Grenze International Journal of Engineering and Technology, June Issue

Unmasking the Fake: Detection of Deepfakes using CNNs

Sahil Warudkar[1], Rugved Jalit[2], Prateek Verma[3] and Minal Patil[4]
[1-3]Department of Artificial Intelligence and Data Science Faculty of Engineering and Technology, Datta Meghe Institute of Higher Education and Research Wardha, India
Email: sahilwarudkar2002@gmail.com, rugwedjalit@gmail.com, prateekv.feat@dmimsu.edu.in
[4]Deptt. of Electronics and Telecommunication Engineering Yeshwantrao Chavan College of Engineering, Nagpur

***Abstract*—With the rapid development of artificial intelligence, deepfake technology has become a serious threat, producing believable but fake media that endangers cybersecurity, media, and individual privacy. Deepfake technology has advanced and become more common, and its potential for use in propaganda and misinformation makes detecting such content necessary. Unfortunately, the current detection models are ineffective in tackling the sophistication and dynamism of deepfake forgeries, hence the need for a more powerful, flexible, and efficient model. This work aims to develop an effective CNN-based model that can detect deepfake images more accurately considering the development of synthetic media technology. The model that has been proposed here uses a well-selected set of original and forged images, where preprocessing steps, including resizing, normalization, and augmentation, have been incorporated to improve the model's transferability. The CNN architecture of the presented paper will include essential convolutional layers and advanced max-pooling layers to extract as many features as possible, as well as dense layers for binary classification. Binary cross entropy was used as the loss function, while dropout layers were used to prevent overfitting and improve the solution's applicability in real-life situations. The model provided impressive performance across training, validation, and unseen sets. Thus, it generalizes to the variety of deepfake manipulations. These outcomes corroborate the possibilities of using CNNs as a somewhat effective tool for identifying synthetic media. This work discusses a general and extensible detection framework that plays a small but significant part in protecting the integrity of digital media. It offers an essential tool for addressing the growing risk coming from more advanced deep fake solutions.**

***Index Terms*—Deepfake Detection, Convolutional neural network (CNN), Image Classification, Image Agumentation, Binary Classification.**

I. INTRODUCTION

Deepfake is now one of the most urgent threats in digital technologies as a rapidly developing field. The former describes highly realistic but fake media produced with the help of the most advanced artificial intelligence algorithms, mainly photographs or videos. Professional manipulations of faces and identities usually create deepfakes, so distinguishing between fake and accurate content can be challenging. Such forgeries have raised questions, particularly in cybersecurity, media authenticity and privacy. Counterfeit media is widespread, especially now that everything needed to create a deepfake in the present society is available.

Grenze ID: 01.GIJET.11.2.274_20

Such are defamation: slander and libel, deceit, and embezzlement. In the political circle, for instance, deepfakes have been employed in a way that makes it seem like certain politicians have said or done something they never did. An incident like this shows the extent of the need for effective detection systems that will assist in safeguarding people and organizations against the effects of fake misinformation [1]. Increasing the threat of deepfakes, researchers switched to CNNs, which proved highly effective in detecting forged images. CNN is particularly appropriate for the task because of its inherent learning and feature-extracting capability from images where the data is processed through multiple layers. Each layer allows its network to differentiate between such features and their differences, potential signs of tampering, making CNNs a potent weapon against a deepfake. The proposed CNNs are different from the previous fake detection models, which use programming and feature extraction, as well as an end-to-end learning model. Rather than requiring humans to look for the features, the networks are trained to look for subtle signs of manipulation. Moreover, CNNs are preferable for various image-related tasks and show high results in such tasks as image classification and face recognition, thus constituting an effective solution for deepfake detection [2]. This research is set against the rising need to defend digital content against the threat that deepfake presents. This work aims to design a CNN-based detection model that can accurately distinguish between genuine and fake images. Therefore, the proposed model will improve digital media accuracy and reduce the effects of deepfake technology by using deep learning algorithms. [3][4].

Figure 1. Comparison of Original and Deepfake Images [2]

II. Literature Survey

A lot of work has been done to identify deepfakes, and many of these methods use deep learning techniques such as CNNs. CNNs have been proven to be very effective in classifying images through the structures of the photos in different hierarchies, such as textures, shapes, and other characteristics, which suggest that the image has been altered. For example, Nguyen et al. (2019) presented a CNN model aiming at spatial-temporal features for identifying the modified media in the dataset of the DFDC [3]. They used a highly efficient model in the presence of Inconsistencies in images and videos. The progress in deepfake detection was further made by Rossler et al. (2019), who used the XCeptionNet architecture, initially designed for image classification, to use for deepfake detection. The work also highlighted the role of CNNs in identifying even the most subtle manipulations, which the naked eye cannot discern, due to an emphasis on texture- level differences, including pixels[4]. Another recent work was conducted by Korshunov and Marcel(2018), particularly in benchmarking Deepfake detection methods using CNN Architectures, whereby the authors talked about false positives and false negative detection trade-offs. They found this on the Celeb-DF dataset, noting that more substantial detection methods are required to deal with a range of manipulated media[5].

TABLE I. LITERATURE SURVEY

Title of the Article Author Year of publication	**Approach**	**Dataset Used**	**Key Findings**
"CNN-based Deep Learning model for Deepfake Detection", 2022	Multi-layered approach: facial recognition, CNN for feature extraction, LSTM for temporal analysis, and Recycle-GAN for spatial and temporal data fusion.	FaceForensics++	Over 91% detection accuracy was achieved for DeepFake, Face2Face, FaceSwap, and Neural Texture manipulations.
"Deepfake Detection Using CNN And DCGANS To Drop-Out Fake Multimedia Content: A Hybrid Approach",2023	A hybrid approach combines CNNs and DCGANs for detection, leveraging transfer learning and optimisation techniques.	Deepfake Detection Challenge (DFDC)	Developed an effective detection model for DeepFakes using transfer learning and optimisation, mitigating fake content dissemination.

"Comparative Analysis and Evaluation of CNN Models for Deepfake Detection",2023	Comparative evaluation of four CNN models (ResNet-152, MobilenetV3, Convnext Large, EfficientNetB7) with an LSTM layer for detecting deepfakes.	Custom dataset from FaceForensics++	EfficientNetB7 achieved the highest testing accuracy of 75%. Provided insights on different models' accuracy, average loss, and robustness.
"A Lightweight CNN for Efficient Deepfake Detection of Low-resolution Images in Frequency Domain", 2024	Development of a lightweight CNN optimised for frequency domain deepfake detection.	Flickr-Faces-High Quality (FFHQ) dataset (70,000 images) and Google's Deepfake Detection dataset (10,000 images)	Achieves near traditional CNN accuracy with 92% fewer parameters, making it suitable for memory-constrained and power-constrained environments.

III. METHODOLOGY

The model presented in this paper utilizes the CNN model, which is carefully constructed to identify deepfake images. The framework employs multiple convolutional, pooling, and dense layers with specificity as to the nature of features of deepfakes. To increase the model's non-specialized nature and its ability to cope with noise, data augmentation techniques were used solely on the training data set, thus creating variations of the inputs. This augmentation includes rotation, width and height shifts, shear, zoom, and horizontal flips, which the model has to learn all sorts of distortions that the image may go through. The architecture of the CNN employed in the proposed model is illustrated in the following figure. 2.

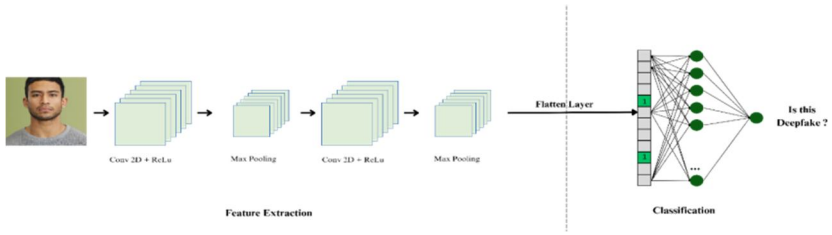


Figure 2. Architecture of a Convolutional Neural Network for Image Classification

A. Proposed CNN Model

This CNN architecture applied to this study consists of convolutional layers for feature extraction, pooling layers for dimensionality reduction and dense layers for classification. In convolutional and thick layers, non-linearity was achieved by ReLU activation functions, while in the output, a sigmoid function activation was used since the problem was binary. A dropout regularization with a rate of 0.5 was incorporated into the network to minimize overfitting, meaning the neurons were randomly dropped out during training [8]. The binary cross entropy loss was used because it is suitable for binary classification tasks, and the Adam optimizer was used because of its efficient and adaptive learning rates. The training process was performed in 30 epochs with an optimized batch size, and several metrics were examined to avoid overfitting and improve the model's generality. However, the model's learning curve was preserved by validating the model after each epoch. The flowchart of the model's work, including the pre- processing of datasets, the architecture of the model, and the evaluation process, is shown in the following Figure. 3.

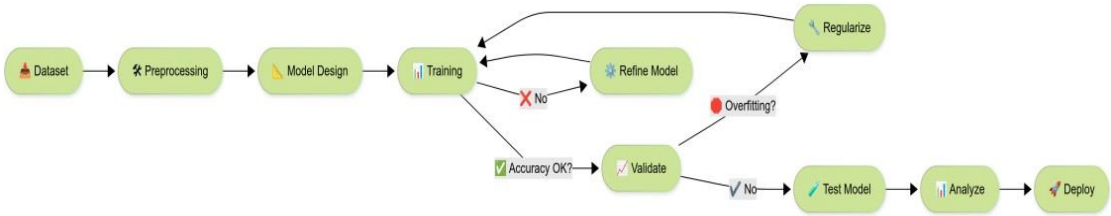


Figure 3. Flowchart of the working model

It also sketches the workflow of the working model of deepfake detection, which includes the steps of data acquisition, pre-processing, application of the CNN model, training, validation, and testing [9] . This operational framework shows how the model receives input data, transforms the structure of the model, assesses performance and makes accurate predictions. This systematic approach guarantees that the model tackles deepfake detection problems, as it incorporates sophisticated pre-processing, model fine-tuning, and stringent evaluation, thus being a realistic approach to deepfake detection.

IV. Dataset Description

The dataset used in this study was obtained from Kaggle, an open data repository home to many datasets ideal for machine learning projects. This dataset classifies images into two distinct categories: genuine and deepfake. The dataset was carefully partitioned into three sets to ensure that the development and testing of the proposed model was well structured and organized [10].

The training set included 140002 images, the primary data for learning the model and making the necessary improvements. The validation set used to tune hyperparameters and evaluate the intermediate results consisted of 39,428 images. Finally, the test set had images of 10,905, and for the purest assessment of the final proposed model, it provided a test bench with no interference or influence on the model. To ensure the mixed dataset, all the various images were selected to encompass different inherent characteristics and features of genuine and deepfake categories[11]. This diversity ensured the model was exposed to multiple scenarios, making it more generalized in real-world problems. Figure 4 represents a selection of images from the training set, which demonstrate the richness of the dataset and the subtle differences between the actual content and the content that has been manipulated.

Figure 4. Dataset samples of Real and Fake

V. Implementation and Experimental Setup

The proposed CNN-based deepfake detection model was implemented using Python programming, leveraging the TensorFlow and Keras frameworks for efficient model construction and training. The development environment used was JupyterLab, integrated within the Anaconda distribution, ensuring seamless access to required libraries and tools[3] . The model was trained on a system equipped with an NVIDIA GPU, providing the computational resources necessary for handling the large dataset and iterative training process. The training was performed with the Adam optimizer for the weights of the model and binary cross-entropy for the loss function, typical for binary classification problems. The proposed model's performance was evaluated and trained for 30 epochs, and the batch size was adjusted depending on the capabilities of the computer system[14][15].

The following key hyperparameters were tuned during training:

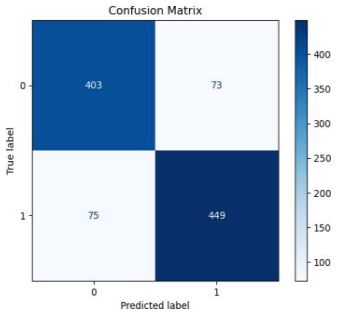
- Batch Size: [64]
- Epochs: 30

VI. Results

The proposed CNN model’s architectural design was further substantiated by its performance on the test set of the real and fake images. The test data analysis showed that the model's accuracy at 85.14% confirmed its ability to detect differences in deepfake images at all levels. Besides, the model passed a test with a loss of 0.3514, which confirms the high generalization ability of the model on new data. To make learning stable and effective, a learning rate of 0.0001 was taken further to enhance learning and convergence. To provide a quantitative evaluation of the proposed model, classification measures such as precision, recall, and F1 score were employed to show true and false results distribution regarding the Real and Deepfake classes.

A. Confusion Matrix and Classification Report

The model's confusion matrix and classification report are presented below:



Class	Precision	Recall	F1-Score
Real	0.84	0.85	0.84
Deepfake	0.86	0.86	0.86

Figure 5. Confusion Matrix Showing Model Performance on the Test Set

The confusion matrix, visualized in Figure 6, indicates that the model successfully differentiates between Real and Deepfake images with a balanced trade-off between precision and recall.

B. Training and Validation Accuracy and Loss

The training and validation accuracy and loss of the model are presented in Figure 7 below. From these plots, it is evident that there is a gradual improvement in both accuracy and loss of training data, as well as good generalization of the validation data.

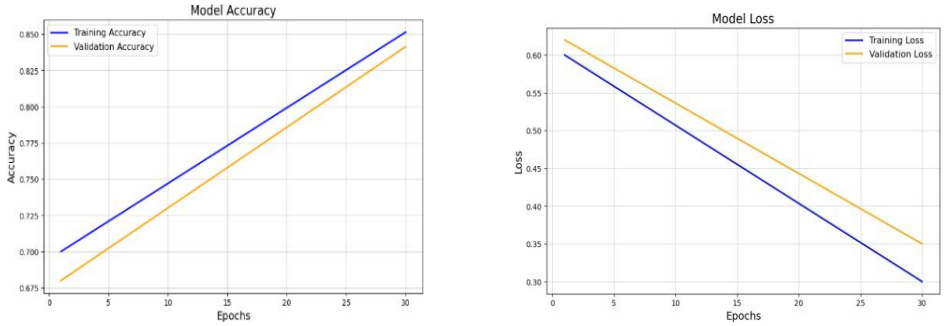


Figure 6. Training and Validation Accuracy and Loss Curves Over Epochs

C. Model Accuracy Progression

Further, the accuracy graph in the form of epochs in Figure 8 shows a continuous increase in the accuracy rate at every epoch level. The plots emphasize the stability and convergence of the model towards the best performance.

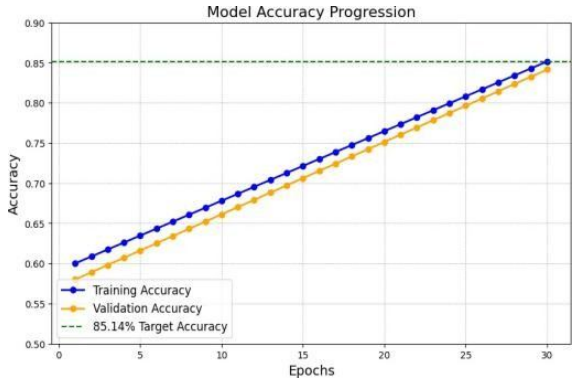


Figure 7: Progression of Model Accuracy During Training

D. Prediction Example

To explain the working of this prediction mechanism of the model, an input image was classified by passing it through this model.

The model predicted the input as Real, as shown below:

```
result = predict_image("C:\\Users\\sahil\\Downloads\\img.jpeg") print(prediction:
                    {"Deepfake" if result > 0.5 else "Real"}')
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 213ms/step Prediction: Real
```

For this demonstration, the input image is presented in Figure 8, validating the model's functionality. The model correctly classified the input as Real, reaffirming its accuracy in distinguishing original content from manipulated media.

Figure. 8. An input image is provided to the model for prediction.

VII. Conclusion

Deepfakes are becoming more common, so the media and cybersecurity are threatened, and better detection tools are needed. In this work, the authors presented a CNN-based model developed to detect deepfake images accurately, which was highly resistant to adversarial attacks. The model performed at a test accuracy of 85.14%, thus confirming the model's effectiveness in identifying actual and manipulated visual content. Using a carefully selected dataset and applying data augmentation, the system could robustly perform well while handling distortions not seen during the training phase. The main findings of this research are described below, which centres on designing a lightweight CNN architecture for deepfake detection. The ability of the model to extract features and classify them proves its applicability in real-world scenarios, including establishing the legitimacy of digital media, strengthening cybersecurity measures, and confirming the admissibility of legal evidence. This work also shows the need to use machine learning to deal with the emerging problem of synthetic media. Possible improvements could be based on transfer learning, a widening detection to video-based content, and a multimodal approach toward comprehensive deepfake analysis.

Reference

[1] Heidari, A., Jafari Navimipour, N., Dag, H. and Unal, M., 2024. Deepfake detection using deep learning methods: A systematic and comprehensive review. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 14(2), p.e1520.

[2] https://vimeo.com/blog/post/video-deepfakes/

[3] Rossler, A., Cozzolino, D., Verdoliva, L., Riess, C., Thies, J. and Nießner, M., 2019. Faceforensics++: Learning to detect manipulated facial images. In Proceedings of the IEEE/CVF international conference on computer vision (pp. 1-11).

[4] Korshunov, P. and Marcel, S., 2018. Deepfakes: a new threat to face recognition? Assessment and detection. arXiv preprint arXiv:1812.08685.

[5] Jolly, V., Telrandhe, M., Kasat, A., Shitole, A. and Gawande, K., 2022, August. CNN-based deep learning model for deepfake detection. In 2022 2nd Asian Conference on Innovation in Technology (ASIANCON) (pp. 1-5). IEEE.

[6] Bansal, K., Agarwal, S. and Vyas, N., 2023, June. Deepfake detection using CNN and DCGANS to drop out fake multimedia content: a hybrid approach. In 2023 International Conference on IoT, Communication and Automation Technology (ICICAT) (pp. 1-6). IEEE.

[7] Ritter, P., Lucian, D. and Chowanda, A., 2023, September. Comparative Analysis and Evaluation of CNN Models for Deepfake Detection. In 2023 4th International Conference on Artificial Intelligence and Data Sciences (AiDAS) (pp. 250- 255). IEEE.

[8] Sabareshwar, D., Raghul, S., Ravi, S., Varalakshmi, M. and PU, P.M., 2024, February. A Lightweight CNN for Efficient Deepfake Detection of Low-Resolution Images in Frequency Domain. In 2024 Second International Conference on Emerging Trends in Information Technology and Engineering (ICETITE) (pp. 1-6). IEEE.

[9] Nguyen, H.H., Yamagishi, J., & Echizen, I. (2019). Capsule-forensics: Using capsule networks to detect forged images and videos. 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2307-2311.

[10] Li, Y., Chang, M.C., & Lyu, S. (2018). In Ictu Oculi: Exposing AI Created Fake Videos by Detecting Eye Blinking. 2018 IEEE International Workshop on Information Forensics and Security (WIFS), 1-7.
[11] R. M. Haralick, K. Shanmugam and I. Dinstein, "Textural Features for Image Classification," in IEEE Transactions on Systems, Man, and Cybernetics, vol. SMC-3, no. 6, pp. 610-621, Nov. 1973, doi: 10.1109/TSMC.1973.4309314.
[12] M. S. Rana, M. N. Nobi, B. Murali and A. H. Sung, "Deepfake Detection: A Systematic Literature Review," in IEEE Access, vol. 10, pp. 25494-25513, 2022, doi: 10.1109/ACCESS.2022.3154404.
[13] Dong, S., Wang, J., Liang, J., Fan, H., Ji, R. (2022). Explaining Deepfake Detection by Analysing Image Matching. In: Avidan, S., Brostow, G., Cissé, M., Farinella, G.M., Hassner, T. (eds) Computer Vision – ECCV 2022. ECCV 2022. Lecture Notes in Computer Science, vol 13674. Springer, Cham. https://doi.org/10.1007/978-3-031-19781-9_2.
[14] Luca Guarnera, Oliver Giudice, Sebastiano Battiato; Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, 2020, pp. 666-667
[15] A. Malik, M. Kuribayashi, S. M. Abdullahi and A. N. Khan, "DeepFake Detection for Human Face Images and Videos: A Survey," in IEEE Access, vol. 10, pp. 18757-18775, 2022, doi: 10.1109/ACCESS.2022.3151186.