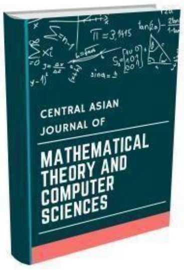


CENTRAL ASIAN JOURNAL OF MATHEMATICAL THEORY AND COMPUTER SCIENCES

https://cajmtcs.centralasianstudies.org/index.php/CAJMTCS

Volume: 06 Issue: 01 | January 2025 ISSN: 2660-5309

Article

Fake Images Identification on Social Media Application Via Generated Adversarial Networks and DenseNet

Noor Fahem Sahib[*1]

1. Department of Food Science and Technology, College of Food Science, Al-Qasim Green University, Babylon, Iraq

* Correspondence: noor@fosci.uoqasim.edu.iq

Abstract: The rapid spread of fake images on social media has become a major concern for individuals, organizations, and governments. These images are often generated using advanced techniques, such as Generative Adversarial Networks (GANs), to manipulate public opinion and spread misinformation. Given the high visual quality of GAN-generated fake faces, detecting them has become increasingly challenging. If misused for image tampering, these synthetic images could lead to serious ethical, moral, and legal issues. Therefore, developing automated detection tools is essential to identify and mitigate the risks associated with synthetic media. In this study, we introduce a DenseNet based approach to detect GAN-generated fake images. Experimental results demonstrate that our proposed model achieves high accuracy, exceeding 98.1%, in distinguishing real and fake faces. Also, it acquired high Sensitivity of 1.00%, Specificity of 97. 6% and F1 scores of 98.6%.

Keywords: Fake Images detection, Generative Adversarial Networks (GAN), Deep Learning (GAN-CNN)

Citation: Sahib, N.F. Fake Images Identification on Social Media Application Via Generated Adversarial Networks and DenseNet. Central Asian Journal of Mathematical Theory and Computer Sciences 2025, 6(1), 119-127.

Received: 18th Feb 2025
Revised: 20th Feb 2025
Accepted: 25st Feb 2025
Published: 27th Feb 2025

1. Introduction

Online social networks such as Facebook, Instagram, and Twitter have grown in popularity as information sharing and dissemination tools in recent years. These platforms allow for fast communication and enable users to share various forms of multimedia, including images, videos, and audio [1]. However, their large user base and open accessibility have also made them a target for cybercriminals, who exploit these networks for malicious purposes. Fake content is often used to manipulate public opinion and spread harmful agendas. Due to their potentially serious consequences, fake images have now become a significant social threat [2].

AI has seen major and rapid growth in recent years, with deep neural networks as the first instrument to solve different problems such as speech recognition, object detection, and image classification. Unlike traditional methods that rely on handcrafted features, deep learning utilizes stacked layers to automatically learn hierarchical representations from input data. Advanced models such as Convolutional Neural Networks (CNNs) and Generative Adversarial Networks (GANs) [3] have been widely explored and have achieved remarkable success in various image-related tasks, including image style transfer and super-resolution [4]. GANs generate highly realistic visuals by modeling probability distributions, where the generator and discriminator compete to improve sample quality. However, GANs are also heavily used in deepfake technology, which, if misused, can lead to harmful consequences. Deepfake images, often indistinguishable from real ones, can be exploited to spread misinformation about public figures such as politicians and celebrities [5]. To address this issue, digital forensic experts are actively developing detection techniques using CNN-based models and adaptive learning environments to accurately identify and differentiate manipulated digital content [6].

"The subsequent portions of this work are structured as follows: Section 2 presents related works that employed GAN-based algorithms for facial generation. Section 3 presents the proposed detection strategy based on GANs and DenseNet. Section 4 presents the experimental data and a discussion. Section 5 concludes the work with significant findings and proposes potential directions for additional inquiry".

Related Work

In the related work, several studies have been proposed regarding the detection of fake images on social media application. Recent studies have demonstrated that GAN models can generate fake face images with high visual quality. Since these synthetic faces can deceive human perception, detecting fake images has become a crucial challenge in image forensics.

Marra, et al. [7] A pioneering study on deep network-based detectors was conducted in this study, demonstrating that state-of-the-art pre-trained CNNs such as Xception, Inception, and DenseNet achieve outstanding performance in detecting GAN-generated images. Notably, these models outperform custom-designed CNN architectures developed specifically for forensic applications, particularly in complex and challenging detection scenarios.

Frank, et al. [8] conducts a frequency-domain analysis to identify artifacts present across various network architectures, datasets, and resolutions. These artifacts serve as distinguishing features to differentiate generated images from real ones. Specifically, a CNN-based classifier is trained using Fourier spectra extracted from both authentic images and their synthetic counterparts, which are produced using an adversarial auto encoder.

McCloskey, Scott, and Michael Albright, [9] Current research on detecting GAN-generated fake images primarily focuses on signal-level features to identify forgeries. This study analyzes GAN generators and finds that saturated pixel frequencies are limited, and RGB channels collapse using weights that differ from the spectral sensitivities of digital cameras. By examining the frequency of over- and under-exposed pixels, a basic forensic method is applied to distinguish GAN-generated images from real camera images. Additionally, the study introduces intensity noise histograms as a tool for classifying authentic and synthetic images.

Wang, et al. [10] A universal detection technique was proposed to identify CNN-generated fake faces. The study highlights that CNN-generated images contain systematic flaws that prevent them from being completely indistinguishable from real images. Despite using different CNN generators, these synthetic images retain detectable fingerprints. A well-trained image classifier can learn to recognize these CNN fingerprints. The research employed ResNet-50 as a classifier, training it on the ProGAN dataset. Additionally, various data augmentation techniques were applied to enhance the detection of post-processing manipulations.

Choi et al. [11] A CNN-based method was proposed for composite forgery detection. The proposed CNN-based technique detects three common image forgery attacks and identifies their simultaneous application. It learns statistical changes caused by manipulation to classify forged images accurately. By analyzing composite manipulations, it distinguishes altered images from originals. The method is practical as real-world attacks often involve multiple manipulations. Experimental results confirm its high reliability and superior performance over traditional forensic approaches.

Li, Chuqiao, et al. [12] The study introduces a Continuous Deepfake Detection Benchmark (CDDB) designed for ongoing deepfake identification. It incorporates multiple models, appropriate evaluation metrics, and multiclass incremental learning techniques, which are widely used in continuous visual recognition, to tackle the challenge effectively.

Jaiswal and Srivastava [13] A logistic regression model was used to detect image splicing by first converting images to grayscale and extracting four feature sets: LBP, Laws Texture Energy (LTE), HoG, and DWT. The model was trained on 142 feature vectors and achieved over 98% accuracy on CASIA 1.0, CASIA 2.0, and Columbia datasets. However, its performance declines on severely downscaled images due to loss of texture and clarity.

Karras, Tero, et al. [14] A progressive strategy was introduced to develop and train GANs for producing high-quality images. Instead of training the entire GAN on high-resolution images from the start, the process begins with a basic GAN trained on low-resolution images. Gradually, more layers are added to adapt the model to higher-resolution images during training. Experimental results show that most fake face images generated at 1024 × 1024 resolution using this method are nearly indistinguishable to the human eye.

2. Materials and Methods

The proposed system contains three phases, the first phase is the preprocessing phase which begins with converting images into fixed (256×256) size, the "Contrast Limited Adaptive Histogram Equalization (CLAHE)" technology was applied to enhance contrast and demonstrate the feature of image and the normalization to divide the image elements by 255 to make the values between 0 and 1 to speed up the processing process. The second phase, The GAN consists of two neural networks: the generator and the discriminator. DenseNet model is then trained on this merged dataset, ensuring its adaptability and robustness in detecting deepfakes in dynamic environments

The proposed approach is demonstrated in Fig 1.

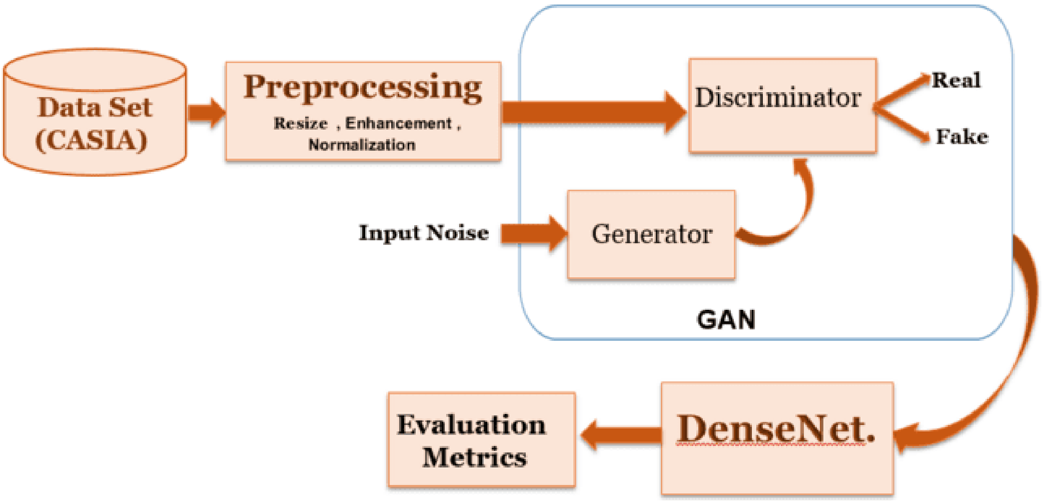


Figure 1. Flowchart of the Proposed System.

Data set

The CASIA dataset is a face dataset created by the Chinese Academy of Sciences Institute of Automation (CASIA) for face recognition research. It is widely used in the field of computer vision and biometrics. It is Natural color image repository with realistic tampering operations, available for the public for research (http//forensics.idealtest.org/ 16M). It is often involving assembling a dataset of social media images, including both real and potentially manipulated images. we used the "Real and Fake" dataset in a ratio of 80:20 for training and testing purposes.

Preprocessing Phase

Preprocessing is a critical phase in fake image detection, preparing data for subsequent feature extraction and emphasizing disease-specific characteristics to enhance model accuracy. This process involves several key steps: Standardizing images to a uniform size ensures consistency across the dataset. Applying Contrast Limited Adaptive Histogram Equalization (CLAHE) improves image contrast, highlighting important features. Adjusting pixel values to a standard range facilitates effective model training.

Resize CT images.

All images are resized to a fixed 256×256 resolution to ensure compatibility with deep learning systems. This scaling step helps reduce computational costs while enhancing processing efficiency. Resizing the images to 256×256 pixels optimizes them for further analysis and accelerates deep learning model training.

Images enhancement (CLAHE).

In this phase, image preprocessing is conducted using the Contrast Limited Adaptive Histogram Equalization (CLAHE) technique. CLAHE is a potent method for enhancing image contrast, effectively increasing the distinction between features, reducing noise and blurring, and highlighting details such as edges and boundaries without altering the image's inherent structure. The core concept involves dividing the input image into equally sized, non-overlapping regions known as tiles. Unlike global histogram equalization, CLAHE computes histograms for each tile, relying on local histograms. To mitigate the issue of noise over-amplification, the algorithm clips the histogram at predefined values and redistributes the excess uniformly across other tiles before calculating the cumulative distribution function. Subsequently, the cumulative histogram is used to perform the equalization. CLAHE operates with two primary parameters: the clip limit (CL), a numerical value that specifies the threshold for noise amplification, and the number of tiles (NT), which defines the count of non-overlapping sub-regions [15].

Normalization

In image processing, normalization—also known as contrast stretching or histogram stretching—is a technique that adjusts the range of pixel intensity values to enhance image contrast. This method is particularly useful for images with poor contrast due to factors like glare. In broader data processing contexts, such as digital signal processing, this technique is referred to as dynamic range expansion. The primary objective of normalization is to transform an image or signal into a range that is more familiar or normal to human perception, thereby improving its interpretability. When applied to a collection of data, signals, or images, normalization aims to maintain a consistent dynamic range across the set.

GANs and DenseNet method to detect fake images.

The GAN consists of two neural networks: the generator and the discriminator. The generator is responsible for producing synthetic data that closely resembles real data, while the discriminator's role is to differentiate between genuine and artificially generated samples. These networks undergo adversarial training, where the generator continuously refines its outputs to deceive the discriminator, and the discriminator enhances its ability to classify real and fake data accurately.

The architecture relies on convolutional layers, avoiding max pooling and fully connected layers. Instead, convolutional strides and transposed convolutions are utilized for downsampling and upsampling. Deep Convolutional GAN specifically incorporates convolutional layers in both the generator and discriminator to improve performance. The architectural layers of the network generator and discriminator are shown in table 1.

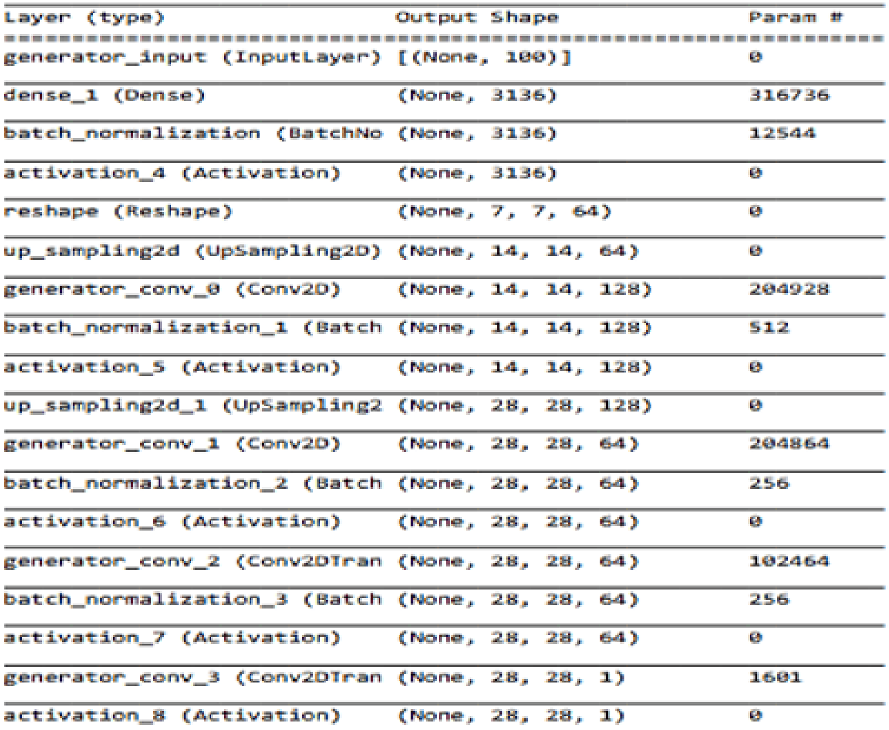
Two distinct loss functions govern this approach—one for the generator and another for the discriminator. These functions measure the difference between the generator's output and actual data. The model employs a min-max loss function based on crossentropy, where the generator aims to minimize the loss by generating highly realistic data, while the discriminator attempts to maximize it by correctly identifying fake and real instances.

The GAN model is trained using a CASIA dataset of real images after preprocessing steps. The model's GAN component learns to generate synthetic data that closely mimics the original data. Once trained, the GAN is used to produce fake samples that mirror real world images and is provided with a dimensionality-reduced feature, these fake samples are saved for future use.

The DenseNet. is a very deep CNN architecture [16], based on the residual network (ResNet), which pursues effectively feature propagation and reuse within the network. Unlike in early CNN architectures, for each layer of DenseNet, the feature-maps of all preceding layers are used as inputs, and its own feature-maps are used as inputs into all subsequent layers. By creating shortcuts between input and output, feature propagation and reuse is encouraged, and the vanishing-gradient problem is much alleviated. Moreover, the number of parameters is much reduced ensuring faster training. DenseNet are shown to ensure excellent performance for GAN image detection.

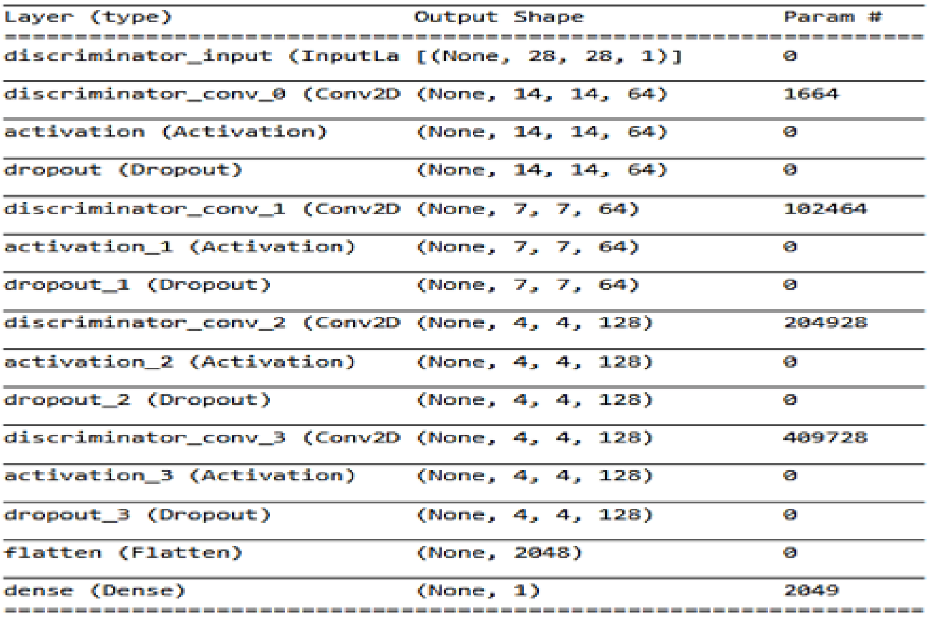
The fake samples from the GAN trained are combined with the real samples to produce a mixed dataset. The DenseNet model is then trained on this merged dataset, ensuring its adaptability and robustness in detecting deepfakes in dynamic environments.

Table 1. (a) shows the architectural layers of the network generator



```
Layer (type)                 Output Shape              Param #
=================================================================
generator_input (InputLayer) [(None, 100)]             0

dense_1 (Dense)              (None, 3136)              316736

batch_normalization (BatchNo (None, 3136)              12544

activation_4 (Activation)    (None, 3136)              0

reshape (Reshape)            (None, 7, 7, 64)          0

up_sampling2d (UpSampling2D) (None, 14, 14, 64)        0

generator_conv_0 (Conv2D)    (None, 14, 14, 128)       204928

batch_normalization_1 (Batch (None, 14, 14, 128)       512

activation_5 (Activation)    (None, 14, 14, 128)       0

up_sampling2d_1 (UpSampling2 (None, 28, 28, 128)       0

generator_conv_1 (Conv2D)    (None, 28, 28, 64)        204864

batch_normalization_2 (Batch (None, 28, 28, 64)        256

activation_6 (Activation)    (None, 28, 28, 64)        0

generator_conv_2 (Conv2DTran (None, 28, 28, 64)        102464

batch_normalization_3 (Batch (None, 28, 28, 64)        256

activation_7 (Activation)    (None, 28, 28, 64)        0

generator_conv_3 (Conv2DTran (None, 28, 28, 1)         1601

activation_8 (Activation)    (None, 28, 28, 1)         0
```

Table 2. (b) shows the architectural layers of the network discriminator



```
Layer (type)                 Output Shape              Param #
=================================================================
discriminator_input (InputLa [(None, 28, 28, 1)]       0
_________________________________________________________________
discriminator_conv_0 (Conv2D (None, 14, 14, 64)        1664
_________________________________________________________________
activation (Activation)      (None, 14, 14, 64)        0
_________________________________________________________________
dropout (Dropout)            (None, 14, 14, 64)        0
_________________________________________________________________
discriminator_conv_1 (Conv2D (None, 7, 7, 64)          102464
_________________________________________________________________
activation_1 (Activation)    (None, 7, 7, 64)          0
_________________________________________________________________
dropout_1 (Dropout)          (None, 7, 7, 64)          0
_________________________________________________________________
discriminator_conv_2 (Conv2D (None, 4, 4, 128)         204928
_________________________________________________________________
activation_2 (Activation)    (None, 4, 4, 128)         0
_________________________________________________________________
dropout_2 (Dropout)          (None, 4, 4, 128)         0
_________________________________________________________________
discriminator_conv_3 (Conv2D (None, 4, 4, 128)         409728
_________________________________________________________________
activation_3 (Activation)    (None, 4, 4, 128)         0
_________________________________________________________________
dropout_3 (Dropout)          (None, 4, 4, 128)         0
_________________________________________________________________
flatten (Flatten)            (None, 2048)              0
_________________________________________________________________
dense (Dense)                (None, 1)                 2049
=================================================================
```

3. Results

This section provides a detailed evaluation of the proposed approach's findings. The experiments were conducted on the Google Colab, utilizing its GPU-enabled backend with 16GB of RAM. The implementation employed TensorFlow with Keras as the backend.

Evaluation Metrics

Several criteria have been employed to assess the performance of the model:

a. The accuracy checks the number of correctly classified instances, whether positive or negative instances. **Accuracy (Acc) = (T P + T N / T P + T N + F P + F N).**

b. Sensitivity is the rate of identification of positive samples rightly.

   Sensitivity (Sen) = (T P / T P + F N).

c. Specificity is the percentage of identification of negative examples rightly.

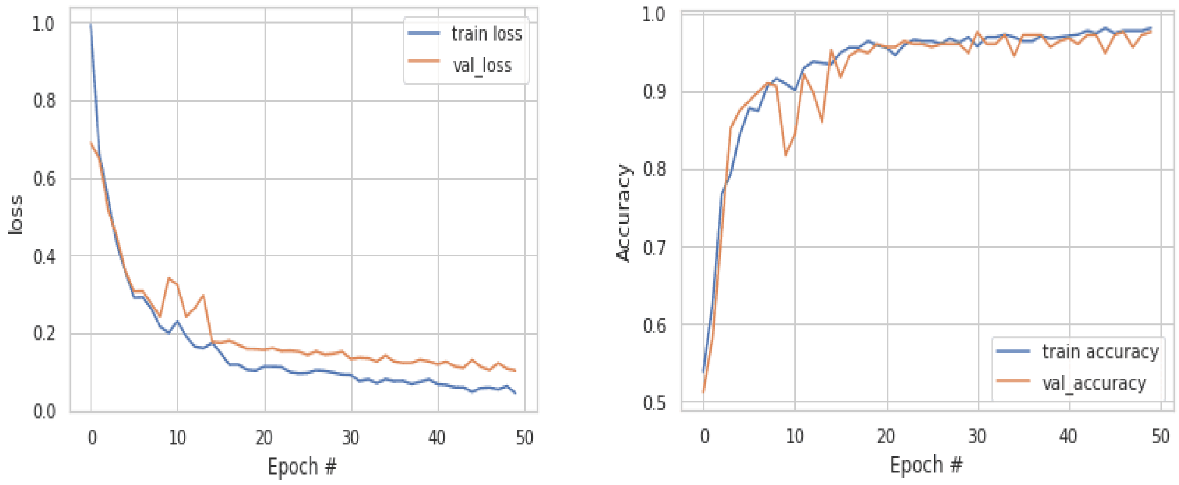
   Specificity (Spe) = (T N / T N + F P)

d. F1-score shows a compound of precision and sensitivity for computing a balanced mean output.

   F1-score = (2 * TP) / (2 * TP + FP + FN).

The Results Over The Casia Data Set

The GAN-Dense net's performance results will be discussed in this section. The training process indicates that the deepfake detection model's performance significantly enhances as the number of epochs increases. With each epoch, the model becomes more proficient at accurately identifying authentic and deepfake images, leading to higher accuracy and reduced loss. This improvement is evident in various models; for instance, after 48 epochs, the model achieved a validation accuracy of 98.1% with minimal training loss. The learning curves (loss of training and validation) plot (Figure 2 (a)) above illustrates the good fit condition because the loss of training and the loss of validation depreciate to a point of equilibrium, and the gap is a little between the two-loss values. It is potentially persistent training of a good fit that would produce a problem of overfitting. Figure 2 (b) illustrates the evaluation of model performance in accuracy score is the total amount of errors the model predicted



(a) **(b)**

Figure 2. (a) the loss of train and validation, (b) the accuracy of train and validation.

The model's performance on a validation or test dataset is evaluated using accuracy and loss. As training progresses, both metrics improve, as seen in Figure 2, indicating that the model is becoming more effective at differentiating between real and deepfake images. A significant decrease in loss further confirms that the model's predictions are becoming more precise. Table 2 presents the test results, showcasing the model's performance in terms of Specificity, Sensitivity, F1-score, and Accuracy.

Table 2. illustrates the performance measures of GAN-Dense net.

Method	Sensitivity	Specificity	F1-score	Accuracy
GAN-Dense net on test data	100%	97.6%	98.5%	98.1 %

Compared to previous methods for detecting real and fake faces, the GAN-DenseNet model achieves an impressive 98.1% accuracy, surpassing other approaches. It outperforms Afchar et al.'s FaceForensics++ Voting Ensemble (94.05%) [17], Huaxiao Mo, et al CNN model (98%) [18], Wang, Yaqing, et al. EANN (82.70%) [19], and Singhal, Shivangi, et al. SpotFake (89.23) [20]. This highlights GAN-DenseNet's superior performance in deepfake detection, showcasing its robustness in this complex task.

4. Discussion

The proposed GAN-DenseNet model demonstrates a remarkable performance in detecting GAN-generated fake images on social media platforms. Achieving an accuracy of 98.1%, with 100% sensitivity, 97.6% specificity, and an F1-score of 98.5%, the model significantly outperforms existing approaches such as FaceForensics++ Voting Ensemble, CNN models, and EANN frameworks. These results highlight the robustness and efficiency of the GAN-DenseNet approach in handling complex deepfake detection tasks. The use of DenseNet architecture, which promotes effective feature propagation and reuse, plays a crucial role in overcoming the vanishing-gradient problem and reducing the number of parameters, thus ensuring faster and more accurate training. Additionally, the gradual improvement in accuracy and decrease in training loss over multiple epochs illustrate the model’s capability to generalize well without overfitting.

The superior performance of the GAN-DenseNet model can be attributed to its innovative integration of GAN-generated synthetic data and DenseNet’s deep convolutional architecture. Unlike previous methods that relied heavily on handcrafted features or shallow network architectures, this model leverages the adversarial training dynamics of GANs, allowing it to learn complex patterns and subtle discrepancies between real and fake images. Moreover, the incorporation of CLAHE and normalization in the preprocessing phase significantly enhances image contrast and consistency, further contributing to the model's detection precision. The comparison with state-of-the-art methods indicates that GAN-DenseNet not only excels in detecting visually realistic deepfakes but also remains resilient to diverse post-processing manipulations. However, future work could explore cross-dataset evaluations and real-world adversarial scenarios to assess the model's adaptability and scalability across various social media contexts.

5. Conclusion

This study proposed a deepfake detection method for social media images using a GANs-DenseNet model with implementing a generative replay technique. The approach involves generating and storing samples from previous tasks and replaying them during new training sessions, significantly enhancing DenseNet's ability to detect deepfakes. The key findings show that the model achieved 97.6% specificity, 100% sensitivity, 98.5% F1-score, and 98.1% accuracy, demonstrating its effectiveness in identifying fake images under dynamic training conditions. Future work will focus on evaluating the model across diverse datasets and ensuring its adaptability to evolving data distributions and adversarial threats in real-world applications.

REFERENCES

[1] Raturi, Rohit. "Machine learning implementation for identifying fake accounts in social network." International Journal of Pure and Applied Mathematics 118.20 (2018): 4785-4797.

[2] Schmidhuber, Jurgen. "Multi-column deep neural networks for image classification." Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2012.

[3] Goodfellow, Ian, et al. "Generative adversarial nets." Advances in neural information processing systems 27 (2014).

[4] Dumoulin, Vincent, Jonathon Shlens, and Manjunath Kudlur. "A learned representation for artistic style." arXiv preprint arXiv:1610.07629 (2016).

[5] Aggarwal, Alankrita, Mamta Mittal, and Gopi Battineni. "Generative adversarial network: An overview of theory and applications." International Journal of Information Management Data Insights 1.1 (2021): 100004.

[6] Cong, Yulai, et al. "Gan memory with no forgetting." Advances in Neural Information Processing Systems 33 (2020): 16481-16494.

[7] Marra, Francesco, et al. "Detection of gan-generated fake images over social networks." 2018 IEEE conference on multimedia information processing and retrieval (MIPR). IEEE, 2018.

[8] Frank, Joel, et al. "Leveraging frequency analysis for deep fake image recognition." International conference on machine learning. PMLR, 2020.

[9] McCloskey, Scott, and Michael Albright. "Detecting gan-generated imagery using color cues." arXiv preprint arXiv:1812.08247 (2018).

[10] Wang, Sheng-Yu, et al. "CNN-generated images are surprisingly easy to spot... for now." Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020.

[11] Choi, Hak-Yeol, et al. "Detecting composite image manipulation based on deep neural networks." 2017 international conference on systems, signals and image processing (IWSSIP). IEEE, 2017.

[12] Li, Chuqiao, et al. "A continual deepfake detection benchmark: Dataset, methods, and essentials." Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. 2023.

[13] Jaiswal, Ankit Kumar, and Rajeev Srivastava. "A technique for image splicing detection using hybrid feature set." Multimedia Tools and Applications 79.17 (2020): 11837-11860.

[14] Karras, Tero, et al. "Progressive growing of gans for improved quality, stability, and variation. arXiv 2017." arXiv preprint arXiv:1710.10196 (2018): 1-26.

[15] Campos, Gabriel Fillipe Centini, et al. "Machine learning hyperparameter selection for contrast limited adaptive histogram Equalization." EURASIP Journal on Image and Video Processing 2019.1 (2019): 1-18. doi.org/10.1186/s13640-019-0445-4

[16] Huang, Gao, et al. "Densely connected convolutional networks." Proceedings of the IEEE conference on computer vision and pattern recognition. 2017.

[17] Afchar, Darius, et al. "Mesonet: a compact facial video forgery detection network." 2018 IEEE international workshop on information forensics and security (WIFS). IEEE, 2018.

[18] Mo, Huaxiao, Bolin Chen, and Weiqi Luo. "Fake faces identification via convolutional neural network." Proceedings of the 6th ACM workshop on information hiding and multimedia security. 2018.

[19] Wang, Yaqing, et al. "Eann: Event adversarial neural networks for multi-modal fake news detection." Proceedings of the 24th acm sigkdd international conference on knowledge discovery & data mining. 2018.

[20] Singhal, Shivangi, et al. "Spotfake: A multi-modal framework for fake news detection." 2019 IEEE fifth international conference on multimedia big data (BigMM). IEEE, 2019.