Comparative Analysis of Deepfake Detection Models on Diverse GAN-Generated Images

Original Scientific Paper

Medha Wyawahare*
Vishwakarma Institute of Technology,
Department of Electronics and Telecommunication
Pune, Maharashtra, India
medha.wyawahare@vit.edu

Siddharth Bhorge
Vishwakarma Institute of Technology,
Department of Electronics and Telecommunication
Pune, Maharashtra, India
siddharth.bhorge@vit.edu

Milind Rane
Vishwakarma Institute of Technology,
Department of Electronics and Telecommunication
Pune, Maharashtra, India
milind.rane@vit.edu

*Corresponding author

Vrinda Parkhi
Vishwakarma Institute of Technology,
Department of Electronics and Telecommunication
Pune, Maharashtra, India
vrinda.parkhi@vit.edu

Mayank Jha
Vishwakarma Institute of Technology,
Department of Electronics and Telecommunication
Pune, Maharashtra, India
mayank.jha20@vit.edu

Narendra Muhal
Vishwakarma Institute of Technology,
Department of Electronics and Telecommunication
Pune, Maharashtra, India
narendra.muhal20@vit.edu

Abstract *– Advancement in Artificial intelligence has resulted in evolvement of various Deepfake generation methods. This subsequently leads to spread of fake information which needs to be restricted. Deepfake detection methods offer solution to this problem. However, a particular Deepfake detection method which gives best results for a set of Deepfake images (generated by a particular generation method) fails to detect another set of Deepfake images (generated by another method). In this work various Deepfake detection methods were tested for their suitability to decipher Deepfake images generated by various generation methods.*

We have used VGG16, ResNet50, VGG19, and MobileNetV2 for deepfake detection and pre-trained models of StyleGAN2, StyleGAN3, and ProGAN for fake generation. The training dataset comprised of 200000 images, 50 % of which were real and 50% were fake. The best performing Deepfake detection model was VGG19 with more than 96 percent accuracy for StyleGAN2, StyleGAN3, and ProGAN-generated fakes.

Keywords*:* *CNN, GAN, VGG19, StyleGAN3, Deepfake*

Received: March 22, 2024; Received in revised form: August 20, 2024; Accepted: August 21, 2024

1. INTRODUCTION

The deepfake image synthesis and detection field has attracted significant research due to the convergence of computer vision and artificial intelligence. This multidisciplinary area, which attracts academicians, researchers and business experts, focuses on the automated creation and recognition of modified visual content. Deepfake image generation and detection have practical repercussions in a variety of fields, from digital forensics and content verification to maintaining user confidence in computer-human interactions. Furthermore, it has the power to fundamentally alter how society interacts with visual information. The creation of models that can not only create but also recognize real images from altered ones is at the core of this endeavor. Similar to how image captioning seeks to describe scenes, the main goal in this case is to create material that fools or imitates reality. This technology offers inventive ways to create digital content but also presents difficulties that call for strict safeguards against misuse and false information.

Modern deep learning techniques serve as inspiration for the architecture supporting deepfake image production and detection. Modern methods usually use the encoder-decoder paradigm, which is appropriate for both facets of this topic. To encode the source's unique features, the encoder must convert them into little feature vectors. The decoder then makes use of these vectors to create them or determine their legitimacy. The core of the encoder component is a convolutional neural network (CNN). The use of well-known CNN architectures like VGG, ResNet, and MobileNet, is common in the encoder. The model's capacity to spot subtle patterns in real and altered images is aided by this larger viewpoint.

The use of four specialized detection models demonstrates our commitment to excellence in deepfake image production and detection. Every model has been painstakingly designed to handle particular aspects of deepfake identification, improving the model's overall accuracy and adaptability. We explore the world of Generative Adversarial Networks (GANs), a powerful method for producing deepfake content, as a complement to these detection attempts. We seek to advance the authenticity and realism of the created images by utilizing adversarial training, adding to the continuing arms race between creation and detection.

In this paper, we conduct a comprehensive evaluation of eight different CNN models, such as VGG16, ResNet50, VGG19, and Xception, among others, for deepfake detection. Initially, we train these CNN models on the OpenForensics dataset, a widely used benchmark dataset in the field of deepfake detection. To evaluate these models performance and generalizability, we test them using a recently created dataset that contains a wide variety of deepfakes. Furthermore, to enhance the robustness of our evaluation, we augment the OpenForensics dataset by incorporating our own generated data, thereby expanding the dataset's diversity and realism. Subsequently, we retrain the CNN models on this augmented dataset, leveraging the enriched data to improve the model's performance.

Finally, we rigorously evaluate the trained models by testing them on three distinct sets of GAN-generated data: ProGAN, StyleGAN2, and StyleGAN3 [1]. The newly generated dataset from these GANs is available on Kaggle for public access. We hope to shed light on how well CNN models perform in identifying deepfakes using a variety of datasets and GAN architectures by using this thorough approach.

2. LITERATURE REVIEW

In 2019, Yadav et al. [2] put forth that deepfake images are man-made media, especially edited videos or photos, produced by cutting-edge machine learning algorithms that can accurately replicate real human expressions and activities. They examine many strategies, from conventional GAN-based techniques to more complex variants, like conditional GANs and cycle-consistent GANs. To create extremely realistic facial forgeries, the proposed deepfake generation model uses a conditional GAN architecture, where the generator is conditioned on both the input and the target identity. To remove the potential misuse of deepfake, they added watermarks on the deepfakes. Sanjana et al. [3] gave a thorough analysis of the current deepfake detection methods to stop the spread of false information and protect the integrity of multimedia content. Detection techniques like CNNs and GANs can spot deepfake face swapping, in which one person's face is swapped out for the face of another. To increase the effectiveness of deepfake detection, transfer learning techniques that use pre-trained models for related tasks (e.g., facial recognition) are used.

Malik et al. [4] provided a thorough analysis of the various techniques and procedures employed for deepfake detection. They look at a variety of strategies, computer vision approaches, and deep learning-based solutions in particular. The survey examines the advantages and disadvantages of various detection techniques and covers both text-based and video-based deepfake. Rana et al. [5] and Paul et al. [6] show significant progress in developing robust deepfake detectors capable of fending off ever-more complex manipulation techniques using GANs for adversarial training. The study produces encouraging results in audio-based deepfake detection using recurrent neural networks, concentrating on minute acoustic artifacts created during speech synthesis to distinguish altered audio from real recordings.

Nguyen et al. [7] investigated various deep learning architectures, such as GANs, autoencoders, and others, that are used to produce deepfake content. To maintain visual integrity, autoencoders, a form of unsupervised deep learning model, have been used for deepfake production. By training on small samples of recently emerging deepfake content, one-shot learning algorithms have demonstrated potential for identifying unique deepfake variants. Datasets like Face Forensics++ and the deepfake Detection Dataset (DFDC) have been significant in advancing research and testing performance in the deepfake detection field. Shen et al. [8] have investigated the technical aspects of how GANs are utilized to create deepfakes, including training the networks, selecting suitable datasets, and fine-tuning the models to produce realistic results, which are probably covered fully in the study. To improve convergence and generation quality, the Wasserstein GAN algorithm variation with a deep convolutional architecture is used to train the generator network. They use the Structural Similarity Index (SSIM) metric and Peak Signal-to-Noise Ratio (PSNR) to objectively analyze the similarity between the created content and the ground truth data to assess the performance of our deepfake generation. Giudice et al. [9] outline a technique to spot abnormalities in the Discrete Cosine Transform (DCT) domain of GAN-generated. This work focuses on preventing deep fakes. The DCT is frequently used for image compression, including JPEG encoding, and GANs also employ it for creation. By examining anomalies in the DCT coefficients of images created using GANs, the authors of this research take a fresh approach to deepfake identification. They make a distinction between real content and content that has been altered by using statistical metrics and machine learning classifiers to identify specific DCT artifacts connected to deepfake image.

Shad et al. [10] conducted a comparative analysis of the performance of eight CNN architectures. They have used images from the Flickr dataset for training the models. Fake images for training were generated using StyleGAN. They evaluated these models on five different evaluation metrics, such as accuracy, precision, recall, etc. VGGFace and ResNet50 performed best with an accuracy of 99% and 97%, respectively. Saxena et al. [11] gave a thorough introduction to GANs, noting the difficulties in training and using them, suggesting different ways to solve these problems, and providing suggestions for possible future research paths. The study contributes to a deeper knowledge of GANs and acts as an invaluable resource for scholars and practitioners in the field of artificial intelligence by carefully examining existing research and methodologies.

Ali et al. [12] tested the generalization of the fake face detection methods. Two types of fake face detection methods have been tested in this paper. The first is texture-based Local Binary Patterns (LBP), and the second is using different CNN architectures such as Alexnet, VGG19, ResNet50, etc. These methods are tested on known and unknown data, and the results show that their performance drops for the unknowns. These results indicate the lack of transferability of the learned classifiers to the general face-forgery classification cases. Patashnik et al. [13] proposed StyleCLIP, a powerful framework for manipulating s generated by StyleGAN using natural language descriptions. By aligning the CLIP model's textual embeddings with StyleGAN's latent space, users can apply targeted modifications to generated text simply by providing descriptive text. StyleCLIP allows users to create diverse and specific visual outputs, offering an exciting approach to interactive and intuitive synthesis and manipulation.

Kumar et al. [14] reviewed various techniques for implementing and detecting deepfake images, focusing on Deep Convolution-based GAN models. A comparative analysis of the proposed GAN model with existing models is performed using parameters such as Inception Score (IS) and Fréchet Inception Distance (FID). Deepfake images present a substantial threat to biometric security and facilitate counterfeiting and fraudulent activities.

Tiwari et al. [15] discuss the use of GANs in creating highly realistic deepfakes and their role in both generating and detecting fake content through discriminator networks. CNNs are highlighted for their effectiveness in classification and detecting subtle anomalies in images and videos, making them a primary method for deepfake detection. RNNs and LSTMs are noted for their capability to handle sequential data, making them suitable for analyzing video content and identifying temporal inconsistencies indicative of deepfakes. Recent advancements in attention mechanisms and transformers show promise for improving deepfake detection accuracy through sophisticated feature extraction and analysis. The authors evaluated deepfake detection models using the Inception Score (IS) and Fréchet Inception Distance (FID) to quantify the quality of the generated data and the effectiveness of detection algorithms.

Nowroozi et al. [16] describe the application of GAN-based CNN models for deepfake detection, highlighting their effectiveness in distinguishing real from artificial faces. The effectiveness of the CNN models, which are Cross-Co-Net and Co-Net, was compared to alternative approaches. It showed superior accuracy, which underscores the robustness of combining GAN-generated data with CNN for deepfake.

Sharma et al. [17] presented the effectiveness of GANs for deepfake detection, leveraging a GAN-based CNN model. Using the Indian actor's dataset, demonstrates how GANs may be used to expand training datasets, hence improving the robustness of the model. The suggested approach outperforms existing techniques and demonstrates its potential for useful applications in digital forensics and image recognition. Sergi et al. [18] investigated the human ability to identify deepfakes created using the StyleGAN2 algorithm. Three intervention tactics were tested for their efficacy in detecting deepfakes through an online poll that attracted 280 participants. Following the evaluation of twenty images, the participant's accuracy score ranged from 60% to 64% depending on the situation, indicating that deepfake images produced by StyleGAN2 are difficult for humans to detect. Notably, interventions did not significantly improve detection accuracy. The findings highlight the difficulty in detecting deepfake images and underscore the urgent need for enhanced detection methods and public awareness.

3. RESEARCH GAP

There has been a lot of intensive research and development in this field in the last few years as a result of the rise of Artificial Intelligence (AI) and Deep Learning (DL) technologies. In the literature that we reviewed, there are several gaps in the current state of deepfake detection research. While the majority of research to date has focused on GAN-based methods and specific designs, there are noticeably few comprehensive comparative studies that look at a larger variety of GAN variants. Moreover, reliance on established datasets, such as Face Forensics++ and DFDC, restricts the understanding of model effectiveness across diverse data sources, indicating a need for research that examines model's performance on more varied and less curated datasets.

Furthermore, the challenge of generalization persists, with many models demonstrating effectiveness on known data but struggling to maintain accuracy in the face of new deepfake techniques or unknown data. The field lacks sufficient exploration of the robustness of detection models against adversarial attacks, highlighting a critical gap in ensuring the reliability of detection methods in real-world scenarios. The scarcity of work comparing different detection models on fakes generated using different GANs is evident.

An organized research for identifying the most robust detection algorithm capable of performing well on all types of deepfakes across various GAN architectures is essential. The lack of established evaluation metrics and benchmarks makes it difficult to compare detection models consistently, which emphasizes the necessity of research projects targeted at creating accurate and consistent evaluations of model performance.

4. PROBLEM STATEMENT

The objectives of our research are to

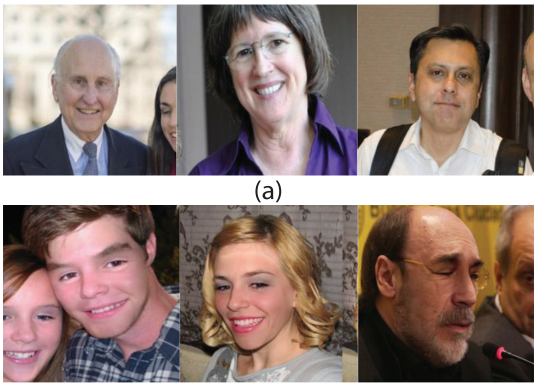
- Generate deepfake images using three different GANs, so that we have diverse fake images to test our detection methods.
- Train eight different CNN models on the dataset to detect fake images, which will be our detection models.
- Compare the performance of deepfake detection models when they are tested on the diverse fake images that are generated different GANs in order to suggest the best performing deepfake detection method.

5. METHODOLOGY

Fig. 3 shows the general preprocessing and detection flow that the model is going through.

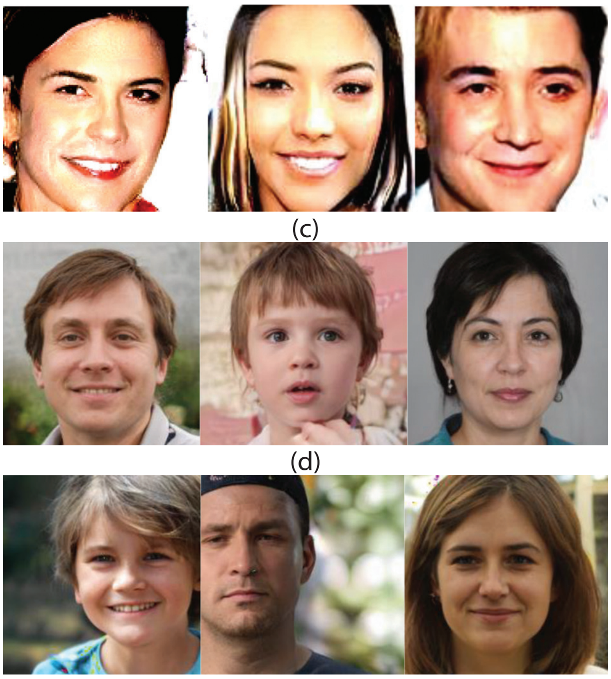
5.1. DATASET

The dataset used comprised of both real and fake images Fig. 1 shows sample images, real and fake, along with the fake images generated using ProGAN, StyleGAN2, and StyleGAN3.



(a)

(b)



(c)

(d)

(e)

Fig. 1. Dataset containing (a) Real , Fake from (b) OpenForensics, (c) ProGAN, (d) StyleGAN2, and (e) StyleGAN3. [19]

We used the Openforensics dataset [19] which is an open dataset and contains approximately 200,000 images. It was split in the ratio 70:20:10 (70% training, 20% validation, and 10% testing). The quantity of images used in the datasets for training, testing, and validation is displayed in Table 1.

A dataset of 15,000 fake images was generated from the three pre-trained GAN models, five thousand from each. We added 5,000 and 1,400 fake generated images from each GAN model for training, testing and validation. This increased the robustness, diversity, and overall quality of the dataset before it was used for training.

Table 1. The Dataset Utilized

Datasets	Number of images		
	Train	Validation	Test
Real	70001	19787	5413
Fake	70001	19641	5492

5.2. GENERATION OF DEEPFAKES

Generative Adversarial networks (GANs) are mostly used to generate fake media. A GAN consists of two parts. The first is the generator, which generates the fakes. It starts with a random vector and keeps improving until it generates an image of the desired quality. The discriminator in the second section determines whether the data produced by the generator is real or fake based on real training data. If the discriminator correctly classifies the generator's fake as fake, then the generator updates its model weights to generate better fakes, and if the discriminator fails to recognize the fake of the generator, then the discriminator updates its model weights to better distinguish between real and fake. Both the generator and discriminator keep updating their models in a loop until the generator can generate fake images good enough to fool the discriminator. Fig. 2 depicts the GAN architecture nicely in a pictorial manner. It shows how the two parts work together, as mentioned above.

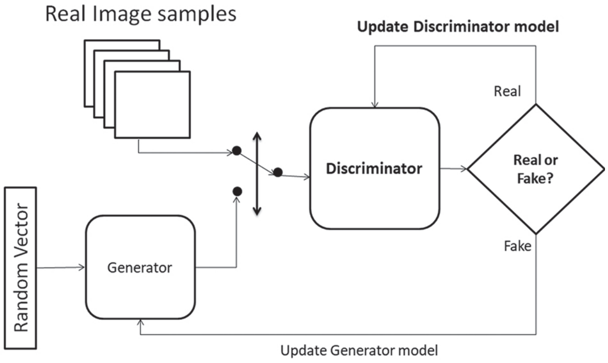


Fig. 2. Block diagram of GAN

For generating fakes, we used three types of pre-trained GAN models: StyleGAN2, StyleGAN3, and ProGAN. A total of 5000 tests were generated for each GAN model to test its detection methods. ProGAN, short for Progressive GAN, was trained on the 'CelebA' dataset and produced images with a resolution of 128x128 pixels [20]. Its progressive training approach starts with low-resolution and gradually increases the resolution, allowing it to capture fine details as it progresses.

In contrast, StyleGAN2 and StyleGAN3 are both high-resolution GANs. With a 256x256 model, they were trained on the 'FFHQ' dataset [21], which includes human faces and is known for generating exceptionally high-quality images. Since SyleGAN3 is advancement over StyleGAN2, it generated the best fake images of them all.

Algorithm 1 outlines the process of generating fake images using pre-trained GAN models. Initially, the algorithm loads the pre-trained GAN model from a specified file and extracts the generator network responsible for generation. Afterward, it sets parameters such as the number of fakes to generate and the truncation factor for controlling quality. Through a loop, the algorithm generates each fake image by creating a random latent space vector, feeding it into the generator network, and converting the output into a recognizable format. These generated fake images are then saved to a designated directory. By systematically iterating through these steps, the algorithm efficiently produces a set of fake images, leveraging the capabilities of pre-trained GAN models.

Algorithm 1 - Deepfake generation using GAN

```
Input:
•   Pretrained model
•   Truncation factor (truncation_psi) or latent dimension for controlling quality
Output:
•   Fake generated images stored in specified directory.

Load Pretrained GAN Model:
•   Load the pre-trained GAN model from the specified file.
•   Extract the generator network (G) from the loaded model.
Create Output Directory
Generate fake images:
•   Loop for each :
a.  Generate a random latent space vector using torch.randn.
b.  Generate an  using the generator network (G) with the specific latent space and conditioning.
c.  Convert the PyTorch tensor to a PIL .
d.  Save the generated  in the output directory with a unique filename.
End
```

5.3. DETECTION OF DEEPFAKES

CNN models are frequently used for detection tasks and usually use an encoder-decoder design. The CNN encoder creates a condensed feature vector after processing the inputs. The desired output is then produced by a CNN decoder using this feature representation. In this system, a CNN model is used for training the datasets and has the best accuracy.

The goal is to ascertain the relative performance of each model in identifying deepfake content. Models such as ResNet50V2, DenseNet121, VGG16, VGG19, InceptionNetV3, InceptionResNetV2, Xception, and MobileNetV2 are being examined in greater detail. In this manner, we may truly learn about their distinct advantages and disadvantages in terms of spotting deepfakes.

We may choose the model or combination of models that works best for our deepfake detection task by evaluating each model's accuracy independently. By using this technique, we can improve the deepfake detection system and make it more dependable and capable of handling the rapidly changing deepfake technology landscape.

The basis for constructing and optimizing the eight different CNNs is our training dataset, which consists of more than 140,000 images.

Hyperparameter and Training Settings:

- Learning Rate: 0.0001

- Activation Method: ReLu
- Optimizer: Adam optimizer
- Batch Size: 64
- Number of Epochs: 10

The model was trained with a learning rate of 0.0001 and the Adam optimizer, which combines the benefits of AdaGrad and RMSProp. With a batch size of 64 to fit GPU memory, the model was trained for 10 epochs to balance training time and performance.

After the training phase, we use a testing dataset of about 10,000 images to thoroughly evaluate the model's performance. Each model is tasked with determining whether a given image is real or fake throughout this review. After testing the model, we predict whether it is real or fake and then calculate the accuracy of the model.

Algorithm 2 outlines the steps involved in building, training, and evaluating a deepfake detection model using a generic CNN architecture. The flexibility of using CNN allows for customization based on specific requirements and facilitates the development of an effective deepfake detection system.

```
Algorithm 2: Deepfake Detection using CNN
Input:
  • datasets for training, validation, and testing (real and fake images)
  • Hyperparameters for the CNN model
  • Number of training iterations
Output:
  • Trained CNN model
Start
  • Import necessary libraries and modules.
  • Set the base path for the dataset.
Prepare the Dataset
  • Load and preprocess the training, validation, and testing datasets.
  • Visualize a sample of s from the training set.
Build the Model
  • Define the architecture for the CNN model.
  • Utilizing the Adam optimizer and categorical cross-entropy loss, compile the model.
Define Callback for Evaluation
  • Create a custom callback (Prediction Callback) to evaluate the model on the validation set after each epoch.
Train the Model
  • Set the number of training steps and validation steps based on batch size and dataset size.
  • Train the model using the training and validation datasets.
  • Utilize the custom callback for evaluating the model's performance on the validation set.
Save the Model
  • Save the trained CNN model for future use.
End
```

Fig.3. shows the block diagram of our detection model, where it shows how we train the model and then preprocess the dataset, after preprocessing the model, it is trained on the different CNN architectures. After training the model has been exported and tested on the test dataset which consists of 10,000 images containing both real and fake. Then the accuracy of the model has been calculated.

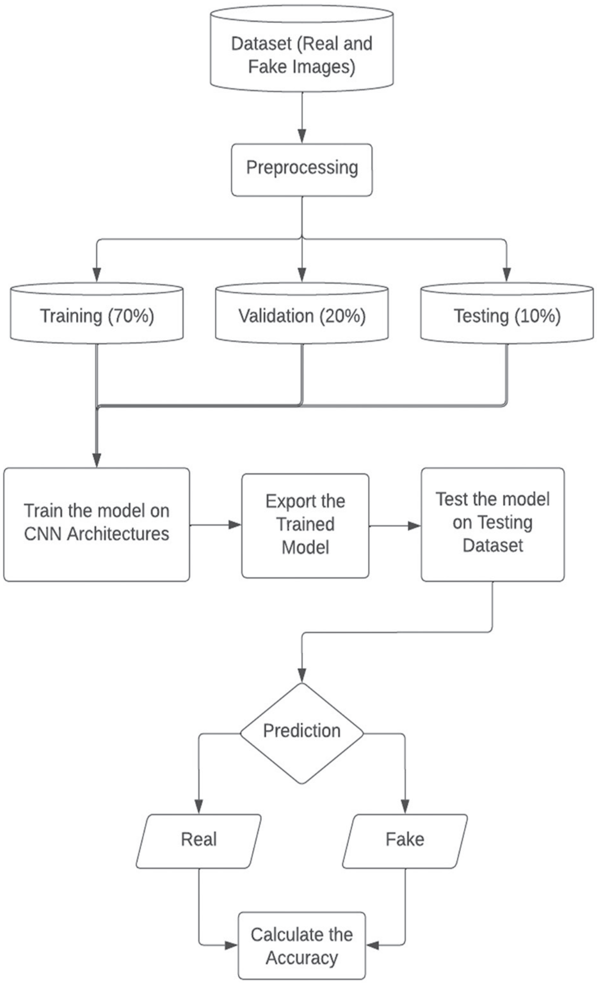


Fig. 3. Block diagram of detection model

Our approach to deepfake detection follows a structured and well-thought-out flow. It all starts with the data collection of both real and fake images that form the basis of our system. To make this data useful, we take a step called preprocessing, where we divide it into three key parts: the training set, the validation set, and the test set. We distribute them in a balanced way, with 70% for training, 20% for validation, and 10% for testing. CNNs are an effective technique that we utilize to train the model using the training dataset.

We used exported models for deepfake detection in the testing set. This indicated the true effectiveness of our deepfake detecting algorithm. It served as performance evaluation of the system, demonstrating its dependability and efficiency in exposing misleading material across a range of contexts.

6. RESULT AND DISCUSSION

Training and testing of the models was done on cloud infrastructure. It featured dual Intel Xeon Silver 4114 CPUs with 40 cores, 128GB of DDR4-2666 ECC Memory, and an Nvidia Tesla V100 GPU with 32GB VRAM. With 4TB of HDD storage and a 100 Mb/s Ethernet interface, it was well-equipped for demanding Deep learning tasks.

A dataset containing 200,000 distinct real and fake images were used to train the model. Random images were fed into the testing process to determine whether or not they were real.

Despite initial success with the OpenForensics dataset, testing on deepfake images from StyleGAN2, StyleGAN3, and ProGAN revealed underwhelming accuracies. Table 2. Shows the obtained accuracies, which ranged from 20% to 50%, these accuracies are noticeably low across multiple CNN architectures. These results tell us about how well the model could distinguish between real and fake.

Table 2. Comparison of Detection Accuracies of CNN models tested on various GANs

Models	Comparison of Detection Accuracies of CNN models tested on various GANs		
	Style_GAN_2_ FFHQ_256	Style_GAN_3_ FFHQ_256	ProGAN_ CelebA_128
VGG16	29.620%	21.342%	35.305%
VGG19	35.343%	26.355%	42.397%
DenseNet121	30.650%	23.165%	38.525%
MobileNetV2	29.420%	20.270%	33.447%

In the initial stage when we evaluated the performance, we realized that the model needed to be trained on all of the datasets, including ProGAN, StyleGAN2, and StyleGAN3. Then a calculated choice was made to add more images to the training dataset created by each of the three GAN models—Style_GAN_3, Style_GAN_2, and ProGAN to improve accuracy.

The objective of this addition was to ensure that the model was exposed to better quality fake images and a wider variety of fake images by adding more diversity and balance to the dataset. So, to address the initial low accuracy rates, the newly generated fake images were subsequently included in the training and validation sets of the dataset.

After making this modification, the model's performance was significantly improved. Across all three GAN datasets, the re-trained models showed a notable increase in accuracy after being trained on better fake images. Experiments with different activation strategies and learning rates were conducted to achieve even better results. ReLU activation and 0.0001 learning rate were found to work best for the model.

Fig.4. shows the graph of loss in training and loss in validation vs the epochs. Graphs of all the eight models have been included in the figure. Optimal configurations of 64 batch sizes and 10 epochs were determined through systematic testing for all CNN models.

In Fig.4 it can be observed that VGG16 has the best training-validation loss curve as it has good training loss convergence and the validation loss also converges close to training loss with little fluctuations. Some of the model's validation graphs were smooth with less fluctuation, and good convergence and some had spikes and variation as compared to training loss. This indicates how the models performed on unseen data as compared to seen data and help determine which model works best on unseen data. Some models do not have a good training validation graph, it is because the model has not generalized well that is it has not performed well on unseen data, its accuracy is bad and the other reason is that sometimes the validation data differs in quality to that of the training data. The models that have better validation graphs have generalized well on unseen data.

Table 3. Shows the accuracy of summarizing different CNNs on the difficult job of detecting manipulated images part of our thorough review of deepfake detection approaches.

Table 3. Comparison of Detection Accuracy for Various GAN Models Using Different CNNs

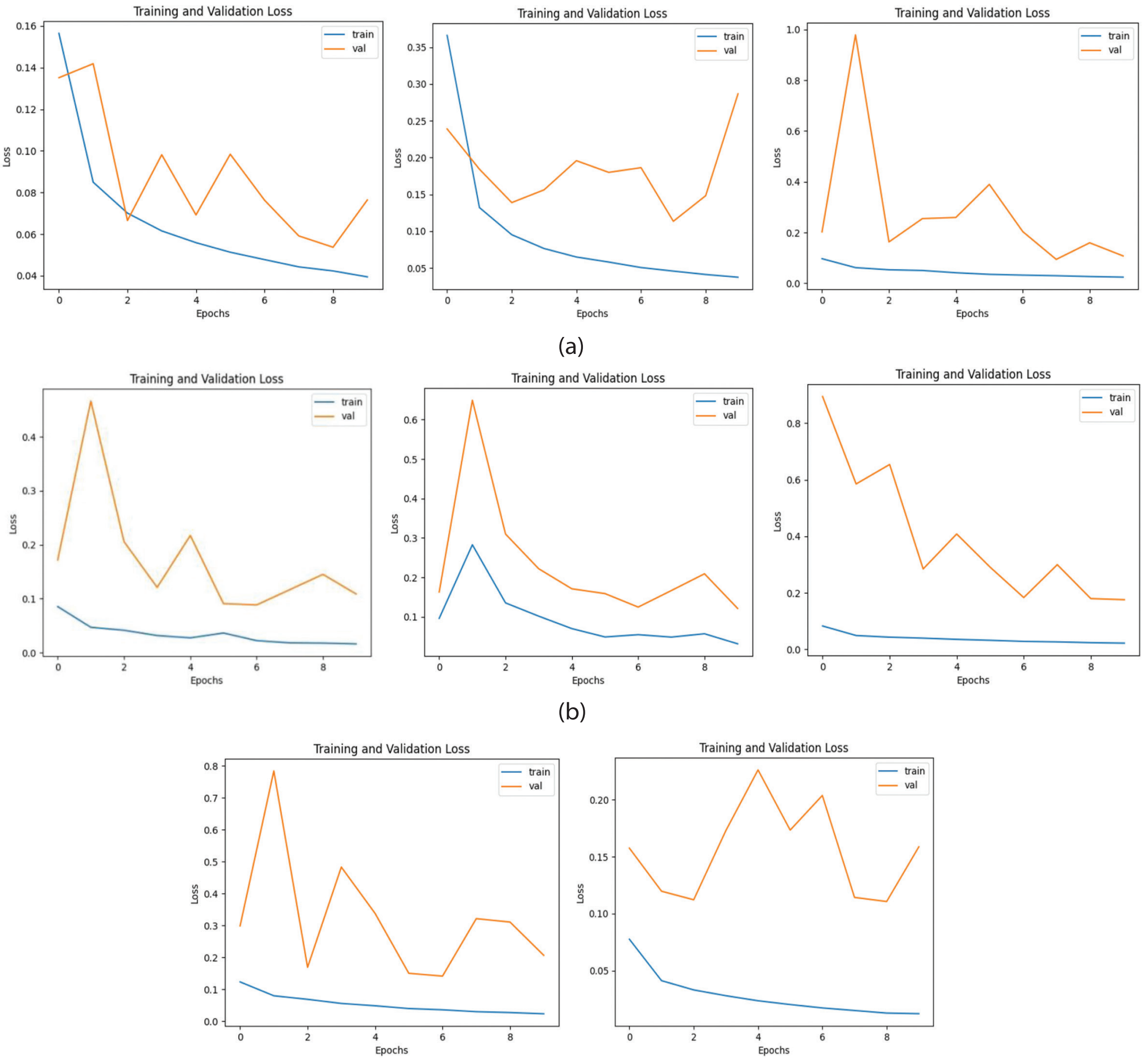
Models	Comparison of Detection Accuracy for Various GAN Models Using Different Convolutional Neural Networks (CNNs)		
	Style_ GAN_2_ FFHQ_256	Style_ GAN_3_ FFHQ_256	ProGAN_ CelebA_128
VGG16	95.038%	94.901%	94.987%
VGG19	97.983%	96.744%	97.397%

DenseNet121	95.650%	95.045%	95.525%
InceptionResNetV2	96.495%	96.380%	96.447%
InceptionV3	97.052%	96.668%	96.831%
MobileNetV2	95.707%	95.592%	95.659%
ResNet50V2	95.150%	93.230%	95.304%
Xception	96.956%	96.908%	96.898%

In Table 3. Three different deepfake datasets are used to thoroughly evaluate each CNN's performance in differentiating between real and fake content: Style_GAN_2_FFHQ_256, Style_GAN_3_FFHQ_256, and ProGAN_CelebA_128. The accuracy in percentage values of the models under consideration—VGG16, VGG19, DenseNet121, InceptionResNetV2, InceptionV3, MobileNetV2, ResNet50V2, and Xception is achieved.

VGG19 emerges as a strong contender with the highest accuracy of 97.983% on the Style_GAN_2_FFHQ_256 dataset and 97.397% on ProGAN_CelebA_128. Xception demonstrates its robustness on the Style_GAN_3_FFHQ_256 dataset by attaining the highest accuracy of 96.908%. Some of the models like InceptionResNetV2 and InceptionV3 had accuracy above 96% in all of the datasets and they were consistent along with the other models.

Overall, for all the models the accuracy ranged between 94% and 98%. Models gave the least accuracy on StyleGAN3 images because they were of the best quality amongst the images created by three GANs. ProGAN and StyleGAN2 images were comparatively detected with greater accuracy.



(a)

(b)

(c)

Fig. 4. Loss in training and validation where **(a)** VGG16, VGG19, and DenseNet121, **(b)** InceptionResNetV2, InceptionV3, and MobileNetV2, and **(c)** ResNet50V2 and Xception

7. CONCLUSION

Our study shows that different models work well in different situations for spotting deepfake images. We tested the eight CNN models against fake images from three GANs: StyleGAN2, StyleGAN3, and ProGAN. VGG19 and VGG16 do great in some cases, while InceptionV3 and Xception are consistently good giving an accuracy above 96.6% for all three GANs. The best-performing model however is VGG19 since it has the best overall accuracy across the three GANs. So our study based on the performances of the CNN models concludes that VGG19 is the better alternative to detect deepfake images coming from various sources.

With more powerful GPUs and CPUs, we can generate and detect deepfakes more efficiently. Advanced systems enable the use of models like EfficientNet, a highly effective CNN architecture, further enhancing our deepfake detection capabilities.

With the rise of artificial intelligence, the quality of deepfakeimages is only going to increase thus making their detection a continuous research topic. Our goal was to find a model that works well on fake s generated through diverse sources thus making it a reliable tool for countering the ever-evolving deepfake creation.

8. REFERENCES:

[1] M. Kumar, N. Muhal, “Fake Face s Generated From Different GANs”, https://www.kaggle.com/datasets/mayankjha146025/fake-face-s-generated-from-different-gans (accessed: 2024)

[2] Y. Digvijay, S. Salmani, "Deepfake: A survey on facial forgery technique using the generative adversarial network", Proceedings of the International Conference on Intelligent Computing and Control Systems, Madurai, India, 15-17 May 2019, pp. 852-857.

[3] S. Sanjan, P. Thushara, P. C. Karthik, M. P. A. Vijayan, A. Wilson, “Review of Deepfake Detection Techniques”, International Journal of Engineering Research & Technology, Vol. 10, No. 5, 2021, pp. 813-816.

[4] A. Malik, M. Kuribayashi, S. M. Abdullahi, A. N. Khan, "DeepFake detection for human faces and videos: A survey", IEEE Access, Vol. 10, 2022, pp. 18757-18775.

[5] M. S. Rana, M. N. Nobi, B. Murali, A. H. Sung, "Deepfake Detection: A Systematic Literature Review", IEEE Access, Vol. 10, 2022, pp. 25494-25513.

[6] O. A. Paul, "Deepfakes Generated by Generative Adversarial Networks", Georgia Southern University, Honors College Theses, 2021.

[7] T. T. Nguyen, Q. V. 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 Understanding, Vol. 223, 2022, p. 103525.

[8] T. Shen, R. Liu, J. Bai, Z. Li. "'Deep fakes' using generative adversarial networks (GAN)", Noiselab, University of California, San Diego, 2018, Report 16.

[9] O. Giudice, L. Guarnera, S. Battiato, "Fighting deepfakes by detecting GAN DCT anomalies", Journal of Imaging, Vol. 7, No. 8, 2021, p. 128.

[10] H. S. Shad, M. M. Rizvee, N. T. Roza, S. M. Hoq, M. M. Khan, A. Singh, A. Zaguia, S. Bourouis, "Comparative analysis of deepfake detection method using convolutional neural network", Computational Intelligence and Neuroscience, Vol. 2021, 2021.

[11] D. Saxena, J. Cao, "Generative adversarial networks (GANs) challenges, solutions, and future directions", ACM Computing Surveys, Vol. 54, No. 3, 2021, pp. 1-42.

[12] A. Khodabakhsh, R. Ramachandra, K. Raja, P. Wasnik, C. Busch, "Fake face detection methods: Can they be generalized?", Proceedings of the International conference of the biometrics special interest group, Darmstadt, Germany, 26-28 September 2018, pp. 1-6.

[13] O. Patashnik, Z. Wu, E. Shechtman, D. Cohen-Or, D. Lischinski, "StyleCLIP: Text-Driven Manipulation of StyleGAN Imagery", Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 10-17 October 2021, pp. 2065-2074.

[14] M. Kumar, H.K. Sharma, "A GAN-based model of deepfake detection in social media", Procedia Computer Science, Vol. 218, 2023, pp. 2153-2162.

[15] A. Tiwari, R. Dave, M. Vanamala. "Leveraging deep learning approaches for deepfake detection: A review", Proceedings of the 7th International Conference on Intelligent Systems, Metaheuristics & Swarm Intelligence, 2023, pp. 12-19. 2023.

[16] E. Nowroozi, Y. Mekdad. "Detecting high-quality GAN-generated faces using neural networks", Big

Data Analytics and Intelligent Systems for Cyber Threat Intelligence, River Publishers, 2023, pp. 235-252.

[17] S. Preeti, M. Kumar, H. K. Sharma. "Robust GAN-Based CNN Model as Generative AI Application for Deepfake Detection", EAI Endorsed Transactions on Internet of Things, Vol. 10, 2024.

[18] B. D. Sergi, S. D. Johnson, B. Kleinberg, "Testing human ability to detect 'deepfake' s of human faces", Journal of Cybersecurity, Vol. 9, No. 1, 2023.

[19] T. N. Le, H. H. Nguyen, J. Yamagishi, I. Echizen, "Openforensics: Multi-face Forgery Detection and Segmentation In-the-wild Dataset [V.1.0.0]", https://doi.org/10.5281/zenodo.5528418 (accessed: 2023)

[20] CelebA-Dataset, "Large-scale CelebFaces Attributes (CelebA) Dataset", https://mmlab.ie.cuhk.edu.hk/projects/CelebA.html (accessed: 2024)

[21] FFHQ, "NVlabs/ffhq-dataset: Flickr-Faces-HQ Dataset (FFHQ)", https://github.com/NVlabs/ffhq-dataset (accessed: 2024)