Deepfake AI Detection System Using ML

Yuvraj Singh Pawar*, Rushikesh Agarwal*, Soham Dahatonde*
Under the guidance of "Dr. Anant Kaulage"
**School of Computing, MIT Art, Design and Technology University, Pune, India*

***Abstract*—The increasing number of AI-generated contents raises questions related to the veracity of the information provided by it and the overall trust of people. With the development of modern deepfakes technology, it becomes possible to create highly realistic images that become almost impossible to identify as synthetic. In this paper, we propose a machine learning approach for detecting AI-generated images through a deep convolutional neural network model. Our model is designed as an EfficientNet-B0 structure that was initially pretrained on ImageNet and then fine-tuned for binary classification of images into two classes: authentic images and synthetic. The algorithm utilizes additional heuristic approaches for increasing robustness, such as frequency domain analysis with the use of FFT transform, as well as statistical analysis. The whole project was developed and presented using a FastAPI backend with a web interface for image uploads and real-time processing. Combining deep learning algorithms with auxiliary techniques and providing a convenient web platform allows us to develop an efficient and practical approach towards detection of deepfakes images.**

***Index Terms*—Deepfake Detection, EfficientNet, Machine Learning, AI-Generated Images, Image Classification, Computer Vision.**

I. INTRODUCTION

Deepfakes are synthetic or tampered media created by artificial intelligence methods, resulting in visually convincing but false media. The fast development of generative models has made the proliferation of this type of content possible and opened new opportunities for misuse, including spreading false news, identity fraud, and tampering with digital content [4].

The widespread usage of digital channels has made images a common tool for communication and information exchange. However, it has also made them susceptible to abuse. Therefore, content producers, reporters, and social media users face difficulties in validating the credibility of the presented visual content, underlining the necessity for accurate recognition approaches [6].

There are many types of deepfake images, such as face-swap photos, completely artificial faces, or enhanced media that mimics the real-world data closely. As this technology advances, discerning between real and fake images based on visual analysis or elementary algorithms becomes increasingly complicated.

To tackle this problem, methods relying on artificial intelligence have been widely considered. Neural network architectures, such as convolutional neural networks (CNNs), can be trained to recognize sophisticated patterns and features distinguishing genuine images from those created by AI algorithms [1].

The objective of this paper is to introduce a system for detecting images generated via artificial intelligence with the help of deep learning algorithms, namely an EfficientNet-B0 model [5]. It will be trained to distinguish between real images and images generated through artificial intelligence means by utilizing transfer learning for the purpose of binary classification. Additionally, heuristic methods will be employed.

II. METHODOLOGY

The following section discusses how this proposed detection system is designed and implemented. This is done through a set of curated data that includes real images labelled as “Real” and artificial intelligence generated images labelled as “Fake.” The algorithm works by doing binary classification, which can be used for the real-time detection of images and videos.

2.1 Dataset Description

The dataset utilized in this research comprises more than 18,000 labelled pictures, which are separated into two categories:

- **Labelled real images**
- **Labelled AI-generated images**

These images are split into three sets: training, validation, and test datasets for accurate evaluation of the models [6].

Two modes of operations have been implemented in this design:

- **Training workflow:** The algorithm trains the deep learning model to discriminate between real and synthetic images on the labelled dataset.
- **Inference workflow:** Users can upload their images or video for online prediction via the pre-trained model.

This approach facilitates efficient model training as well as real-life application.

2.2 Preprocessing

Appropriate preprocessing plays a crucial role in preparing the data for deep learning modeling.

2.2.1 Image Resizing and Normalization

All images are resized to 224 × 224 pixels according to the input requirements of the model. Normalization is performed based on the default ImageNet mean and standard deviation [5].

2.2.2 Data Cleaning

Corrupted or invalid image files are automatically removed from the dataset to prevent possible errors during training.

2.2.3 Dataset Partitioning

Approximately 80% of the data is used for training purposes, while validation and testing sets account for 10% each.

2.2.4 Data Augmentation

Several transformations are applied to the dataset during the training process to improve generalization capability.

2.2.5 Feature Representation

The proposed model is pixel-based, and there is no need for any feature engineering. Feature representation for CNNs happens automatically from the data fed to the model [1].

2.3 Deep Learning Model

The primary deep learning model chosen for this application is EfficientNet-B0. It is a convolutional neural network known for being efficient and highly accurate for image classification tasks [5]. The transfer learning technique involves using the pre-trained ImageNet weights for initializing the model. The feature extraction part of the network is frozen, while the classification head is customized to perform binary classification.

2.3.1 Model Training

The input images are classified via the EfficientNet-B0 model that learns patterns that differentiate between real and AI-generated pictures. The classification head has the following layers:

- **Fully connected layers**
- **Dropout layers for regularization**
- **Batch normalization for stable training**

Training is done using the Adam optimizer and cross-entropy loss function. Learning rate scheduling is performed based on validation results.

2.3.2 Model Prediction

Once the model is trained, it can be used to evaluate images on validation and test sets. The output of the model on each input image is a probability score that shows whether the input image is AI generated or not.

2.3.3 Performance Metrics

The following criteria are adopted for measuring classification performance of the model:

- **Accuracy:** Rate of successful classification of all test images.
- **Precision:** Indicator of the level of false-positive suppression.
- **Recall (Sensitivity):** Metric describing classifier sensitivity.
- **F1-Score:** Metric balancing precision and recall.

They are widely applied in classification evaluation of the deepfake detection models [6].

2.3.4 Confusion Matrix Analysis

Confusion matrix analysis is applied to visualize the results of classification performance estimation and identify possible misclassification instances.

2.4 Auxiliary Heuristic Analysis

Besides machine learning-based image classification, the algorithm uses heuristics to increase the reliability of its operation:

- **FFT-based Analysis:** Features in frequency domain are estimated using Fast Fourier Transform technique to identify abnormal high frequency artifacts.
- **Statistical Analysis:** Natural vs unnatural images' distributions can be assessed through pixel variance or mean value comparison.

These mutually beneficial approaches are consistent with previous literature in the field of media forensics and improve accuracy [4].

2.5 Video Detection Using the System

For video processing, the system captures frames from videos at equal intervals. Frames are evaluated using the trained model independently. A majority voting scheme is adopted to derive the final outcome in terms of video using the results from the frames.

2.6 System Implementation and Interface

The implementation process uses FastAPI as the back-end architecture, which provides the following APIs: image prediction, video analysis, snapshot-based detection, and history logging. A web-based interface allows users to upload media files and visualize results in real time.

2.6.1 System Usability

The system is built to be scalable and lightweight to facilitate smooth implementation, user-friendly to

allow easy use for both technical and non-technical users, and highly efficient to allow real-time predictions. Output presentation makes the system ideal for use by professional journalists, researchers, and others in content validation.

III. RESULTS AND ANALYSIS

This section presents the evaluation of the proposed EfficientNet-B0-based deepfake detection model and analyzes its performance on unseen data using standard classification metrics and qualitative observations.

3.1 Model Evaluation

The model has been trained using around 80% of the dataset and validated on the other validation and testing datasets. The important evaluation metrics are:

- **Accuracy:** It is the total ratio of correct predictions made by the model. High accuracy means that the model is successfully trained for distinguishing between the real and fake images.
- **Precision:** The percentage of predicted images out of all AI-generated images. It measures the number of false positive cases.
- **Recall (Sensitivity):** The ratio of correctly detected AI-generated images out of all AI-generated images. It is highly essential to detect any deepfake image.
- **F1-Score:** It is the harmonic average of precision and recall values. It is one of the important metrics for evaluating imbalanced classes.
- **Loss:** The cross-entropy loss can be calculated using the predicted probability and actual class labels. The decreasing trend in the loss value means that the model is efficiently trained.

3.2 Confusion Matrix Analysis

A confusion matrix is created from the validation dataset to illustrate the classification accuracy for both classes (real and fake). The confusion matrix shows accuracy in classifying real and fake images and errors made in predicting the correct category. By analyzing the confusion matrix, the efficiency of the model can be assessed, helping determine whether the model favors any particular class or has difficulty with certain input images.

3.3 Sample Predictions and Qualitative Analysis

For further assessment of the model, sample validation images are chosen to compare their predicted labels and ground truths. Images were first processed (resized and normalized) before being sent for analysis in the EfficientNet-B0 model. This analysis offers valuable information on the model's behavior, helping uncover patterns within wrong classifications. Most wrong predictions were linked to low-quality images, uneven textures or lighting conditions, or confusing visual data.

3.4 Video Detection Performance

The method involves extending image-based classification using video frames by first capturing frames at regular intervals, followed by classification of these frames. The majority voting technique is used to ascertain the output for the entire video. This ensures that even if some frames have errors during classification, the outcome will be unaffected. From empirical observations, the algorithm exhibits consistent performance when dealing with videos.

3.5 Real-World Applicability

Model training is done by the application developed using the FastAPI framework, which allows online inference of images and videos. Main functions include image uploading (supports JPG, PNG), image preprocessing (resizing and normalization), image prediction (classification into real or fake category), result visualization on the web interface, and results exportation for further processing. The deployment is practical for applications such as verification and filtering of uploaded content.

3.6 Summary

An EfficientNet-B0 model is an efficient tool in detecting the difference between real photos and those generated by AI. Key findings include: high accuracy of prediction with balanced recall and precision; reliable detection confirmed by confusion matrix results; good performance on both images and video files; and practical implementation in the web environment. The outcome suggests that the proposed system is feasible and scalable in identifying AI-generated images.

IV. CONCLUSION

The current work presents a web application for detecting AI-created media using a convolutional neural network built with EfficientNet-B0 architecture. This project aims to address practical problems faced in distinguishing between real photos and generated ones by creating a model trained with a dataset containing both types of media objects [5].

Performance evaluation shows promising results since the model achieved high accuracy, precision, recall, and F1-scores when processing unseen validation and testing images. Hence, the current work is in line with modern developments in this area of study, where the use of machine learning algorithms allows obtaining reliable predictions [6].

The web app uses a FastAPI-powered backend and a simple front-end interface that enables uploading media files and receiving predictions in real-time mode. Besides applying deep learning methods, other

tools used in forensic media studies to identify deepfakes are also included into the workflow of the project [4].

The suggested technique proves that a combination of deep learning models with other methods of analysis and user-targeted deployment yields an approach to detect deepfakes that is both feasible and scalable. The model can process different types of input, such as images and video frames.

V. FUTURE WORK

Even though the suggested system provides efficient performance to identify fake AI-created pictures, there are additional improvements that could further increase efficiency and practicality of the system.

First, one could consider real-time deepfake detection, when the system analyzes not static pictures but continuously incoming data in the form of video frames. Such a solution will greatly benefit applications such as live content filtering on social media platforms.

Model explainability is another key improvement. Utilizing technologies like Grad-CAM, SHAP, or LIME will allow identifying specific areas of the picture that affect the decision made by the system [4].

The system can also be extended to analyze various formats of digital media, such as videos, GIFs, and even live streams, to allow for deepfake detection on a variety of different digital platforms. By applying frame-level analysis along with temporal modeling, improvements in the accuracy of detection for video content could be achieved.

Future studies could leverage data augmentation and class weighting approaches for loss functions to address class imbalance and dataset diversity issues [6]. Continuous learning and model updating based on evolving technology is another important research opportunity.

By implementing the system using cloud or distributed computing technology, it is possible to make the system capable of efficiently handling large volumes of data and real-time processes. Improvements can also be made through better visualizations and interface design, including ROC and Precision-Recall graphs and feature activation maps. Lastly, security aspects could be solved through proper management of sensitive information and compliance with various standards.

VI. ACKNOWLEDGMENT

The authors would like to thank the faculty and staff members of the Department of Computer Science at MIT ADT University for all the support provided throughout this research. Particular thanks go to Prof. Anant Kaulage, our faculty guide, whose guidance and suggestions have been extremely valuable throughout the entire process of this research. Additionally, the open source libraries PyTorch, torchvision, and scikit-learn were essential to constructing the deepfake detection system described in this paper.

REFERENCES

[1] A. Rössler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Nießner, “FaceForensics++: Learning to Detect Manipulated Facial Images,” in Proc. IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, South Korea, Oct. 2019, pp. 1–11.

[2] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, “Analyzing and Improving the Image Quality of StyleGAN,” in Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, June 2020, pp. 8110–8119.

[3] S. Verdoliva, “Media Forensics and DeepFakes: An Overview,” IEEE Journal of Selected Topics in Signal Processing, vol. 14, no. 5, pp. 910–932, Aug. 2020.

[4] Y. Liu, X. Chen, J. Wang, and Z. Wang, “Deepfake Detection: Current Challenges and Future Directions,” IEEE Access, vol. 9, pp. 147–167, 2021.

[5] M. Tan and Q. Le, “EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks,” in Proc. International Conference on Machine Learning (ICML), Long Beach, CA, USA, June 2019, pp. 6105–6114.

[6] H. Wang, Z. Wu, X. Li, Z. Wang, and S. Wang, “CNN-Generated Images Are Surprisingly Easy to Spot… For Now,” in Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, June 2020, pp. 8695–8704.

[7] T. A. Dal Pozzolo, O. Caelen, R. A. Johnson, and G. Bontempi, “Calibrating Probability with Undersampling for Unbalanced Classification,” in Proc. IEEE Symposium Series on Computational Intelligence (SSCI), Cape Town, South Africa, Dec. 2015, pp. 159–166.