iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** III **Month of publication:** March 2026

DOI: https://doi.org/10.22214/ijraset.2026.78867

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue III Mar 2026- Available at www.ijraset.com

Deepfake Detection Using Convolutional Neural Networks (CNN)

Vijay Chakole[1], Akshita Lanjewar[2], Astha Jadhao[3], Pallavi Chikate[4], Mayuri Sawalakhe[5]
Dept of Electronics and Telecommunication, KDK College of Engineering, Maharashtra, India

Abstract: ***Deepfake technology has developed into a powerful tool capable of producing highly realistic synthetic media, which raises significant concerns related to misinformation and digital security. This study proposes a deep learning–based method for identifying deepfake images using multiple Convolutional Neural Network (CNN) architectures. The system utilizes five different models, including GoogLeNet, InceptionV3, VGG16, DenseNet121, and Xception, to perform binary classification between authentic and manipulated images. The dataset undergoes preprocessing and enhancement through various data augmentation techniques to improve the generalization ability of the models. Transfer learning is employed to take advantage of pretrained weights, thereby reducing computational cost. The models are trained and evaluated using standard performance metrics such as accuracy, precision, recall, and F1-score. The experimental findings indicate that VGG16 achieves the highest accuracy among all models, while DenseNet121 and Xception also show strong performance. This study demonstrates the effectiveness of CNN-based techniques in detecting manipulated media and highlights the significance of selecting appropriate models to improve detection accuracy.***

Keywords: ***Deepfake Detection, Convolutional Neural Networks (CNN), Transfer Learning, Image Classification, Data Augmentation, Synthetic Media Analysis***

I. INTRODUCTION

Over the past few years, significant developments in artificial intelligence and deep learning have led to the emergence of highly realistic synthetic media, commonly known as deepfakes [4],[11]. These images and videos are created using advanced generative models that can realistically modify or substitute facial features. While this technology offers advantages in domains like entertainment and virtual reality, it also introduces serious concerns such as identity misuse, misinformation, and breaches of privacy [4].

Identifying deepfake content has become increasingly difficult due to the continuous improvement of generative methods. Traditional image analysis techniques often fail to detect the subtle alterations present in such media. Consequently, deep learning-based approaches—particularly Convolutional Neural Networks (CNNs)—have gained widespread attention because of their ability to automatically capture intricate visual patterns and features [6]–[10].

The objective of this study is to develop a robust deepfake detection framework using multiple CNN architectures. Through a comparative evaluation of different models, the research aims to determine the most effective architecture for achieving accurate and reliable classification of manipulated images..

II. LITERATURE REVIEW

Hany Farid and his team have carried out significant work in the field of digital image forensics, introducing techniques that identify manipulated media by examining irregularities in visual patterns. However, such conventional approaches show limitations when applied to highly convincing deepfake images [2].

Andreas Rossler and colleagues developed the FaceForensics++ dataset and conducted evaluations using various deep learning models for deepfake detection. Their findings revealed that CNN-based methods achieve superior performance compared to traditional techniques, although the results are strongly influenced by the quality of the dataset [3].

Karen Simonyan and Andrew Zisserman introduced the VGG16 architecture, which has become widely used in image classification tasks. Its deep layered structure enables efficient extraction of detailed features, making it effective for identifying subtle facial alterations [8].

Christian Szegedy and his collaborators proposed the Inception architecture, including GoogLeNet and InceptionV3, which utilizes multi-scale convolution filters. This design enhances computational efficiency and has proven useful in deepfake detection applications [6],[7].

Gao Huang introduced DenseNet, a network architecture where each layer is directly connected to every other layer in a feed-forward manner. This structure promotes feature reuse and improves gradient propagation, resulting in enhanced performance for image classification tasks [9].

III. PROBLEM STATEMENT

The rapid advancement of deepfake technology has made it increasingly challenging to differentiate between authentic and manipulated images using conventional techniques. This issue poses significant risks across various domains, including social media, journalism, and cybersecurity. Therefore, there is a strong need for an automated solution capable of accurately identifying deepfake images and reducing the spread of misleading information.

IV. OBJECTIVES

To preprocess and organize a dataset consisting of real and fake images
To utilize data augmentation methods to enhance model performance
To implement various CNN architectures for the detection of deepfake images
To train and assess the models using suitable evaluation metrics
To compare the performance of different models and determine the most effective architecture

V. METHODOLOGY

The proposed approach follows a multi-stage pipeline designed for effective deepfake image detection.

A. Data Collection and Organization

The dataset used in this study consists of images divided into two categories: real and fake. It is systematically arranged into three subsets:
Training set
Validation set
Testing set
Such an arrangement ensures efficient training and reliable evaluation of the models [1].

B. Data Preprocessing

Prior to training, all images are resized according to the input specifications of different CNN architectures:
224 × 224 × 3 (for most models)
299 × 299 × 3 (for Xception)
In addition, pixel values are scaled to the range [0,1], which helps improve training stability and convergence speed.

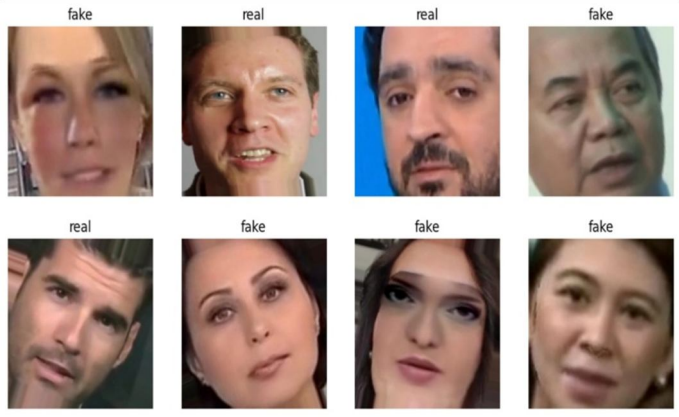


Fig 1 – Data visualisation weather the image is real or Fake.

C. Data Augmentation

To increase dataseTt variability and minimize overfitting, multiple augmentation techniques are applied using ImageDataGenerator, including:

Rotation
Width and height shifts
Zoom operations
Shear transformations
Horizontal flipping
These transformations enable the model to perform better on unseen data by improving generalization [1].

D. Data Loading Using Generators

The dataset is loaded through Keras ImageDataGenerator, which supports real-time augmentation and optimized memory utilization. Data is supplied to the models in batches, enhancing training efficiency.

E. Model Development

Five Convolutional Neural Network (CNN) architectures are implemented using transfer learning. Each model uses pretrained ImageNet weights, with the base layers frozen and additional custom layers added for binary classification [6]–[10].

1) GoogLeNet

GoogLeNet is built on the Inception architecture, where multiple convolution filters (1×1, 3×3, 5×5) operate in parallel to capture features at different scales. This improves efficiency while maintaining performance.
Pretrained on ImageNet
Base layers frozen

Added layers:
Global Average Pooling
Dropout
Dense layer (Sigmoid)
GoogleNet Classification report:

Classification Report:

	precision	recall	f1-score	support
fake	0.63	0.43	0.51	1541
real	0.57	0.75	0.65	1562
accuracy			0.59	3103
macro avg	0.60	0.59	0.58	3103
weighted avg	0.60	0.59	0.58	3103

Fig 2 – Classification report of GoogleNet

GoogleNet Confusion Matrix:

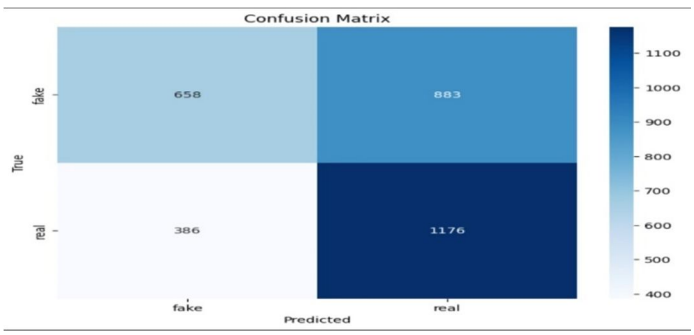


Fig 3 – Confusion Matrix of GoogleNet

GoogleNet Accuracy Graph:

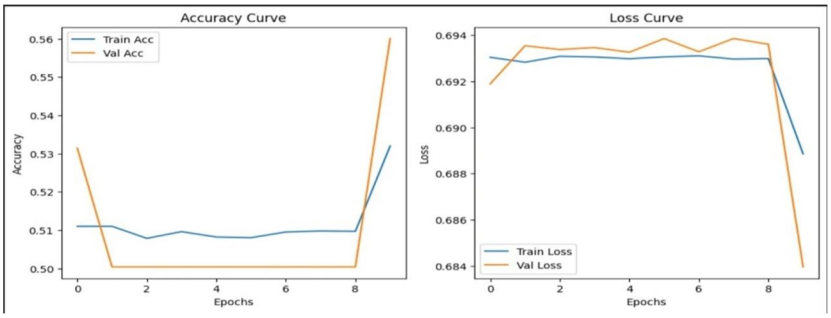


Fig 4 – Accuracy Graph of GoogleNet

2) InceptionV3

InceptionV3 enhances GoogLeNet by introducing factorized convolutions and improved optimization strategies, leading to better feature extraction with reduced computational cost.

Pretrained on ImageNet

Base layers frozen

Added layers:

Global Average Pooling

Dropout

Dense layer (Sigmoid)

Inception Classification Report:

Classification Report:

	precision	recall	f1-score	support
fake	0.65	0.64	0.64	1541
real	0.65	0.66	0.65	1562
accuracy			0.65	3103
macro avg	0.65	0.65	0.65	3103
weighted avg	0.65	0.65	0.65	3103

Fig 5 – Classification report of Inception

Inception Confusion Matrix:

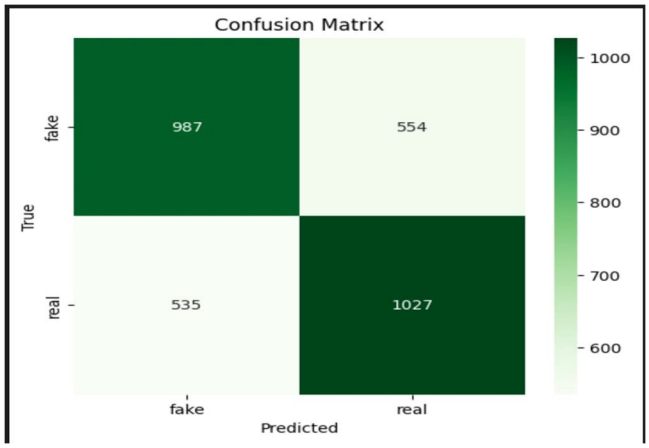


Fig 6 – Confusin Matrix of Inception

Inception Accuracy Graph:

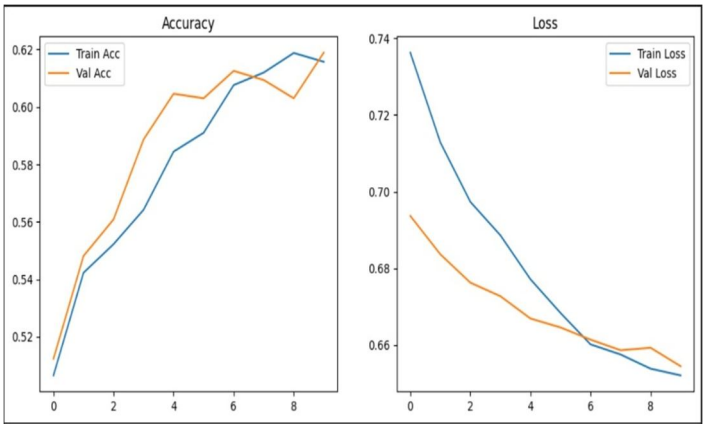


Fig 7 – Accuracy Graph of Inception

3) VGG16

VGG16 is a deep network with 16 layers that uses small 3×3 convolution filters. It is widely recognized for its strong capability in extracting detailed features.

Pretrained on ImageNet

Base layers frozen

Added layers:

Flatten

Dense (ReLU)

Dropout (0.5)

Output layer (Sigmoid)

VGG16 Classification Report:

```
Classification Report:
              precision    recall  f1-score   support

        fake       0.78      0.61      0.69      1541
        real       0.68      0.83      0.75      1562

    accuracy                           0.72      3103
   macro avg       0.73      0.72      0.72      3103
weighted avg       0.73      0.72      0.72      3103
```

Fig 8 – Classification report of VGG16

VGG16 Confusion Matrix:

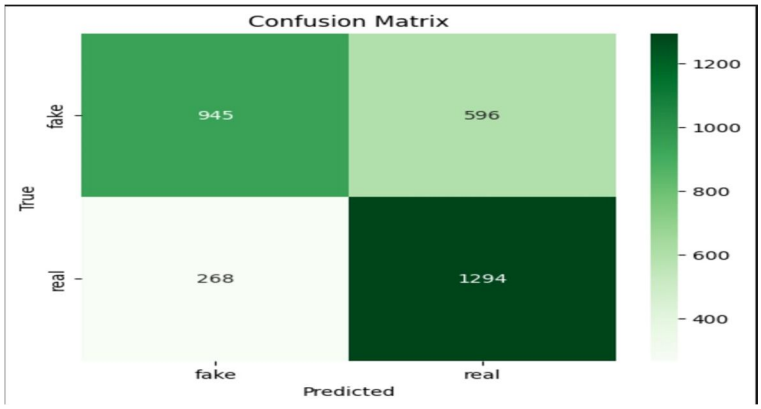


Fig 9 – Confusion Matrix VGG16

VGG16 Accuracy Graph:

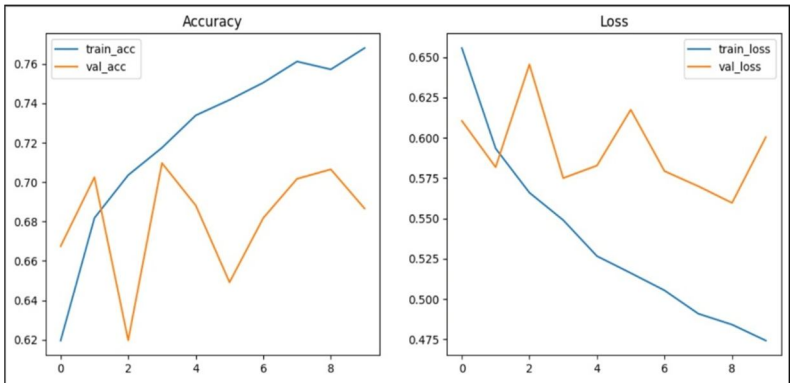


Fig 10 – Accuracy Graph of VGG16

4) DenseNet121

DenseNet121 connects each layer with all preceding layers, enabling efficient feature reuse and improved gradient flow, which enhances performance.
Pretrained on ImageNet
Base layers frozen

Added layers:
Global Average Pooling
Dropout (0.5)
Dense layer (Sigmoid)

DenseNet121 Classification Report:

```
Classification Report:
              precision    recall  f1-score   support

        fake       0.76      0.57      0.65      1541
        real       0.66      0.83      0.73      1562

    accuracy                           0.70      3103
   macro avg       0.71      0.70      0.69      3103
weighted avg       0.71      0.70      0.69      3103
```

Fig 11 – Classification report of DenseNet1121

DenseNet121 Confusion Matrix:

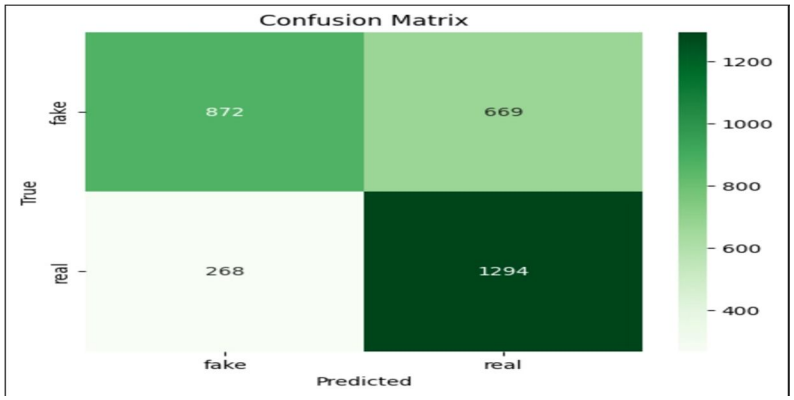


Fig 12 – Confusion Matrix of DenseNet1121

DenseNet121 Accuracy Graph:

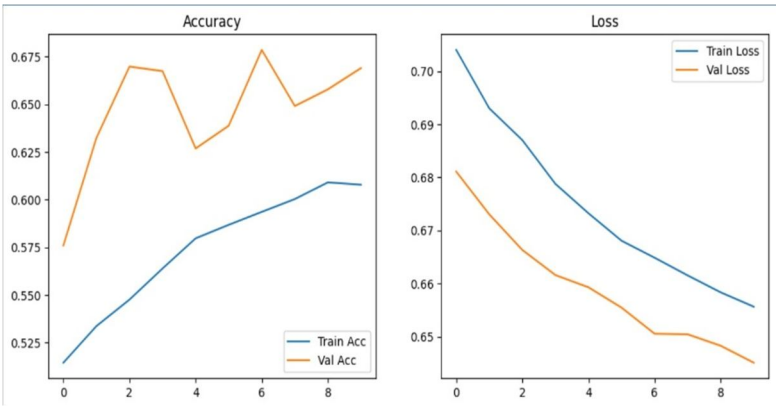


Fig 13 – Accuracy Graph of DenseNet1121

5) Xception

Xception replaces standard convolutions with depthwise separable convolutions, resulting in improved efficiency and performance.
Input size: 299 × 299 × 3
Pretrained on ImageNet
Base layers frozen

Added layers:
Global Average Pooling
Dropout (0.5)
Dense layer (Sigmoid)

Xception Classification Report:

```
Classification Report:
              precision    recall  f1-score   support

        fake       0.63      0.84      0.72      1541
        real       0.77      0.52      0.62      1562

    accuracy                           0.68      3103
   macro avg       0.70      0.68      0.67      3103
weighted avg       0.70      0.68      0.67      3103
```

Fig 14 – Classification report of Xception

Xception Confusion Matrix:

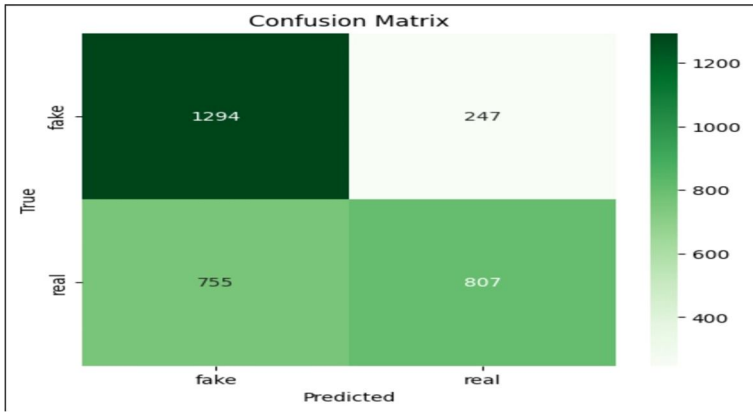


Fig 15 – Confusion Matrix of DenseNet1121

Xception Accuracy Graph:

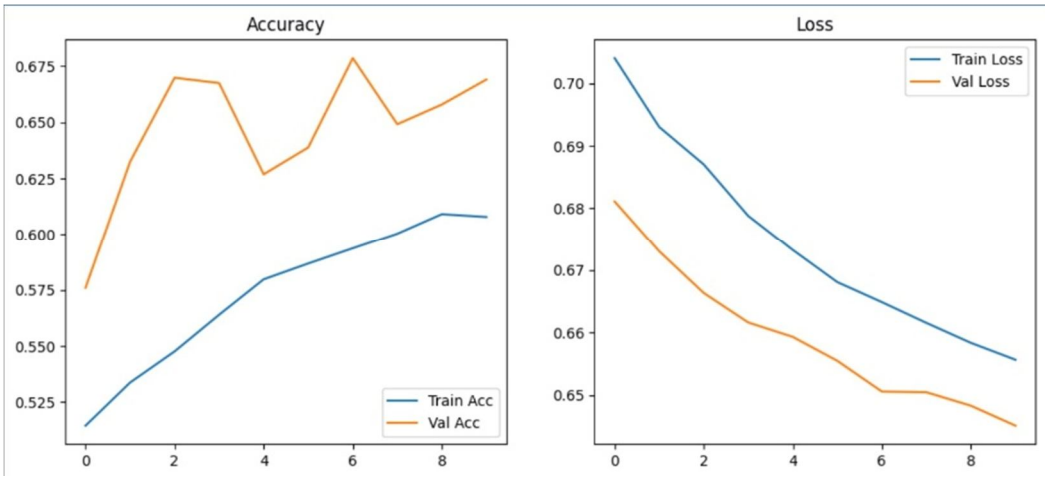


Fig 16 – Accuracy Graph of DenseNet1121

F. Model Training

All models are trained using the following configuration:
Optimizer: Adam
Learning rate: 0.0001
Loss function: Binary Crossentropy
Epochs: 10
Batch size: 32
Validation data is used during training to monitor performance and reduce overfitting.

G. Model Evaluation

The trained models are evaluated using the following metrics:
Test Accuracy
Confusion Matrix
Classification Report (Precision, Recall, F1-score)
These measures provide a detailed assessment of model performance [12].

H. Prediction

The trained model is applied to classify new input images as real or fake. Predictions are generated based on probability values obtained from the sigmoid activation function.

I. Deployment Using Flask

After evaluating all models, the best-performing model is selected and deployed using the Flask framework. Flask enables the creation of a simple web interface for real-time predictions.
The deployment process involves:
Saving the trained model
Developing a Flask application
Creating an interface for image upload
Processing input images and generating predictions
Displaying results as Real or Fake
This implementation allows the system to function as a practical tool for detecting deepfake images.

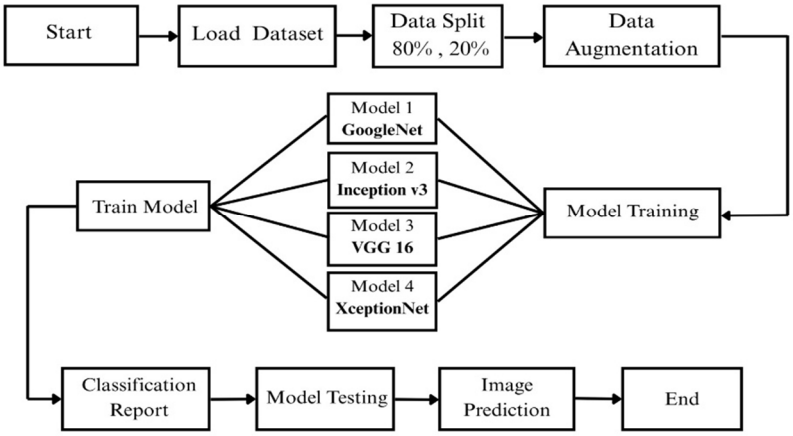


Fig 17 – Workflow of whole process.

VI. RESULTS AND ANALYSIS

The performance of all models is compared based on accuracy:

Model	Accuracy
GoogLeNet	~59%
InceptionV3	~65%
VGG16	~72.16%
DenseNet121	~69–70%
Xception	~67.7%

A. Analysis

VGG16 achieved the highest accuracy due to its deeper structure and strong ability to extract detailed features. DenseNet121 and Xception also delivered good performance because of efficient feature reuse and optimized convolution operations. Although GoogLeNet showed lower accuracy, it provided faster computation compared to other models..

VII. CONCLUSION

The proposed system for deepfake detection demonstrates the effectiveness of Convolutional Neural Networks in identifying manipulated images. Multiple CNN architectures, including GoogLeNet, InceptionV3, VGG16, DenseNet121, and Xception, were implemented and evaluated for performance comparison.

Among these, VGG16 achieved the best accuracy, mainly due to its deep architecture and strong feature extraction capability. DenseNet121 and Xception also showed competitive results, emphasizing the importance of advanced architectures and efficient feature utilization. The findings indicate that transfer learning plays a key role in improving performance while reducing training time and computational requirements [6]–[10].

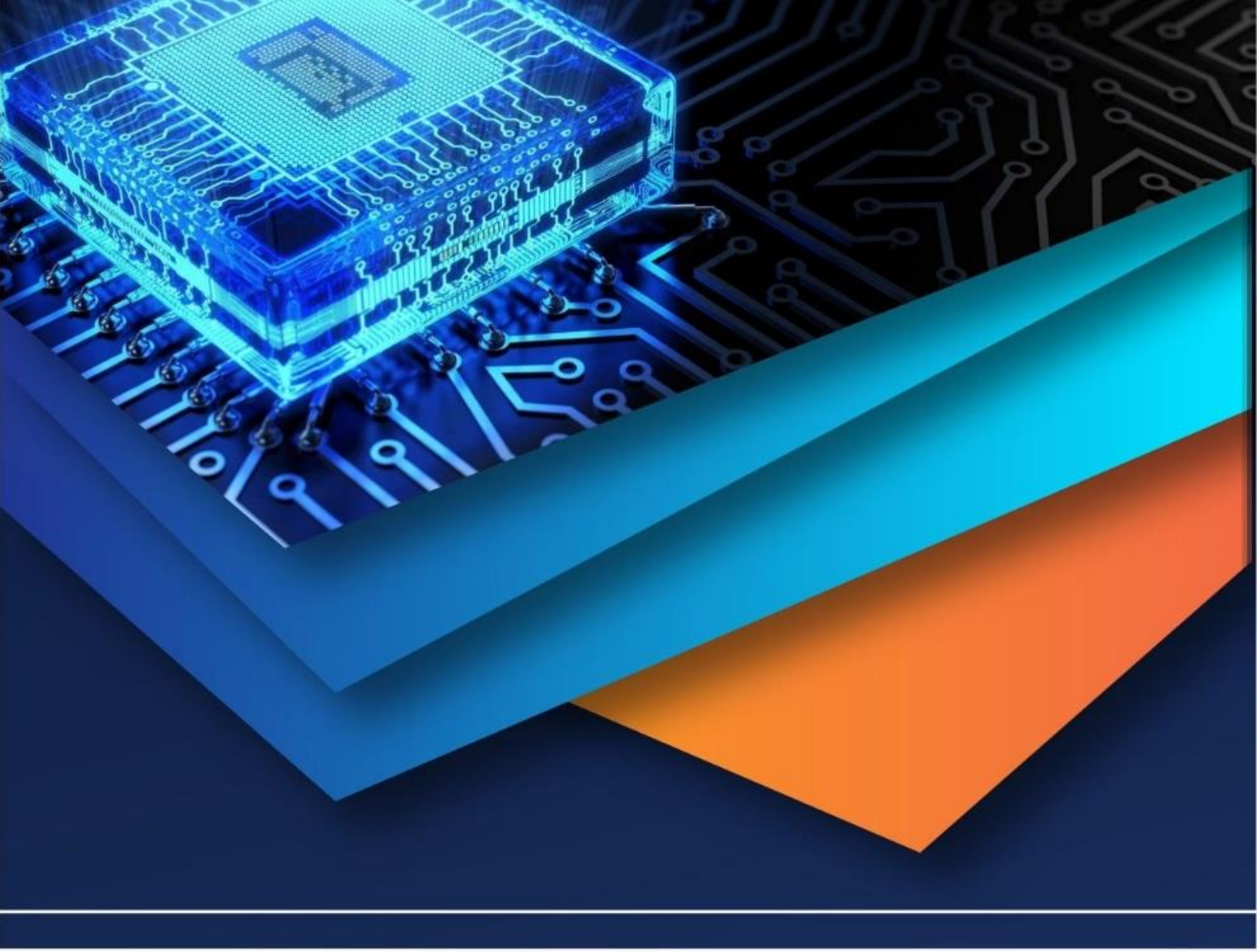
The application of data augmentation further improved model generalization by increasing variability within the dataset. Evaluation metrics such as confusion matrix, precision, recall, and F1-score provided a comprehensive analysis of performance [12].

Additionally, the project was extended to a real-world application by deploying the trained model using Flask. This enables users to upload images and receive predictions instantly, making the system practical and user-friendly.

Overall, this study highlights the potential of deep learning methods in addressing deepfake-related challenges and underscores the importance of selecting suitable architectures for improved detection accuracy [4], [5].

REFERENCES

[1] J. J. Bird and A. Lotfi, "CIFAKE: Image Classification and Explainable Identification of AI-Generated Synthetic Images," IEEE Access, vol. 12, 2024.
[2] H. Farid, "Image Forgery Detection: A Survey," IEEE Signal Processing Magazine, vol. 26, no. 2, pp. 16–25, Mar. 2009.
[3] A. Rössler et al., "FaceForensics++: Learning to Detect Manipulated Facial Images," Proc. IEEE ICCV, 2019, pp. 1–11.
[4] Y. Li and S. Lyu, "Exposing DeepFake Videos By Detecting Face Warping Artifacts," Proc. IEEE CVPR Workshops (CVPRW), 2019.
[5] D. Güera and E. J. Delp, "Deepfake Video Detection Using Recurrent Neural Networks," Proc. IEEE AVSS, 2018, pp. 1–6.
[6] C. Szegedy et al., "Going Deeper with Convolutions," Proc. IEEE CVPR, 2015.
[7] C. Szegedy et al., "Rethinking the Inception Architecture for Computer Vision," Proc. IEEE CVPR, 2016.
[8] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," Proc. ICLR, 2015.
[9] G. Huang et al., "Densely Connected Convolutional Networks," Proc. IEEE CVPR, 2017.
[10] F. Chollet, "Xception: Deep Learning with Depthwise Separable Convolutions," Proc. IEEE CVPR, 2017.
[11] B. Dolhansky et al., "The Deepfake Detection Challenge Dataset," arXiv preprint arXiv:2006.07397, 2020.
[12] H. Nguyen, J. Yamagishi and I. Echizen, "Capsule-Forensics: Using Capsule Networks to Detect Forged Images and Videos," Proc. IEEE ICASSP, 2019.
[13] X. Yang, Y. Li and S. Lyu, "Exposing Deep Fakes Using Inconsistent Head Poses," Proc. IEEE ICASSP, 2019.
[14] S. Agarwal et al., "Protecting World Leaders Against Deepfakes," Proc. IEEE CVPR Workshops, 2019.
[15] Z. Wang et al., "CNN-generated Images are Surprisingly Easy to Spot… for Now," Proc. IEEE CVPR, 2020.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY