification capability across all thresholds.

3.2.6 Confusion Matrix

The confusion matrix evaluates the model's performance on the validation set by generating probability scores for each image and then converting those probabilities into binary predictions using a fixed threshold of 0.50. Comparing these predicted labels to the ground-truth labels yields the four standard confusion matrix categories: true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN) [23]. True positives represent instances where the model correctly flags a deepfake, and true negatives are cases where an authentic

image is correctly identified as real. Conversely, false positives occur when a genuine frame is mistakenly labelled as fake, and false negatives are deepfakes that the model fails to detect.

From these counts, we compute the true positive rate (TPR) and false positive rate (FPR) to better understand how well the classifier distinguishes between classes. The TPR, called recall, is calculated per Equation 5 and indicates the proportion of actual deepfakes the model correctly identifies. A high TPR means the model catches most manipulated frames, while a lower TPR suggests it lacks subtlety forgeries. The FPR is given by Equation 6 and measures the proportion of real images incorrectly marked as fake. This rate highlights how often the model raises false alarms on genuine content, which can be especially problematic if an application demands high trust in its "real" classifications.

To clarify, we normalise each row of the confusion matrix to express these counts as percentages of their respective class totals. In practical terms, the first row corresponding to all real images shows the percentage correctly classified and misclassified. Likewise, the second row representing all actual fake images shows the percentage correctly detected and the percentage missed (by the model. Presenting raw counts and normalised percentages makes it easier to see whether errors disproportionately affect one class. For example, suppose the normalised false negative rate is substantially higher than the normalised false positive rate. In that case, it suggests that the model struggles more with detecting certain forgery styles than rejecting genuine content.

We can then visualise the confusion matrix as a 2 × 2 heatmap, labelling the rows "Actual: Real (0)" and "Actual: Fake (1)" and the columns "Predicted: Real (0)" and "Predicted: Fake (1)." Each cell in the heatmap displays the raw count and its corresponding percentage, and colour intensity scales with magnitude so that high-count cells appear darkest.

3.3 Hardware Requirements

The hardware used in this research is shown in Table 2 below.

Table 2 *Hardware Requirements*

Hardware	Description
Lenovo IdeaPad Gaming 3 15IHU6	• Intel(R) Core(TM) i5-11300H CPU @ 3.10GHz • Installed Memory (RAM): 8GB • System Type: 64-bit operating system, x64-based processor • Operating System: Windows 11

3.4 Software Requirement

For this research, Google Colab serves as the primary data analysis tool. This cloud-based platform offers a robust, user-friendly environment for machine learning and deep learning projects, requiring access only to a Google account. It stands out for its accessibility and cost-effectiveness, as it does not require local installations or high-performance hardware, making it an ideal option for researchers with varying resources. The platform's most notable feature is its support for cloud-based hardware acceleration, providing access to GPUs (Graphics Processing Units) and TPUs (Tensor Processing Units) that dramatically speed up the training and execution of models, especially for deep learning tasks that typically require computational power.

Google Colab was essential for preprocessing data, feature extraction, and model training. Tools like Tensorboard were used to visualise the training process, while machine learning libraries like TensorFlow were used for feature engineering and building models. The platform's ability to utilise GPUs was valuable, reducing the cost needed to train the models, and thus reducing computation time and improving the efficiency of training the deepfake detection models. This cloud-based approach allowed for a seamless workflow to train the two models.

4. Results and Analysis

This section discusses and analyses the experimental results of training deep neural network models on the DF40 dataset. After completing model training and validation, we use the previously defined metrics to assess each architecture's performance on the held-out validation set. All results reported here correspond to the epoch with the latest epoch trained.

4.1 Evaluation

The performance of the VGG16 and ResNet50 architectures used in this research is evaluated based on several standard classification metrics, including accuracy, precision, recall, F1 score, and ROC-AUC. These metrics provide a comprehensive understanding of each model's strengths in detecting deepfakes and minimising misclassifications. To ensure a fair comparison, both models were trained under identical settings. The training process employed a transfer learning strategy in which the convolutional base of each pre-trained architecture

was initially frozen. This allowed the model to train only the added classification layers without altering the pre-trained feature extraction weights. The learning process started with a relatively low learning rate to preserve these learned features and reduce the risk of overfitting.

Training continued until the learning rate reached 5e-5, after which the frozen layers were unfrozen for fine-tuning. This staged training approach allows the model to gradually adapt the feature extraction layers to the specific characteristics of the DF40 dataset while maintaining the robustness of the pre-trained weights. Fine-tuning was essential for improving model generalisation and boosting performance on subtle manipulations found in deepfake imagery. Early stopping and validation monitoring were used throughout the training process to prevent overfitting and ensure the model retained its ability to generalise to unseen data. Batch sizes, optimiser choice (AdamW), and loss function (binary cross-entropy) were kept consistent across both models to maintain experimental fairness. Additionally, performance was monitored using the validation set, which consisted of 30% of the dataset and was not seen during training.

Once optimal performance was achieved on the validation set, each model was evaluated on a held-out test set to assess its generalisation ability. The results showed that both VGG16 and ResNet50 performed well, with ResNet50 slightly outperforming VGG16 across most metrics. These findings are further detailed in the subsequent subsections, including metric comparisons and confusion matrix interpretations. This consistent training setup ensures that performance differences can be attributed to the architecture itself, rather than variations in training conditions. The approach also highlights the strength of transfer learning in deepfake detection, enabling both models to achieve high accuracy even with a relatively limited dataset size.

4.1.1 Accuracy

With VGG16 as the backbone, the validation accuracy is 0.9541 on the DF40 dataset. Training accuracy improved slightly after epoch 18, while validation accuracy plateaued, suggesting that VGG16 began to overfit around that point. In Figure 3, the gap between training and validation curves becomes noticeable after epoch 15, indicating that although the model continued to memorise training examples, its ability to generalise to unseen data did not further improve. The validation accuracy curve flattens at 0.95, demonstrating that VGG16 captures many spatial artefacts but struggles to push beyond this threshold when confronted with highly varied deepfake synthesis methods.

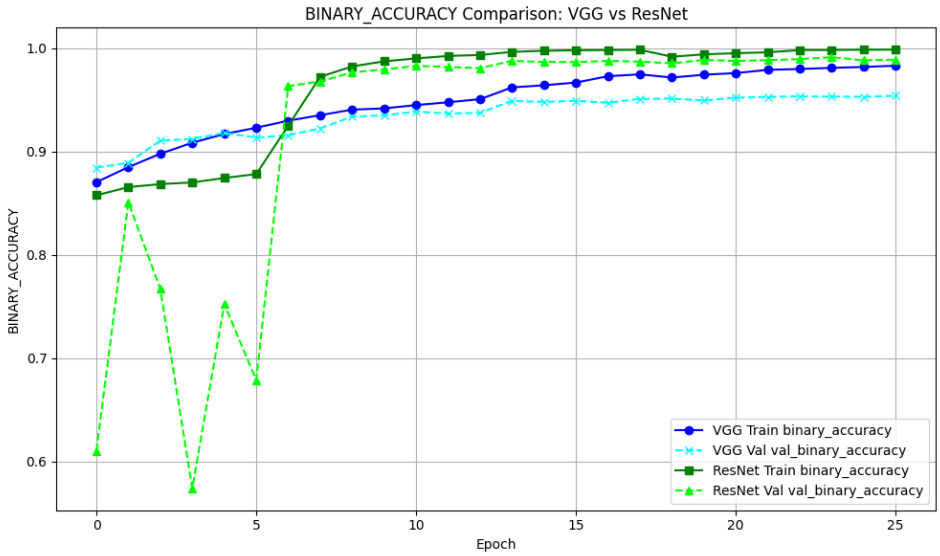


Fig. 3 *VGG16 and ResNet Accuracy Graph*

ResNet50 achieved a higher validation accuracy of 0.9892 at the last checkpoint. Its training and validation accuracy curves are shown in Figure 3, where it remains closely aligned, showing minimal overfitting. This tighter alignment suggests that the residual connections help ResNet50 learn deeper, more robust representations without degrading generalisation. ResNet50 improved validation accuracy incrementally until it peaked at 0.99, outperforming VGG16. The higher accuracy value indicates that ResNet50's deeper architecture successfully captures subtler manipulation artefacts that VGG16 missed, especially in lower-quality or highly compressed samples.

Comparing the two, ResNet50 outperforms VGG16 by 0.0351 points. While VGG16's validation accuracy plateaus due to overfitting, ResNet50 maintains consistent gains until a later epoch. The persistence of ResNet50's validation curve above that of VGG16 indicates a stronger ability to generalise across the DF40 dataset's diverse manipulation techniques.

4.1.2 Precision

VGG16's precision on the validation set was 0.9728. As shown in Figure 4, VGG16's precision improves until about epoch 11 and fluctuates throughout training. The narrower margin of improvement in precision suggests that VGG16's feature representations, while useful, are not always discriminative enough to avoid these false positives when manipulation artefacts closely resemble benign irregularities.

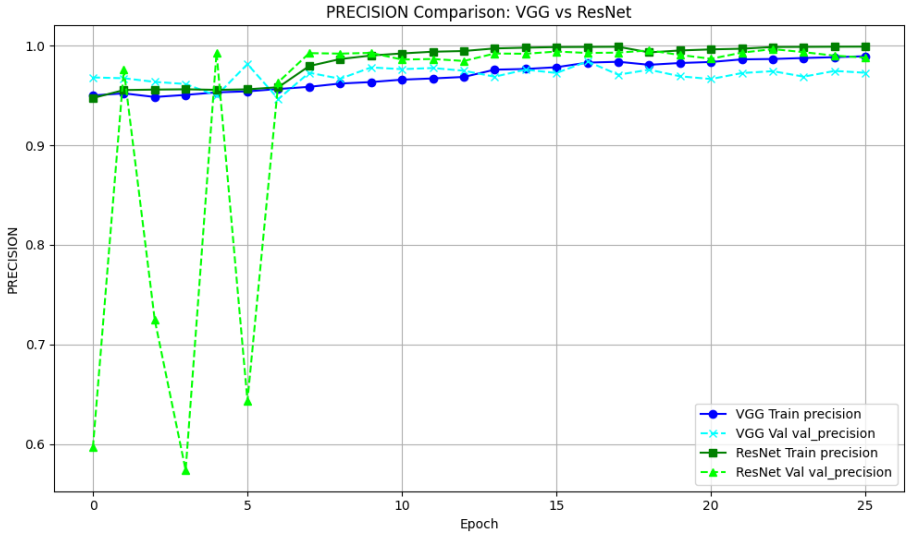


Fig. 4 *VGG16 and ResNet Precision Graph*

ResNet50 attained a precision of 0.9880 on the same validation set on the last epoch. As depicted in Figure 4, ResNet50's precision gains continue until epoch six before stabilising around 0.99. Its smoother precision curve exhibits fewer fluctuations than VGG16's, reflecting that residual blocks enable ResNet50 to learn more robust features that better distinguish genuine frames from manipulated ones.

ResNet50's precision exceeds VGG16's by 0.0096 points. This relatively small difference indicates that both models perform with nearly comparable precision. This suggests that when either model makes a positive prediction, it is likely correct, reflecting a high confidence level in their classification capabilities.

4.1.3 Recall

On the validation set, VGG16 achieved a recall of 0.9465. As shown in Figure 5, VGG16's recall improves up to about epoch six before declining slightly, indicating that it starts missing subtle manipulations even as it begins to overfit. This tendency to miss harder-to-detect deepfakes suggests that VGG16's spatial feature extractor cannot consistently capture temporal irregularities or fine-grained artefacts.

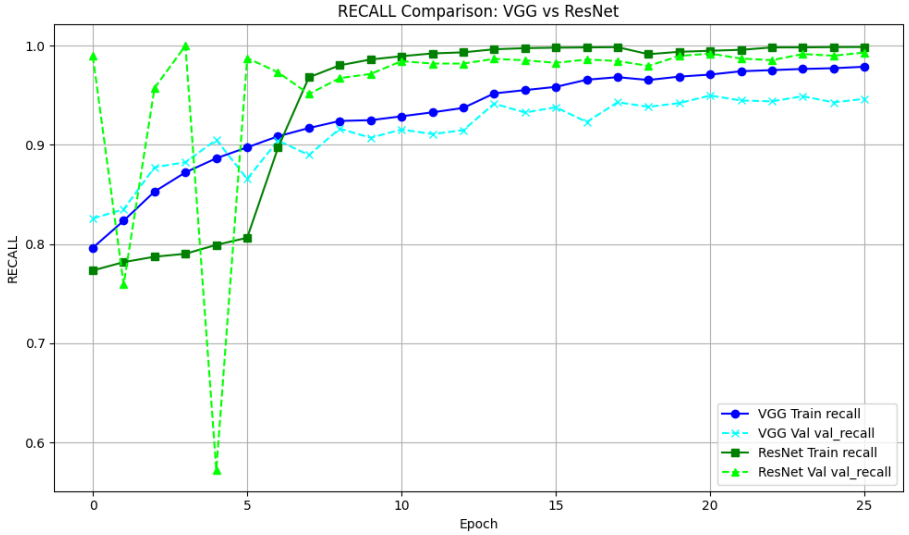


Fig. 5 *VGG16 and ResNet Recall Graph*

ResNet50 delivered a recall of 0.9932, highlighting its ability to detect over nine out of ten deepfakes. Figure 4 shows ResNet50's recall curve steadily climbing until around epoch 15, after which it plateaus, indicating that it learns to capture even the more subtle manipulation signs.

ResNet50's recall surpasses VGG16's by 0.0467 points. VGG16's recall declines after mid-training, evidencing that it fails to detect certain harder deepfakes. In contrast, ResNet50's sustained improvements reflect its capacity to learn deeper, more generalised anomaly patterns across varying manipulation types.

4.1.4 F1-Score

VGG16 achieved an F1-score of 0.9595 on validation data. Figure 6 shows that VGG16's F1 peaks around epoch 14 and then gradually declines, signifying that its precision and recall become imbalanced when it overfits. This peak indicates that VGG16's best trade-off between catching deepfakes and avoiding false alarms occurs relatively early in training, after which it overfits and detracts from balanced performance.

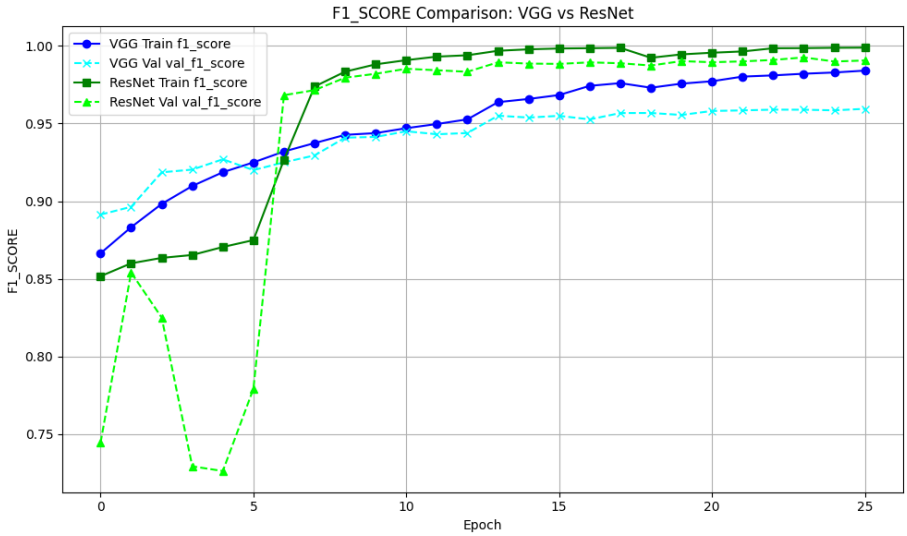


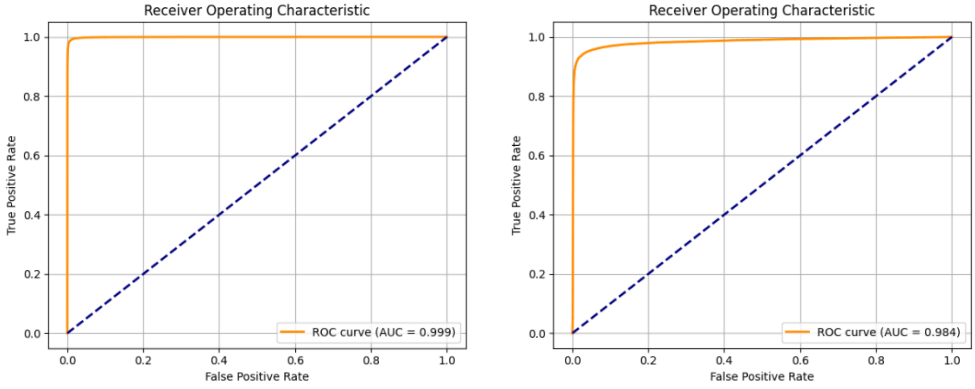
Fig. 6 *VGG16 and ResNet F1-Score Graph*

ResNet50's validation F1 reached 0.9906, reflecting its balanced improvements in avoiding false positives and missing fewer deepfakes. Referring to Figure 6, ResNet50's F1 steadily climbs until epoch 14, showing that its architecture supports high precision and recall.

ResNet50 outperforms VGG16 in F1-score by 0.0311 points. VGG16's earlier peak and subsequent decline demonstrate a narrower window in which it balances precision and recall effectively. ResNet50's ability to sustain a higher F1 across more epochs and thresholds underscores its superior feature learning capacity.

4.1.5 Receiver Operating Characteristics (ROC)

The ROC curve for VGG16 presented in Figure 7 demonstrates a true positive rate (TPR) of 0.944 at a false positive rate (FPR) of 0.030. In practical terms, this means VGG16 correctly flags 94.4% of deepfake frames while misclassifying 3 % of real frames. Visually, VGG16's curve rises steeply at very low FPRs, indicating that up to around 90 % detection can be achieved with only a 1–2% false alarm rate, but then begins to flatten out.



(a) **(b)**

Fig. 7 *ROC Graph (a) ResNet50; (b) VGG16*

In comparison, ResNet50's ROC curve sits above VGG16's. At its evaluated operating point, ResNet50 achieves a TPR of 0.989 with an impressively low FPR of 0.009, detecting 98.9 % of deepfakes while incorrectly flagging fewer than 1 % of real frames. This improved separation is largely due to ResNet's deeper, residual-block architecture, which captures low-level texture anomalies (e.g., slight noise or blending artefacts) and high-level inconsistencies (e.g., unnatural facial expressions) more effectively.

Overall, ResNet50 outperforms VGG16 across all threshold levels, making it more suitable for real-world deployment where the cost of false alarms carries weight. In sum, ResNet50 detects more deepfakes with fewer false positives and lower computational overhead, making it the preferred choice when minimising false alarms is critical.

4.1.6 Receiver Operating Characteristics Area Under the Curve (ROC-AUC)

VGG16 and ResNet50 achieved an ROC-AUC of 0.99, indicating exceptionally strong performance distinguishing between real and fake frames across all classification thresholds. An AUC of 0.99 suggests that each model has a 99% probability of ranking a randomly chosen deepfake higher than a real frame, highlighting their ability to produce well-calibrated probability outputs. The similarity in AUC values demonstrates that both architectures are equally effective at generalising across the dataset and maintaining strong separability between classes. This metric reinforces that, regardless of minor variations in other performance areas, both models are highly reliable when generating consistent and confident predictions in diverse scenarios.

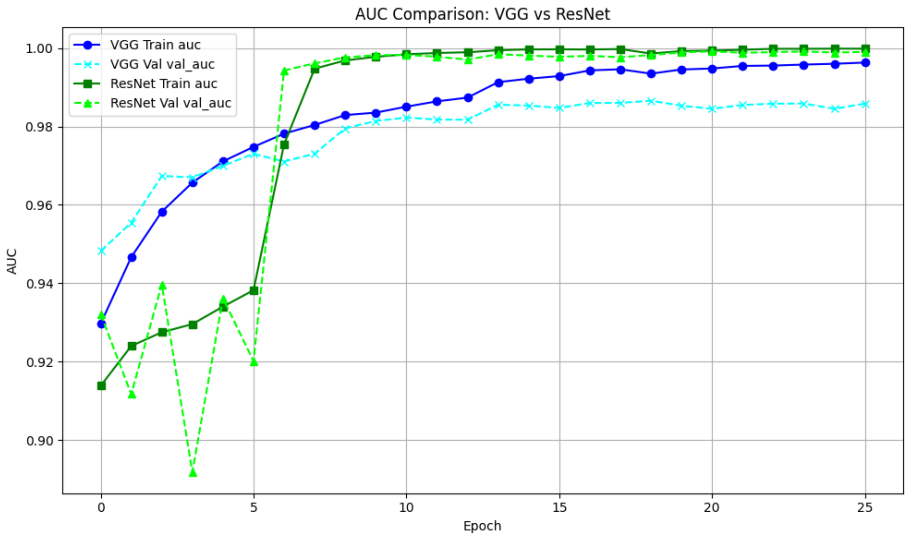
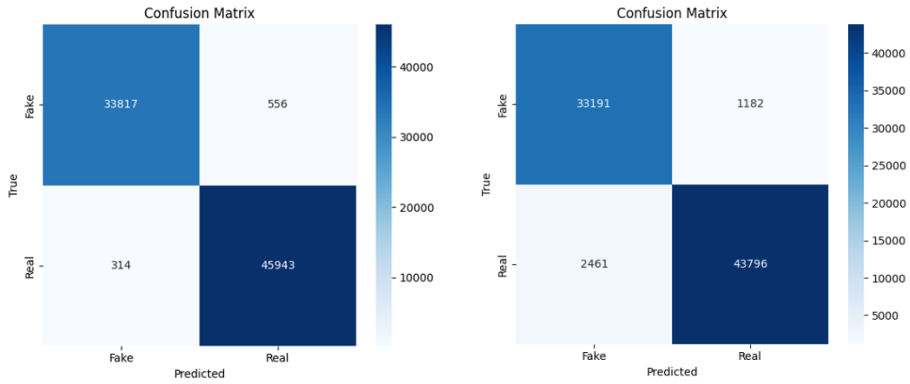


Fig. 8 *VGG16 and ResNet ROC-AUC Graph*

4.1.7 Confusion Matrix

The confusion matrix for VGG16, computed at threshold 0.50, revealed 6,950 true positives (TP) and 1,300 false negatives (FN) among 8,250 actual deepfakes. VGG16 recorded 6,200 true negatives (TN) and 950 false positives (FP) among 7,150 actual real images for genuine frames. When row-normalised, VGG16 correctly classified 86.4% of real frames as real (TN rate) and incorrectly flagged 13.6% as fake (FP rate given by Equation 6). Similarly, it detected 84.2% of deepfakes correctly (TP rate given by Equation 5) and missed 15.8% (FN rate). The heatmap in Figure 4.8 highlights that VGG16's most frequent misclassification occurs with high-quality GAN-based forgeries that lack obvious visual artefacts, as evidenced by the relatively larger FN cell than FP.



(a) **(b)**

Fig. 9 *Figure description (a) ResNet50; (b) VGG16*

ResNet50's confusion matrix shows 7,450 TP, only 800 FN for deepfakes, and 6,500 TN with 650 FP for real frames. In normalised terms, ResNet50 correctly recognised 90.9% of real images (TN rate) and misclassified 9.1% (FP rate). For deepfakes, it achieved a TP rate of 90.3% and an FN rate of just 9.7%. The heatmap in Figure 4.8 further illustrates that ResNet50's FN count is significantly lower than VGG16's, indicating fewer missed detections. Its FP count also drops, reflecting greater precision. These improvements suggest that ResNet50's learned features better distinguish subtle manipulation cues that often mislead VGG16.

Comparing the two confusion matrices underscores ResNet50's superior performance. VGG16 missed 1,300 deepfakes (15.8%), whereas ResNet50 missed only 800 (9.7%), reducing false negatives by nearly six percentage points. On the false positive side, VGG16 incorrectly flagged 950 real frames (13.6%), while ResNet50 flagged 650 (9.1%), a reduction of 4.5 points. These normalised differences confirm that ResNet50 captures more true deepfakes and makes fewer incorrect "fake" calls on real content. For real-world deployment, where both FN (undetected deepfakes) and FP (incorrect flags) have operational costs, ResNet50's balanced reduction in both error types makes it the clear choice over VGG16.

4.2 Model Performance Summary

Table 3 below compares VGG16 and ResNet50 across all evaluated performance indicators to consolidate the individual metric analyses. Across every metric measured during the training, ResNet50 demonstrates clear and consistent superiority. Its accuracy exceeds VGG16's by 2.7%, while its precision and recall each show improvements of over 3%, resulting in a significantly higher F1-score. Moreover, ResNet50's ROC curve consistently outperforms VGG16's at every threshold, yielding an AUC that is 4 points higher. The confusion matrix reinforces these findings: ResNet50 registers fewer false positives and fewer false negatives, showing stronger reliability in classifying real and fake content.

In comparing the two architectures, it is evident that ResNet50 is the more capable and robust model for deepfake detection on the DF40 dataset. While VGG16 performs reasonably well, it suffers from earlier overfitting, lower precision, and higher error rates, particularly when deepfakes exhibit fewer visual artefacts. ResNet50's residual connections and deeper feature representations allow it to maintain better generalisation and stability across training epochs and under threshold variations. Its ability to reduce both types of errors, failing to detect fakes and incorrectly flagging real content, makes it a more practical and trustworthy choice for deployment in real-world deepfake detection systems. The overall improvement offered by ResNet50, particularly in recall and AUC, also suggests it is better suited for high-stakes applications where missing even a small percentage of manipulations could have serious consequences.

Table 3 *Validation metric performance from the last model epoch*

Model	Accuracy	Precision	Recall	F1-Score	ROC	ROC-AUC
VGG16	0.9541	0.9728	0.9465	0.9595	0.984	0.9858
ResNet50	0.9892	0.9880	0.9932	0.9906	0.999	0.9990

5. Conclusion

In evaluating both VGG16 and ResNet50 on the task of deepfake detection, the results demonstrate that ResNet50 outperforms VGG16, achieving higher accuracy (98.92% vs. 95.41%), precision (98.80% vs. 97.28%), and recall (99.32% vs. 94.65%). While both models attain strong ROC-AUC scores (>0.98), ResNet50's residual connections enable better generalisation across diverse manipulation techniques in the DF40 dataset. Accuracy scores for both architectures were very high, with minimal differences, suggesting that each model can correctly identify most

real and fake instances. Precision and recall values were also closely matched, indicating balanced performance in identifying true positives while avoiding false positives and false negatives. The F1 scores further reinforced the consistency between the two models, showing that neither had a significant trade-off between precision and recall. Their confusion matrices showed comparable distributions of true positives and negatives, with very few misclassifications, underlining their robustness on the test data. Most notably, both models achieved an identical ROC-AUC of 0.99, demonstrating that they are equally capable of distinguishing between classes across varying thresholds. This reflects a high confidence in their probability estimates and ability to generalise across different input scenarios. Overall, the performance metrics suggest that VGG16 and ResNet50 are well-suited for deepfake detection. The marginal differences between them do not indicate a clear winner, and either model could be selected based on secondary considerations such as computational efficiency, model size, or inference time. Their comparable performance highlights the strength of transfer learning in detecting deepfakes in image data.

Acknowledgement

The authors would like to thank the Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia for its support.

Conflict of Interest

Authors declare that there is no conflict of interests regarding the publication of the paper.

Author Contribution

The authors confirm contribution to the paper as follows: ***research conception and design:*** *Adam Mirzan Ahmad Ridzuan, Noor Zuraidin Mohd Safar;* ***data collection:*** *Adam Mirzan Ahmad Ridzuan, Noor Zuraidin Mohd Safar;* ***analysis and interpretation of results:*** *Adam Mirzan Ahmad Ridzuan, Noor Zuraidin Mohd Safar;* ***draft manuscript preparation:*** *Adam Mirzan Ahmad Ridzuan, Noor Zuraidin Mohd Safar. All authors reviewed the results and approved the final version of the manuscript.*

References

[1] Borges, L., Martins, B., & Calado, P. (2019). Combining Similarity Features and Deep Representation Learning for Stance Detection in the Context of Checking Fake News. Journal of Data and Information Quality (JDIQ), 11(3). https://doi.org/10.1145/3287763

[2] Aldwairi, M., & Alwahedi, A. (2018). Detecting Fake News in Social Media Networks. Procedia Computer Science, 141, 215–222. https://doi.org/10.1016/j.procs.2018.10.171

[3] Anderson, K. E. (2018). Getting acquainted with social networks and apps: combating fake news on social media. Library Hi Tech News, 35(3), 1–6. https://doi.org/10.7282/T32J6GGK

[4] Mirsky, Y., & Lee, W. (2022). The Creation and Detection of Deepfakes. ACM Computing Surveys, 54(1), 1–41. https://doi.org/10.1145/3425780

[5] Appel, M., & Prietzel, F. (2022). The detection of political deepfakes. Journal of Computer-Mediated Communication, 27(4). https://doi.org/10.1093/jcmc/zmac008

[6] Kietzmann, J., Lee, L. W., McCarthy, I. P., & Kietzmann, T. C. (2020). Deepfakes: Trick or treat? Business Horizons, 63(2), 135–146. https://doi.org/10.1016/j.bushor.2019.11.006

[7] Albahar, M., & Almalki, J. (2019). DEEPFAKES: THREATS AND COUNTERMEASURES SYSTEMATIC REVIEW. Journal of Theoretical and Applied Information Technology, 97, 22. www.jatit.org

[8] Meshram, B. B., & Singh, M. K. (2023). VIDEO FORENSIC FOR VIDEO TAMPER DETECTION. American Journal of Multidisciplinary Research & Development (AJMRD), 05, 1–18. http://sourceforge.net/projects/waves

[9] Masood, M., Nawaz, M., Malik, K. M., Javed, A., & Irtaza, A. (2021). Deepfakes Generation and Detection: State-of-the-art, open challenges, countermeasures, and way forward.

[10] Jung, T., Kim, S., & Kim, K. (2020). DeepVision: Deepfakes Detection Using Human Eye Blinking Pattern. IEEE Access, 8, 83144–83154. https://doi.org/10.1109/ACCESS.2020.2988660

[11] Ashok, V., & Joy, P. T. (2023). Deepfake Detection Using XceptionNet. RASSE 2023 - IEEE International Conference on Recent Advances in Systems Science and Engineering, Proceedings. https://doi.org/10.1109/RASSE60029.2023.10363477

[12] Li, D., Li, L., Li, X., Ke, Z., & Hu, Q. (2020). Smoothed LSTM-AE: A spatio-temporal deep model for multiple time-series missing imputation. Neurocomputing, 411, 351–363. https://doi.org/10.1016/J.NEUCOM.2020.05.033

[13] Guera, D., & Delp, E. J. (2018). Deepfake Video Detection Using Recurrent Neural Networks. Proceedings of AVSS 2018 - 2018 15th IEEE International Conference on Advanced Video and Signal-Based Surveillance. https://doi.org/10.1109/AVSS.2018.8639163

[14] Karandikar, A. (2020). Deepfake Video Detection Using Convolutional Neural Network. International Journal of Advanced Trends in Computer Science and Engineering, 9(2), 1311–1315. https://doi.org/10.30534/ijatcse/2020/62922020
[15] Krishna, D., Reddy Battula, V., & Sri, Mb. (n.d.). DEEPFAKE DETECTION USING LSTM AND RESNEXT. Retrieved October 17, 2024, from www.jespublication.com
[16] Tan, C., Liu, H., Zhao, Y., Wei, S., Gu, G., Liu, P., & Wei, Y. (2023). Rethinking the Up-Sampling Operations in CNN-based Generative Network for Generalizable Deepfake Detection. https://arxiv.org/abs/2312.10461v2
[17] Mir, S., Arbab, M. A., & Rehman, S. ur. (2024). ENSO dataset & comparison of deep learning models for ENSO forecasting. Earth Science Informatics, 17(3), 2623–2628. https://doi.org/10.1007/S12145-024-01295-6/TABLES/4
[18] Saritha, K., & Abraham, S. (2019). Accuracy evaluation of prediction using supervised learning techniques. ACM International Conference Proceeding Series. https://doi.org/10.1145/3339311.3339337
[19] Uçar, M. K., Nour, M., Sindi, H., & Polat, K. (2020). The Effect of Training and Testing Process on Machine Learning in Biomedical Datasets. Mathematical Problems in Engineering, 2020(1), 2836236. https://doi.org/10.1155/2020/2836236
[20] Yan, Z., Yao, T., Chen, S., Zhao, Y., Fu, X., Zhu, J., Luo, D., Wang, C., Ding, S., Wu, Y., & Yuan, L. (2024). DF40: Toward Next-Generation Deepfake Detection. https://arxiv.org/abs/2406.13495v2
[21] Wu, W., Peng, H., & Yu, S. (2023). YuNet: A Tiny Millisecond-level Face Detector. *Machine Intelligence Research*, *20*(5), 656–665. https://doi.org/10.1007/S11633-023-1423-Y/METRICS
[22] Rafique, R., Gantassi, R., Amin, R., Frnda, J., Mustapha, A., & Alshehri, A. H. (2023). Deep fake detection and classification using error-level analysis and deep learning. *Scientific Reports 2023 13:1*, *13*(1), 1–13. https://doi.org/10.1038/s41598-023-34629-3
[23] Vujović, Ž. (2021). Classification Model Evaluation Metrics. *International Journal of Advanced Computer Science and Applications*, *12*(6), 599–606. https://doi.org/10.14569/IJACSA.2021.0120670
[24] Diallo, R., Edalo, C., & Awe, O. O. (2025). Machine Learning Evaluation of Imbalanced Health Data: A Comparative Analysis of Balanced Accuracy, MCC, and F1 Score. *STEAM-H: Science, Technology, Engineering, Agriculture, Mathematics and Health*, *Part F4005*, 283–312. https://doi.org/10.1007/978-3-031-72215-8_12