International Journal of Multidisciplinary Research and Growth Evaluation.

Efficient Deepfake Image Detection Model Based on MobileNetV2

K Sahithya [1*], J Keerthana [2], K Sunil Joshi [3], R Khadar Basha [4]
[1-4] Student, Department of Information Technology, Kallam Haranadhareddy Institute of Technology (Autonomous), Guntur, Andhra Pradesh, India

* Corresponding Author: **K Sahithya**

Article Info

ISSN (online): 2582-7138
Volume: 06
Issue: 02
March-April 2025
Received: 04-02-2025
Accepted: 27-02-2025
Page No: 650-653

Abstract
Deepfake images have emerged as a notable concern in today's digital era. They pose significant threat to digital privacy, security and integrity. Deepfake images have profound impact on society. Deepfake images are used to create false content, which is then disseminated through social media apps. It is essential to detect them to combat the growing threat of deepfake technology. Deepfake image detection is a challenging issue, as advancements in artificial intelligence have led to the creation of highly realistic images that are difficult to identify. Several machine learning and convolutional neural network-based models have been proposed for deepfake image detection; however, their accuracy remains limited. In this paper, we propose a deepfake detection model based on MobileNetV2 to improve classification accuracy and efficiency. The proposed model is lightweight and efficient, making it well-suited for deepfake detection by effectively capturing complex patterns and inconsistencies in images. Our results and comparative analysis demonstrate that the proposed MobileNetV2-based model exhibits better performance than theexisting VGG16 and ResNet models in terms of accuracy.This work highlights the potential of MobileNetV2 in addressing the growing challenge of deepfake image detection.

DOI: https://doi.org/10.54660/.IJMRGE.2025.6.2.650-653

Keywords: Deepfake Images, CNN, Machine Learning, Artificial Intelligence and MobileNetV2

1. Introduction

The domain of artificial intelligence (AI) is continuously evolving, driving significant advancements in the creation of deepfake images through sophisticated deep learning models. AI based deepfake image generators are designed to learn complex patterns and features from real images, enabling them to produce highly realistic synthetic images [1, 3]. These generators producing fake images that are increasingly difficult for both humans and machines to distinguish from real ones. This rapid progress in AI-driven image generation raises serious concerns about misinformation, identity theft, and the overall trustworthiness of digital content. It is essential to address the challenges posed by deepfake images. Many researches proposed machine learning and deep learning-based models to detect deepfake images [4]. Existing models often struggle with effective pattern extraction for identifying deepfake images due to the high level of realism and complexity involved in their generation. Deepfake creators use advanced techniques which continuously improve their ability to mimic facial expressions, textures, and lighting conditions, making it challenging for detection models to isolate distinguishing features [5]. Traditional models may fail to capture subtle artifacts, inconsistencies, and unnatural patterns introduced during the generation process, leading to increased false positives and false negatives. Enhancing pattern extraction capabilities through more sophisticated feature mapping, multi-scale analysis, and adaptive learning mechanisms is crucial for improving the accuracy and reliability of deepfake detection systems.

In this work, we used MobileNetV2 as a base model with frozen weights, that serves as a powerful feature extractor for deepfake image detection. By freezing the weights, the pre-trained model retains the learned representations from large datasets, allowing it to extract high-level features such as edges, textures, and patterns from input images without further modification. The extracted high-level features are forwarded to additional classification layers, enabling the model to identify subtle artifacts and

inconsistencies present in deepfake images. This transfer learning strategy not only speeds up training but also improves generalization, as the model benefits from the robust feature extraction capabilities of MobileNetV2.

2. Related Work

This section describes the existing techniques used for detecting deepfake images along with their pros and cons.

A. A. Maksutov *et al* [6] proposed machine learning based deepfake detection model. The efficiency and accuracy of machine learning-based deepfake detection models are generally low. These models struggle with large datasets and high-resolution images, leading to slower inference and higher memory usage.

Zahra NazemiAshani *et al* [7] performed a comparative analysis of three CNN models—VGG16, VGG19, and ResNet50 used for the deepfake image detection using a dataset of 1,200 images. This work employed transfer learning to train these models and evaluated their performance based on accuracy, recall, precision, and F1 score. The limitation of this approach is that the high accuracy achieved by VGG19 might be specific to deepfakes generated by FaceApp and may not generalize well to deepfakes created using different tools or techniques.

Raza A *et al* [8] proposed DFP approach to detect deepfake images. DFP model is a hybrid model based on VGG16 and CNN and uses transfer learning. In this work, DFP was compared with existing machine learning and CNN based models. DFP produced around 95% of accuracy in detecting the deepfake images. This study does not provide analysis of computational resources required for training and deploying DFP model.

3. Proposed Methodology

This section demonstrates our proposed model based on MobileNetV2, which is designed for the efficient detection of deepfake images. The architecture of the proposed deepfake detection model based on MobileNetV2 is summarized as follows:

Step 1: MobileNetV2 base model (Frozen Weights):

- The MobileNetV2 model is used as the backbone for feature extraction.
- Its pre-trained weights are frozen during the initial training phase to retain the knowledge acquired from large-scale image datasets.
- This allows the model to extract high-level spatial and contextual features from input images effectively.

Step 2: Global average pooling (GAP) layer:

- After extracting features, a Global Average Pooling layer reduces the spatial dimensions of the feature maps.
- This layer minimizes the number of parameters, reducing overfitting and computational complexity while preserving essential information

Step 3: Fully connected layers:

- The pooled features are passed through a series of fully connected (dense) layers.
- ReLU activation is applied to introduce non-linearity and improve learning capacity.
- Batch normalization is used to stabilize training by normalizing layer inputs, accelerating convergence.
- Dropout layers are introduced to prevent overfitting by randomly deactivating a fraction of neurons during training.

Step 4: Output Layer:

- A final dense layer with a SoftMax activation function is used to classify the input image into two categories: real or fake.
- The SoftMax function outputs a probability distribution, enabling the model to make confident predictions.

Learning rate adjustment strategy:

The model's learning rate is dynamically adjusted using a Learning Rate Scheduler callback to ensure smooth and stable training:

- **Initial Phase:** The learning rate is set to 0.001 to enable rapid learning during the early stages of training.
- **Mid Training Phase:** After a few epochs, the learning rate is reduced to 0.0001 to refine the learned features and avoid overshooting the optimal solution.
- **Fine-Tuning Phase:** In the later epochs, the learning rate is further reduced to 0.00001 to fine-tune the model, enhancing its ability to capture subtle patterns and improving overall performance.

The algorithm steps are illustrated in the figure 1.

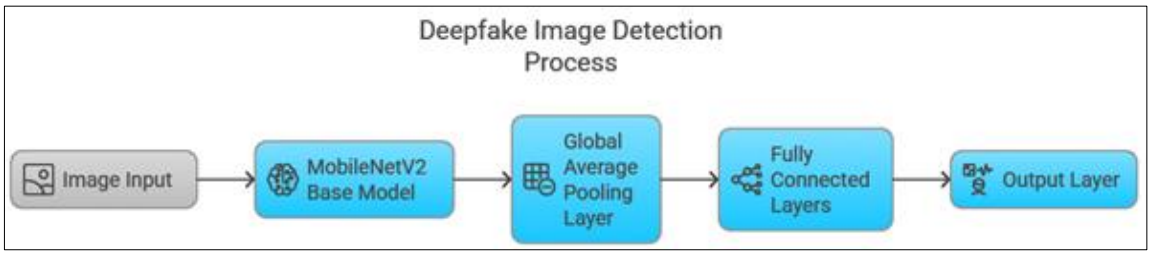

Fig 1: Proposed Model's Architecture

The steps outlined below are essential for improving the performance, accuracy, and generalization of the deepfake detection model.

Preprocessing and data preparation

Before feeding images into the CNN, extensive preprocessing is performed to enhance the model's generalization capability. In deep learning applications, the quality of input data significantly impacts accuracy and helps reduce overfitting.

a) Image resizing and normalization

- The first step in preprocessing involves resizing all input images to a fixed resolution of 96 × 96 pixels to maintain uniformity across the dataset.
- Since deep learning models expect numerical inputs, the pixel values of images are normalized using the rescale=1. /255 parameter in Image Data Generator. This scales the pixel intensity values to the range (0, 1), which helps accelerate convergence and prevents large gradients from destabilizing training.

b) Data Augmentation

To enhance generalization and prevent overfitting, data augmentation techniques are applied. This ensures that the model learns to recognize deepfake patterns despite variations in input data. The following augmentation techniques are used:

- **Horizontal Flip:** Randomly flips images along the horizontal axis to simulate variations in facial orientation.
- **No Vertical Flip:** Vertical flipping is avoided as it creates unrealistic face images.
- **Random Rotations, Scaling, and Shifts:** Introduces slight changes in position, size, and rotation to prevent the model from over-relying on fixed features.

By augmenting the dataset, the model becomes more resilient to adversarial manipulations and learns to identify deepfake artifacts under varying conditions.

c) Data loading and splitting

- After preprocessing, the dataset is loaded into memory using the flow from directory () method, which automatically assigns labels based on the directory structure. The dataset is split into 80% training and 20% validation subsets, ensuring that the model's performance is evaluated on unseen data during training. This helps in early detection of overfitting and allows tuning of hyperparameters accordingly.

d) Dataset visualization and bias detection

- OpenCV (cv2) is used for loading and visualizing sample images from the dataset. Real and fake face samples are displayed to help researchers analyze the dataset distribution and detect potential biases. Detecting biases is crucial because CNNs may unintentionally learn dataset-specific artifacts rather than genuine deepfake patterns. A model that exploits such biases could fail when tested on deepfakes generated using different architectures.

e) Training the Model

- The model. Fit () function is used to train the model through backpropagation, updating the network weights iteratively. The model is trained for 20 epochs, meaning the entire dataset is passed through the network 20 times. Loss and accuracy values are recorded at each epoch, providing insights into the model's learning behavior and convergence trends. If the training loss decreases while validation loss increases, it indicates overfitting. This can be mitigated using dropout layers and early stopping mechanisms.

f) Saving and Deployment

- After training, the model is saved. The saved model can be loaded and used for real-time inference on new images or videos without retraining. This makes the model a valuable tool for detecting synthetic media and combating misinformation in real-world applications.

4. Results and Discussions

The proposed model is executed using TensorFlow for building, training, and inference. OpenCV is used for loading and visualizing the results, helping to analyze the dataset and identify patterns or biases. We have designed a user-friendly interface that allows users to easily interact with the deepfake detection system. The interface accepts an image as input, processes it using the proposed MobileNetV2-based model, and outputs whether the image is deepfake or real. This intuitive design ensures a smooth user experience, making the detection process accessible and efficient even for non-expert users.

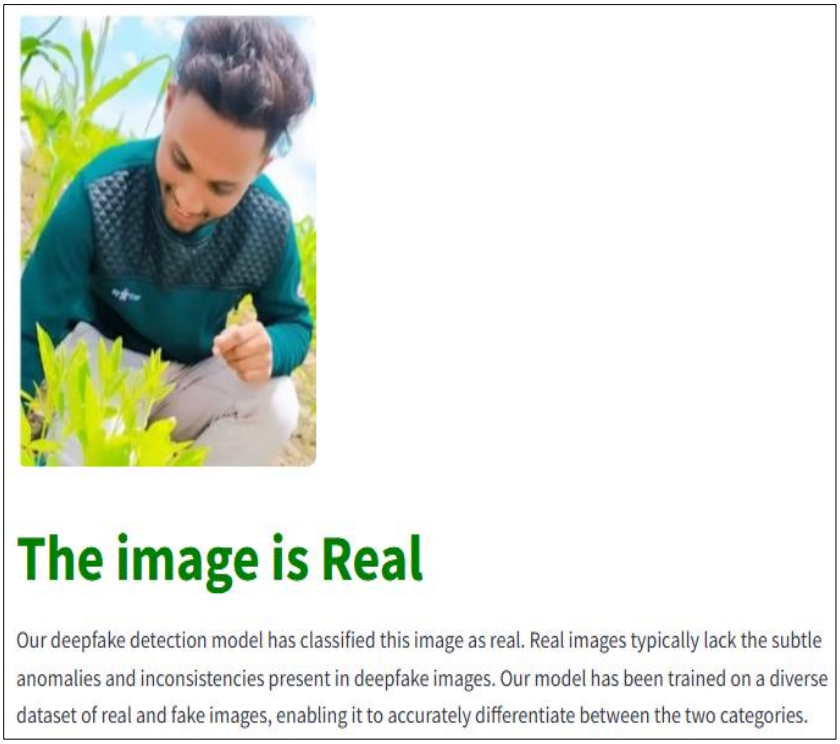

Fig 2: Sample Output: Real Image

Fig 3: Sample Output: Fake Image

The performance of the proposed model is compared with the existing VGG16 and ResNet models using various evaluation metrics. The comparative analysis is summarized in Table 1.

Table 1: Performance Comparison of MobileNetV2, VGG16, and ResNet Models

Model	Precision	Accuracy	F1-Score	Inference Speed
VGG16	88%	90%	89%	Slow (large model size)
ResNet	94%	96%	95%	Moderate
Proposed Model	92%	95%	93%	Fast (Optimized Operations)

The results demonstrate that the proposed model based on MobileNetV2 is highly efficient, offering superior accuracy and precision. Its lightweight architecture and faster inference speed make it suitable for real-time applications, especially on resource-constrained devices like mobile phones.

5. Conclusion

In this work, we proposed a deepfake detection model based on MobileNetV2 to classify given image as real or deepfake image. The proposed model demonstrated high accuracy, precision, and F1-score, then the existing models such as VGG16 and ResNet. Furthermore, proposed model exhibits better performance in terms of inference speed and computational efficiency making it ideal for real-time applications. Additionally, the user-friendly interface enhances accessibility by providing a seamless experience for users to detect deepfake images accurately.

6. References

1. Malik M, Kuribayashi S, Abdullahi SM, Khan AN. DeepFake detection for human face images and videos: A survey. IEEE Access. 2022;10:18757-75. doi: 10.1109/ACCESS.2022.3151186.
2. Kumar N, P P, Nirney V, G V. Deepfake image detection using CNNs and transfer learning. 2021 International Conference on Computing, Communication and Green Engineering (CCGE); 2021; Pune, India. p. 1-6. doi: 10.1109/CCGE50943.2021.9776410.
3. Mary A, Edison A. Deepfake detection using deep learning techniques: A literature review. 2023 International Conference on Control, Communication and Computing (ICCC); 2023; Thiruvananthapuram, India. p. 1-6. doi: 10.1109/ICCC57789.2023.10164881.
4. Khalil HA, Maged SA. Deepfakes creation and detection using deep learning. 2021 International Mobile, Intelligent, and Ubiquitous Computing Conference (MIUCC); 2021; Cairo, Egypt. p. 1-4. doi: 10.1109/MIUCC52538.2021.9447642.
5. Nirkin Y, Wolf L, Keller Y, Hassner T. DeepFake detection based on discrepancies between faces and their context. IEEE Trans Pattern Anal Mach Intell. 2022;44(10_Part_1):6111-21.
6. Maksutov AA, Morozov VO, Lavrenov AA, Smirnov AS. Methods of deepfake detection based on machine learning. 2020 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus); 2020; St. Petersburg and Moscow, Russia. p. 408-11. doi: 10.1109/EIConRus49466.2020.9039057.
7. NazemiAshani Z, *et al* Comparative analysis of deepfake image detection methods using VGG16, VGG19, and ResNet50. J Adv Res Appl Sci Eng Technol. 2025;47(1):16-28. doi: 10.37934/araset.47.1.1628.
8. Raza A, Munir K, Almutairi M. A novel deep learning approach for deepfake image detection. Appl Sci. 2022;12:9820. doi: 10.3390/app12199820.
9. Soudy AH, Sayed O, Tag-Elser H, *et al* Deepfake detection using convolutional vision transformers and convolutional neural networks. Neural Comput Appl. 2024;36:19759-75. doi: 10.1007/s00521-024-10181-7.
10. Almars A. Deepfakes detection techniques using deep learning: A survey. J Comput Commun. 2021;9:20-35. doi: 10.4236/jcc.2021.95003.