ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Volume 12, Issue 07, July 2025

DeepFake Image Detection Using Transfer Learning and Attention-Enhanced EfficientNet

[1] Sarvesh S. Gharat, [2] Dr. Puja Padiya, [3] Dr. Amarsinh Vidhate

[1] MTech Student, D Y Patil Deemed to be University, Navi Mumbai, India.
[2] Professor, D Y Patil Deemed to be University, Navi Mumbai, India.
[3] HOD, D Y Patil Deemed to be University, Navi Mumbai, India.
Corresponding Author Email: [1] sar.gha.rt23@dypatil.edu, [2] puja.padia@dypatil.edu

Abstract— ***Deepfake detection refers to the process of identifying synthetically generated or manipulated facial images that closely resemble real human faces. These images, often created using advanced generative adversarial networks (GANs) like StyleGAN2, present a significant challenge for traditional detection systems due to their high realism and subtle artifacts. Traditional machine learning models and shallow neural networks often fall short in effectively distinguishing real from fake faces, primarily because they lack the capacity to capture intricate pixel-level features and contextual semantics within images. This study addresses those limitations by applying advanced deep learning techniques, including convolutional neural networks (CNNs) and several state-of-the-art pretrained models VGG16, InceptionResNet, Xception, MobileNet, and EfficientNet-B2 leveraging transfer learning for improved performance. Extensive experiments were conducted using a robust image dataset containing both real and synthetic faces. Each model was fine-tuned for binary classification (Real vs. Fake), and evaluated using precision, recall, F1-score, accuracy, and confusion matrix. Among all, EfficientNet-B2 enhanced with an attention mechanism emerged as the best-performing model, achieving an impressive 83% accuracy. The integration of attention allows the model to focus more effectively on distinguishing facial features, making it particularly robust against complex deepfakes. This research introduces a novel and efficient framework for real-time, high-accuracy deepfake detection.***

Index Terms— ***DeepFake Detection, CNN, Transfer Learning, EfficientNet, Attention Mechanism, Image Classification.***

I. INTRODUCTION

DeepFake is an artificial intelligence approach that allows precise creation [1] of convincing images, videos, or sound using deep learning and technologies called Generative Adversarial Networks (GANs) [2]. At the beginning, DeepFakes were used for entertainment and artistic endeavors, but now they are common in spreading false news and secretly stealing people's identities as well as committing online crimes [3]. There are various DeepFakes, for example, face-swapping, lip-syncing, voice cloning, and even making whole-body versions [4]. Even though technology brings benefits such as creating similar-looking avatars, boosting films, and aiding patients with artificial speech, its adaptation can seriously harm people's privacy, safety, and trust.

DeepFake technology is expected to bring major changes to content making, learning, gaming, and virtual reality [5]. At the same time, there are bounds and issues associated with them. Main concerns involve ethics, legal matters, and telling real media apart from synthetic media that is now very hard to spot. Besides, it is becoming clear that traditional detection solutions are not enough, as DeepFakes are often successful.

II. RELATED WORKS

In their paper, [6] sought to urge people to be aware of the danger posed by deepfake technologies, as these can easily create realistic images and videos and cause security or misinformation problems. The main goal of the study was to tell real images apart from deepfake ones with the help of DL algorithms. For this purpose, the authors came up with a CNN-based model that adds Dense, MaxPooling, and Dropout layers to better identify and regulate important features. First, the videos were processed to extract the frames, then the features of the faces were taken, after preprocessing and classification were done. Other than the VGG model, a comparison was done using a standard CNN and a combination of an MLP and CNN. Customizing the CNN gave the best results, since it achieved 91.4% accuracy, a loss of only 0.342, and had an excellent AUC of 0.92, compared to the other models. Distinguishing between real pictures and fake ones made by top-notch tools has become quite hard. The method done by CNN was accurate, yet this approach only looks at single frames, which may make it harder to work with varied types of deepfake generation or picture resolutions. A further step would be to use information from when events happen in videos and learn from various types of datasets to make the model stronger.

In [7], the authors suggested a new and better Deep CNN design meant for exact detection of deepfake media, due to its rising use to alter the public's view and ruin reputations with highly accurate fake images and videos created by GANs. The main goal of the study was to design a model for detecting deepfakes that is accurate and can work with various datasets and ways of making deepfakes. Many sources were used to collect images, and then they were rescaled before being fed into the D-CNN to strengthen the model. The model was optimized using Adam and binary cross-entropy loss was used in its training. Data for deepfake and real images consisted of 5000 and 10000 images that come from GAN challenges. Its almost perfect performance was seen in all the datasets. Many previous approaches could not catch inconsistencies between different frames in media, but this problem is partly solved by the model's better generalization capability. Although it is very accurate, it has one shortcoming: it depends only on images, not on video sequences, so it may not resist newer advanced types of deepfakes or gain useful insights into dynamics employed over time in the videos.

In [8], the authors proposed a DL-based framework to enhance the detection of deepfakes. The study aimed to develop a robust detection system combining multiple artificial intelligence techniques to improve reliability and accuracy. CNNs are used for detecting critical facial regions such as the eyes and nose, while a hybrid CNN-ViT (Vision Transformer) model is employed for comprehensive face detection. For prediction, a majority voting mechanism combines the outputs from three individual models trained on different facial features to arrive at a final decision. The model was trained and tested using the FaceForensics++ and DFDC datasets and evaluated using multiple performance metrics. The CNN-based model achieved an impressive 97% accuracy, while the CNN-ViT hybrid reached 85%, demonstrating significant improvement over previous studies. A major challenge addressed in this study was the high similarity between deepfake and real videos, which the multi-feature ensemble approach helped to mitigate. However, a key limitation lies in the reliance on handcrafted feature detection (such as eyes and nose), which may be sensitive to variations in facial orientation, occlusions, or lighting conditions. Further enhancements could involve integrating temporal dynamics and more end-to-end deep models for increased generalization.

In the work [9], the authors discussed the challenge of finding fake human faces made by GANs, since they are easy to create with mobile apps and share on social networks, thus threatening privacy, prevention of fraud, and people's trust in each other. Authors looked into employing SVM classifiers in combination with and without PCA to separate genuine from fake facial images through machine learning. First, the approach converts RGB images to the YCbCr color space, then applies gamma correction and edges are found by the Canny filter to help with the classification of facial features. There were two detection systems examined: one that using SVM and PCA, and a second one that used SVM by itself. From the results, it was shown that SVM with PCA achieved an accuracy of 96.8%, much higher than the 72.2% achieved by standalone SVM, confirming that using PCA for feature reduction helped SVM work better. The biggest problem dealt with was the design of a light method that is strong enough to function well with visually deceiving fake images. Still, one problem with the model is that its accuracy could be influenced by the manual steps taken before training and is not guaranteed to work for different or unseen datasets. Work in the future might look into making SVMs use deep learning techniques or changing their structure to include dynamic features for more flexibility.

In [10], the researchers concentrated on tackling how inauthentic images and videos used to be spread online, leading to misinformation and harms to various people, especially influential public figures. In this study, a brand-new model was introduced by using SVM and CNN for detecting deepfakes. This model was trained and tested using the publicly available dataset called 140k Real and Fake Faces to find out if images were real or fake. This way, the weaknesses of each classifier are corrected, reaching a perfect accuracy of 88.33%. The toughest thing to overcome was making sure fake faces cannot be misidentified as real ones by the standard tools used. Nevertheless, because its accuracy is not as high as in more complex AI architectures, it might not be as effective in real-life, large-scale, or changing situations with deepfakes. Work could be done in the future by bringing in advanced neural network designs or time-based analysis methods for video deepfakes to strengthen their effectiveness.

In [11], it was reported that deepfake videos are now a real concern because they can more easily lead to misinformation and illegal acts involving faces. It used a Triplet based detection strategy along with two classifiers: Random Forest and Stochastic Gradient Descent. Facial embeddings were obtained by applying the MTCNN approach, which has the ability to detect and properly align faces. It was developed using 600 videos that included video frames with 30, 50, and 70 frames. Results pointed out that the Triplet Loss model combined with Random Forest achieved a higher accuracy of 84%, AUC of 0.8987, EER of 0.1776, precision of 0.9694, recall of 0.8115, and an F1 score of 0.8441 when paired with the Random Forest classifier in comparison to the SGD classifier. Handling the differences in the number of frames and getting quality feature embedding is the biggest difficulty in this approach. Using this method results in lesser accuracy, and it relies on capturing each frame instead of time-based changes, which could cause missed or different results. In the future, temporal sequence testing could be carried out on various large-scale and diverse data so that it is more effective.

In this paper [12], the authors dealt with the trouble of detecting fake voices, which are created by AI and can now go unnoticed by many. The purpose of the study was to boost the detection of audio deepfakes using both ML and DL methods and the Fake-or-Real dataset, which is made up of audio samples from text-to-speech tools, and sorted into four datasets differing in length and bitrate: for-rece, for-2-sec, for-norm, and for-original. The main method used to find important features was using Mel-frequency cepstral coefficients (MFCCs). Among all the classifiers, Support Vector Machine (SVM) did the best on the for-rece and for-2-sec datasets, and Gradient Boosting was the best choice for the for-norm dataset. VGG-16 worked the best on the for-original data and surpassed the achievements of other contemporary techniques. The main problem was correctly telling real sounds from synthetic ones in every type of audio length and quality. The main issue is that this approach uses only static aspects of audio and does not use any information about time frames or surrounding context, which could cause it to struggle with more sophisticated deep picture file of sound media. In the future, scientists could test detection with time series models and in messy external conditions to increase its robustness.

III. METHODOLOGY

A. Dataset Description

To guarantee that the deepfake detection system is evaluated fully, this study used the public "Real and Fake Face Detection" and "Fake-Vs-Real-Faces (Hard)" datasets. They both have the purpose of separating real from fake images of human faces. CNN-BendyRealvsFake builds its main dataset, Real and Fake Face Detection, with facial images created by using StyleGAN2, which makes it difficult for people to tell the real ones from the fake ones. Facial images are taken using the Unsplash API and then cropped with OpenCV to make sure the facial region is always represented in the same way. Every image in the dataset is JPEG and of the same size, 300x300, to ease the training process for the model. To make sure the model is strong and variable, the Hard dataset is used as an additional test data set. All in all, it has 1288 images made up of 700 fake faces and 589 real ones, where both were produced by StyleGAN2 and the real ones were selected from a variety of people to ensure a mix in age, gender, makeup, and ethnicity. Unlike most other datasets, this one has variety in the key features of the fake class. This is intended to resemble real situations in which good synthetic images may easily trick regular image classifiers. With this combination of datasets, the model gets familiarized with both well-structured information and hypothetical real-life tests, increasing its ability to deal with new situations. Images are given labels and their annotations are placed in a csv file that maps each image ID to its class label. With this approach, the DeepFake detection model becomes more durable and precise in all kinds of situations.

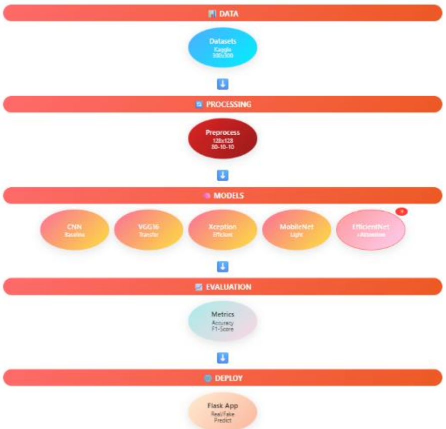

Figure 1. System Architecture Diagram

B. Data Preprocessing

The dataset used for DeepFake detection comprises two distinct image repositories sourced from Kaggle. The first dataset includes real and fake facial images, where real faces are collected via the Unsplash API and fake faces are generated using StyleGAN2, a state-of-the-art generative model known for producing highly realistic synthetic faces. The second dataset consists of even more challenging "hard fake" images, enhancing the dataset's robustness for real-world application. All images are in JPEG format with a standard size of 300×300 pixels as shown in Figure 1. To prepare the dataset for model training, the images were loaded using OpenCV and resized to 128×128 pixels using the cv2.resize() function to meet the input size requirements of various deep learning models. Additionally, label encoding was performed using LabelBinarizer to convert the categorical labels ('Real' and 'Fake') into a binary numerical format, enabling compatibility with classification models.

C. Data Visualisation

Figure 2 includes three images of human faces taken from the training dataset. Putting these images next to each other helps us understand that the 'Real' class includes different people of all ages, genders, and facial features. The images were read from training_real and showed side by side using subplot from matplotlib; all images were put in a single row. Every image does not have axes to make sure the face is clearly seen and emphasized. Visualising the data is highly important in EDA because it helps you confirm the quality of the data before you start with deep learning.

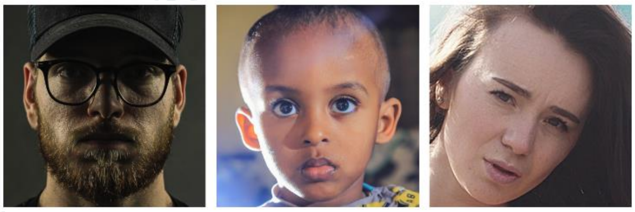

Figure 2. Sample Real Face Images from the Training Dataset

Figure 3 provides three examples of the 'Fake' class in the training dataset that are computer-generated portraits of people. Images from training_fake directory were loaded using matplotlib in a layout with one line and three columns. StyleGAN2 was used to make the faces by creating very lifelike but computer-generated facial features. With the help of the figure, you can see and study the small signs that may show the images are generated instead of captured by a camera. This move is done during exploratory data analysis as it supports our ability to explain the model and deal with the dataset.

Figure 3. Sample Fake Face Images from the Training Dataset

IV. IMPLEMENTATION

A. Implementation of CNN

The classification of DeepFake images was done by implementing CNN using the Keras Sequential API. The network was arranged so that it can identify, extract, and learn organized features from images that were resized and had three color channels. It starts off with three convolutional layers, and every layer has 128 filters with a kernel size of 3×3 and 'same' padding, so the spatial dimensions do not change. The last layer is made up of a single neuron and a sigmoid activation function, as it is used to tell whether a coin is Real or Fake.

B. Implementation of Mobilenet

By making use of transfer learning, the MobileNetV2 model helps classify whether images containing faces are DeepFakes or not. MobileNetV2 is loaded from the ImageNet data and not given its top classification layers when include_top is set to False, giving you the chance to use the model for the specified binary classification task. The fully connected layer takes images that are resized to 128×128×3 and helps extract important features. Besides, a GlobalAveragePooling2D layer is placed after the main model to shrink the size of the maps and still keep vital data.

C. Implementation of InceptionResnet

With the InceptionResNetV2 model, the use of Inception and residual structures together helps detect both major and finer details in a person's face. The architecture was transferred from the Keras applications in TensorFlow, and because we did not want to use the top layer, we turned it off (include_top=False). The temporary input shape was altered to fit the 128x128x3 resized images. InceptionResNetV2 was connected in sequence and maintained as trainable in order to be able fine-tune all the model's layers. So that the output from the base model would be less complex, GlobalAveragePooling2D was used to transform the spatial feature maps into a one-dimensional representation that reduces overfitting.

D. Implementation of VGG16

To detect DeepFake images, VGG16 takes advantage of transfer learning to separate real from fake facial pictures. Since the model is first trained on ImageNet data, it is set to not use the classification layers for top and instead use them for binary classification. The model takes images at 128×128×3 to coincide with the resized samples. All layers except the last two family groups are lazily updated, since they were trained extensively already on a huge ImageNet dataset. On top of the basic layer, the output feature maps are put into a flattening layer and then processed by a dense layer with 64 ReLU-activated neurons to allow the network to learn complex features.

E. Implementation of Xception

Xception's separate convolutions allowed it to function well as a major component for DeepFake classification, making it very efficient in spotting patterns hidden in the image. As the Xception base uses pre-trained ImageNet weights and includes_top=False, developers simply added a basic and effective classification head to use the network for binary classification. The images were first made 128×128 and were set to three color channels to meet the model's expectations.

F. Implementation of EfficientNet B2

The implementation of the EfficientNet-B2 model for DeepFake detection leverages the transfer learning capability of TensorFlow's Keras API. EfficientNet-B2, a member of the EfficientNet family known for its balance between accuracy and computational efficiency, is initialized with pre-trained ImageNet weights and used as the base model without its top classification layers (include_top=False). The model architecture includes 7,771,387 total parameters, out of which 7,703,812 are trainable, while the remaining 67,575 are frozen (non-trainable) from the pre-trained base. This architecture combines high representational power and low parameter count, making it well-suited for DeepFake detection with both accuracy and efficiency.

G. Implementation of EfficientNet B2 with attention

It was implemented to enhance the model's ability to focus on the most informative regions of facial images, significantly improving DeepFake detection accuracy. Feature maps extracted from EfficientNetB2 undergo batch normalization to stabilize training. An attention mechanism is introduced using a series of Conv2D layers with ReLU activation and dropout regularization. This stack processes the feature maps and generates an attention map via a sigmoid-activated convolution, producing a single-channel mask that highlights important spatial areas. To align this attention map with the base model's feature depth, a frozen 1×1 convolution layer is used to replicate the attention weights across all channels. This attention map is then applied to the normalized feature maps through element-wise multiplication, resulting in masked features. These are processed using Global Average Pooling (GAP), and a rescaling operation is performed to normalize the weighted features and prevent numerical instability. Finally, the output passes through dropout and dense layers for classification.

The model achieves a balance of high accuracy and efficiency, with over 8.1 million parameters, most of which are trainable.

V. RESULT ANALYSIS

A. Results of CNN

Figure 4 presents the confusion matrix for the CNN model's performance in classifying DeepFake images. The model predicted all inputs as class 0, failing to recognize any samples from class 1. Specifically, it correctly identified 166 real images (True Negatives) but misclassified all 167 fake images as real (False Negatives), resulting in zero True Positives and False Positives. This imbalance indicates the model's inability to learn discriminative features for the "Fake" class, highlighting significant limitations in its generalization capacity. The result exposes poor recall and precision for the fake category, emphasizing the need for more robust models or additional training refinements.

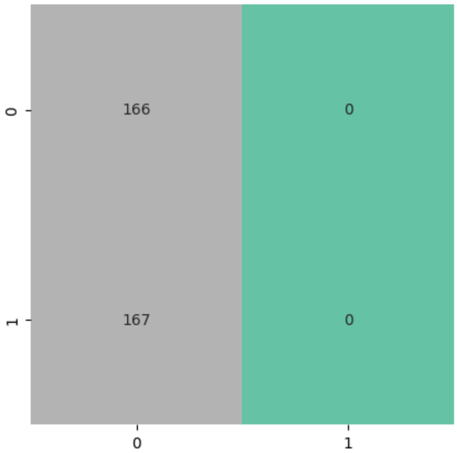

Figure 4. Confusion Matrix

B. Results of Mobilenet

Figure 5 displays the confusion matrix representing the performance of the MobileNet model on the DeepFake classification task. MobileNet demonstrates a reasonable balance in classification, the high number of false negatives suggests that fake images often resemble real ones, making them harder to detect. Despite its lightweight nature, the model needs further optimization to enhance its ability to identify manipulated images with higher recall.

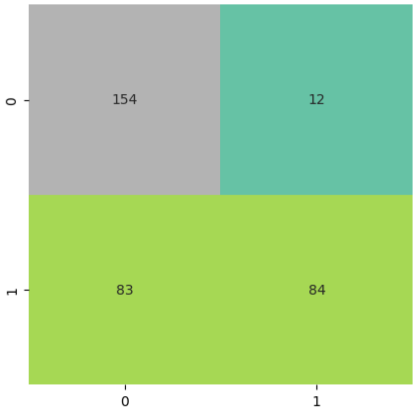

Figure 5. Confusion Matrix

C. Results of InceptionResnet

Figure 6 illustrates the confusion matrix for the InceptionResNet model applied to the DeepFake classification task. The model successfully predicted 120 real images and 142 fake images.

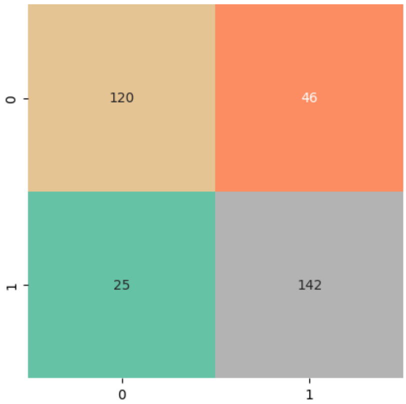

Figure 6. Confusion Matrix

D. Results of VGG16

Figure 7 shows the confusion matrix of the VGG16 model's performance on the DeepFake classification task. The model correctly classified 126 real images and 118 fake images.

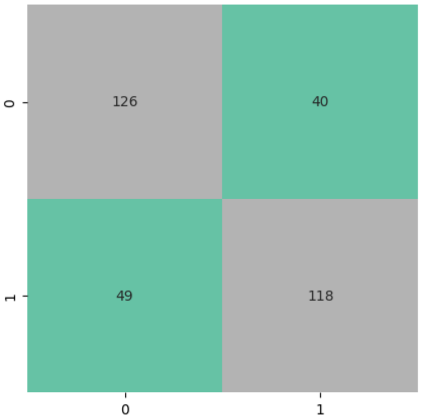

Figure 7. Confusion Matrix

E. Results of Xception

Figure 8 presents the confusion matrix for the Xception model applied to the DeepFake classification task. The model accurately predicted 142 real images and 125 fake images.

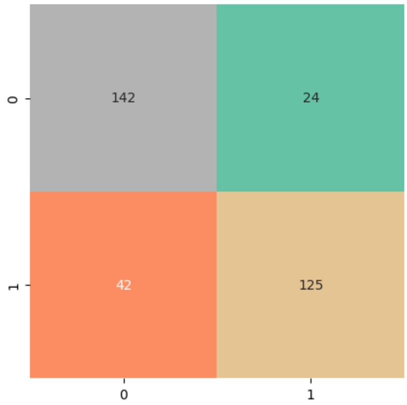

Figure 8. Confusion Matrix

F. Results of EfficientNet B2

Figure 9 illustrates the confusion matrix for the EfficientNet-B2 model used in DeepFake image classification. The model correctly classified 141 real images and 127 fake images. The diagonal values (141 and 127) reflect the true positives for each class, while the off-diagonal values (25 and 40) show the misclassifications. These results highlight the model’s robustness and efficiency in distinguishing between real and fake faces.

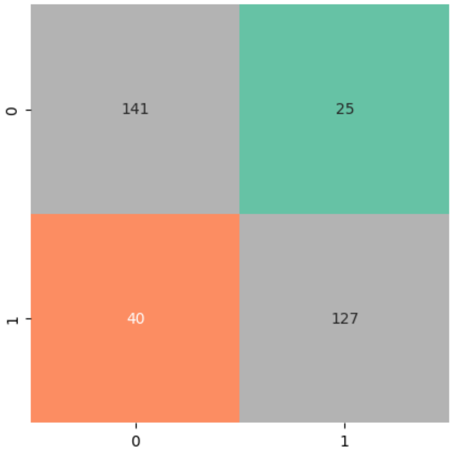

Figure 9. Confusion Matrix

G. Results of EfficientNet B2 with attention

Figure 10 presents the confusion matrix for the EfficientNet-B2 model enhanced with an attention mechanism. The model demonstrates improved performance by correctly classifying 123 real images and 153 fake images. The significant increase for the fake class (153) and the noticeable reduction in false negatives (14) suggest that the attention mechanism effectively enhances feature discrimination.

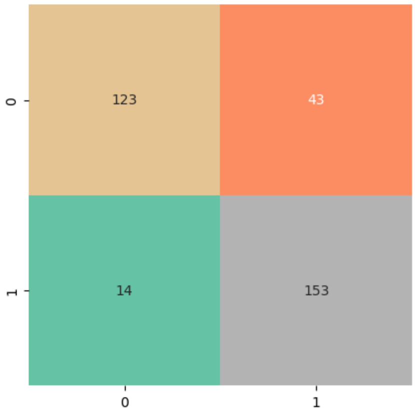

Figure 10. Confusion Matrix

H. Comparative Analysis of Models

The table 1 below presents a comparative analysis of different deep learning models applied to the DeepFake image classification task. Among all models, the baseline CNN achieved the lowest accuracy at 50%, indicating limited feature learning. MobileNet and VGG16 showed moderate performance at 71% and 73% respectively. InceptionResNet and Xception provided better accuracy, both nearing 80%. EfficientNet-B2 matched Xception’s performance with 80% accuracy but with improved efficiency. The best-performing model was EfficientNet-B2 with Attention Mechanism, achieving 83% accuracy. This novel approach enhances feature sensitivity by guiding the model’s focus to important regions in the image, significantly improving classification results.

Table 1: Comparison Table

Model	Accuracy (%)
CNN	50
MobileNet	71
InceptionResNet	79
VGG16	73
Xception	80
EfficientNet-B2	80
EfficientNet-B2 with Attention (Novelty)	**83**

VI. CONCLUSION

In this research, we presented a comprehensive approach to DeepFake image detection by leveraging multiple DL models. This study also proposed a novel enhancement by integrating an attention mechanism into EfficientNet-B2, which significantly improved classification performance. The models were trained and evaluated on a curated dataset of real and fake faces, with fake images generated using advanced GAN techniques like StyleGAN2. Preprocessing steps such as label encoding, resizing, and balanced dataset splitting ensured robust training. Through extensive evaluation this study observed that traditional models like

CNN and VGG16 provided basic benchmarks, while deeper models like EfficientNet and Xception delivered higher accuracy. Notably, the attention-enhanced EfficientNet-B2 outperformed all other models, demonstrating that adding attention mechanisms enables the model to focus better on important features, thereby enhancing detection reliability.

To bring our work closer to practical deployment, we also developed a web application using the Flask framework. This user-friendly interface allows users to upload an image as input, and the trained model processes it to predict whether the face is real or fake. This component demonstrates the feasibility of integrating AI models into real-world applications for on-the-fly DeepFake detection and verification. By combining model experimentation with deployment capability, this study bridges the gap between academic research and practical implementation. Moving forward, further improvements can include real-time video analysis, adversarial robustness, and training on even more diverse datasets to improve generalizability. Overall, this work contributes significantly to the fight against digital misinformation by providing a technically sound and application-ready solution for DeepFake detection.

REFERENCES

[1] Hannun, H. R. T., Zhang, T., 2022. Deepfake generation and detection, a survey. *Multimedia Tools and Applications*, *81*(5), pp.6259-6276.

[2] Masood, M., Nawaz, M., Malik, K.M., Javed, A., Irtaza, A. and Malik, H., 2023. Deepfakes generation and detection: State-of-the-art, open challenges, countermeasures, and way forward. *Applied intelligence*, *53*(4), pp.3974-4026.

[3] Singh, P. and Dhiman, D.B., 2023. Exploding AI-Generated Deepfakes and misinformation: A threat to global concern in the 21st century. *Available at SSRN 4651093*.

[4] Goyal, H., Wajid, M.S., Wajid, M.A., Khanday, A.M.U.D., Neshat, M. and Gandomi, A., 2025. State-of-the-art AI-based Learning Approaches for Deepfake Generation and Detection, Analyzing Opportunities, Threading through Pros, Cons, and Future Prospects. *arXiv preprint arXiv:2501.01029*.

[5] Ahmed, N.U.R., Badshah, A., Adeel, H., Tajammul, A., Duad, A. and Alsahfi, T., 2024. Visual Deepfake Detection: Review of Techniques, Tools, Limitations, and Future Prospects. *IEEE Access*.

[6] Kosarkar, U., Sarkarkar, G. and Gedam, S., 2023. Revealing and classification of deepfakes video's images using a customize convolution neural network model. *Procedia Computer Science*, *218*, pp.2636-2652.

[7] Patel, Y., Tanwar, S., Bhattacharya, P., Gupta, R., Alsuwian, T., Davidson, I.E. and Mazibuko, T.F., 2023. An improved dense CNN architecture for deepfake image detection. *IEEE Access*, *11*, pp.22081-22095.

[8] Soudy, A.H., Sayed, O., Tag-Elser, H., Ragab, R., Mohsen, S., Mostafa, T., Abohany, A.A. and Slim, S.O., 2024. Deepfake detection using convolutional vision transformers and convolutional neural networks. *Neural Computing and Applications*, *36*(31), pp.19759-19775.

[9] Altaei, M.S.M., 2023. Detection of deep fake in face images-based machine learning. *Al-Salam Journal for Engineering and Technology*, *2*(2), pp.1-12.

[10] Pryor, L., Dave, R. and Vanamala, M., 2023. Deepfake detection analyzing hybrid dataset utilizing cnn and svm. *arXiv preprint arXiv:2302.10280*.

[11] Setyaningrum, A.H. and Saputro, A.E., 2024, October. Deepfake Video Classification Using Random Forest and Stochastic Gradient Descent with Triplet Loss Approach Algorithm. In *2024 12th International Conference on Cyber and IT Service Management (CITSM)* (pp. 1-6). IEEE.

[12] Hamza, A., Javed, A.R.R., Iqbal, F., Kryvinska, N., Almadhor, A.S., Jalil, Z. and Borghol, R., 2022. Deepfake audio detection via MFCC features using machine learning. *IEEE Access*, *10*, pp.134018-134028.