Deepfake Image Detection & Classification using Conv2D Neural Networks

Debasish Samal[1,†], Prateek Agrawal[1,2,*,†] and Vishu Madaan[1,†]

[1]*Lovely Professional University, Punjab, INDIA*
[2]*Shree Guru Gobind Singh Tricentenary University, Gurugram, Haryana, INDIA*

Abstract

From past few years, rapid advancement of generative AI and fake image creation has evolved, using deep learning. These AI generated fake images are still incredibly challenging to detect. A generative adversarial network (GAN) can create realistic looking fake multimedia, such as images, audio, and videos. So, the spreading of fake media creates panic in social communities and can damage the reputation of a person or community by manipulating public sentiments and opinions towards a person or community. Current studies have suggested using the convolution neural network (CNN) as an effective tool to fight against deepfakes. This paper presents an improved CNN architecture, the Conv2D Model which is trained on 1,40,000 images containing 70,000 real images and 70,000 deepfake images while most of the approaches are using image datasets containing small number of images and pre-trained models to show the fake detection accuracy. A sparse-categorical cross entropy and adam optimizer are applied to enhance the CNN model's learning rate. The proposed model produces an accuracy of 94.54% in OpenForensics: Large-Scale Challenging Dataset For Multi-Face Forgery Detection and Segmentation in-the-wild[1].

Keywords

Deepfake, CNN, Conv2D, Convolutional Neural Network, real image, fake image, GAN

1. Introduction

Artificial intelligence (AI) has made significant developments in various fields, such as computer vision, speech analysis and generation in industries. In an equivalent way, deep learning generative techniques have brought about a revolutionary change in audiovisual processing. Recently, a relatively new phenomenon called deepfakes (DF) has appeared, enabling the generation of artificial (fake) content based on digitally captured images & videos of individuals. Deepfake involves capturing a person's facial expressions, lip movements, and eye movements, and overlaying them onto a different background to create a lifelike simulation of that person in a fabricated scenario. As the global population becomes increasingly interconnected & reliant on social media platforms, Deepfakes are being used more often to generate synthetic data of

ACI'23: Workshop on Advances in Computational Intelligence at ICAIDS 2023, December 29-30, 2023, Hyderabad, India
*Corresponding author.
†These authors contributed equally.
debasishsamal01@gmail.com (D. Samal); dr.agrawal.prateek@gmail.com (P. Agrawal); dr.vishumadaan@gmail.com (V. Madaan)
https://www.linkedin.com/in/debasish-samal-50b906299/ (D. Samal); https://www.drprateekagrawal.com/ (P. Agrawal)
0000-0002-0217-5221 (D. Samal); 0000-0001-6861-0698 (P. Agrawal); 0000-0002-9127-4490 (V. Madaan)

politicians, communities, actors, and media. This, in turn, contributes to the proliferation and dissemination of fake news on social media.[2].

Figure 1: Deepfake image example 1, (left) Real image and (right) Fake image of Indian actress Rashmika Mandanna, Source: https://indianexpress.com/article/

The deepfakes are so popular in use that every day these deepfake contents are releasing and making headlines, interfering the privacy and destroying the reputation of the person. As shown in Figure 1, the news went viral when indian actress Rashmika Mandanna caught in a deepfake video showing exact facial expression and appearance which is hard to recognize if at all known that it was a deepfake.

Figure 2: Deepfake image example 2, (left) Real image and (right) Fake image of Hollywood actor Tom Cruise, Source: https://www.vice.com/en/

With the widespread use of platforms like Telegram, Instagram, Reddit, WhatsApp, and Wikipedia for sharing images, it has become increasingly difficult to distinguish between authentic photos and those that have been manipulated. The use of diverse photo-editing

software complicates the process of verifying an image's authenticity. Picture forgeries are commonly created through splicing and copy-movement techniques. In photo montages' copy-move forgery, a section of an image is illegally replaced with another section to obscure significant details. By cutting and pasting a section from one image onto another, image splicing forms a novel digital image. The main objective of forgery detection is to distinguish similar regions in copy-move forgery and distinct areas in spliced images.

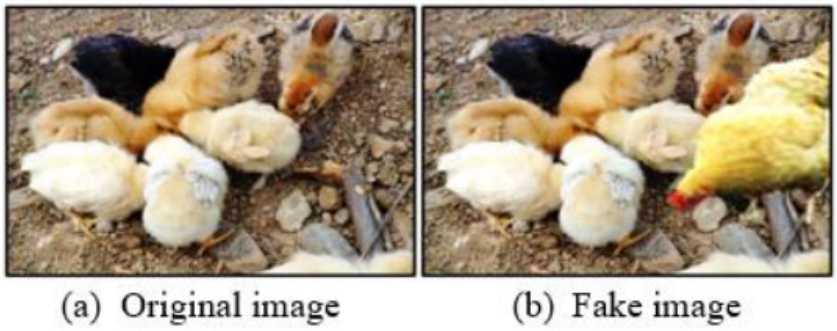

Figure 3: Samples of CASIA_V2 [3], (a) Real image and (b) Fake image

An effective deepfake detection system can accurately identify manipulated and synthetic content that differs from authentic content. Current research publications emphasize the development of a resilient deepfake detection scheme where, many existing approaches in the literature exhibit weaknesses in terms of resilience, effectiveness in formulating the deepfake detection model, & the incorporation of generalizability and legibility within the model.

The ability of a deepfake detection system to accurately find manipulation in both high-quality and poor-quality image or video contents is crucial for its robustness. It is important that the system's effectiveness is not compromised by the resolution of the content being analyzed. Typically, deepfake detection systems tend to perform less effectively when analyzing low-quality content. Generalizability is achieved when each deepfake generation tool employs unique methods to detect the deepfake contents. Interpretability is a critical aspect within the domain of deepfake detection, where a model must have the capability to find the authentic and manipulated regions within an image (such as a person's face) and assign fake probability labels to the corresponding face regions. This feature is essential as it empowers a system to comprehend the complexities of artificially synthesized content and provides a clear rationale for identifying differences in the images. Consequently, there exists a pressing need for robust deepfake detection models that can strike a balance between the aforementioned criteria. Several notable examples of Deepfake tools include Faceswap, DeepFaceLab, Faceswap-GAN, DFaker, StyleGAN, StarGAN, and Face Swapping GAN (FSGAN), among several others[4].

1.1. Main Contribution

Below are the main contributions of the paper:

- We delve into different strategies for detecting deepfake images through the utilization of the improved frameworks, emphasizing both their strengths and drawbacks.

- We give a fundamental study on the deepfake images, their creation and advancements following detection works.
- We provide a convolutional neural network model (Conv2D) architecture to classify deepfake images. The proposed model is trained over 1,40,002 training images and 39,428 testing images with 10,905 validation images used for image classification using the whole Open Forensics dataset [1].

The following sections of the document are organized accordingly: In Section 2, we provide a literature review of related works, featuring various deep learning models and discuss existing deepfake detection approaches. In Section 3, we present current state of the art benchmark data sets used widely for better accuracy of deepfake detection. In section 4, we tested our proposed CNN Model and discussed the results in Section 5. Lastly, we summarize our findings in Section 6 and propose potential avenues for future studies to conclude the paper.

2. Literature Review

2.1. Deepfake

Deepfakes have gained widespread recognition primarily because of the convenience and accessibility of various mobile applications and algorithms. These applications heavily rely on deep learning methodologies, although there are also alternative approaches used. The implementation of deep learning for data representation is a prevalent and extensively employed method in modern times.

Deepfakes can jeopardize individuals' and governments' privacy and societal security. Moreover, they are a grave threat to national security, with democracies increasingly at risk. Various methods and strategies have been developed to deal with the impact of deepfakes, enabling the detection of such content and the implementation of necessary measures[5]. In contrast to the identification of video deepfakes, which comprise a series of images, the primary objective of deepfake image detection is to distinguish any image as fake or real. Recent research[6],[7],[3] examines various biological indicators to identify deepfake images, specifically focusing on eye and gaze properties that distinguish them. Additionally, the scientists integrated these attributes to create unique signatures, enabling a comparison between genuine and manipulated images. This analysis encompassed geometric, visual, metric, temporal, and spectral variances.

2.2. Deep learning models

2.2.1. Autoencoder

The Autoencoder was the first technology employed in the generation of deepfakes[8]. The purpose of the model is to reproduce images it has been taught. The output is generated through three successive stages: encoding, latent space, and decoding. The encoder compresses the input pixels, encoding specific attributes like skin texture, color, facial expressions, open/closed eyes, head pose and fine details, resulting in a smaller compressed image. The latent space processes the compressed image, revealing patterns and structural similarities among the data points. The decoder reconstructs an output by decomposing and interpreting the information

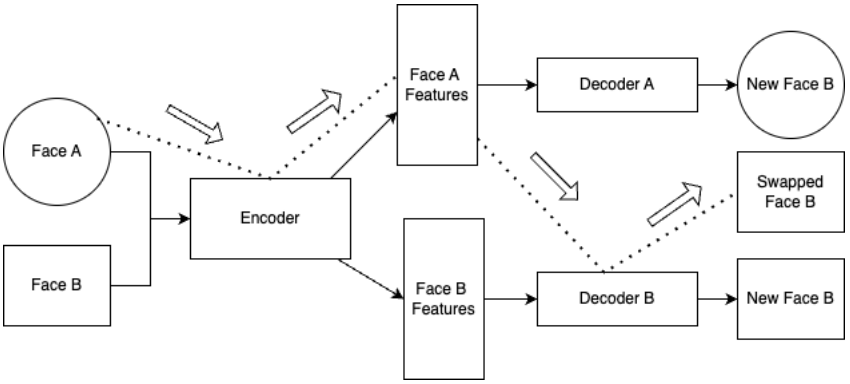

Figure 4: Autoencoder Architecture

from the latent space. The decoder aims to reproduce an image as similar as possible to the original.

An autoencoder can be utilized to exchange two faces as shown in Figure 4. Face B is reconstructed to look like Face A by tracing the route indicated by the red arrows. Both faces were encoded identically. Encoding common features enables similar positioning of faces in the latent space for the encoder. Autoencoders can swap faces in the same image. To accurately reconstruct Face B as similar to Face A, the decoder uses Face A's latent space as reference. This technique is used in DeepFaceLab, DFaker, TensorFlow-based deepfakes, and other deepfake technologies[7].

2.2.2. CNN

The convolutional neural network (CNN) is a specific type of neural network that is designed to learn feature engineering by optimizing filters. This regularization technique allows CNN to automatically capture relevant features from input data without the need for manual feature engineering. As mentioned in Fig 5, CNNs consist of convolution layers, pooling layers, and output layers. CNNs are commonly used in tasks such as fake photo detection and object recognition because they excel in extracting features using principles of linear algebra, particularly matrix multiplication, to identify patterns in images.

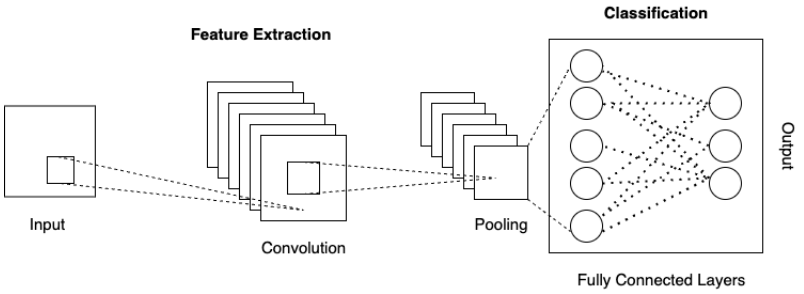

Figure 5: General Architecture of CNN

Studies of [2] show an improved dense CNN model which focuses on high generalizability and detection accuracy over GAN generated image datasets. Similarly, Zhu et al.[9] proposed a deep learning model for detecting deepfake images using CNNs to extract frame-level features and detect forgeries. The method was evaluated using a large dataset of forged images from various sources, and it yielded favorable results for the project. Earlier research by Wang et al.[10] present an approach that reveals images containing synthetic faces generated by deep neural network models. By analyzing the entire image, the convolution network initially extracts several low-level features through multiple layers, which subsequently combine to form more intricate features via a succession of convolution layers. CNNs can capture more comprehensive information from images due to the composition of their high-level features from multiple low-level features.

2.2.3. GAN

GAN is one of the best techniques for artificial image detection in computer vision. Their core principle is based on game theory[11], In a generator-versus-discriminator competition, the generator generates the samples. The discriminator's task is to differentiate between real and generated samples.

In GANs, both the generator and discriminator learn concurrently: the generator generates artificial images following the dataset distribution, while the discriminator distinguishes between real and fake images. After numerous training iterations, the generator network produces images that closely resemble real images, while the discriminator network learns to distinguish between these produced images and real ones.

The discriminator model within Generative Adversarial Networks (GANs) is responsible for classifying an input example from the problem domain, whether it is a real instance or one that has been generated. Its main task is to predict a binary label, distinguishing between real and fake. Generative Adversarial Network (GAN) architecture is the progression of arranging and planning the structure of GANs to enhance their performance.

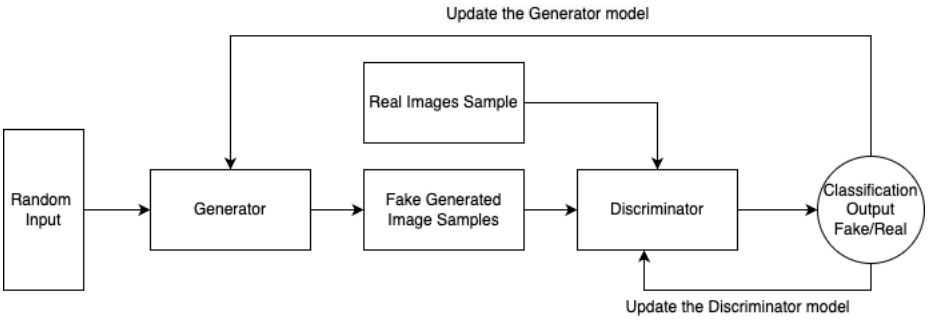

Figure 6: Basic Architecture of Generative Adversarial Network

This technology, called GANs, was first invented in 2014 by [12], see Fig 8. In the past few years, people have made big advancements in generating and detecting gan produced fake pictures.In the work by [13], the authors present an approach for detecting GAN-generated images

through the generalization of an unsupervised domain adaptation model. The results shows significant generalization accuracy improvement over StyleGan[14], StarGan[15], StyleGAN2[16] and PGGAN[17].Zhang et al.[18] developed AutoGAN, a system capable of replicating the synthetic imperfections found in GAN-generated images. This model incorporates upsampling techniques. Also In 2023, Monkam et al.[19] introduced the G-JOB GAN model which achieved 95.7% accuracy using a 4096-image dataset from the CelebA dataset.

Table 1
A qualitative analysis of existing deepfake image detection approaches

Sr. No	Author	Year	Journal	Dataset	Methodology	Accuracy in %	Remarks
1	Patel et al.[2]	2023	IEEE Access	StyleGAN	Deep-CNN	95.33	Proposed model detects specific GAN model datasets only
2	Zhu et al.[9]	2023	IEEE Transactions on Information Forensics and Security	CELEB A, LFW	IAP	94	Limitedly tested on two datasets majorly and focuses more on face protection
3	Ju et al.[20]	2023	IEEE Transactions on Multimedia	DF3 CUSTOM DATASET, LSUN	GAN-DCT	90.6	It combines multi scale global features with informative local features of images
4	Wang et al[21]	2023	Mathematics MDPI	FF++ CELEB-DF UADFV	CNN, Frequency Domain Analysis	72.27	It only uses on low quality compressed fake images and Generalized AUC score is just 72 percent.
5	Khalil et al.[22]	2023	IEEE Access	AttGAN GDWCT StyleGAN StyleGAN2	Deep-CNN	94.67	only GAN generated image datatsets were taken and focused on high generalizability
6	Panigrah et al.[3]	2023	Revue d'Intelligence Artificielle	CASIA v2	Ensemble approach of Pre-trained CNN models	90.09	Methothology is optimal for Copy-move and splicing type images
7	Raza et al.[23]	2022	Applied Sciences MDPI	AKAGGLE DATASET by The Dept. of Computer Science, Yonsei University	Deepfake predictor DFP based on a hybrid of VGG16 and CNN	95	The suggested approach outdid transfer learning based techniques.
8	Guarnera et al.[24]	2023	Journal of Imaging	Combined Images from CelebA FFHQ GAN	EfficientNet DCT	89.30	The dataset consumed is very complex for various attacks on images like scaling,JPEG compression, rotation etc.
9	Tang et al.[25]	2021	Security and Communication Networks	CELEB A	Fake Image Discriminator FID DWT StyleGAN2	91.22	The study mainly examines GAN synthesized images where as earlier studies worked on small sized datasets.
10	Shad et al.[4]	2021	Computational Intelligence and Neuroscience	Flickr StyleGAN	CNN models DenseNet VGGNet ResNet VGGFace	90	The study helps to detect deepfake images from a large dataset of over one lakh forty thousand images with implimentation of eight CNN architectures.

3. State-of-the-Art Datasets for Deepfake Image Detection

A variety of Deepfake visuals have been created over last few years utilizing different frameworks including AttGAN, StarGAN, GDWCT, StyleGAN, and StyleGAN2. The image datasets listed below are primarily utilized for deepfake image detection purposes:

1. *CelebA* 202,599 high-quality images of celebrities are present in the dataset, accompanied by detailed annotations.[26]
2. *The FF++* The dataset comprises 1000 authentic video sequences that were modified using four automated face manipulation techniques: FaceSwap, Face2Face, Deepfakes, and NeuralTextures.[27]
3. *LSUN* The dataset contains around one million face images that have been labeled.[28]
4. *The CELEB-DF*, 590 YouTube videos representing a mix of ages, ethnicities, and genders make up the dataset. Furthermore, it contains 5639 DeepFake videos that replicate the original content.[29]
5. *The HFF* The dataset contains a significant amount of synthetic facial images, consisting of more than 155,000 face photos.[30]
6. *The DigiFace-1M* The dataset consists of an extensive collection of over one million artificial facial images, covering a broad spectrum of diversity.[31]
7. *AttGAN* is a well-built dataset of over 30,000 images.[32]
8. *StyleGAN* image dataset contains over 7000 synthesised images[33]
9. *StyleGAN2* dataset covers 100,000 fake face images[34]
10. *OpenForensics: Multi-Face Forgery Detection And Segmentation In-The Wild* dataset [1] consists on over 1,90,000 real and fake images.

4. Proposed Conv2D Model

This section describes the CNN architecture proposed herein Fig 7, which combines convolutional and pooling layers. Convolutional layers extract image features, while pooling layers reduce the dimensionality of the feature maps. After being processed by the convolutional layers, the feature maps are flattened and combined into a one-dimensional array for input to the fully connected layer. The output layer determines the subsequent class after the fully connected layer processes the input image. In a similar way, the proposed Conv2D model presented in this study is designed for binary image classification tasks.

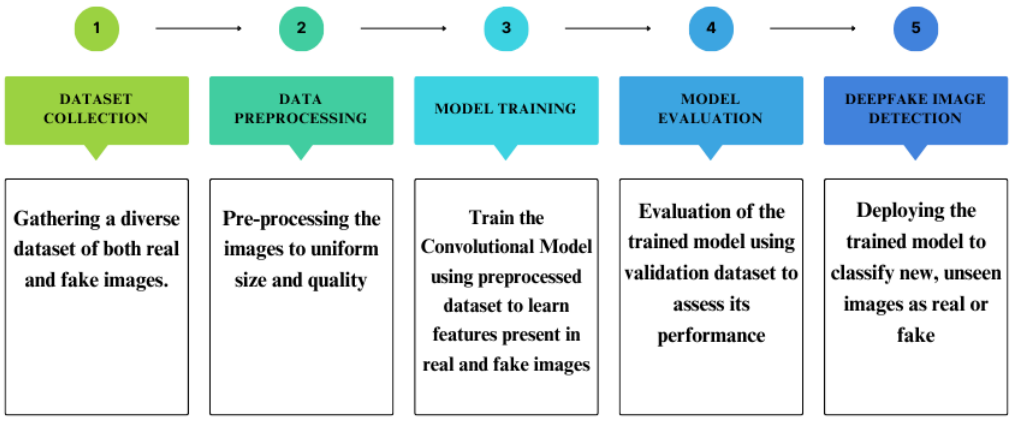

Figure 7: Flow Diagram of the Conv2D model

The Conv2D model carries out the deepfake image detection divided into five phases.

1. *Dataset Collection* involves the collection of authentic data as the initial task of any deep learning model.
2. *Data Preprocessing* begins with resizing the images and augmentation to increase the diversity of the dataset.
3. *Model Training* starts by using the preprocessed images to train the deep learning model, which is the Conv2D model here. So that the model should learn to distinguish between the features present in real and fake images.
4. *Model Evaluation* is the fourth phase, where we evaluate the trained model using a separate validation dataset to recognize its performance in distinguishing between real and fake images. Metrics such as accuracy can be used for evaluation.
5. *Deepfake Image Detection* happens once the model demonstrates satisfactory performance, then we deploy it to classify new and unseen images as real or fake.

5. Results & Discussions

In this section, we discuss the effectiveness of the suggested design and the outcomes obtained.

Each layer of the model helps in efficient training in the following ways:

Table 2
Conv2D model output dimensions and no. of parameters for each layer

Layer	Type of layer	Output Dimension	No. of Parameters
1	Conv2D	(None, 148, 148, 32)	896
2	Max Pooling2D	(None, 74, 74, 32)	0
3	Conv2D	(None, 72, 72, 64)	19496
4	Max Pooling2D	(None, 36, 36, 64)	0
5	Conv2D	(None, 34, 34, 128)	73856
6	Max Pooling2D	(None, 17, 17, 128)	0
7	Flatten	(None, 36992)	0
8	Dense-1	(None, 1064)	39360552
9	Dense-2	(None, 2)	2130

Total params	39,455,930
Trainable params	39,455,930
Non-trainable params	0

- A 3x3 convolutional layer containing 32 filters initiates the model. The activation function for this layer is ReLU (Rectified Linear Unit). This layer receives a 150x150 RGB image as input.
- The second layer is a max pooling operation with a pool size of 2x2. This layer compresses the input's spatial dimensions by selecting the maximum value within a given window determined by pool size.
- A 3×3 convolutional layer with 64 filters and ReLU activation is implemented as the third layer.
- The fourth layer is another max pooling layer with a pool size of 2×2.
- The fifth layer comprises a convolutional structure with 128 filters of size 3×3, all applying the ReLU activation function.
- The sixth layer is a max pooling layer with a 2x2 pool size.
- Flatten Layer is the seventh layer that converts the 2D matrix into a 1D vector.
- The eighth layer consists of 1064 neurons, fully-connected to the previous layers, and employs the ReLU activation function.
- The final dense layer comprises 2 neurons representing the two classes, 'Fake' and 'Real', and is activated by the softmax function to deliver probabilities.

This design takes advantage of CNNs to extract hierarchical features from images, which are then utilized for the binary classification task. By incorporating multiple convolutional and pooling layers, the model is able to grasp complex patterns in the data. The dense layers at the end of the model carry out the final classification by leveraging these learned features.

5.1. Simulation Setup

The model's training, testing, and implementation was done using the TensorFlow and Keras libraries in Python, which is carried out on an Intel Core i7-11th generation CPU. We conducted the trials using a graphics processing unit (GPU) from NVIDIA GEFORCE RTX 3060 equipped with 16 gb of random-access memory (RAM).

5.2. Dataset Description

We have chosen to utilize the OpenForensics dataset [1], which has data on over 1,90,000 real and fake images.OpenForensics is the initial extensive dataset that poses a significant challenge. This dataset is designed with face-specific rich annotations explicitly for face forgery detection and segmentation. The OpenForensics dataset has great value for research in both deepfake elimination and general artificial face detection because of its rich annotations. It is a balanced dataset of resolution 256×256 pixels. The training, testing, and validation images are divided into two classes namely real and fake containing a total of 140002, 39428 & 10905 images respectively.

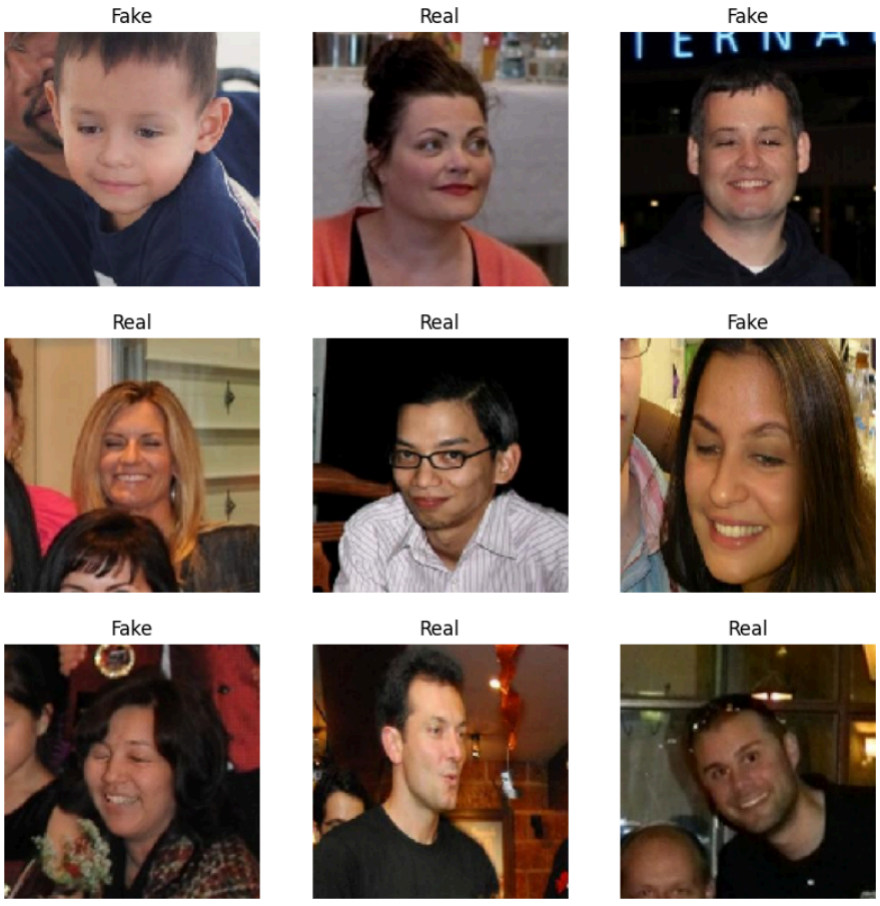

Figure 8: Example real and fake Images from Dataset[1]

5.3. Evaluation metrics and Discussions

In our experimentation, we use Accuracy scores to measure the model's performance.Accuracy is one of the most used evaluation metric in machine learning, especially for classification problems like image classification using Convolutional Neural Networks (CNNs).

The proportion of correct predictions to total number of predictions is the measure of accuracy. Mathematically, it can be expressed in equation 1:

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}}.$$

In a binary categorization problem, this can also be written as:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{1}$$

where:

- TP represents correctly identified positive instances.
- The count of correctly identified negative instances is referred to as TN.
- False positives, FP, represent instances classified as positive when they should have been negative.
- False negatives, FN, represent incorrectly identified negative instances.

Accuracy is a straightforward metric that provides a general measure of how well a model is performing across all classes. In the context of image classification with CNNs, accuracy can give us a quick understanding of how well our model is able to correctly classify images. For our Conv2D model, we have used a sparse categorical cross entropy and the adam optimizer to increase the model's learning rate. Spanned over 10 epochs and validation batch size of 50 we were able to achieve 99.36% training accuracy and 94.54% validation accuracy as shown in Figure 9 . The model loss over time is shown in Figure 10.

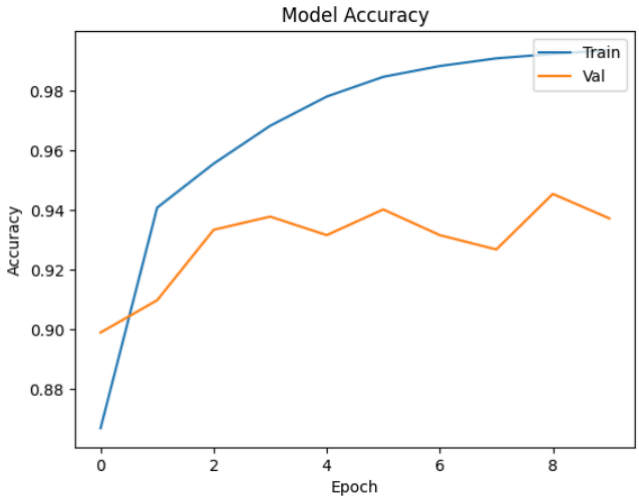

Figure 9: Conv2D model Accuracy

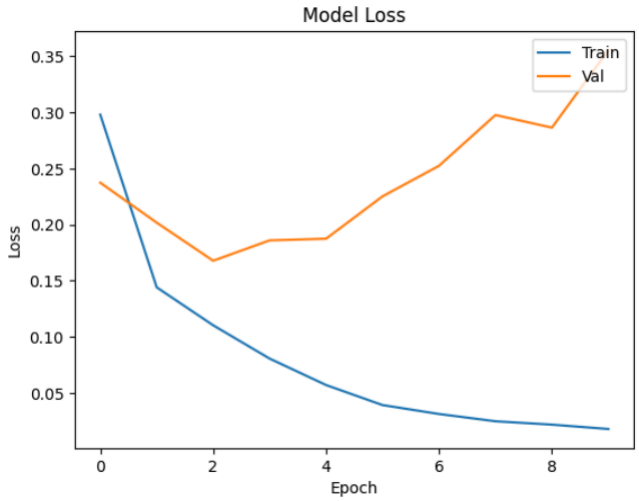

Figure 10: Conv2D model Loss

Different DeepFake detection models [3, 24, 4] are trained on various datasets containing noticeable artifact characteristics like low resolution, color discrepancies, and visible boundaries. These learned features might not be effective when applied to the high-quality DeepFake dataset like OpenForensics[1], leading to a decrease in performance. In addition, from the experimental results it can be observed our model's accuracy maintains over an average of 90% which can be considered reasonable with respect to the latest deepfake image detection models. Our model is able to achieve such reasonable accuracy over such a large scale dataset.

6. Conclusion and Future Work

Detecting deepfake content has always been a challenging task due to its unique level of abstraction. Traditionally, the problem is categorized as a binary classification issue, distinguishing between prestine and deepfake labels. To address this issue, a CNN-based Conv2D architecture has been proposed in our research to effectively identify deepfake images. The architecture has demonstrated an impressive accuracy of 94.54% when trained on the extensive OpenForecsics dataset[1], which consists both class of real and fake images. Despite observing an increase in model loss over time, the accuracy of the model remains excellent over validation data. Furthermore, this work can be expanded to classify open image datasets and video deepfake content. For video deepfake detection, the model can process each frame by extracting the face, cropping it, and then applying the model to detect deepfake falsifications. A pipeline can be created to implement this process for handling video data. The proposed CNN-based model, which utilizes diverse data augmentation methods, demonstrates strong performance and equilibrium across the dataset.

References

[1] T.-N. Le, H. H. Nguyen, J. Yamagishi, I. Echizen, Openforensics: Large-scale challenging dataset for multi-face forgery detection and segmentation in-the-wild, in: International Conference on Computer Vision, 2021.

[2] Y. Patel, S. Tanwar, P. Bhattacharya, R. Gupta, T. Alsuwian, I. E. Davidson, T. F. Mazibuko, An Improved Dense CNN Architecture for Deepfake Image Detection, IEEE Access 11 (2023) 22081–22095. URL: https://ieeexplore.ieee.org/document/10057390/. doi:10.1109/ACCESS.2023.3251417.

[3] G. R. Panigrah, P. K. Sethy, S. P. R. Borra, N. K. Barpanda, S. K. Behera, Deep Ensemble Learning for Fake Digital Image Detection: A Convolutional Neural Network-Based Approach, Revue d'Intelligence Artificielle 37 (2023) 703–708. URL: https://iieta.org/journals/ria/paper/10.18280/ria.370318. doi:10.18280/ria.370318.

[4] H. S. Shad, M. M. Rizvee, N. T. Roza, S. M. A. Hoq, M. Monirujjaman Khan, A. Singh, A. Zaguia, S. Bourouis, Comparative Analysis of Deepfake Image Detection Method Using Convolutional Neural Network, Computational Intelligence and Neuroscience 2021 (2021) e3111676. URL: https://www.hindawi.com/journals/cin/2021/3111676/. doi:10.1155/2021/3111676.

[5] H. F. Shahzad, F. Rustam, E. S. Flores, J. Luís Vidal Mazón, I. De La Torre Diez, I. Ashraf, A Review of Image Processing Techniques for Deepfakes, Sensors 22 (2022) 4556. URL: https://www.mdpi.com/1424-8220/22/12/4556. doi:10.3390/s22124556.

[6] D. Wan, M. Cai, S. Peng, W. Qin, L. Li, Deepfake Detection Algorithm Based on Dual-Branch Data Augmentation and Modified Attention Mechanism, Applied Sciences 13 (2023) 8313. URL: https://www.mdpi.com/2076-3417/13/14/8313. doi:10.3390/app13148313.

[7] T. T. Nguyen, Q. V. H. Nguyen, D. T. Nguyen, D. T. Nguyen, T. Huynh-The, S. Nahavandi, T. T. Nguyen, Q.-V. Pham, C. M. Nguyen, Deep Learning for Deepfakes Creation and Detection: A Survey, Computer Vision and Image Understanding 223 (2022) 103525. URL:

http://arxiv.org/abs/1909.11573. doi:10.1016/j.cviu.2022.103525, arXiv:1909.11573 [cs, eess].

[8] R. Katarya, A. Lal, A Study on Combating Emerging Threat of Deepfake Weaponization, in: 2020 Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), IEEE, Palladam, India, 2020, pp. 485–490. URL: https://ieeexplore.ieee.org/document/9243588/. doi:10.1109/I-SMAC49090.2020.9243588.

[9] Y. Zhu, Y. Chen, X. Li, R. Zhang, X. Tian, B. Zheng, Y. Chen, Information-Containing Adversarial Perturbation for Combating Facial Manipulation Systems, IEEE Transactions on Information Forensics and Security 18 (2023) 2046–2059. URL: https://ieeexplore.ieee.org/document/10086559/. doi:10.1109/TIFS.2023.3262156.

[10] S.-Y. Wang, O. Wang, R. Zhang, A. Owens, A. A. Efros, CNN-generated images are surprisingly easy to spot... for now, 2020. URL: http://arxiv.org/abs/1912.11035, arXiv:1912.11035 [cs].

[11] M. Mohebbi Moghaddam, B. Boroomand, M. Jalali, A. Zareian, A. Daeijavad, M. H. Manshaei, M. Krunz, Games of GANs: game-theoretical models for generative adversarial networks, Artificial Intelligence Review 56 (2023) 9771–9807. URL: https://doi.org/10.1007/s10462-023-10395-6. doi:10.1007/s10462-023-10395-6.

[12] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative Adversarial Nets (2014). URL: https://dl.acm.org/doi/10.5555/2969033.2969125.

[13] M. Zhang, H. Wang, P. He, A. Malik, H. Liu, Improving GAN-Generated Image Detection Generalization Using Unsupervised Domain Adaptation, in: 2022 IEEE International Conference on Multimedia and Expo (ICME), IEEE, Taipei, Taiwan, 2022, pp. 1–6. URL: https://ieeexplore.ieee.org/document/9859763/. doi:10.1109/ICME52920.2022.9859763.

[14] T. Karras, S. Laine, T. Aila, A Style-Based Generator Architecture for Generative Adversarial Networks, in: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Long Beach, CA, USA, 2019, pp. 4396–4405. URL: https://ieeexplore.ieee.org/document/8953766/. doi:10.1109/CVPR.2019.00453.

[15] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, J. Choo, StarGAN: Unified Generative Adversarial Networks for Multi-domain Image-to-Image Translation, in: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, IEEE, Salt Lake City, UT, 2018, pp. 8789–8797. URL: https://ieeexplore.ieee.org/document/8579014/. doi:10.1109/CVPR.2018.00916.

[16] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, T. Aila, Analyzing and Improving the Image Quality of StyleGAN, 2020, pp. 8110–8119.

[17] T. Karras, T. Aila, S. Laine, J. Lehtinen, Progressive Growing of GANs for Improved Quality, Stability, and Variation, 2018. URL: http://arxiv.org/abs/1710.10196. doi:10.48550/arXiv.1710.10196, arXiv:1710.10196 [cs, stat].

[18] X. Zhang, S. Karaman, S.-F. Chang, Detecting and Simulating Artifacts in GAN Fake Images, in: 2019 IEEE International Workshop on Information Forensics and Security (WIFS), IEEE, Delft, Netherlands, 2019, pp. 1–6. URL: https://ieeexplore.ieee.org/document/9035107/. doi:10.1109/WIFS47025.2019.9035107.

[19] G. Monkam, W. Xu, J. Yan, A GAN-based Approach to Detect AI-Generated Images, in: 2023 26th ACIS International Winter Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD-Winter), IEEE, Taiyuan, Taiwan,

2023, pp. 229–232. URL: https://ieeexplore.ieee.org/document/10223798/. doi:10.1109/SNPD-Winter57765.2023.10223798.

[20] Y. Ju, S. Jia, J. Cai, H. Guan, S. Lyu, GLFF: Global and Local Feature Fusion for AI-synthesized Image Detection, IEEE Transactions on Multimedia (2023). URL: https://ieeexplore.ieee.org/abstract/document/10246417, https://github.com/littlejuyan/GLFF.

[21] B. Wang, X. Wu, Y. Tang, Y. Ma, Z. Shan, F. Wei, Frequency Domain Filtered Residual Network for Deepfake Detection, Mathematics 11 (2023) 816. URL: https://www.mdpi.com/2227-7390/11/4/816. doi:10.3390/math11040816, number: 4 Publisher: Multidisciplinary Digital Publishing Institute.

[22] A. H. Khalil, A. Z. Ghalwash, H. A.-G. Elsayed, G. I. Salama, H. A. Ghalwash, Enhancing Digital Image Forgery Detection Using Transfer Learning, IEEE Access 11 (2023) 91583–91594. URL: https://ieeexplore.ieee.org/document/10226188/. doi:10.1109/ACCESS.2023.3307357.

[23] A. Raza, K. Munir, M. Almutairi, A Novel Deep Learning Approach for Deepfake Image Detection, Applied Sciences 12 (2022) 9820. URL: https://www.mdpi.com/2076-3417/12/19/9820. doi:10.3390/app12199820.

[24] L. Guarnera, O. Giudice, F. Guarnera, A. Ortis, G. Puglisi, A. Paratore, L. M. Q. Bui, M. Fontani, D. A. Coccomini, R. Caldelli, F. Falchi, C. Gennaro, N. Messina, G. Amato, G. Perelli, S. Concas, C. Cuccu, G. Orrù, G. L. Marcialis, S. Battiato, The Face Deepfake Detection Challenge, Journal of Imaging 8 (2022) 263. URL: https://www.mdpi.com/2313-433X/8/10/263. doi:10.3390/jimaging8100263, https://iplab.dmi.unict.it/deepfakechallenge/.

[25] G. Tang, L. Sun, X. Mao, S. Guo, H. Zhang, X. Wang, Detection of GAN-Synthesized Image Based on Discrete Wavelet Transform, Security and Communication Networks 2021 (2021) 1–10. URL: https://www.hindawi.com/journals/scn/2021/5511435/. doi:10.1155/2021/5511435, https://github.com/peterwang512/CNNDetection.

[26] CelebFaces Attributes (CelebA) Dataset, 2015. URL: https://www.kaggle.com/datasets/jessicali9530/celeba-dataset.

[27] ondyari, FaceForensics++: Learning to Detect Manipulated Facial Images, 2023. URL: https://github.com/ondyari/FaceForensics, original-date: 2018-04-13T12:47:46Z.

[28] F. Yu, LSUN, 2023. URL: https://github.com/fyu/lsun, original-date: 2015-04-02T01:06:34Z.

[29] Y. Li, Celeb-DF: A Large-scale Challenging Dataset for DeepFake Forensics, 2023. URL: https://github.com/yuezunli/celeb-deepfakeforensics, original-date: 2019-10-02T00:31:06Z.

[30] Z. Guo, G. Yang, J. Chen, X. Sun, Fake face detection via adaptive manipulation traces extraction network, Computer Vision and Image Understanding 204 (2021) 103170. URL: https://linkinghub.elsevier.com/retrieve/pii/S107731422100014X. doi:10.1016/j.cviu.2021.103170, https://github.com/EricGzq/AMTENnet.

[31] DigiFace-1M Dataset, 2022. URL: https://github.com/microsoft/DigiFace1M, original-date: 2022-09-15T09:35:25Z.

[32] E. Y.-J. Lin, AttGAN-PyTorch, 2023. URL: https://github.com/elvisyjlin/AttGAN-PyTorch, original-date: 2018-11-28T08:56:52Z.

[33] NVlabs/stylegan, 2023. URL: https://github.com/NVlabs/stylegan, original-date: 2019-02-04T15:33:58Z.

[34] NVlabs/stylegan2, 2023. URL: https://github.com/NVlabs/stylegan2, original-date: 2019-11-26T20:52:23Z.