Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 10, Issue 1, Januari 2026
DOI : https://doi.org/10.33395/sinkron.v10i1.15582
e-ISSN : 2541-2019
p-ISSN : 2541-044X

Enhanced Performance Evaluation of VGG16 and ResNet50 for Deepfake Detection Using Local Ternary Pattern

Ghifari Ferdian Rizqullah[1)*], Puspa Eosina[2)], Andik Eko Kristus Pramuko[3)]
[1)2)3)]Teknik Informatika, Universitas Ibn Khaldun, Indonesia
[1)]ghifaririzqullah0@gmail.com, [2)] puspa.eosina@ft.uika-bogor.ac.id, [3)] andik.eko@ft.uikabogor.ac.id

Submitted :Nov 13, 2025 | **Accepted** : Jan 23, 2026 | **Published** : Jan 25, 2026

Abstract:.Deepfake video generation has become increasingly sophisticated, posing challenges for detection methods that rely solely on convolutional neural networks (CNNs without explicit texture enhancement). Many existing approaches have limited robustness in capturing subtle texture inconsistencies caused by manipulation, compression, and noise. This study investigates the integration of Local Ternary Pattern (LTP)–based texture enhancement with transfer learning models for deepfake video detection. Specifically, VGG16 and ResNet50 architectures are evaluated using the Celeb-DF (v2) dataset. LTP is employed to extract fine-grained texture features due to its higher robustness to illumination variations and noise compared to conventional descriptors such as Local Binary Pattern (LBP). Video frames are processed individually and used to train CNN classifiers, followed by evaluation at both frame and video levels. Experimental results show that ResNet50 outperforms VGG16, achieving a test accuracy of 93% with a validation loss of 0.2228, while VGG16 reaches an accuracy of 88% with a validation loss of 0.2636. Further testing on 20 withheld videos demonstrates that ResNet50 correctly classifies all samples, whereas VGG16 misclassifies two real videos, indicating lower robustness to real-video misclassification. These results demonstrate that LTP-based texture enhancement effectively supports CNN-based deepfake detection and that deeper architectures benefit more from enriched texture representations. This study provides empirical insights into improving robustness and reliability in deepfake video classification.

Keywords: Convolutional Neural Networks, Deepfake, Local Ternary Pattern, Texture Enhancement, Transfer Learning.

INTRODUCTION

The rapid advancement of artificial intelligence (AI) has significantly accelerated the generation of digital content, including the emergence of deepfake technology. Deepfake refers to AI-based techniques capable of manipulating facial appearance, expressions, and identity in images, videos, or audio with a high degree of realism (Maulana, 2023; Zain, 2024). Although deepfake technology can be used for beneficial purposes such as entertainment and education, its misuse has led to serious issues including fraud, misinformation, and identity manipulation in various countries, including Indonesia. These risks pose major challenges to information authenticity, personal reputation, and digital security, thereby emphasizing the urgent need for reliable and automated deepfake detection systems.

Technically, deepfake generation relies on complex AI models that alter facial regions and expressions, producing highly realistic forged content that is difficult to distinguish from authentic media (Malik, Kuribayashi, Abdullahi, & Khan, 2022; Rahayu & Santoso, 2023). The high visual quality of deepfakes often conceals manipulation artifacts from human perception. As a result, traditional inspection approaches are insufficient, and digital forensic systems must depend on computational models capable of identifying subtle inconsistencies embedded within manipulated facial textures (Rana, Nobi, Murali, & Sung, 2022). This challenge highlights the importance of texture-based detection approaches, as deepfake manipulation processes frequently introduce micro-texture anomalies that conventional models tend to overlook.

To facilitate deepfake detection research, several benchmark datasets have been introduced, including FaceForensics++, DFDC, and Celeb-DF (Dolhansky et al., 2020; Li, Yang, Sun, Qi, & Lyu, 2020; Rossler et al.,

*name of corresponding author

2019). Among these, Celeb-DF (v2) is widely adopted due to its high visual quality, diverse facial expressions, and complex illumination conditions. However, despite the availability of large-scale datasets, many deepfake classification models still struggle to capture subtle facial texture details such as wrinkles, skin folds, and local noise inconsistencies, which are crucial forensic indicators for distinguishing genuine and manipulated faces (Xu, Liu, Liang, Lu, & Zhang, 2021).

Most state-of-the-art deepfake detection methods rely on convolutional neural networks (CNNs) to automatically learn discriminative features. While CNNs are effective at capturing high-level semantic information, they often fail to preserve fine-grained texture details due to repeated downsampling and pooling operations. Consequently, subtle manipulation artifacts may be suppressed, leading to reduced detection performance. This limitation has motivated the integration of handcrafted texture descriptors as complementary features for deep learning-based forensic systems.

Local Binary Pattern (LBP) has been widely used as a texture descriptor in deepfake detection studies; however, it is sensitive to noise and illumination variations. To address these limitations, Local Ternary Pattern (LTP) extends LBP by introducing a three-level encoding scheme that improves robustness to noise while preserving discriminative texture variations (L, V, & M, 2022). By effectively capturing fine-grained facial texture patterns, LTP has strong potential to enhance the detection of subtle deepfake artifacts. Nevertheless, the effectiveness of LTP when combined with modern deep CNN architectures has not been sufficiently investigated, particularly on challenging deepfake video datasets.

In addition to feature representation, deepfake detection faces challenges related to high computational costs and limited labeled data. Transfer learning has therefore been widely adopted to leverage pre-trained models and improve training efficiency (Kamal & Ez-zahraouy, 2023). Popular architectures such as VGG16 and ResNet50 have demonstrated strong performance in image classification tasks. VGG16 offers a simple yet effective convolutional structure for feature extraction (Albashish, Al-Sayyed, Abdullah, Ryalat, & Ahmad Almansour, 2021), while ResNet50 employs residual connections to mitigate the vanishing gradient problem and enable deeper representations (Fathur Rozi, Adiwijaya, & Swasono, 2023).

Previous studies have proposed diverse deepfake detection approaches, including semantic segmentation, frequency-domain analysis, and hybrid deep learning models (Arini, Bahaweres, & Al Haq, 2022; Khalil et al., 2021; Kohli & Gupta, 2021; Nirkin et al., 2020; Wodajo et al., 2023). Despite promising results, most existing works either rely solely on deep semantic features or utilize traditional texture descriptors without systematically evaluating their integration with widely used transfer learning architectures.

To the best of our knowledge, this study is the first to systematically evaluate the effectiveness of Local Ternary Pattern (LTP)-based texture feature enhancement integrated with VGG16 and ResNet50 transfer learning architectures for deepfake video classification on the Celeb-DF (v2) dataset using a balanced sampling strategy. The objective of this research is to compare the performance of these architectures and analyze the contribution of LTP in improving the detection of fine-grained facial manipulation artifacts.

METHOD

The methodological framework of this study encompasses several main processes, namely data collection, data preprocessing, model training, evaluation, and prediction. In the data collection stage, a video dataset from Celeb-DF (V2), consisting of both real facial videos and deepfake manipulations, was gathered. Next, in the preprocessing stage, frame extraction, facial feature extraction, size normalization, and the application of the LTP method were performed. The processed data were then divided into three subsets—training, validation, and testing—to ensure that the trained models not only memorized patterns from the training data but also generalized effectively to unseen data. The subsequent stage involved training the models using two transfer learning architectures, VGG16 and ResNet50, which were then evaluated using a confusion matrix and AUC-ROC. Finally, the model prediction process was conducted, wherein both trained models were tested on 20 videos to assess their respective performance. Overall, the workflow of these processes is illustrated in the methodological framework shown in Figure 1.

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 10, Issue 1, Januari 2026
DOI : https://doi.org/10.33395/sinkron.v10i1.15582
e-ISSN : 2541-2019
p-ISSN : 2541-044X

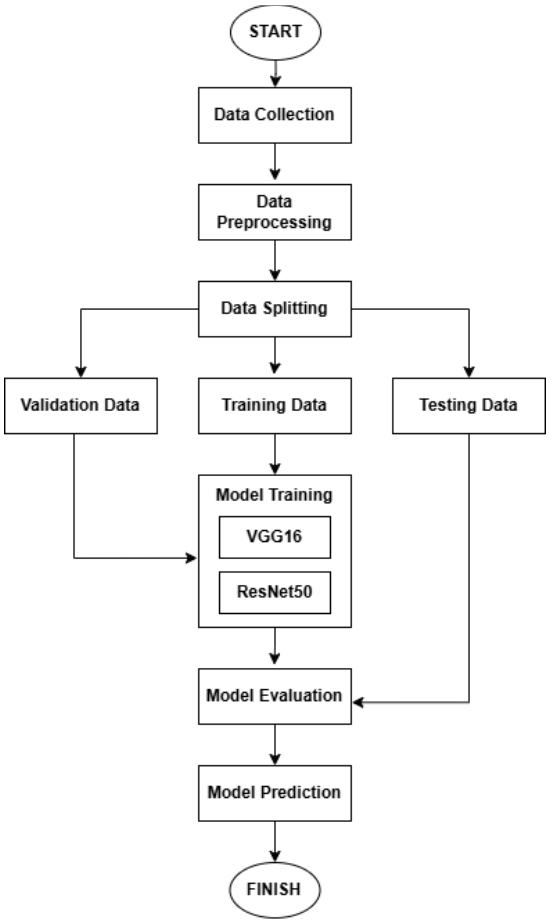

Figure 1 Methodological Framework

Data Collection

The first stage is data collection, in which this study utilizes the Celeb-DF (V2) deepfake dataset. This dataset consists of 590 real videos, 5,369 deepfake videos, and 300 videos sourced from YouTube. The average duration of each video is 13 seconds, with a standard frame rate of 30 frames per second. The real videos were obtained from YouTube and feature interviews with 59 celebrities, varying in gender, age, and ethnic background (Li et al., 2020).

For this study, a subset of 400 videos was selected, comprising 200 real videos and 200 deepfake videos. This equal distribution aims to ensure that the models trained in this study can learn in a balanced manner from both types of data, namely real and deepfake videos.

Data Preprocessing

After data collection, the next step is data preprocessing. This stage is carried out to ensure that the data are in the appropriate format required for model development. The preprocessing steps include frame extraction, facial feature extraction, application of the LTP method, and data augmentation.

The first step is frame extraction, where video data are converted into images. Ten frames are extracted from each video at intervals determined by the ratio between the total number of frames and the number of frames to be selected. This process aims to obtain a visual representation of each video to be used in the subsequent processing stages.

Next, facial feature extraction is performed using the Multi-task Cascaded Convolutional Networks (MTCNN) method. MTCNN is employed to accurately detect faces and remove irrelevant parts of the image(Jin, Li, Pan, Ma, & Lin, 2021). This step is crucial to ensure that the model focuses solely on the facial region, thereby improving accuracy in deepfake classification.

The following stage is the implementation of LTP on the detected facial regions. LTP divides pixel values into three categories (negative, zero, positive), which helps detect finer texture variations compared to LBP (Tan et al., 2011). Similar to LBP, the image is divided into small pixel-based blocks, and each pixel in a block is compared with its central pixel (Tan et al., 2011). LTP uses a threshold system to determine the comparison between the central pixel and its neighboring pixels; in this study, each central pixel has eight neighboring pixels. A threshold

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 10, Issue 1, Januari 2026
DOI : https://doi.org/10.33395/sinkron.v10i1.15582
e-ISSN : 2541-2019
p-ISSN : 2541-044X

value of T = 2 is applied, following prior studies that reported this value to be effective in capturing fine-grained texture variations while maintaining robustness to noise (H. A. Putra, Wihandika, & Rahman, 2022). After the LTP calculation, the resulting values are converted into 1, 0, and -1, which are then transformed into two binary patterns: positive and negative. For the positive pattern, each -1 value is converted to 0, while for the negative pattern, each -1 value is converted to 1, and all other values are converted to 0. For further illustration, the implementation of LTP is shown in Figure 2.

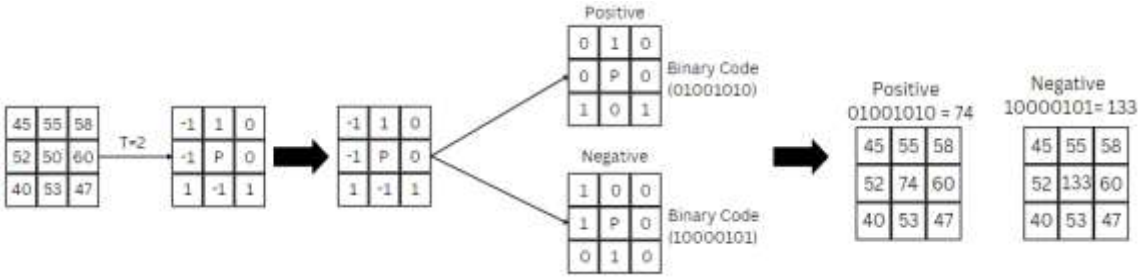

Figure 2 LTP Calculation Implementation

After the LTP calculation, each image produces two outputs: one representing the positive pattern and the other representing the negative pattern.

The final stage in this study is data augmentation, which is performed to increase the variability of the training data, thereby enhancing the model's generalization capability to unseen data. The augmentation process includes transformations such as rotation, horizontal flipping, and image resizing. This step aims to simulate various real-world conditions in deepfake videos, making the model more robust to data variations. Additionally, augmentation serves to efficiently expand the training dataset without increasing the number of original samples, which is particularly important given the limitations of the available dataset.

Data Splitting

After all the data have completed the preprocessing stage, the next step is data splitting. In this study, the data were divided into three subsets: training, testing, and validation, using the “train_test_split” library. From a total of 4,000 images per class, the data were allocated with a ratio of 80% for training, 10% for testing, and 10% for validation.

Model Training

In this stage, a transfer learning approach is employed, which utilizes pre-trained weights from models previously trained on large-scale datasets to improve training efficiency (A. E. Putra, Naufal, & Prasetyo, 2023). This approach was selected because it can achieve high accuracy even with a relatively limited amount of training data (A. E. Putra et al., 2023).

This study uses two popular architectures, VGG16 and ResNet50, as the base models for transfer learning. The VGG16 architecture employs 3×3 filters in each convolutional layer and is known for effectively capturing detailed visual features (Ashani et al., 2025), while ResNet50 utilizes shortcut connections to preserve feature information and address the vanishing gradient problem (Victor Ikechukwu, Murali, Deepu, & Shivamurthy, 2021). By applying these architectures, it is expected that classification performance can be enhanced without requiring a large amount of training data.

In this study, the transfer learning models are frozen (trainable = False), meaning the pre-trained weights will not be updated during training. Only the convolutional layers (include_top=False) are trained, allowing the researchers to add new classifier layers.

Hyperparameters such as the number of dense layers, number of epochs, batch size, and optimizer type have a significant impact on model performance. Various hyperparameter combinations are tested in this study to obtain the most optimal results. The detailed hyperparameter settings used for each architecture are presented in Table 1.

Table 1 Training Model Scenarios

Model	Dense Layer	Epoch	Dropout	Batch Size
VGG16	128,32	50	0,2	16
VGG16	128,32	50	0,2	32
VGG16	128,32	50	0,2	64

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 10, Issue 1, Januari 2026
DOI : https://doi.org/10.33395/sinkron.v10i1.15582
e-ISSN : 2541-2019
p-ISSN : 2541-044X

Model	Dense Layer	Epoch	Dropout	Batch Size
VGG16	512,128	50	0,2	16
VGG16	512,128	50	0,2	32
VGG16	512,128	50	0,2	64
VGG16	256,128,64	50	0,2	16
VGG16	256,128,64	50	0,2	32
VGG16	256,128,64	50	0,2	64
VGG16	256,128,32	50	0,2	16
VGG16	256,128,32	50	0,2	32
VGG16	256,128,32	50	0,2	64
ResNet50	512,128,32	50	0,2	16
ResNet50	512,128,32	50	0,2	32
ResNet50	512,128,32	50	0,2	64
ResNet50	1024,512,128	50	0,2	16
ResNet50	1024,512,128	50	0,2	32
ResNet50	1024,512,128	50	0,2	64
ResNet50	512,128	50	0,2	16
ResNet50	512,128	50	0,2	32
ResNet50	512,128	50	0,2	64
ResNet50	1024,256	50	0,2	16
ResNet50	1024,256	50	0,2	32
ResNet50	1024,256	50	0,2	64

Model Evaluation

In this stage, the performance of the two trained architectures, VGG16 and ResNet50, is evaluated. The evaluation process is conducted using a confusion matrix and ROC-AUC. The confusion matrix is a commonly used method for assessing model performance, as it shows the number of correct and incorrect predictions. This method consists of four main components: True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). From the confusion matrix, evaluation metrics such as accuracy, recall, precision, and F1-Score can be calculated (Kurniadi, Shidiq, & Mulyani, 2025).

Accuracy represents the proportion of the model's predictions that match the actual data and is calculated using Equation 1(Kurniadi et al., 2025).

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

Precision indicates how accurately the model predicts the positive class, defined as the ratio of correctly predicted positive instances to the total number of positive predictions, as shown in Equation 2 (Kurniadi et al., 2025).

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

Recall represents the model's ability to identify all existing positive instances, as described in Equation 3 (Kurniadi et al., 2025).

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

To obtain a more balanced evaluation between precision and recall, the F1-Score is calculated as the harmonic mean of these two metrics, as described in Equation 4 (Kurniadi et al., 2025).

$$F1-Score = \frac{2\ x\ precision\ x\ recall}{precision+recall} \quad (4)$$

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 10, Issue 1, Januari 2026
DOI : https://doi.org/10.33395/sinkron.v10i1.15582
e-ISSN : 2541-2019
p-ISSN : 2541-044X

This study not only utilizes the confusion matrix but also employs the ROC Curve and AUC as additional evaluation methods. The ROC Curve is a two-dimensional plot illustrating the effectiveness of a classification system, where the x-axis represents the false positive rate and the y-axis represents the true positive rate. The AUC (Area Under the Curve) measures the area under the ROC curve, reflecting the model's ability to distinguish between positive and negative classes (Yang & Berdine, 2017).

Model Prediction

Model prediction on new data represents the final stage of this study. The trained models are tested using 10 deepfake videos and 10 real videos that were not previously used in the training or testing datasets. The prediction results are compared with the original labels of each video to assess accuracy, ensuring that the models perform well not only on the training data but also generalize effectively to unseen data.

RESULT

Preprocessing Data

The preprocessing stage represents a crucial initial step in preparing the data according to the model's requirements. This process began with frame extraction from .mp4 videos, where ten frames were extracted from each video at intervals adjusted based on the total number of frames to ensure variability in the samples. A total of 4,000 frames were used for model training.

Next, face detection and extraction were performed using MTCNN to obtain the facial regions, which serve as the primary focus for deepfake classification. After face detection, LTP with a threshold of 2 was applied to reduce noise while extracting essential texture patterns. LTP produced two types of patterns, positive and negative, effectively doubling the dataset to 8,000 images. The results of the LTP process are illustrated in Figure 3.

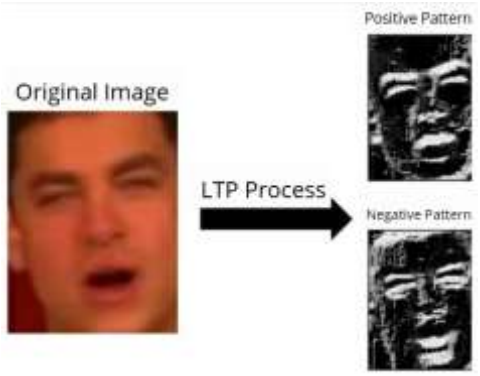

Figure 3 LTP Implementation

The next stage is data augmentation. This process utilizes ImageDataGenerator, a Keras library commonly used for image data augmentation. The parameters applied in the data augmentation stage of this study are presented in Table 2.

Table 2 ImageDataGenerator Parameters

Parameter	Value
Horizontal_flip	True
Vertical_flip	True
Shear_range	True
Target_size	224x224

Data Splitting

At this stage, the dataset was divided into three subsets: training, validation, and testing. Seventy percent of the data were allocated for training to train the model, 10% were used as validation to monitor model performance during training and prevent overfitting, and the remaining 10% were used for testing to evaluate the trained model. The detailed dataset split is presented in Table 3.

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 10, Issue 1, Januari 2026
DOI : https://doi.org/10.33395/sinkron.v10i1.15582
e-ISSN : 2541-2019
p-ISSN : 2541-044X

Table 3 Data Splitting Results

Data	Training	Testing	Evaluation	Total
Fake	3200	400	400	4000
Real	3200	400	400	4000
Total	6400	800	800	8000

Model Training

The next stage is model training, utilizing two transfer learning architectures: VGG16 and ResNet50. Each model was trained using various configurations to ensure optimal performance and prevent overfitting. The configurations included a learning rate of 0.0001, the Adam optimizer, and a binary cross-entropy loss function. The training process was conducted according to the scenarios presented in Table 1. The results of the trained models are shown in Table 4.

Table 4 Model Training Results

Model	Scenario	Batch Size	Train Accuracy	Loss Accuracy	Val Accuracy	Val Loss
VGG16	2	16	0.9357	0.1643	0.8913	0.2636
ResNet-50	4	16	0.9686	0.0885	0.9325	0.1926

Based on the testing results presented in Table 4, the VGG16 model achieved its best performance in Scenario 2 with a batch size of 16, yielding a training accuracy of 0.9357 and a validation accuracy of 0.8913. Meanwhile, the ResNet50 model reached its optimal performance in Scenario 4, also with a batch size of 16, achieving a training accuracy of 0.9686 and a validation accuracy of 0.9325. Overall, ResNet50 demonstrated superior performance compared to VGG16, due to its more complex convolutional layer architecture and its ability to handle higher-level image feature representations.

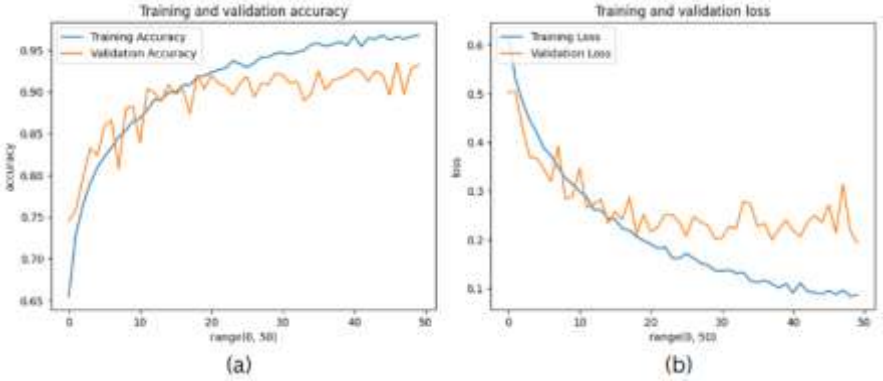

Figure 4 ResNet50 Training Graph

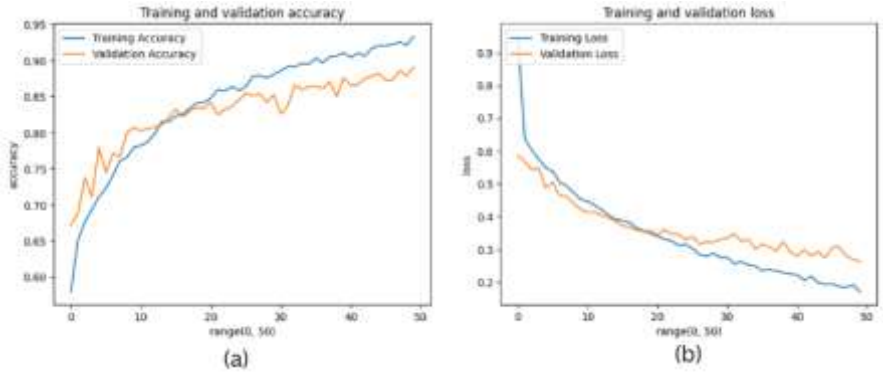

Figure 5 VGG16 Training Graph

Based on Figure 4, the ResNet50 architecture demonstrated a stable increase in training accuracy, while validation accuracy reached a peak before exhibiting fluctuations, indicating potential overfitting. The training

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 10, Issue 1, Januari 2026
DOI : https://doi.org/10.33395/sinkron.v10i1.15582
e-ISSN : 2541-2019
p-ISSN : 2541-044X

loss consistently decreased, whereas the validation loss fluctuated, reflecting the model's limitations in generalizing to the validation data.

Meanwhile, in Figure 5, the VGG16 architecture also showed a stable increase in both training and validation accuracy up to approximately the 20th epoch, after which a divergence between the two emerged, indicating overfitting. Nevertheless, the validation loss continued to decrease significantly, suggesting that the model was still able to adapt to new data during the later stages of training.

Model Evaluation

After the training process was completed, the next stage was the evaluation of the developed models. This evaluation aimed to assess the models' performance in classifying previously unseen data. The data used for evaluation were drawn from the validation subset prepared during the dataset splitting stage. The evaluation was conducted using two methods: ROC-AUC and the confusion matrix. The results of the model testing using ROC-AUC are presented in Table 5.

Table 5 ROC-AUC Evaluation Results

Model	Scenario	Batch Size	AUC-ROC
VGG16	2	16	0.95
ResNet50	1	16	0.98

Based on the ROC-AUC evaluation of the developed models, it can be observed that the ResNet50 model outperformed the VGG16 model overall. The highest AUC-ROC value for the VGG16 model was obtained in Scenario 2 with a batch size of 16, achieving 0.95, while the highest AUC-ROC value for the ResNet50 model was achieved in Scenarios 1 and 3, also with a batch size of 16, reaching 0.98.

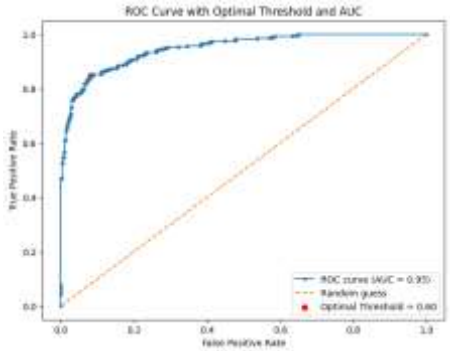

Figure 6 VGG16 ROC-AUC Curve

Figure 6 illustrates the ROC-AUC curve for the VGG16 model, showing the relationship between the false positive rate (FPR) and true positive rate (TPR) across various thresholds. The red point on the graph indicates the optimal threshold of 0.60, which provides the best balance between sensitivity (TPR) and specificity (1 − FPR). The diagonal orange line represents a random guess as a reference baseline.

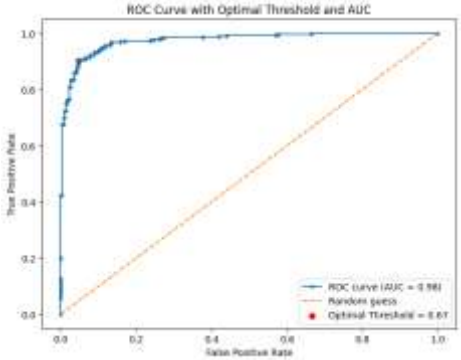

Figure 7 ResNet50 ROC-AUC Curve

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 10, Issue 1, Januari 2026
DOI : https://doi.org/10.33395/sinkron.v10i1.15582
e-ISSN : 2541-2019
p-ISSN : 2541-044X

Figure 7 shows the ROC-AUC curve with an optimal threshold value of 0.67. The ROC-AUC value of ResNet50 is higher than that of VGG16, indicating that, based on the ROC-AUC evaluation, the ResNet50 model outperforms VGG16 in generalizing to new data.

After the model evaluation using ROC-AUC, the resulting TPR and FPR values are used to determine the most optimal threshold for evaluating the model with a confusion matrix. The optimal threshold obtained from ROC-AUC is applied to each scenario for both the ResNet50 and VGG16 models.

Table 6 Confusion Matrix Evaluation Results

Model	Best Scenario	Batch Size	Accuracy	Recall	Precision	F1-Score
ResNet50	Scenario 1	16	0.93	0.93	0.93	0.93
VGG16	Scenario 2	16	0.88	0.88	0.88	0.88

Based on Table 6, the ResNet50 model achieved its best performance in Scenario 1 with a batch size of 16, attaining an accuracy, precision, recall, and F1-score of 0.93. Meanwhile, the VGG16 model achieved its best results in Scenario 2 with a batch size of 16, with an accuracy, precision, recall, and F1-score of 0.88.

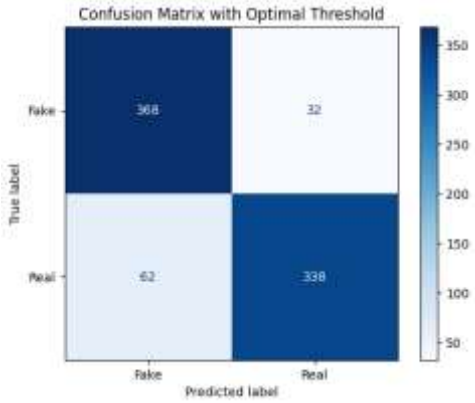

Figure 8 VGG16 Confusion Matrix Results

Based on the results shown in Figure 8, the VGG16 model correctly classified 368 “Fake” samples and 338 “Real” samples, while 32 Fake samples were incorrectly predicted as Real and 62 Real samples were incorrectly predicted as Fake. In terms of class-wise performance, the Fake class achieved a precision of 0.86 and a recall of 0.92, indicating that most manipulated samples were successfully detected, although a small portion of real samples was misclassified as Fake. Meanwhile, the Real class obtained a higher precision of 0.91 but a lower recall of 0.84, suggesting that while predictions labeled as Real were generally reliable, the model showed a tendency to incorrectly classify some genuine samples as Fake. Overall, these results demonstrate that the VGG16 model exhibits strong sensitivity toward deepfake detection, which is desirable for forensic applications, albeit at the cost of slightly reduced recall for real samples.

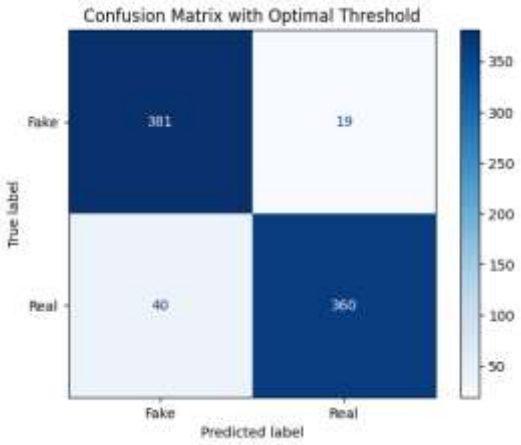

Figure 9 ResNet50 Confusion Matrix Results

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 10, Issue 1, Januari 2026
DOI : https://doi.org/10.33395/sinkron.v10i1.15582
e-ISSN : 2541-2019
p-ISSN : 2541-044X

Meanwhile, Figure 9 shows that the ResNet50 model correctly classified 381 “Fake” samples and 360 “Real” samples, while 19 Fake samples were incorrectly predicted as Real and 40 Real samples were incorrectly predicted as Fake. In terms of class-wise performance, the Fake class achieved a precision of 0.90 and a recall of 0.95, indicating that the majority of manipulated videos were accurately detected with only a small number of false negatives. The Real class obtained a precision of 0.95 and a recall of 0.90, reflecting a strong ability to correctly identify genuine videos while minimizing false alarms. Compared to VGG16, ResNet50 demonstrates more balanced precision and recall across both classes, suggesting improved robustness in distinguishing real and manipulated facial textures.

Although both models achieve strong classification performance, several misclassifications can still be observed. These errors predominantly occur in frames affected by low illumination, heavy video compression, and motion blur, which obscure fine-grained facial texture details. In such conditions, the effectiveness of LTP in capturing discriminative micro-texture patterns is reduced, leading to incorrect predictions. Additionally, extreme head poses and partial facial occlusions further limit the visibility of key facial regions, such as the eyes and mouth, where manipulation artifacts are typically present. Compared to VGG16, ResNet50 shows fewer misclassifications, indicating better robustness to degraded visual conditions, likely due to its deeper architecture and residual connections. Nevertheless, both models remain sensitive to severe visual distortions, highlighting directions for future improvement.

Model Prediction

The final stage is video prediction, in which the best-performing model from each architecture is tested on video data. The prepared dataset consists of 10 real videos and 10 fake videos from the Celeb-DF V2 dataset that were not used during model development. The video data undergo several preprocessing steps to match the requirements of the trained models. After preprocessing, each frame extracted from the videos is predicted individually. The final classification of a video is determined based on the dominant class or the majority of frames within that video.

Table 7 Model Predictions on Video Data

No	Video Name	True Label	VGG16 Prediction	ResNet50 Prediction
6	id22_0004.mp4	Real	Fake	Real
3	id23_0001.mp4	Real	Fake	Real

Based on the testing results presented in Table 7, the VGG16 model misclassified two Real-labeled videos as Fake, while all Fake videos were correctly detected. Analysis of the misclassified frames shows that these errors are mainly caused by challenging visual conditions, including uneven lighting, strong facial shadows, and compression artifacts that introduce artificial texture patterns. Such conditions can mimic manipulation cues, leading VGG16 to incorrectly label genuine videos as Fake. In contrast, ResNet50 exhibits better robustness under similar conditions, indicating stronger generalization capability in deepfake detection.

DISCUSSIONS

The experimental results indicate that ResNet50 consistently outperforms VGG16 for deepfake video classification when combined with LTP-based texture preprocessing. ResNet50 achieved stronger overall performance, with accuracy, precision, recall, and F1-score reaching 0.93, while VGG16 obtained 0.88. At the class level, ResNet50 shows more balanced results, where for the Fake class it achieved precision 0.90, recall 0.95, and F1-score 0.93, and for the Real class it achieved precision 0.95, recall 0.90, and F1-score 0.92. In contrast, VGG16 showed less balanced performance between classes, particularly with lower recall on the Real class (0.84), indicating a higher tendency to misclassify genuine videos as fake. These findings suggest that the residual connections in ResNet50 enable more effective utilization of LTP-extracted texture cues, supporting deeper feature learning while maintaining important information.

The AUC-ROC results further support this observation, where ResNet50 achieved a higher value of 0.98 compared to 0.95 for VGG16, indicating stronger discriminative capability and more stable generalization. From a threat model perspective, LTP-based preprocessing enhances sensitivity to spatial manipulation artifacts commonly produced by deepfake synthesis, making the approach relevant for real-world applications such as content moderation, digital forensics, and identity verification systems. Compared to many state-of-the-art deepfake detection approaches that often rely on large-scale data, complex temporal modeling, or transformer-based architectures, these results highlight that texture-based enhancement combined with transfer learning can still provide competitive performance in limited-data conditions.

*name of corresponding author

However, this study is limited by the use of a single dataset and restricted diversity of manipulation patterns, which may affect robustness against unseen deepfake techniques. Future work may focus on improving robustness against emerging deepfake generation methods through adaptive or multi-scale texture representations, without increasing model complexity.

CONCLUSION

This study evaluated the performance of VGG16 and ResNet50 transfer learning architectures for deepfake video classification using the Celeb-DF (V2) dataset with Local Ternary Pattern (LTP)–based texture preprocessing. The contribution of this work lies in the implementation and evaluation of LTP-based texture representation within both architectures under limited data conditions. Experimental results show that ResNet50 outperformed VGG16, achieving a higher test accuracy (0.93) and lower validation loss (0.2228), while VGG16 obtained a test accuracy of 0.88 with a validation loss of 0.2636. At the video level, ResNet50 correctly classified all evaluated samples, whereas VGG16 misclassified a small number of real videos.

The main limitation of this study is the use of a relatively limited amount of data from a single dataset, which may restrict the diversity of deepfake patterns. However, the proposed approach still achieved competitive performance, indicating its potential applicability in scenarios where large-scale labeled datasets are unavailable. Future work should consider multi-dataset evaluation, temporal feature modeling, and the integration of LTP with more advanced architectures such as transformer-based or hybrid CNN–transformer models to improve robustness.

REFERENCES

Albashish, D., Al-Sayyed, R., Abdullah, A., Ryalat, M. H., & Ahmad Almansour, N. (2021). Deep CNN Model based on VGG16 for Breast Cancer Classification. *2021 International Conference on Information Technology, ICIT 2021 - Proceedings*, (July), 805–810. https://doi.org/10.1109/ICIT52682.2021.9491631

Arini, A., Bahaweres, R. B., & Al Haq, J. (2022a). Quick Classification of Xception And Resnet-50 Models on Deepfake Video Using Local Binary Pattern. *2021 International Seminar on Machine Learning, Optimization, and Data Science, ISMODE 2021*, (January), 254–259. https://doi.org/10.1109/ISMODE53584.2022.9742852

Arini, A., Bahaweres, R. B., & Al Haq, J. (2022b). Quick Classification of Xception And Resnet-50 Models on Deepfake Video Using Local Binary Pattern. *2021 International Seminar on Machine Learning, Optimization, and Data Science, ISMODE 2021*, (June), 254–259. https://doi.org/10.1109/ISMODE53584.2022.9742852

Ashani, Z. N., Syafidza, I., Ilias, C., Ng, K. Y., Kamel, M. R., Jarno, A. D., & Zamri, N. Z. (2025). *Comparative Analysis of Deepfake Image Detection Method Using VGG16 , VGG19 and ResNet50*. *1*(1), 16–28.

Dolhansky, B., Bitton, J., Pflaum, B., Lu, J., Howes, R., Wang, M., & Ferrer, C. C. (2020). *The DeepFake Detection Challenge (DFDC) Dataset*. Retrieved from http://arxiv.org/abs/2006.07397

Fathur Rozi, M. I., Adiwijaya, N. O., & Swasono, D. I. (2023). Identifikasi Kinerja Arsitektur Transfer Learning Vgg16, Resnet-50, Dan Inception-V3 Dalam Pengklasifikasian Citra Penyakit Daun Tomat. *Jurnal Riset Rekayasa Elektro*, *5*(2), 145. https://doi.org/10.30595/jrre.v5i2.18050

Jin, R., Li, H., Pan, J., Ma, W., & Lin, J. (2021). *Face Recognition Based on MTCNN and FaceNet*. Retrieved from www.aaai.org

Kamal, & Ez-zahraouy, H. (2023). A comparison between the VGG16 , VGG19 and ResNet50 architecture frameworks for classi cation of normal and CLAHE processed medical images. *Research Square*, 0–16.

Khalil, S. S., Youssef, S. M., & Saleh, S. N. (2021). Article icaps-dfake: An integrated capsule-based model for deepfake image and video detection. *Future Internet*, *13*(4). https://doi.org/10.3390/fi13040093

Kohli, A., & Gupta, A. (2021). Detecting DeepFake, FaceSwap and Face2Face facial forgeries using frequency CNN. *Multimedia Tools and Applications*, *80*(12), 18461–18478. https://doi.org/10.1007/s11042-020-10420-8

Kurniadi, D., Shidiq, R. M., & Mulyani, A. (2025). *Comparison of Optimizer Use in White Blood Cell Classification Employing CNN*. *14*(February), 77–86.

L, S. K. B., V, S. N., & M, S. K. (2022). *Enhanced Local Ternary Pattern method for Face Recognition*. *66*(2), 139–143. https://doi.org/10.37398/JSR.2022.660218

Li, Y., Yang, X., Sun, P., Qi, H., & Lyu, S. (2020). Celeb-DF: A Large-Scale Challenging Dataset for DeepFake Forensics. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 3204–3213. https://doi.org/10.1109/CVPR42600.2020.00327

Malik, A., Kuribayashi, M., Abdullahi, S. M., & Khan, A. N. (2022). DeepFake Detection for Human Face Images and Videos: A Survey. *IEEE Access*, *10*(January), 18757–18775. https://doi.org/10.1109/ACCESS.2022.3151186

Maulana, G. (2023). Beredar Video Jokowi Fasih Mandarin, Kominfo: Editan AI Menyesatkan! Retrieved from detikNews website: https://news.detik.com/berita/d-7003320/beredar-video-jokowi-fasih-mandarin-

*name of corresponding author

e-ISSN : 2541-2019
p-ISSN : 2541-044X

kominfo-editan-ai-menyesatkan

Nirkin, Y., Wolf, L., Keller, Y., & Hassner, T. (2020). *DeepFake Detection Based on the Discrepancy Between the Face and its Context*. 1–10. Retrieved from http://arxiv.org/abs/2008.12262

Putra, A. E., Naufal, M. F., & Prasetyo, V. R. (2023). *Klasifikasi Jenis Rempah Menggunakan Convolutional Neural Network dan Transfer Learning*. *9*(1), 12–18.

Putra, H. A., Wihandika, R. C., & Rahman, M. A. (2022). *Ekstraksi Ciri Tekstur Local Ternary Pattern dan Klasifikasi Naïve Bayes untuk Deteksi Penggunaan Masker Wajah*. *6*(8), 4065–4071.

Rahayu, R. A. S., & Santoso, H. (2023). Analysis of Fake Face Images: Detecting the Authenticity of Manipulated Images Using Variational Autoencoder Methods and Deep Neural Network Forensics. *Sibatik Journal | Volume*, *2*(9), 2701–2726. Retrieved from https://publish.ojs-indonesia.com/index.php/SIBATIK

Rana, M. S., Nobi, M. N., Murali, B., & Sung, A. H. (2022). Deepfake Detection: A Systematic Literature Review. *IEEE Access*, *10*, 25494–25513. https://doi.org/10.1109/ACCESS.2022.3154404

Rossler, A., Cozzolino, D., Verdoliva, L., Riess, C., Thies, J., & Niessner, M. (2019). FaceForensics++: Learning to detect manipulated facial images. *Proceedings of the IEEE International Conference on Computer Vision*, *2019-Octob*, 1–11. https://doi.org/10.1109/ICCV.2019.00009

Tan, X., Triggs, W., Tan, X., Triggs, W., Local, E., Feature, T., … Triggs, B. (2011). *Enhanced Local Texture Feature Sets for Face Recognition under Difficult Lighting Conditions*.

Victor Ikechukwu, A., Murali, S., Deepu, R., & Shivamurthy, R. C. (2021). ResNet-50 vs VGG-19 vs training from scratch: A comparative analysis of the segmentation and classification of Pneumonia from chest X-ray images. *Global Transitions Proceedings*, *2*(2), 375–381. https://doi.org/10.1016/j.gltp.2021.08.027

Wodajo, D., Atnafu, S., & Akhtar, Z. (2023). *Deepfake Video Detection Using Generative Convolutional Vision Transformer*. (Dl). Retrieved from http://arxiv.org/abs/2307.07036

Xu, B., Liu, J., Liang, J., Lu, W., & Zhang, Y. (2021). DeepFake Videos Detection Based on Texture Features. *Computers, Materials and Continua*, *68*(1), 1375–1388. https://doi.org/10.32604/cmc.2021.016760

Yang, S., & Berdine, G. (2017). *The receiver operating characteristic (ROC) curve*. *5*(19), 34–36. https://doi.org/10.12746/swrccc.v5i19.391

Zain, R. A. (2024). Perusahaan Raksasa Inggris Jadi Korban Penipuan Deepfake, Kerugian Tembus Rp 400 Miliar. Retrieved from liputan6 website: https://www.liputan6.com/tekno/read/5598161/perusahaan-raksasa-inggris-jadi-korban-penipuan-deepfake-kerugian-tembus-rp-400-miliar

*name of corresponding author