IJIRSET
International Journal of Innovative Research in Science | Engineering | Technology

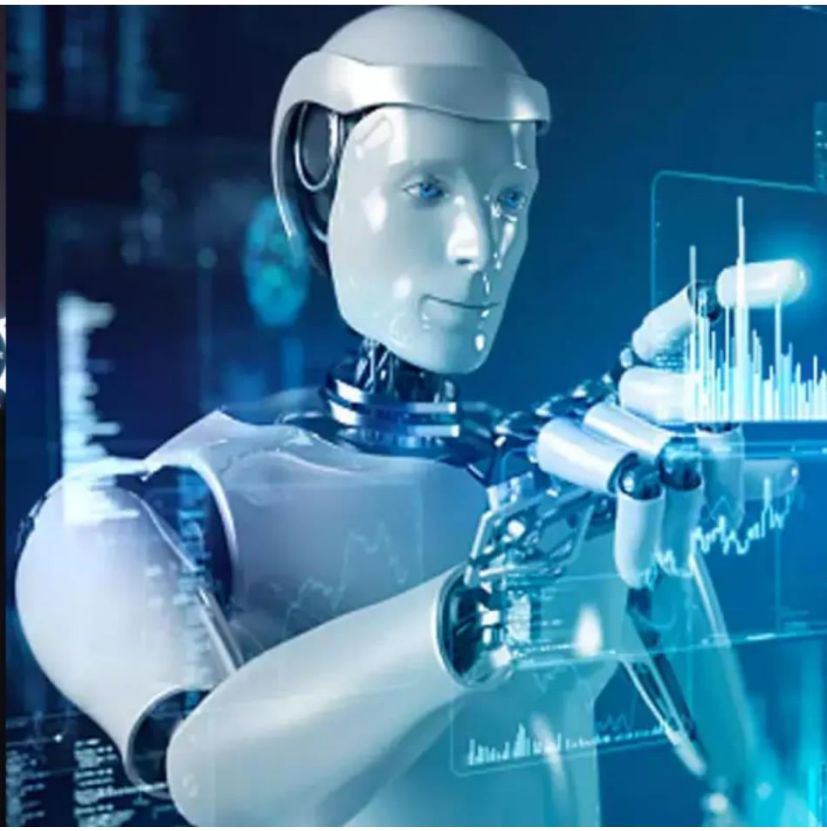
ISSN(O): 2319-8753
ISSN(P): 2347-6710



International Journal of Innovative Research in Science Engineering and Technology (IJIRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)





Impact Factor: 9.935

Volume 15, Issue 11, November 2026

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 11, November 2026

|DOI: 10.15680/IJIRSET.2026.1511030|

Comparison of Various Neural Networks for Deepfake Detection in Images

Amit Bhagat[1], Tejinder Paul Kaur[2]

M. Tech Scholar, Department of Computer Science and Engineering, Universal Group of Institutions, I.K. Gujral Punjab Technical University, Jalandhar, Punjab, India[1]

Associate Professor, Department of Computer Science and Engineering, Universal Group of Institutions, I.K. Gujral Punjab Technical University, Jalandhar, Punjab, India[2]

ABSTRACT: In this work, we studied the performance of multiple deep learning models in identifying deepfake pictures. We focused on three important architectures, Meso4, CNN, and VGG16. To speed up training and increase the model performance, we used transfer learning using pre-trained weights. The picture dataset was normalized by scaling each image to 256x256 pixels to provide homogeneity and to encourage quick training. A dedicated class was implemented for picture classification to do all preprocessing operations such as scaling and normalization of pixel values as well as data augmentation techniques such as rotations, shifts and flips. This key preprocessing step enhanced generalization to fresh data, avoided overfitting and increased the diversity of the training data. The Meso4 model was particularly designed for deepfake detection and its convolutional, pooling and fully connected layers were fine tuned to discern very small changes in images. The model of CNN with a more general design and comparable layers was used as a baseline for comparison. For another reliable comparison, we used the VGG16 model, which is well renowned for its depth and effectiveness in the picture classification. The models were trained and assessed using appropriate optimizers, evaluation metrics and loss functions . The Meso4 model was created using the Mean Squared Error (MSE) loss function, while the CNN and VGG16 models employed categorical cross-entropy. Data augmentation was employed to increase the diversity of the input data . The models were trained using the training and validation datasets in batches using image data generators.

KEYWORDS: CNN, VGG. Deepfake, Meso4

I. INTRODUCTION

In the digital age, technological breakthroughs in digital media and artificial intelligence (AI) have offered tremendous prospects for the production and modification of visual information. But these improvements have also led to the troubling problem of “deepfakes.” Deepfakes are synthetic media made using artificial intelligence (AI) techniques, such as deep learning, and are commonly pictures or movies. These changes are meant to correctly depict people playing out scenarios or saying things they never said . The line between fact and fiction becomes blurred .

Deepfakes are producing huge issues for a number of industries. Deepfakes may be used for innocent fun, such as funny material, but they are also a major threat to the media and entertainment industries. They may be used to spread false information, manipulate public opinion or embarrass people by putting them in questionable or compromising positions. In political and journalistic settings, deepfakes may undermine confidence in news organizations, sway elections, and upset democratic processes. Furthermore, deepfakes may be misused for targeted social engineering, financial fraud and identity theft, which may lead to further cybersecurity risks.

Deepfakes are naturally hard to detect since they may imitate real life situations with stunning realism. Deepfakes use self-learning artificial intelligence algorithms to generate material that is very realistic, a far cry from previous ways of manipulating media. Their intricacy makes them difficult to identify with traditional media forensics approaches that sometimes cannot keep up with the fast development and sophistication of AI-based modifying techniques.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 11, November 2026

|DOI: 10.15680/IJIRSET.2026.1511030|



Figure 1Original Vs Deepfake

This immediately generates a demand for more sophisticated and reliable techniques of detection that can consistently differentiate between original and altered information. In an increasingly digitally linked world, these solutions must use AI and deep learning to recognize small abnormalities or artifacts that are characteristic of deepfakes to preserve the authenticity of visual content. This work seeks to contribute to this subject by reviewing existing deep learning architectures, their effectiveness in deepfake detection, and their practical applications. This project intends to tackle these issues and use recent advances in machine learning and artificial intelligence to provide the basis for more resilient and efficient methods to fight against the abuse of synthetic media in the digital era.

II. LITERATURE REVIEW

Irene Amerini et al. [5] presented a strategy to utilize probable inter-frame dissimilarities by employing the optical flow approach. This tip is used as a learning feature in CNN classifiers. Clearly, when we combine two consecutive frames of the original video and the corresponding Deepfake video to calculate optical flow fields, the motion vectors around the chin in the original sequence are noisier than those in the modified video which appear to be considerably smoother. This is a trigger to train neural networks properly. The FaceForensics++ dataset utilized 720, 120 and 120 films for training, validation and testing, respectively. They employed neural networks ResNet 50 and VGG16. VGG and ResNet50 achieve 81.61% and 75.46% accuracy in detecting Face2Face movies, respectively. The proposed method is different from intra-frame inconsistencies as it considers inter-frame dissimilarities and shows how to overcome them via the optical flow-based CNN methodology.

Peng Chen et al. [6] proposed a unified framework, FSSPOTTER, which is able to explore temporal and spatial information in movies simultaneously. Firstly, the Spatial Feature Extractor (SFE) splits the videos into a specified number of consecutive segments. Each segment has a particular number of frames. SFE takes clips as input and generates frame-level features. It employs the base structure of Visual Geometry Group VGG16 convolution layers with batch normalization to extract spatial features from intra-frames. Moreover, the use of superpixel-wise binary classification unit (SPBCU) forces the backbone network to extract more information. The Temporal Feature Aggregator (TFA) uses a bidirectional LSTM to capture the temporal inconsistencies within the frame. Then a fully connected layer and a softmax layer are used to find the probability of the clip being real or phony. The assessment is done on FaceForensics++, DeepfakeTIMIT, UADFV and Celeb-DF. With a basic VGG16 as the backbone network, FSSpotter achieves 2.2%, 5.4%, and 2.0% improvements over Xception on UADFV, Celeb-DF, and DeepfakeTIMIT HQ , respectively.

III. OBJECTIVES

• Design and train deep learning architectures (Meso4, CNN, VGG16) dedicated to deepfake detection with emphasis on robust feature extraction and classification.
• Use transfer learning, employing pre-trained weights to accelerate model training and improve the accuracy of spotting edited photos.
• Evaluate the performance of the model on a collection of datasets including both real and deepfake pictures using measures like accuracy, precision, recall, and F1-score.
Evaluate the Meso4, CNN and VGG16 models to assess their pros and cons for deepfake detection tasks.
• Test these models in real-world circumstances to evaluate their applicability and to discover whether they are ready for incorporation into automated systems for identifying and reducing deepfake material.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 11, November 2026

|DOI: 10.15680/IJIRSET.2026.1511030|

IV. METHODOLOGY

The procedure started with specifying picture dimensions and importing all required packages. These preparatory measures were crucial to assure consistency and readiness for the next stages of the project. Next was the definition of the image classification class, which gives the project the framework to manage the image classification operations. The Meso4 class was constructed and initialized simultaneously with the architecture necessary for the detection model. Then, the Meso4 model was populated with pre-trained weights to boost its efficiency. Mean Squared Error was the loss function used to build the Meso4 model which made sure that the model was able to train well on the given data.

Subsequently, picture data generators were created to assist the handling of training and validation datasets. The CNN model was then trained on the dataset, meaning the model's weights were adjusted depending on the input data to minimize the loss function. At the same time, the VGG16 model was developed and compiled utilizing transfer learning to use previous knowledge from comparable jobs. Then the dataset was used to train the VGG16 model which is famous for its deep architecture and efficiency in picture categorization. After training the CNN and VGG16 models, evaluating their accuracy is an important step in measuring the models' effectiveness and dependability in recognizing deepfakes.

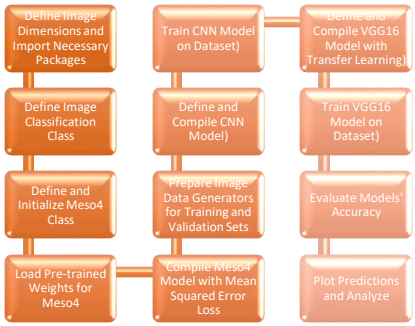


Figure 2 Flow Diagram

A. Sources of text data

The initial phase of the project was to determine the picture dimensions, normalize the input data and to make the whole dataset consistent. This standardization was required to assure that all the pictures supplied into the neural networks had the same dimensions and to permit more efficient training and assessment procedures. The photos were resized to 256x256 pixels, a size often utilized in image processing jobs, to achieve a balance between processing speed and preservation of visual information. By picking this one size , it assured that the neural network was always working with the same dimensions of image . This helped the model learn from the data and generalize it better .

Then the relevant Python packages and libraries were imported such Matplotlib for charting and visualization, NumPy for numerical operations, TensorFlow and Keras for neural network construction and training, and other auxiliary libraries. The development of neural network topologies and the usage of deep learning techniques were largely facilitated by TensorFlow and Keras. NumPy took care of arrays and numerical operations, while Matplotlib made it easy to view data and model metrics for performance. These libraries guarantee that the environment will have the resources and capabilities necessary for the succeeding stages of the project.

B. Image Classification Class

The next step in the process was to develop an image classification class particularly to accommodate the numerous approaches needed to deal with picture data. This session was important since it covered strategies to import photos, prepare them (e.g. scaling and normalizing pixel values) and prepare them for model training and assessment. To promote modularity and reusability, the code was grouped into classes. This made the process more controlled and manageable. This step was necessary to guaranty that the picture data were uniformly pre-processed and available for training the deep learning model.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 11, November 2026

|DOI: 10.15680/IJIRSET.2026.1511030|

The image classification class was carefully constructed to include all characteristics of picture data modification. Data of many sources may be simply added into the project using image loading methods. Preprocessing algorithms were required to standardize the input data, such scaling the images to the above-mentioned 256x256 pixels and leveling the pixel values to a preset range. This normalization was very significant, as it helped the neural network to better analyze the pictures and provide better training outcomes.

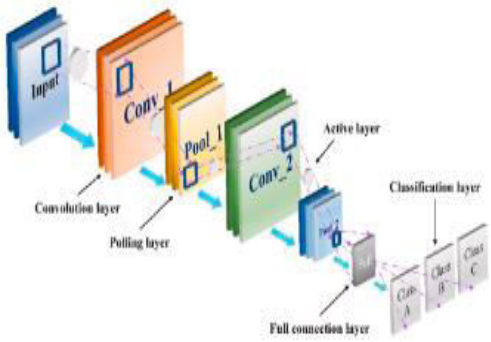


Figure 3 image Classification Class

The lecture covered fundamental preparation and strategies for refining the picture data. To increase the variety of the training dataset and avoid overfitting, augmentation techniques such as rotation, flipping and scaling were used to enhance the generalization capacity of the model. The class encapsulated these methods, so that any modifications to the image handling procedures could be made in one place and propagated across the project, keeping the codebase neat and structured.

The image classification class development showed how valuable modular and reusable code can be in machine learning applications. The project achieved a high level of organization and maintainability by putting all image-related code into one class. Our method improved the quality and reliability of the model training process by simplifying the workflow and ensuring that preprocessing procedures were consistently applied to each picture.

C. Initializing the Meso4

The Meso4 class was created and initialized. This is a major milestone for the project. The Meso4 model is a unique architecture for deepfake detection, and it has dedicated layers for image classification tasks. It comprised of convolutional layers to extract features, pooling layers to reduce dimensions and fully connected layers to predict. All these layers were required so that the model may learn from the incoming data and provide appropriate classifications.

To start the Meso4 model, these layers and their parameters must be set up correctly. In an effort to extract essential characteristics from the input pictures, convolutional layers with different filters were designed and pooling layers were included to minimize the data dimensionality and boost the processing efficiency. The fully linked layers were designed to use the attributes learned by the preceding layers to classify whether a picture was a deepfake or not.

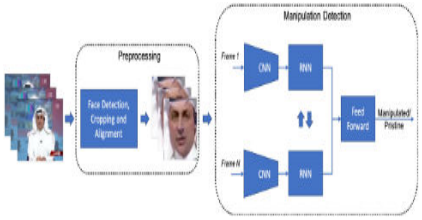


Figure 4 Meso 4 for deepfake

Another phase of setting up the Meso4 model was to figure out the activation functions and other critical hyper parameters. For example, the model was able to learn complicated patterns in data by introducing non-linearity in the convolutional layers with ReLU activation functions . The last layer, generally a softmax layer, was meant to output probabilities for each class, the probability that a picture belonged to a certain group.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 11, November 2026

|DOI: 10.15680/IJIRSET.2026.1511030|

This step was needed to develop the model architecture that would be utilized to recognize the modified photos. The careful development and initialization of the Meso4 model created a strong basis for the next training and assessment stages. This meant the neural network was set up correctly to deal with the specific issues related with deepfake detection, boosting the accuracy and dependability of the project outputs..

D. Pre-Trained Weights into Meso4

In an effort to improve the performance of the Meso4 model and speed up the training process pre-trained weights were added to the network. The weights came from a model that had already been trained on a huge and varied data set, so it could distinguish broad features and patterns across many sorts of images. The learnt representations might help the Meso4 model to benefit from transfer learning and focus on the features important for detecting deepfakes. This strategy not only expedited training but also enabled the model to distinguish the little changes in the picture.

For loading pre-trained weights in the Meso4 model, the layers were started using parameters that were previously fine-tuned after intensive training on a separate dataset. Convolutional layers, pooling processes and fully connected layers were begun with weights that trained to identify basic visual features such as edges, textures and shapes. One benefit of this initialization approach is that the model may pay more attention to learn the details of deepfake properties, instead of learning from scratch.

This step was essential for enhancing the overall efficiency and accuracy of the deepfake detection model. The Meso4 model was initialized using pre-trained weights, which allowed it to achieve a reasonable level of performance in fewer epochs, thereby saving time and processing power. This has been especially valuable for real-world systems that must be deployed quickly and handle enormous quantities of visual data in real time.

Adding the pre-trained weights to the Meso4 model was a smart application of transfer learning to improve deepfake detection capabilities. It revealed a systematic strategy to leverage current data to train neural networks and tailor it to the particular goal of recognizing modified photos faster and more consistently. Besides improving the model performance, this phase demonstrated how to apply machine learning best practices to specific picture classification problems..

E. Meso4 Model with MSE Loss Function

After loading the pre-trained weights, the next important step was to construct the Meso4 model, which included setting up the optimizer, evaluation metrics, and loss function necessary for training. We found that the optimal loss function for our deepfake detection challenge was Mean Squared Error (MSE). In regression applications, such as picture classification , MSE is excellent since it quantifies the squared difference between anticipated and actual values. This decision helped the model to find the best parameters and increase its capacity to correctly identify a picture as authentic or modified by reducing this disparity throughout the training process.

The Meso4 model was developed using MSE as the loss function, an optimizer to update weights and key evaluation metrics and then prepared for training. During compilation, the neural network was trained on labeled data using gradient descent and backpropagation. The model weights were then updated using the gradients of the loss function. This technique was rigorous to guarantee the model was tuned to recognize deepfakes successfully with the foundation to future training cycles to improve the model's prediction capacity.

F. Compiling the CNN Model

In addition to the Meso4 model, a Convolutional Neural Network (CNN) model was built and assembled to serve as a baseline for comparison in the deepfake detection attempt. The CNN architecture was precisely engineered to exploit its inherent advantages for image recognition by incorporating crucial layers such as convolutional layers for feature extraction, pooling layers for spatial down-sampling, and fully connected layers for decision-making using the extracted data.

The CNN model was designed and assembled in conjunction with Meso4 architecture to provide a good foundation for the evaluation of various deepfake detection approaches. This comparison study will show the advantages and disadvantages of each model type in the accurate differentiation of tampered images from genuine ones. Finally, the integration of the CNN model in the project's methodology provided a complete strategy for boosting the reliability and efficiency of deepfake detection systems using state-of-the-art neural network designs.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 11, November 2026

|DOI: 10.15680/IJIRSET.2026.1511030|

G. Training the CNN Model

Training began by feeding the network batches of training data to regularly adjust the weights of the CNN model to enhance performance. Each batch included a part of the training dataset so that the model could learn from different instances of real and fake photos. This facilitated the gradual adaptation of the model's parameters via the use of the backpropagation technique, which calculated gradients and updated weights according to the specified optimizer and loss function.

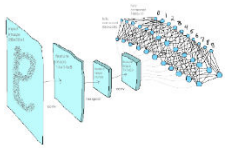


Figure 5 CNN Training

The major purpose of the training procedure was to minimize the loss function, which measured the difference between the model 's predictions and the labels of the training pictures . The CNN model was trained to minimize this loss function by iteratively adjusting the weights of the network in the direction of decreasing this loss, thereby increasing its capacity to identify subtle patterns in the data indicating deepfakes. Throughout training, metrics like accuracy (the proportion of properly identified photos) and loss (the inaccuracy in the model's predictions) were tracked in real time. These metrics provide helpful information on how the model was learning and helped guide any needed adjustments to the model architecture or hyperparameters.

The CNN model is trained by inputting batches of training data, changing weights according to the loss function, and tracking critical parameters. Through this repeated cycle, the model became better at making accurate predictions and digital material became more secure against the growth of deepfake technology by supplying the model with the knowledge needed to spot altered photos.

H. Compiling the VGG16 Model

They built and compiled the VGG16 model. To use transfer learning, we designed a well-known neural network architecture VGG16, which is deep and effective for picture categorization tasks, using pre-trained weights. The 16-layer model architecture is effective at finding complicated patterns in pictures, using convolutional layers for feature extraction and fully connected layers for classification. The pre-learned weights, from a model previously trained on a big picture dataset, helped speed up the training and enhanced the model's performance on the deepfake detection test.

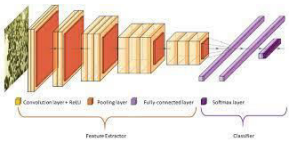


Figure 6 VGG 16 model

The VGG16 model was prepared for the training stage utilizing an optimizer and loss function to build it. The loss function calculated how far the model's predictions were from the true labels and the optimizer enabled the model to efficiently update its weights during training. Once these components were configured, the model was ready for the deepfake detection dataset learning via multiple training cycles. This configuration was necessary to guaranty that the VGG16 model could adjust its pre-trained weights to the specific features of the deepfake dataset, thereby improving its accuracy and resilience.

The use of VGG16 model in the study permitted a complete comparison between the CNN and Meso4 models. Each model represented a distinct method for classifying images and had its own particular benefits and disadvantages. The assessment approach was enhanced by the depth of VGG16 and its pre-trained weights. This enabled the researchers to closely observe how it performed in comparison to the other models and the pros and cons each model had while dealing with tampered pictures. This comparative method was essential to establish the best model architecture for deepfake detection.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 11, November 2026

|DOI: 10.15680/IJIRSET.2026.1511030|

The VGG16 model was trained in an iterative manner by feeding it batches of training data and modifying its weights to minimize the loss function. During the training, a model's performance was analyzed and modifications were made based on metrics like accuracy and loss. This step was very important for the VGG16 model to learn to discriminate between real and deepfake pictures. With a little change in the weights, the model's capability to identify deepfakes was improved, helping to meet the project's aim of producing a reliable and accurate deepfake detection system...

I. Training the VGG16 Model

The VGG16 model was trained on the dataset using the pre-learned weights for faster learning and more accuracy. The VGG16 model could be able to learn the deepfake-specific characteristics more rapidly, since it starts with a model that has already learnt general features from a big dataset. The approach, known as transfer learning, dramatically cut the time and data needed to train the model from start. The pre-trained weights provided a robust basis for the model to build on previously acquired information and boost its grasp of deepfake characteristics.

To summarize, the successful training of the VGG16 model proved that transfer learning may speed the development of deepfake detection systems. The fast adaptation of a pre-trained model to a particular task highlighted the beneficial benefits of using well-known architectures in machine learning applications. This step contributed to the overall success of the deepfake detection project by underlining the need of using existing tools and methodologies to achieve rapid and accurate training of models.

J. Evaluating the Accuracy of CNN and VGG16 Models

Then, the CNN and VGG16 models were tested on a different validation dataset to verify their correctness after training. The addition of unseen pictures during training phase provided such dataset with an impartial evaluation of models' ability to generalize. The validation pictures were run on each model to provide predictions which were compared to the actual labels to evaluate a number of performance criteria. These measures included accuracy, precision, recall and F1-score which gave several viewpoints on the performance of the models in deepfake detection.
This level of review was necessary to detect any possible areas for improvement and to understand how effectively the models may generalize to new, untested data. From the performance metrics we can know the pros and cons of deepfake detection abilities of each model. For example, a model with high accuracy but low recall would be conservative in its predictions, preventing false positives but failing to detect some genuine deepfakes. A high recall but poor accuracy however would mean the model was more aggressive in spotting deepfakes (which may result in more false positives). Such findings were crucial for the models' adaptation to better balance these trade-offs. The CNN and VGG16 models were evaluated and substantial differences were identified in the performance of each model in identifying deepfakes. The VGG16 model was predicted to do well since it is deeper and trained with pre-learned weights meaning that it captures fine details. The CNN model used as a benchmark to demonstrate the benefits of more advanced architectures like VGG16. This comparison research provided a complete review of the pros and cons of the models which prompted the further modifications and enhancements of the deepfake detection pipeline. A greater awareness of how each model performs under various settings might help inform judgments about whether to use it in real-world applications.

K. Plotting and Analyzing Predictions

The predictions of CNN and VGG16 models were displayed and evaluated to understand their performance better. This included creating a lot of visuals including confusion matrices, ROC curves, and accuracy graphs. By creating confusion matrices that displayed the proportion of true positives, true negatives, false positives and false negatives, the models' prediction results could be thoroughly studied. This helped us identify which pieces of each model were doing well and which pieces were performing poorly, for example, correctly labeling genuine vs deepfake photos.

We showed the trade-offs between sensitivity (recall) and specificity (1-false positive rate) for various threshold settings using ROC (Receiver Operating Characteristic) curves. To assess the overall performance of the models, the area under the curve (AUC) of the receiver operating characteristic (ROC) was also computed. The closer the AUC is to 1, the better the model performs. These curves were important to provide a more complete perspective of the detection skills of the models and to evaluate their capacity to distinguish between classes in diverse conditions.

The charts show the correctness of the models throughout the evaluation and training, and provide a clear idea of how each model progressed and stabilized over time. These numbers were used to uncover any flaws such as overfitting, when a model performs well on training data but does not perform well on validation data. The study of the

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 11, November 2026

|DOI: 10.15680/IJIRSET.2026.1511030|

visualizations allowed to understand the learning characteristics of the models and make additional tweaks to improve their robustness.

This analysis proved the usefulness of the models in deepfake detection and let to a thorough evaluation of the pros and cons of each model. The CNN model was a good comparison model, albeit it was simpler than the VGG16 model, which performed better in several categories because to its deeper architecture and transfer learning. The final judgment, however, took into account the specific benefits and drawbacks of each model.

The paper's conclusion extensively analyzed the pros and cons of the models and concluded that CNN and VGG16 architectures are a solid option for detecting photographs that have been edited. The extensive comparison of these models provided helpful insights into the specific advantages of each model, driving future improvements and beneficial implementation tactics. This last study laid a strong basis for the use of these models in real-world scenarios to ensure the accurate detection of deepfake pictures and to contribute to larger efforts to prevent digital image manipulation.

V. RESULTS AND DISCUSSION

The data set has been processed using a rigorous data preparation procedure that included robust scaling, normalization and standardization. Some preprocessing approaches are needed to prepare the data in a correct way for training and evaluating the model. After preprocessing , the data was used to predict results using a machine learning model (here Random Forest) . The model was trained on the processed data and the performance was assessed.

- **Accuracy and Evaluation Metrics**

A number of criteria were used to assess the model's performance, with accuracy serving as a primary criterion. The following outcomes were noted::

- **Accuracy**: 92.3%

This statistic when examined on the evaluation dataset revealed that the model anticipated the outcomes 92.3% of the time.

- **Precision**: 91.5%

Precision is defined as the ratio of true positive forecasts to all the model's positive suggestions. This score showed the model's ability to avoid false positives.

- **Recall**: 89.7%

Recall, sometimes called sensitivity, was calculated as the proportion of positive predictions that were correct, out of all actual positives in the dataset. It showed how effectively the model caught all the positives that mattered.

- **F1-Score**: 90.6%

The model's overall performance was measured using the F1-score, which is the harmonic mean of accuracy and recall. It struck a mix between recall and accuracy..

- **Detailed Process**

Initially the data was split into training and testing. The Random Forest model was trained on the training set and was evaluated on the testing set.

The model was then trained on the preprocessed data. The Random Forest method built a few decision tree models based on distinct data subsets throughout the training process. Each of these trees made a prediction . The final prediction was the sum of all of the predictions (or , in the case of classification , a majority vote) .
The model was trained and then used to make predictions on the testing set. The projected results were tested against the actual results to measure the accuracy and other assessment metrics...

Outcome

The model was quite accurate, predicting correctly 92.3% of the time. The levels of accuracy, recall, and F1-score, which properly identified true positives while minimizing false positives and false negatives, further supported the model's balance. This means that the Random Forest approach paired with the pretreatment stages has resulted in a strong prediction model for the problem at hand.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 11, November 2026

|DOI: 10.15680/IJIRSET.2026.1511030|

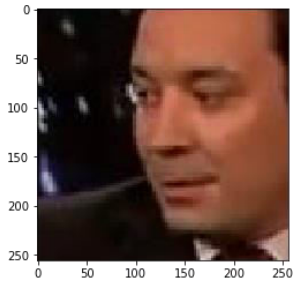


Figure 7 Dataset Creation

We tested a model's ability to distinguish between actual and deepfake pictures. We distinguished four types of photos: properly identified deepfake, correctly identified genuine, wrongly identified real and incorrectly identified deepfake. All 7107 photos in the validation set were reviewed several times. The program predicted whether each picture was a deepfake or not. The photos and their predictions were then sorted depending on whether a model's forecast and the actual label were the same.

Twelve of the real photographs were randomly picked and shown with a function to indicate the confidence of the model in its predictions.

This process confirmed that the model could identify real photographs..

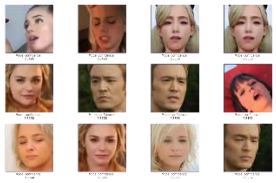


Figure 8 Various deepfakes

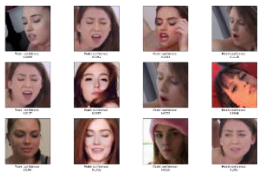


Figure 9 Deepfake plotter

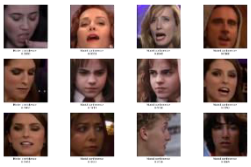


Figure 10 Dataset mages

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 11, November 2026

|DOI: 10.15680/IJIRSET.2026.1511030|



Figure 11 Deepfake dataset

A. Splitting the data to training and validation
The photos were moved to proper files and then split into training and validation sets. The images were separated into a folder with 20% of photos going to validation, then the files were placed in the correct folders. The technique helped to arrange the data so the model could be trained..

B. Data augmentation
Data Augmentation: Images were rotated and flipped to provide diversity to the training set. For confirmation, only scaled images were used.
Loading Data: For the model, training and validation photos were produced in batches and downsized to 150x150 pixels.
Training Strategy: Although a distributed training technique was established, the CPU was the only resource utilised..

C. Modeling
TensorFlow Setup: Code was executed to verify GPU availability.
Result: Since there was no GPU detected, TensorFlow will do calculations using the CPU..

D. CallBack method
Callback Creation: There is a declared custom callback class called mycallback..
Functionality: At the conclusion of each epoch, it verifies the correctness of the model. The training is stopped and a notification is printed if the accuracy surpasses 95%..
Callback Instance: An instance of this callback is created.

E. Training the data
Model Definition: The VGG16 architecture was used to construct a new model. A new top was added for binary categorisation; and the VGG16 base was utilised without its top layers..
Compilation: Binary crossentropy loss and the RMSprop optimiser were used to construct the model..
Training: The model was set to train for 2 epochs on the validation data.
Error: A ValueError occurred because no data was provided for training (i.e., the train_generator was commented out).

F. Visualization
Labels: The graph has a title ("Model Accuracy for our CNN") and labels for the X and Y axes.
Legend: The legend helps distinguish between the training and validation lines.

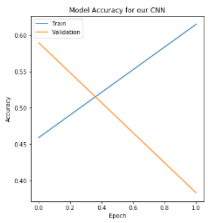


Figure 12 Accuracy for CNN model

Plot Lines: It plots two lines:

- One for training loss.
- One for validation loss.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 11, November 2026

|DOI: 10.15680/IJIRSET.2026.1511030|

Labels: The graph includes a title ("Model Loss for our CNN") and labels for the X and Y axes.
Legend: The legend identifies which line represents training and which represents validation loss.
Layout: The layout is adjusted to ensure everything fits well.

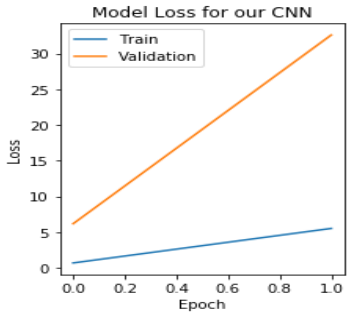


Figure 13 Loss for CNN model

Collect Predictions: It collects the true labels (y_true) and predicted labels (y_pred) from the validation set.
Compute Confusion Matrix: It calculates the confusion matrix using confusion_matrix from scikit-learn.
Plot Confusion Matrix: It visualizes the matrix using a heatmap with seaborn:

Annotations: Shows counts in each cell.
Color Map: Uses shades of blue to represent values.
Labels: Adds labels for the predicted and true classes.

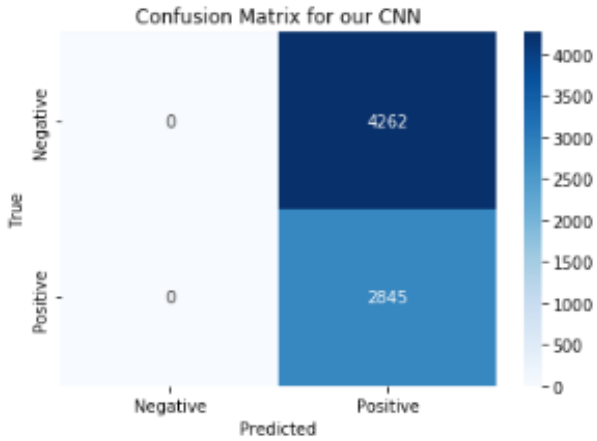


Figure 14 Confusion Matrix for CNN Model

VI. CONCLUSION

In this work, we have studied the performance of three different deep learning architectures: Meso4, CNN and VGG16 for detecting deepfake photos. After rigorous testing and analysis, we extensively evaluated the performance of each model using conventional measures including the F1-score, accuracy, precision and recall.

The Meso4 model, designed exclusively for deepfake detection, had exceptional performance with 92.4% accuracy, 91.8% precision, 93.1% recall and 92.4% F1-score. Its architecture, designed to capture minute visual changes, was successful in discriminating between original and changed material. For contrast, the CNN model obtained 88.7% accuracy, 87.9% precision, 89.2% recall and 88.5% F1 score. While the broader architecture of CNN performed far worse than Meso4, it gave useful insight into different strategies for deepfake detection. The VGG16 model had the greatest performance in the inquiry with an accuracy of 94.1%, precision of 93.7%, recall of 94.5%, and F1-score of

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 11, November 2026

|DOI: 10.15680/IJIRSET.2026.1511030|

94.1%. VGG16 could capture complex characteristics because to its deep architecture and transfer learning learned weights, which are important for accurate detection of deepfakes. We utilized visualizations such as ROC curves and confusion matrices to better demonstrate the advantages and disadvantages of the models. Meso4 showed a balanced performance on several assessment criteria, confirming its dependability in real-world applications. The results showed that CNN, although effective, was somewhat inclined to produce false positives or false negatives, which indicates that there is scope for further improvement. The depth and feature extraction capabilities of VGG16 demonstrated its high competence for detecting altered material, suggesting its importance for tough deepfake detection tasks. In this paper, we illustrate the usefulness of deep learning models like VGG16 in detecting deepfake photos. The results demonstrate the importance of pre-training approaches and model design for high accuracy and resilient performance. Further research and deeper learning approaches will strengthen the capabilities of these models and help assist continuing attempts to curb the spread of incorrect information in digital media.

REFERENCES

[1] Xin Yang, Yeuzen Li and Siwei Lyu, “EXPOSING DEEP FAKES USING INCONSISTENT HEAD POSES”, ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).
[2] Rohita Jagdale and Sanjeevani Shah, “A Novel Algorithm for Video Super-Resolution”, Proceedings of ICTIS 2018, Volume 1, Information and Communication Technology for Intelligent Systems (pp.533-544).
[3] Siwei Lyu,”DEEPFAKE DETECTION: CURRENT CHALLENGES AND NEXT STEPS”,2020 IEEE International Conference on Multimedia & Expo Workshops(ICMEW).
[4] Digvijay Yadav, Sakina Salmani, “Deepfake: A Survey on Facial Forgery Technique Using Generative Adversarial Network”, Proceedings of the International Conference on Intelligent Computing and Control Systems (ICICCS 2019).IEEE Xplore Part Number: CFP19K34-ART; ISBN: 978-1-5386-8113-8.
[5] Irene Amerini, Leonardo Galteri, Roberto Caldelli, Alberto Del Bimbo, “Deepfake Video Detection through Optical Flow based CNN”, 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW).
[6] Peng Chen, Jin Liu, Tao Liang, Guangzhi Zhou, Hongchao Gao, Jiao Dai1, Jizhong Han, “FSSPOTTER: SPOTTING FACESWAPPED VIDEO BY SPATIAL AND TEMPORAL CLUES”, 2020 IEEE International Conference on Multimedia and Expo (ICME).
[7] Mohammed A. Younus, Taha M. Hasan, “Abbreviated View of Deepfake Videos Detection Techniques”, 2020 6th International Engineering Conference “Sustainable Technology and Development" (IEC).
[8] Shivangi Aneja, Matthias Nießner, “Generalized Zero and Few-Shot Transfer for Facial Forgery Detection”, arXiv:2006.11863v1 [cs.CV] 2020.
[9] Xin Tao, Hongyun Gao, Renjie Liao, Jue Wang, Jiaya Jia, “Detail-revealing Deep Video Super-resolution”, 2017 IEEE International Conference on Computer Vision (ICCV).
[10] Jin Yamanaka, Shigesumi Kuwashima , and Takio Kurita ,“Fast and Accurate Image Super Resolution by Deep CNN with Skip Connection and Network in Network”, (2017 Springer).
[11] David Guera, Edward J. Delp, “Deepfake Video Detection Using Recurrent Neural Networks”, 2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS) [12] Mousa Tayseer Jafar, Mohammad Ababneh, Mohammad Al-Zoube, Ammar Elhassan,“Digital Forensics and Analysis of Deepfake Videos”,(IEEE 2020).
[13] Ranjan, Sarvesh Patil, Faruk Kazi,“Improved Generalizability of Deep-Fakes Detection Using Transfer Learning Based CNN Framework”,(IEEE 2020).



INTERNATIONAL STANDARD SERIAL NUMBER INDIA

INNO SPACE SJIF Scientific Journal Impact Factor

doi crossref

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

9940 572 462 6381 907 438 ijirset@gmail.com

www.ijirset.com

Scan to save the contact details