INDIAN INSTITUTE OF INFORMATION TECHNOLOGY, ALLAHABAD

Detection of "Deep Fake" Images using Deep Learning Techniques

Mini Project Report

VII Semester

Submitted By:

AVISHKAR SINGH	IIT2020008
PRATHAM GUPTA	IIT2020026
SHASHWAT KUMAR GAUTAM	IIT2020030
JAY SUTHAR	IIT2020087
AKSHAT GHARIYA	IIT2020505

Under the Guidance of

Dr. Manish Kumar

ABSTRACT

Robust picture recognition techniques are desperately needed to reduce hazards and threats as deepfake techniques become more widespread. This research paper gives a thorough analysis of the use of cutting-edge models, such as DenseNet, Xception, and Generative Adversarial Networks (GANs), in deep fake picture identification. The paper investigates how well these models work to distinguish real images from photoshopped ones, taking into account adversarial techniques that are static or dynamically changing. We carry out comprehensive experiments on various datasets, assessing the effectiveness of the model, its capacity for generalisation, and its vulnerability to adversarial attacks. The results open the door to more robust solutions in the dynamic field of synthetic media by furthering our understanding of deep fake detection processes.

As the principal detection frameworks, Xception, a potent convolutional neural network architecture, and DenseNet, renowned for its dense connection and feature reuse, are used. A vast collection of deepfake photos, including ones with altered faces, substitute objects, and artificially created environments, is used to train these models. The training data is updated to reflect the most recent developments in adversarial picture generation through the use of the GAN model to replicate various deep fake contexts.

In this report we have further used four different datasets on the above mentioned three models and done a comparative study based on the results, and discussed the current research challenges for deepfake image detection using the DenseNet21, Xception and GAN models.And lastly we have provided a future Scope for deep fake detection methodds and concluded our report paper.

TABLE OF CONTENTS

1. INTRODUCTION

1.1. What is DEEP FAKE?

The term ”deepfake” is a combination of ”deep learning” and ”fake.” It describes a category of artificial media, usually audio or video, where very realistic but completely fake content is produced using deep learning and artificial intelligence algorithms. Deepfake technology has become well-known for its capacity to create and alter content that can accurately imitate the voice and appearance of actual people, frequently with the intention of misleading or spreading false information. Deepfake technology is most frequently used to edit videos so that someone appears to be talking or doing something they never actually did. This might have serious repercussions since it could be used to fabricate stories, pose as someone else, or negatively influence well-known people.

Deep neural networks, in particular generative adversarial networks (GANs), are commonly used in the development of deepfakes. Large image and audio datasets are used to train these networks so they can recognize the features of a specific person’s voice or face. After being trained, they may produce fresh content that closely mimics the intended audience.

Deepfake technology raises worries about its potential for misuse, particularly in the areas of fake news, disinformation, and privacy invasion, even while it has certain genuine uses, such as spectacular effects in cinema and entertainment. Many institutions and researchers are attempting to reduce the harmful effects of deepfakes by creating methods to identify and counteract them.

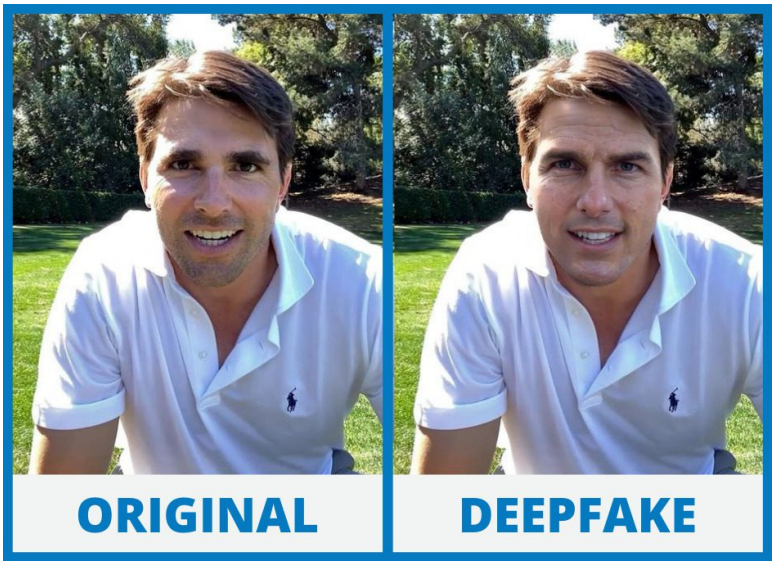

Figure 1.1: Example of deep fake Image

1.2. Types of DEEP FAKE

Deepfakes exist in many shapes and sizes, and they can be divided into distinct groups according on their intent and content. Here are a few typical forms of deep fakes:

- **Face Swaps:** In face-swapping deepfakes, one person's face is substituted for another's in a picture or video. This may give the impression that someone is saying or acting in ways that they have never done.
- **Voice Cloning:** AI algorithms are used in voice deepfakes to imitate a person's voice. These might be employed to give the impression that someone is saying words or phrases they have never said. Technology for voice cloning can also be applied to impersonation.
- **Lip Sync:** The act of lip-syncing Deepfakes entail manipulating an individual's lip movements inside a video to correspond with an other audio track. As a result, it could seem as though the other person is speaking a different language or saying something they didn't.
- **Puppet Master Deepfakes:**In these videos, a person's complete body—rather than simply their face—is altered. When someone uses a deepfake of this kind, it can appear as though they are moving and acting in ways they never have.
- **Speech-to-Text Deepfakes:** Speech-to-text Deepfakes use AI to produce lifelike speech from textual text. With this, audio material that imitates the voice of a certain individual can be produced.
- **2D and 3D Deepfakes:**Deepfakes can be produced in two dimensions or three dimensions. While 3D deepfakes can alter the depth and perspective of a subject in a video, 2D deepfakes only function with flat photos and movies.
- **Impersonation:** Some deepfakes are made for satire or impersonation, in which people pose as famous people, public personalities, or historical figures, usually for humorous or amusing effects.
- **Disinformation and Content Manipulation:** Deepfakes can be used to disseminate false or misleading narratives by manipulating content. They could be employed for malevolent, social, or political ends.
- **Entertainment:** The film and gaming industries, for example, use deepfake technology to create lifelike characters and special effects for legitimate entertainment purposes.

1.3. CONSEQUENCES OF DEEP FAKE

Deepfakes can have a range of consequences, both positive and negative, depending on their use and impact. Here are some of the consequences associated with deepfake technology:

1.3.1. Negative Consequences:

- **Misinformation and Disinformation:** False and misleading information can be disseminated by using deepfakes to produce convincingly fake audio or video recordings. This can erode public confidence in the media and make it harder to tell what is authentic from phony.
- **Privacy Invasion:** Deepfakes are a useful tool for manipulating sensitive or private content. Examples of this include making fictitious revenge porn or jeopardizing people's privacy by placing them in situations they never would have been in.
- **Identity theft and impersonation:** Malevolent actors can pose as real people utilizing deepfake technology, which could damage their reputation or interpersonal connections. Additionally, this may result in fraud and identity theft.
- **Political Manipulation:** Deepfakes are a tool for producing fake content that purports to depict political personalities acting or speaking in ways they never would have. Election results and public perception may be affected by this.
- **Social Engineering assaults:** People may be tricked into divulging private information or acting in ways they otherwise wouldn't by using deepfakes in social engineering assaults.
- **Undermining Trust:** The prevalence of deepfakes may contribute to a general decline in people's confidence in the veracity of information and media, making it increasingly difficult to tell what is fake from what is genuine.
- **Legal and Ethical Issues:** The usage of deepfakes may give rise to legal problems, such as privacy, copyright, and defamation cases. It also brings up moral dilemmas regarding acceptable technological use and its bounds.

1.3.2. Positive Consequences

- **Entertainment and Creative Expression:** The legitimate applications of deepfake technology include producing lifelike digital characters, producing special effects for motion pictures, and fostering artistic expression.

- **Education and Training:** Deepfake technology can be applied to language learning, training simulations, and historical figure simulations for educational purposes.
- **Accessibility:** Text-to-speech deepfakes and voice cloning can be used to produce content that is more accessible to individuals with disabilities. For example, this can be done by giving persons who have speech problems alternate voices.
- **Research and Development:** Deepfake technology is improving computer vision and speech synthesis through applications in artificial intelligence and machine learning.

As the technology surrounding deepfakes continues to advance, there is ongoing debate and research into how to mitigate their negative consequences and ensure responsible use. Efforts are being made to develop detection tools, regulations, and ethical guidelines to address the potential harm caused by deepfakes while preserving their legitimate applications.

1.4. DEEP FAKE Detection

Detecting deepfakes is a challenging task because the technology used to create them continually evolves and becomes more sophisticated. Various methods and techniques have been developed to detect deepfakes, but no single method is foolproof. Detection often requires a combination of approaches[1]. Here are some common methods for deepfake detection:

- **Face and Lip-Sync Analysis:** Inconsistencies in facial expressions, blinks, or lip movements can be indicative of deepfakes. Deep learning models can be used to analyze these aspects.
- **Blink Analysis:** Deepfake videos sometimes lack natural blinks or may exhibit unusual blinking patterns. Blink analysis algorithms can detect such anomalies.
- **Audio Analysis:** Voice analysis can help identify inconsistencies in speech patterns, pitch, or accent that may suggest a deepfake.
- **Metadata Examination:** Metadata in image or video files can reveal inconsistencies, such as unusual editing software used or discrepancies in creation dates.
- **Reverse Image and Video Search:** You can use reverse image or video search engines to see if the same content has been used elsewhere on the internet, indicating possible manipulation.

- **Forensic Analysis:** Experts in digital forensics can conduct a thorough examination of video and audio files for inconsistencies in compression, encoding, or other artifacts that may be indicative of manipulation.
- **Deep Learning Models:** Deep learning models[2] specifically trained to detect deepfakes are becoming more sophisticated. These models analyze patterns and anomalies in visual and audio content to identify fakes.
- **Blockchain Technology:** Some platforms and services are exploring the use of blockchain technology to verify the authenticity of media content, creating a tamper-proof record of the content's origin.
- **Human Expertise:** Experienced human analysts can often identify deepfakes by carefully reviewing content for visual and audio anomalies. However, this method is time-consuming and may not be scalable for large volumes of content.
- **Multi-Modal Analysis:** Combining multiple detection techniques, including face analysis, audio analysis, and other factors, can improve the accuracy of detection.

It's important to note that while these methods can help detect deepfakes, the arms race between creators of deepfakes and those working to detect them continues. Detection methods are not foolproof, and new deepfake techniques may be able to evade current detection techniques.As deepfake detection technology evolves, it is crucial for organizations, platforms, and governments to invest in research and development to stay ahead of the technology curve and implement effective measures to combat the spread of deceptive and malicious deepfakes.

2. MODEL ORIENTED DETECTION METHODS

2.1. CNN (Convolutional Neural Network)

Convolutional Neural Network is what CNN stands for. It's a particular kind of artificial neural network made to handle and examine visual input, such pictures and movies. In a variety of computer vision applications, such as picture categorization, object detection, facial recognition, and more, CNNs have demonstrated remarkable effectiveness. The human visual system served as inspiration for them, and they are distinguished by their capacity to automatically identify and extract hierarchical characteristics from incoming data.[3]

2.1.1. Key components and concepts of CNN

- **Convolutional Layers:** These are the fundamental components of CNNs. Convolutional layers process the incoming data by applying a series of filters, commonly referred to as kernels. These filters "convolve" or slide through the input, multiplying each element individually and combining the output. The goal of this procedure is to extract regional patterns and characteristics from the data. The convolution operation is defined as follows:

$$z[i,j] = (f * x)[i,j] = sum(m)sum(n)f[m,n] * x[i-m, j-n] \tag{2.1}$$

  where f is the filter, x is the input data, z is the output feature map, and m and n are the filter indices.

- **Pooling Layer:** This layer reduces the spatial dimensions of the input data by performing a downsampling operation. The most common type of pooling operation is max pooling, which takes the maximum value in each pooling region.

  The max pooling operation is defined as follows:

$$y[i,j] = max[x[m,n] : i*s <= m < (i+1)*s, j*s <= n < (j+1)*s] \tag{2.2}$$

  where x is the input data, y is the output of the pooling operation, and s is the stride.

- **Fully Connected Layer:** This layer is a commonly and widely used neural network layer that takes the flattened output of the previous layers as input and produces a final output. This layer is used to classify the input data into different classes.

  Teach layer's output is sent via an activation function, such the Rectified Linear Unit (ReLU), which can be given as follows:

$$f(x) = max(0,x) \tag{2.3}$$

- **Dropout:** A Dropout layer is another prominent feature of CNNs. The Dropout layer acts as a mask, eliminating some neurons' contributions to the subsequent layer while maintaining the functionality of all other neurons.

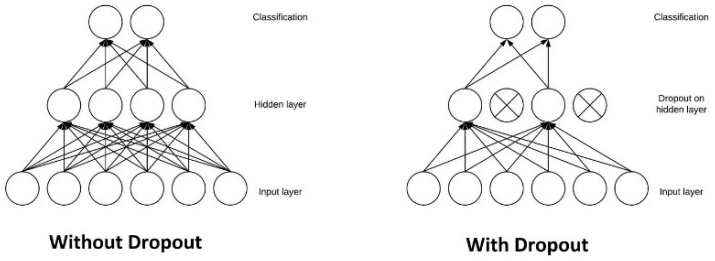

Figure 2.1: Dropout [1]

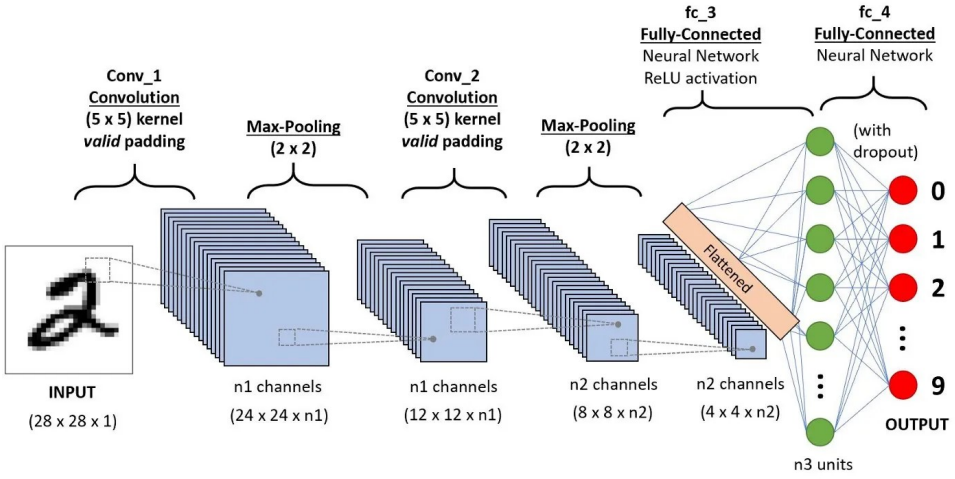

Figure 2.2: CNN [1]

The diagram shows a typical convolution neural network (CNN) architecture. The input image is processed by a convolution layer, which applies a set of filters to generate a feature map. The output of the convolution layer is then passed through a Rectified Linear Unit ReLU activation function to introduce non-linearity. The subsequent pooling layer applies max pooling to the output of the convolution layer, which reduces the spatial dimensions of the feature map. The final stage involves passing the output of the pooling layer through a fully connected layer, which is responsible for generating the network's final output.

2.2. DenseNet121

Information moves progressively from one convolutional layer to the next in a feed-forward convolutional neural network (CNN), where each layer receives and generates an output feature map. Since every layer in these networks is connected to every other layer, the total number of direct connections is equal to the number of layers (L).

But when CNNs go deeper and have more layers, an issue known as the ”vanishing gradient” appears. This problem arises because some information may be lost or diminished as it passes through other levels, impairing the network’s ability to teach new members.

DenseNets simplify the interlayer connectivity and reconfigure the traditional CNN design to solve this issue. Every layer in a DenseNet design is directly connected to every other layer, creating a structure that is dense and interconnected. In a network consisting of ’L’ layers, this leads to L(L+1)/2 direct connections, guaranteeing effective information transfer across layers and assisting in the mitigation of the vanishing gradient issue.

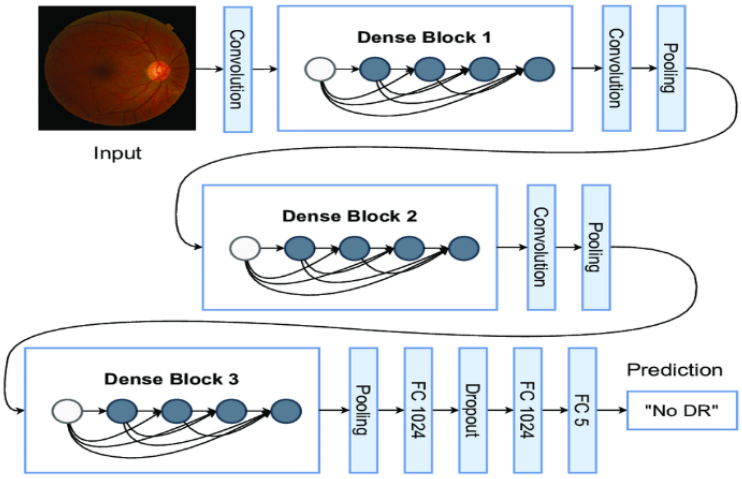

Figure 2.3: Dropout [2]

2.2.1. DenseNet Architecture and Components

- **Connectivity :** In each layer, instead of summing the feature maps from the previous layers, they are concatenated and utilized as inputs. This characteristic reduces the number of parameters required in DenseNets compared to an equivalent traditional CNN, promoting feature reuse by discarding redundant feature maps. Consequently, the lth layer takes as input the feature-maps from all preceding layers, denoted as x0, x1, ..., xl-1, and can be represented as:

$$x_\ell = H_\ell([x_0, x_1, \ldots, x_{\ell-1}]) \tag{2.4}$$

In this equation, [x0, x1, ..., xl-1] represents the concatenation of feature-maps, which encompasses the outputs generated in all layers preceding layer l (0, 1, ..., l-1). To simplify implementation, the multiple inputs to Hl are concatenated into a single tensor.

- **Dense Blocks:** The usage of dense blocks is the primary innovation of DenseNets. A dense block is a series of convolutional layers with feed-forward connections between each layer and every subsequent layer. In contrast, layers in conventional CNNs are usually linked in a sequential manner.

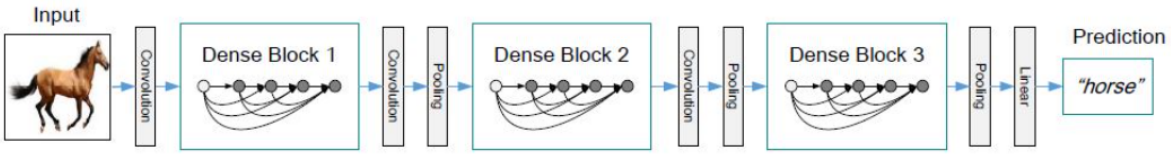

Figure 2.4: XceptionNet orignal Architecture [3]

In order to solve the vanishing gradient problem, the dense connectivity makes sure that gradients may move across the network more readily during training.

- **Growth Rate:**

  One can consider the features as a collective state of the network. The feature map size increases with each pass through a dense layer, where each layer contributes 'K' new features to the existing global state. This parameter 'K' is often referred to as the growth rate of the network, which controls the amount of information added in each network layer. If each function Hl generates k feature maps, then the lth layer will have

$$k_l = k_0 + k * (l - 1) \tag{2.5}$$

  input feature-maps, where k0 represents the initial number of channels in the input layer. Unlike conventional network architectures, DenseNets allow for the creation of relatively narrow layers.

- **Bottleneck layers:** Every layer in a dense block in DenseNet-121 and its variants is made up of a bottleneck layer. A 1x1 convolution and a 3x3 convolution make up a bottleneck layer. This architecture preserves expressiveness while lowering the amount of parameters and processing needed.

2.3. XceptionNet

XceptionNet is a deep learning model that was introduced in a research paper titled ”Xception: Deep Learning with Depthwise Separable Convolutions” by François Chollet, the creator of the Keras deep learning framework. XceptionNet is an extension of the Inception architecture and is designed to improve the efficiency and accuracy of deep neural networks. It achieves this by utilizing depthwise separable convolutions, which significantly reduce the number of parameters in the model while retaining expressive power.

In the context of deepfake detection, XceptionNet can be used as a feature extractor to analyze the content of images and identify patterns associated with manipulated or fake images.

XceptionNet typically consists of 36 convolutional layers, divided into 14 residual modules. It starts with a simple convolutional layer followed by multiple residual modules, which are composed of depthwise separable convolutions, batch normalization, and ReLU activation functions.

Orignal architecture of XceptionNet:

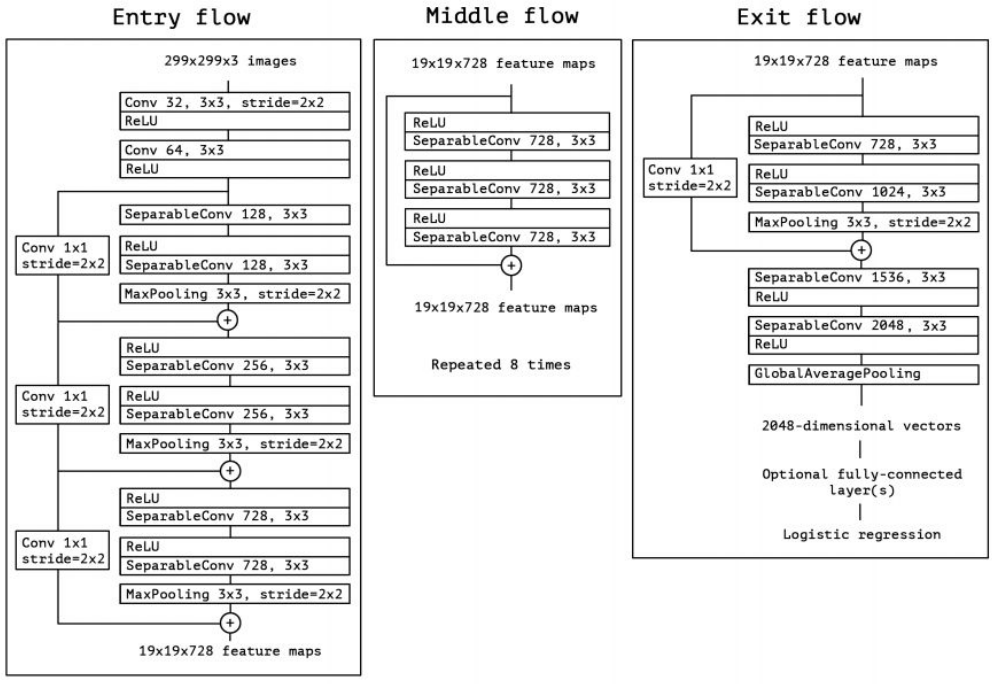

Figure 2.5: Xception orignal Architecture [2]

2.3.1. XceptionNet Architecture:

- Depthwise Separable Convolution: The key innovation in XceptionNet is the depthwise separable convolution operation, which is a combination of depthwise convolution and

pointwise convolution. Let's break down these operations mathematically:

- Depthwise Convolution: Depthwise convolution applies a separate convolution operation for each input channel. For an input tensor X with dimensions (h, w, din) where h and w are the height and width of the input, and din is the number of input channels, the depthwise convolution operation can be defined as follows:

$$\text{DepthwiseConv}(X,K)_{i,j,k} = \sum_{m,n} X_{i+m,j+n,k} \times K_{m,n,k} \tag{2.6}$$

  Here,K is the depthwise convolution kernel with dimensions (k,k,din), and × represents element-wise multiplication.

- Pointwise Convolution: Pointwise convolution applies 1×1 convolutions to the output of the depthwise convolution to combine information from different channels. For an input tensor with dimensions (h,w,dout), where dout is the number of output channels, the pointwise convolution operation can be defined as follows:

$$\text{PointwiseConv}(X,K)i,j,m = \sum_{k} Xi,j,k \times K_{1,1,k,m} \tag{2.7}$$

  Here, K is the pointwise convolution kernel with dimensions (1,1,din,dout).

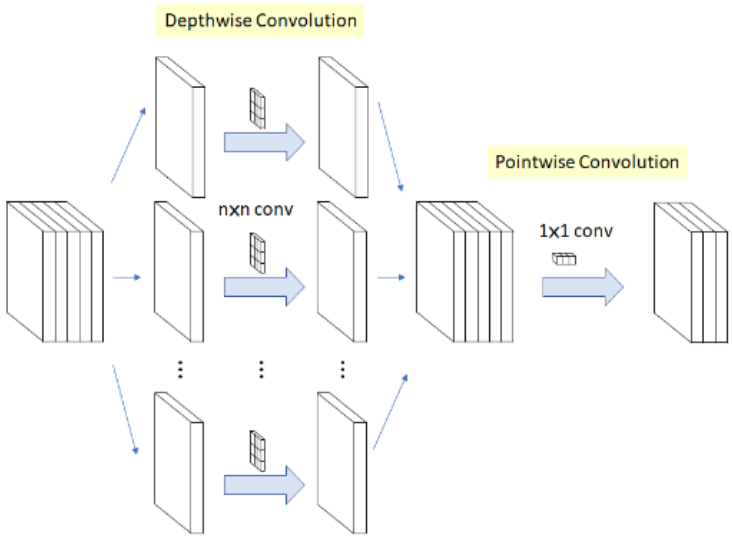

Figure 2.6: XceptionNet [1]

- XceptionNet Block: XceptionNet consists of several Xception blocks, which are composed of depthwise separable convolutions, batch normalization, and skip connections. The overall architecture of an Xception block is as follows:

  - Depthwise Separable Convolution: Apply depthwise separable convolution with

batch normalization and ReLU activation.

$$\text{ConvBlock}(X) = \text{ReLU}\left(\text{BatchNorm}\left(\text{DepthwiseConv}\left(X, K_{\text{depthwise}}\right)\right)\right) \quad (2.8)$$

– Residual Connection: Add the input tensor X to the output of the depthwise separable convolution.

$$\text{Output} = \text{ConvBlock}(X) + X \quad (2.9)$$

2.4. Generative Adversarial Networks (GANs)

2.4.1. What are GANs?

In 2014, Ian Goodfellow and his team introduced a groundbreaking concept known as Generative Adversarial Networks (GANs), revolutionizing the field of generative modeling. GANs are designed to create a new dataset that closely mimics [4] the characteristics of the original training data. They comprise two primary components, often referred to as neural networks: the Generator[5] and the Discriminator, which engage in a competitive process to comprehend and replicate the patterns within a dataset.

To break down the acronym GAN:

- Generative: GANs specialize in learning a generative model that essentially explains how data is generated probabilistically. In simpler terms, they understand the visual process of generating data that closely resembles the training dataset.
- Adversarial: GANs train in an adversarial manner, involving a competitive interplay where two models challenge each other.
- Networks: GANs harness deep neural networks as the foundation for their training process.

When given random input, which is frequently noise, the Generator network creates samples—such as text, audio, or images—that closely resemble the training data it was given. Producing samples that are almost identical to real data is the Generator's main objective.

The Discriminator network, on the other hand, is responsible for differentiating between created and actual samples. Real data from the training set and produced data from the generator are used to teach it. Accurately classifying created data as counterfeit and authentic data as actual is the discriminator's goal.

The Generator and the Discriminator engage in an aggressive game during the training phase. The Discriminator wants to improve its capacity to distinguish between produced and real data, while the Generator wants to create samples that can fool the Discriminator. Because of this adversarial training dynamic, both networks gradually get better at what they do.

As GANs are trained, the Generator gets better at producing realistic samples, while the Discriminator gets better at distinguishing real data from produced data. Ideally, this continuous process results in a convergence where the Generator becomes very good at generating high-quality samples that are difficult for the Discriminator to differentiate from real data.

GAN has demonstrated excellent performance in many disciplines such as text processing, video processing and image processing. They are used for many purposes, including deepfakes, creating realistic images, enhancing negative images, and more. Thanks to GANs, the field of artificial intelligence has been greatly improved, opening up new opportunities for the practical use of artificial intelligence.

2.4.2. Components of Generative Adversarial Networks (GANs)

Generator and discriminator are two elements of GAN. Unlike counterfeiters, manufacturers create fake models that use genuine data to trick people into believing the fake data is real. On the other hand, the observer as a researcher examines the patterns produced by the generator to identify inconsistencies and classify them as true or false. The generator and parser constantly compete until the generator reaches a level of intelligence where it can generate false information that the parser can constantly use.To gain an intuitive understanding of how GANs are trained, let's explore the roles of their two key components:

- Discriminator – Imagine it as a supervisor. The Discriminator acts as a straightforward classifier, responsible for determining whether the data presented is genuine or fake. It goes through training using authentic data and provides feedback to the Generator.
- Generator – The Generator uses unsupervised learning as opposed to the Discriminator. On the basis of the original (actual) data, it is in charge of creating synthetic data. With its hidden layers, activation functions, and loss function, this component performs the operations of a neural network. Its main goal is to generate fake data based on the input it gets, with the ultimate goal being to continuously fool the Discriminator. The training procedure stops when the Generator consistently fools the Discriminator, signaling that a well-trained GAN model has been created.

In this case, the generative model is trained to generate additional models to increase the

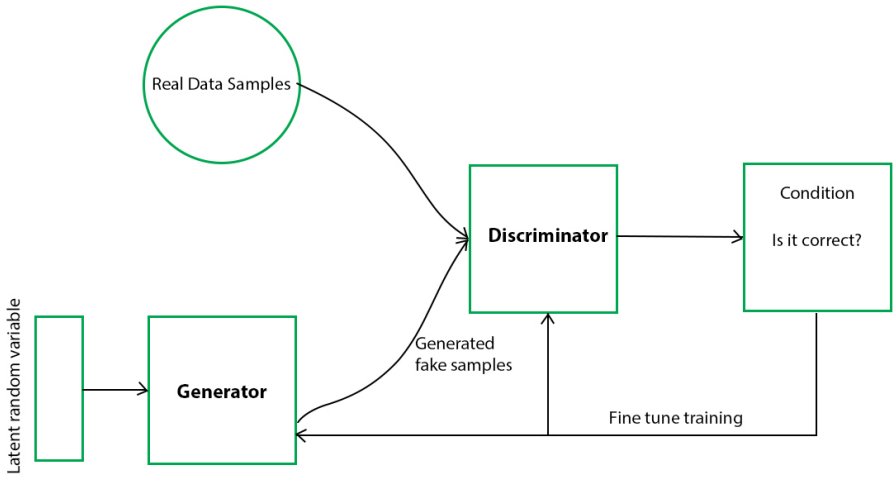

Figure 2.7: GAN [3]

probability of false discrimination and thus maximize discrimination errors). Generative modeling aims to examine the classification of data. Instead, the discriminator follows a model that evaluates the probability that the samples taken are drawn from the training set rather than generated, and attempts to classify them correctly while reducing the accuracy of the GAN.

The Discriminator in the GAN architecture aims to minimize its reward function V(D, G), while the Generator tries to increase the Discriminator's loss. This can be thought of as a minimax game.

If you want to learn more about real-life GANs, including how to design, train, and use two neural networks for prediction, check out the GAN architecture below for an easier understanding of eight.

It is important to note that both GAN frameworks are neural networks. The output of the generator is very close to the input of the estimating discriminator. In response, the engine uses feedback to increase performance and make weight adjustments. In contrast, the discriminator acts as a feedforward neural network.

2.4.3. Training and Prediction of Generative Adversarial Networks (GANs)

Let's delve into the training process of GAN, breaking down the training of the Generator and Discriminator separately:

- Problem Definition: Define the problem you want to solve using a GAN, like generating images, audio, or text.

- Choose GAN Architecture: Select a GAN architecture that suits your problem.
- Training the Discriminator on Real Data: The Discriminator starts by classifying real data, updating its weights based on misclassification.
- Training the Generator: The Generator, using random noise as input, generates fake data. The Discriminator remains inactive during this phase. The Generator's goal is to transform noise into meaningful data.
- Training the Discriminator on Fake Data: The Generator's output is assessed by the Discriminator to distinguish between real and fake data.
- Training the Generator with Discriminator Feedback: The Generator receives feedback from the Discriminator and adjusts to improve its performance.

This cycle iterates until the Generator becomes skilled at consistently fooling the Discriminator.

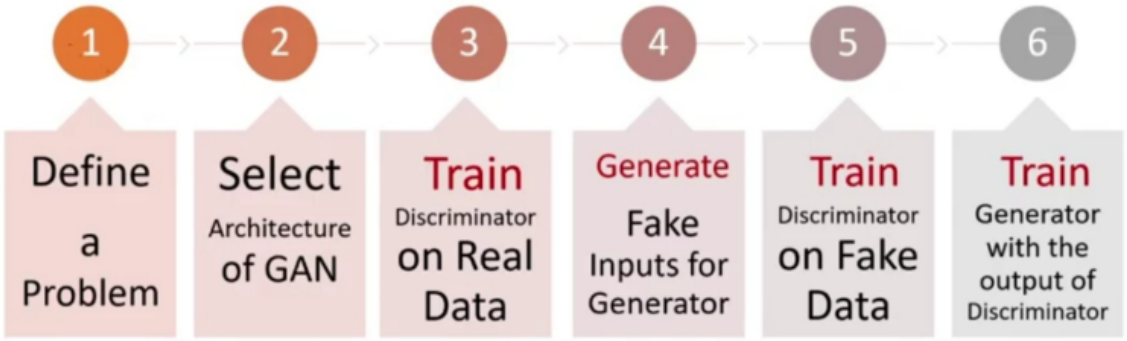

Figure 2.8: CYCLE OF GAN [1]

2.4.4. Generative Adversarial Networks (GANs) Loss Function

Certainly, let's explain the loss function utilized by GANs and how it's involved in the minimization and maximization process during their iterative training. The Generator aims to minimize this loss function, while the Discriminator attempts to maximize it. It's analogous to a minimax game for those familiar with such concepts.

Loss Function for Generative Adversarial Networks (GANs):

$$\min_G \max_D V(D,G)V(D,G) = E_{x\sim p_{\text{data}}(x)}[\log D(x)] + E_{z\sim p_z(z)}[\log(1-D(G(z)))] \tag{2.10}$$

- The discriminator's estimate of the likelihood that the actual data (symbol x) is true is represented by D(x).
- Ex stands for the expected value computed across the whole set of real data instances.
- The output that the generator produces with input noise of z is denoted by G(z).
- The Discriminator's assessment of the likelihood that a phony instance is real is represented by D(G(z)).
- The expected value computed across all random inputs provided to the Generator is represented by Ez, which is effectively a summary of the expected value over all synthetic instances produced by G(z).

The primary objective of the loss function in Generative Adversarial Networks (GANs) is dual-fold: minimizing the Discriminator's efficacy in discerning between real and generated data (maxD) while concurrently maximizing the Generator's proficiency in deceiving the Discriminator (min G). This antagonistic interplay engenders a dynamic equilibrium wherein the Generator progressively crafts data closely mirroring real data distributions, while the Discriminator enhances its discernment between genuine and synthetic instances. This adversarial process drives GANs towards convergence by iteratively refining both networks' performance through a minimax optimization strategy.

Through this iterative refinement, GANs attain a state where the Generator produces synthetic data that intricately emulates the characteristics of authentic data. This convergence fosters the creation of remarkably realistic synthetic samples. The adversarial training mechanism propels the Generator to learn nuanced patterns and structures inherent in real data, thereby enabling the generation of synthetic data indistinguishable from its authentic counterparts.

3. LITERATURE REVIEW

Table 3.1: Continued Literature Survey for Deep Learning Techniques (Part 1).

Authors(year)	Aim	Issues	Methods	Datasets	Findings	Shortcomings/future work
Mika Westerlund[1]	Explore Deepfake Tech	Investigate Ethical, Societal, and Security Aspects	Literature Review, Synthesis of Existing Studies	N/A	Deepfake technology has significant implications in various domains, including politics, entertainment, and cybersecurity.	Limited original research; Dependency on existing studies
Maryam Taeb, Hongmei Chi[2]	Evaluate Deepfake Detection	Assess Effectiveness of Detection Techniques	Deep Learning Methods	Diverse Deepfake Datasets	Comparative analysis of deepfake detection techniques based on deep learning approaches.	Limited focus on non-deep learning methods for comparison.
Siwei Lyu[6]	Examine Deepfake Detection Challenges	Identify Key Challenges in Detection	Literature Review, Expert Opinion	N/A	Identification of major challenges in deepfake detection, including the rapid evolution of deepfake generation techniques and the need for large-scale labeled datasets.	Limited focus on specific deepfake detection methods.

Table 3.2: Continued Literature Survey for Deep Learning Techniques (Part 2).

Authors(year)	Aim	Issues	Methods	Datasets	Findings	Shortcomings/future work
Usha Kosarkar, Gopal Sarkarkar, Shilpa Gedam[3]	Detect and Classify Deepfakes Images	Address Deepfake Misuse Custom	Convolutional Neural Network (CNN) Model	Diverse Image Datasets	Development of a customized CNN model for deepfake image detection and classification.	Lack of comparison with other state-of-the-art deepfake detection methods.
Fatma Ben Aissa, Monia Hamdi, Mourad Zaied, Mahmoud Mejdoub[4]	Provide an Overview of GAN-DeepFakes Detection	Examine Challenges in Detection Proposal, Improvement, Evaluation Methods	GAN-Generated DeepFakes	Comprehensive overview of GAN-based DeepFakes detection techniques, including proposals, enhancements, and evaluations.	Lack of a specific author name.	Limited discussion on real-world deployment and practicality of detection methods.
Taeb, M., and Chi, H., 2022[7]	Compare deepfake detection techniques using deep learning.	Challenges in selecting optimal detection techniques.	Comparative analysis of various deep learning methods.	Unspecified.	Identification of strengths and weaknesses in different deepfake detection techniques.	Need for standardized evaluation metrics, exploration of ensemble methods, addressing adversarial attacks.

Table 3.3: Continued Literature Survey for Deep Learning Techniques (Part 3).

Authors(year)	Aim	Issues	Methods	Datasets	Findings	Shortcomings/future work
Kumar, M., and Sharma, H.K., 2023[8]	Develop a GAN-based model for deepfake detection in social media.	Deepfake proliferation poses a threat to authenticity in social media.	GAN-based model.	Unspecified	Successful deepfake detection in social media.	Lack of specificity regarding datasets, potential bias in model training, scalability concerns.
Parayil, A.M., Ameen Masood, V., Muhammed Ajas, P., Tharun, R., and Usha, K.[9]	Implement Xception and LSTM for deepfake detection.	Challenges in accurate deepfake identification.	Xception and LSTM.	Unspecified.	Improved deepfake detection accuracy with the proposed model.	Limited explanation on dataset characteristics, potential challenges in model generalization, scalability concerns.
Raza, A., Munir, K., and Almutairi, M., 2022[10]	Introduce a novel deep learning approach for deepfake image detection.	Challenges in current deepfake detection techniques.	Novel deep learning approach.	Unspecified.	Improved accuracy and efficiency in deepfake image detection.	Lack of detailed information on datasets used, potential adaptability challenges in diverse scenarios.

Table 3.4: Continued Literature Survey for Deep Learning Techniques (Part 4).

Authors(year)	Aim	Issues	Methods	Datasets	Findings	Shortcomings/future work
Seow, J.W., Lim, M.K., Phan, R.C., and Liu, J.K., 2022[11]	Provide a comprehensive overview of Deepfake technology, covering generation, detection, datasets, and opportunities.	Ethical concerns and technological challenges.	Literature review, analysis of existing methods.	Various datasets for generation and detection.	Summarized existing deepfake technologies, challenges, and opportunities.	Need for continuous updates as technology evolves, potential biases in datasets, exploration of emerging deepfake trends.
Patel, Y., Tanwar, S., Bhattacharya, P., Gupta, R., Alsuwian, T., Davidson, I.E., and Mazibuko, T.F., 2023[12]	Enhance CNN architecture for improved deepfake image detection.	Challenges in accurate deepfake image detection.	Improved Dense CNN architecture.	Unspecified	Enhanced accuracy in deepfake image detection.	Potential limitations in diverse dataset representation, scalability considerations, real-time application challenges.
Kammoun, A., Slama, R., Tabia, H., Ouni, T., and Abid, M., 2022	Survey the use of Generative Adversarial Networks (GANs) in face generation[13].	Challenges and trends in GAN-based face generation.	Review of existing GAN-based methods for face generation.	Various face datasets used in GAN research.	Summarize and categorize GAN-based approaches for face generation.	Lack of standardized evaluation metrics, potential biases in datasets, exploration of GANs in specific face generation applications.

4. DATASETS USED

4.1. Flickr Image dataset:

The Flickr dataset is a widely recognized dataset in the field of computer vision and natural language processing, specifically designed for tasks related to image description and language understanding dataset contains around 150,000 images collected from the photo-sharing platform Flickr, each accompanied by five descriptive sentences. These sentences serve as textual descriptions of the images.

4.2. 190K-Faces:

190,000 authentic and fake photos can be found in this collection. The StyleGAN network was trained on around 29,000 images of 69 distinct models in this library, yielding face images with a flat background.

4.3. iFakeFaceDB Dataset:

A face image dataset called iFakeFaceDB is used to research the detection of synthetic face alteration. It contains approximately 20,000 synthetic face images produced using the Style-GAN model and altered using the GANprintR technique, along with genuine photographs. Every image was shrunk to 256 x 256 and aligned.

4.4. ForgeryNet Dataset:

The ForgerNet dataset includes 221,247 videos and 2.9 million photos, along with seven image-level techniques and eight video-level approaches that are sourced from throughout the globe. Every picture and video has a label attached to it.

5. RESULTS AND DISCUSSIONS

5.1. Results for the Datasets

Models	Datasets							
	Flickr Image dataset				190K-Faces			
	Acc	Prec	R	F1	Acc	Prec	R	F1
CNN	95.1	95.1	95.0	95.0	81.3	74.9	93.8	83.3
DenseNet121	91.6	99.5	83.6	90.8	91.4	94.8	87.5	91.0
Xception	88.2	99.9	76.5	82.2	88.0	95.6	79.4	86.8
GAN(Discriminator)	83.3	81.5	86.0	83.7	78.4	83.1	72.1	75.3

Models	Datasets							
	FakeFaceDB Dataset				ForgeryNet Dataset			
	Acc	Prec	R	F1	Acc	Prec	R	F1
CNN	54.6	55.0	65.2	64.6	58.0	56.4	91.7	69.9
DenseNet121	83.3	75.0	85.6	85.7	70.7	72.8	71.5	72.2
Xception	79.3	70.9	99.3	82.8	68.7	67.4	79.8	73.1
GAN(Discriminator)	90.9	85.3	98.7	91.5	64.2	61.7	84.4	71.3

Note: Acc: Accuracy; Prec: Precision; R-Recall: F1-F1score.

5.2. Result analysis for the Datasets

Performance metrics for various models on the Flickr Image dataset and the 190K-Faces dataset are shown in the provided table. Accuracy (Acc), precision (Prec), recall (R), and F1 score (F1) are among the measurements. Let's talk about each model and dataset's outcomes.

5.2.1. Flickr Image dataset:

- CNN: Achieves a high accuracy of 95.1%Ẇell-balanced precision, recall, and F1 score, indicating robust performance.
- DenseNet: Impressive precision (99.5%) but slightly lower recall (83.6%).Good F1 score (90.8%), suggesting a good balance between precision and recall.
- Xception: High precision (99.9%) but relatively lower recall (76.5%). The F1 score (82.2%) suggests a trade-off between precision and recall.
- GAN (Discriminator): Strong performance with an accuracy of 83.3%.Balanced precision, recall, and F1 score, indicating reliable discrimination.

5.2.2. 190K-Faces dataset:

- CNN: Lower accuracy (81.3%) compared to the Flickr dataset.The F1 score (83.3%) suggests a trade-off between precision and recall.
- DenseNet: Maintains high precision (94.8%) but experiences a decrease in recall (87.5%).Still achieves a good F1 score (91.0%).
- Xception: A slight drop in accuracy (88.0%) compared to the Flickr dataset. Maintains a good balance between precision, recall, and F1 score.
- GAN (Discriminator): Moderate accuracy (78.4%) with balanced precision, recall, and F1 score. Performs consistently across both datasets.

General Discussion

- The dataset selection has a big impact on how well the model works. Compared to the 190K-Faces dataset, the Flickr Image dataset looks to be more difficult, as seen by the lower accuracies and F1 scores.
- The performance of CNN, DenseNet, and Xception varies throughout datasets, highlighting the significance of choosing models that are suited to particular data features.
- Due to its constant high precision, DenseNet is a good fit for situations where reducing false positives is essential.

- Although Xception often achieves good precision, recollection may suffer. In situations where false positives are more worrisome than false negatives, it might be appropriate.
- Strong performance across datasets is demonstrated by GAN (Discriminator), suggesting that it can handle a variety learning picture datasets.

5.2.3. FakeFaceDB Dataset:

- CNN: Moderate accuracy (54.6%) with a relatively balanced F1 score (64.6%). Higher recall (65.2%) compared to precision, indicating a tendency to minimize false negatives.
- DenseNet: Achieves high precision (83.3%) but with a lower recall (75.0%). Balanced F1 score (85.6%) suggests an overall good performance.
- Xception: Lower accuracy (79.3%) compared to DenseNet and CNN. High recall (99.3%) but with a lower precision (70.9%), resulting in a relatively lower F1 score (82.8%).
- GAN (Discriminator): High accuracy (90.9%) with a balanced F1 score (91.5%). Strong recall (98.7%) indicates effective identification of positive cases.

5.2.4. ForgeryNet Dataset:

- CNN: Improved accuracy (58.0%) compared to FakeFaceDB. High recall (91.7%) but with lower precision, leading to a moderate F1 score (69.9%).
- DenseNet: Moderate accuracy (70.7%) and balanced precision, recall, and F1 score. Demonstrates consistency across metrics.
- Xception: Moderate accuracy (68.7%) with a balanced F1 score (73.1%). Similar to the FakeFaceDB results, maintains high recall but with a lower precision.
- GAN (Discriminator): Lower accuracy (64.2%) compared to FakeFaceDB. Balanced precision, recall, and F1 score, indicating consistent performance.

General Discussion

- The models show different performance on different datasets, highlighting the impact of dataset properties on model behavior.
- DenseNet is a dependable option for situations where precision and recall are equally critical because it continuously exhibits balanced performance on both datasets.

- Xception frequently exhibits great recall, but it may sacrifice precision; this is especially evident in the ForgeryNet dataset.
- On the FakeFaceDB dataset, GAN (Discriminator) performs well in terms of accuracy and F1 score, but on the ForgeryNet dataset, accuracy decreases.
- The model selection should take into account the application's particular requirements as well as the trade-off between reducing false positives and false negatives.
- It can take more testing and tweaking to get the best model performance across various datasets and use scenarios.

A thorough examination of deep learning models on various datasets provides important information about the dynamics of their performance. Both sections demonstrate how sensitive the model's efficacy is to the features of the dataset; the Flickr Image dataset presents unique difficulties, as evidenced by the model's lower accuracy and F1 scores. Interestingly, DenseNet shows up as a reliable performer, balancing recall and precision. Xception, on the other hand, introduces a noticeable trade-off with precision across datasets, favoring high recall over accuracy.

The GAN (Discriminator) model's performance, which is reliable in some situations, emphasizes the importance of carefully weighing the difficulties unique to each dataset. The model's inconsistent accuracy between the ForgeryNet and FakeFaceDB datasets emphasizes the value of a careful assessment in a range of scenarios.

Precision-recall trade-offs are a common theme, which highlights how important it is to match model selections to particular application needs. Because DenseNet consistently maintains a balanced trade-off, it is the recommended option when both erroneous positives and false negatives have significant implications.

In order to improve overall resilience and capitalize on the characteristics of many models, future research avenues might investigate ensemble approaches. Furthermore, examining a model's interpretability can help ensure that it is reliable and useful in real-world applications.

The study's conclusion emphasizes the complex interactions that exist between datasets, performance indicators, and deep learning models. A sophisticated comprehension of the properties of the dataset and the intended precision-recall trade-off are essential for selecting the best model. Sustained research endeavors are vital in order to enhance and optimize the performance of models for a variety of datasets and applications.

6. CURRENT RESEARCH CHALLENGES

Promising techniques for identifying and combating the proliferation of deepfakes include deep learning approaches such as DenseNet21, XceptionNet, and Generative Adversarial Networks (GANs) nevertheless, there are still a number of issues with their implementation[6].These challenges are discussed below :

- Adversarial Sophistication: The capacity of deep fake generators to modify their methods in order to avoid discovery is known as adversarial sophistication. Deepfake generators, such as GANs, are widely used and are always improving to create more convincing and lifelike effects. Because of this dynamic nature, detection models, such as those built on top of DenseNet21 and XceptionNet, must continuously enhance their capacity to distinguish between real and altered content.

- Limited Generalization: These architectures, such as DenseNet21 and XceptionNet, are robust, but their usefulness may be limited to certain kinds of deep fakes or samples. Invoice or text-based manipulations could be difficult for a model that was trained on face-swapping deep fakes. The process of creating models that can recognise a broad variety of deepfake methods in a variety of settings and content kinds is necessary to achieve generalisation.

- Lack of Standardized Benchmarks: The comparison of various methods is made more difficult by the lack of established benchmarks for assessing deep fake detection models. It is essential to establish widely recognised datasets with a variety of deep fake scenarios and well-defined evaluation metrics. This makes it possible for researchers to regularly evaluate and contrast the models' performances.

- Interpretability and Explainability: Developing deep fake detection algorithms that are interpretable and explainable is crucial to fostering confidence. Due to their complexity, DenseNet21 and XceptionNet frequently function as "black boxes," making it difficult to comprehend the reasoning behind their decisions. To better understand these models' decision-making processes and interpret their results, their transparency should be improved.

- Data Imbalance and Diversity: Biassed models may result from training dataset imbalances when real samples greatly outweigh deepfake samples. Developing balanced datasets that faithfully capture real-world circumstances is essential to ensuring strong detection. In order to enhance the adaptability of the model, diversity in training data is also required, encompassing a range of content, settings, and creation methods.

- Zero-Day Attacks: The term ”zero-day attacks” describes situations in which deep fakes that have not been observed before but have just been devised are not recognised by detection methods. Things like this could happen if deepfake creation techniques progress quickly. Effectively countering growing threats requires the development of proactive solutions, such as updating models continuously and utilising anomaly detection techniques.

As a result, solving these problems calls for a multidisciplinary strategy that includes developments in model architectures, dataset curation, assessment techniques, and continuous cooperation between industry stakeholders and the academic community. Continual innovation and adaptability in deep fake detection approaches will be crucial as the deep fake creation landscape changes.

References

[1] M. Westerlund, The emergence of deepfake technology: A review, Technology innovation management review 9 (11) (2019).

[2] M. Taeb, H. Chi, Comparison of deepfake detection techniques through deep learning, Journal of Cybersecurity and Privacy 2 (1) (2022) 89–106.

[3] U. Kosarkar, G. Sarkarkar, S. Gedam, Revealing and classification of deepfakes video's images using a customize convolution neural network model, Procedia Computer Science 218 (2023) 2636–2652.

[4] F. Ben Aissa, M. Hamdi, M. Zaied, M. Mejdoub, An overview of gan-deepfakes detection: proposal, improvement, and evaluation, Multimedia Tools and Applications (2023) 1–23.

[5] X. Wang, H. Guo, S. Hu, M.-C. Chang, S. Lyu, Gan-generated faces detection: A survey and new perspectives, arXiv preprint arXiv:2202.07145 (2022).

[6] S. Lyu, Deepfake detection: Current challenges and next steps, in: 2020 IEEE international conference on multimedia & expo workshops (ICMEW), IEEE, 2020, pp. 1–6.

[7] M. Taeb, H. Chi, Comparison of deepfake detection techniques through deep learning, Journal of Cybersecurity and Privacy 2 (1) (2022) 89–106.

[8] M. Kumar, H. K. Sharma, et al., A gan-based model of deepfake detection in social media, Procedia Computer Science 218 (2023) 2153–2162.

[9] A. M. Parayil, V. Ameen Masood, P. Muhammed Ajas, R. Tharun, K. Usha, Deepfake detection using xception and lstm.

[10] A. Raza, K. Munir, M. Almutairi, A novel deep learning approach for deepfake image detection, Applied Sciences 12 (19) (2022) 9820.

[11] V. O. MAKARICHEV, V. V. LUKIN, V. S. KHARCHENKO, Image compression and protection systems based on atomic functions (2023).

[12] 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.

[13] A. Kammoun, R. Slama, H. Tabia, T. Ouni, M. Abid, Generative adversarial networks for face generation: A survey, ACM Computing Surveys 55 (5) (2022) 1–37.