CHAPTER 1: INTRODUCTION

One of the most distinctive features of human beings is their face. The term "deepfakes" is a fusion of "Fake" and "Deep Learning". This phenomenon involves substitution of individuals' faces with those of others in a deceptively authentic manner, facilitated by various algorithms grounded in deep learning technology.

Deep learning techniques combined with artificial intelligence have significantly improved the quality of our daily life. Among these, one of the most significant contributions is in the deep learning models can recognize and categorize a variety of features, including human faces, important points, medical images, and more, in the field of computer vision. Although the concepts of deep learning were first presented in the late 19th century, the practical use of these technologies dates back to the early 20th century.

The availability of a vast amount of data due to the internet explosion and improved computational capacity at a low cost are some of the primary causes of this abrupt increase. The engine of data science and artificial intelligence is data. More precise data must be learned by the model for it to function well, and as the volume of data rises, so does the model's performance.

Despite acknowledging the numerous benefits of artificial intelligence, we must also confront its negative consequences, one of which is the emergence of a highly threatening form of technological fraud known as deepfake.Deepfake technology utilises deep neural networks to create fake images or movies. This method includes superimposing the face of a target individual onto a source video, giving the impression that the target person is present in the original clip. Deepfakes are created using a variety of approaches, including both cloned and non-cloned voices. Incorporating a cloned voice improves the realism of the modified information by combining created visual effects with generated speech. Cloned voices are made by analysing a target person's voice and integrating it with source audio, producing audio content that the target person never spoke.Leveraging deep learning algorithms, the target individual's voice is seamlessly blended with the source audio, making it nearly impossible to distinguish from the real recording.

Deepfake technology has grown significantly in past years, owing to a variety of conditions. For starters, programmes and open-source code have become more easily accessible, making it easier for individuals to create visually convincing deepfakes. Furthermore, the widespread availability of films, videos, and photos has played a crucial impact. Creating high-quality deepfakes takes significant training of the deepfake models, which requires large datasets. Obtaining such datasets used to be difficult, but with the internet revolution, it is now feasible. Furthermore, the widespread availability of high-capacity computer systems has expedited the training process, accelerating the development of deepfake technology.

Deepfake represents a burgeoning field within artificial intelligence wherein the visage of one individual is superimposed onto another person's face. Deep Fakes are forms of artificially generated content within the realm of artificial intelligence. A different way of organizing them is by splitting them into two groups: puppet master and lip-sync. Lip-sync deepfakes are edited videos wherein the mouth movements can be modified to match a recorded sound. Meanwhile, puppet-master deepfakes involve videos featuring a subject individual (puppet) whose animation replicates the eye movements, facial expressions, and head gestures of another individual (master) positioned in front of a camera

While certain deep fakes may be produced through conventional Computer graphics or visual effect methods, the prevailing contemporary method for generating deep fakes involves using deep-learning models, including autoencoders and generative adversarial networks (GANs). The encoders function by extracting all features present in an image, while decoders are employed to produce the fabricated image. Deepfake techniques require a substantial quantity of images and videos for training the deep learning models, which was previously a challenging undertaking. However, in the current era, acquiring a large dataset of images from social media has become relatively easy. The widespread availability of data has consequently spurred the advancement of more sophisticated deepfake techniques.

These models have found widespread application in the domain of computer vision. These models are employed as to analyze the facial expressions and movements of an individual, enabling the synthesis of facial images of another person who replicates similar expressions and movements. In 2017, the first Deep fake material emerged, in which the faces of porn actors and well-known

celebrities were interchanged. There are various positives as well as negatives of the face swap technology or more formally deepfake technologies. Negatives of deep fake are like Corporate level fraud Fraudsters have shifted from attempting to convince employees to transfer funds using deceptive emails within organizations. Instead, they now employ phone calls, mimicking the voice of high-ranking executives such as the CFO or CEO to achieve their fraudulent objectives.

There are various privacy concerns as well because individuals' faces can be swapped with another person to make any false image or derogatory video which is very difficult to detect. Significant improvement in deep fake creation technologies can lead to decreasing trust in digital information. Whereas positives are Entertainment, Creativity, and Digital Effects, deepfake technologies can improve computer-generated images and improve realism. There was a circulating video on the internet featuring former American President Barack Obama, wherein he was depicted saying things he has never actually expressed. Additionally, deepfake technology has been utilized to manipulate footage of Joe Biden, altering images from the US 2020 election to show him sticking out his tongue. As an example, a CEO fell victim to a deepfake, resulting in a loss of $243,000. The onset of software called DeepNude induces additional risks, as it has the capability to transform individuals into non-consensual pornography. Similarly, the Chinese app Zao has gained popularity, allowing uneducated users to swap visages with the bodies of movie actors or actresses and integrate themselves into recognizable scenes from TV clips, movies, and documentaries.

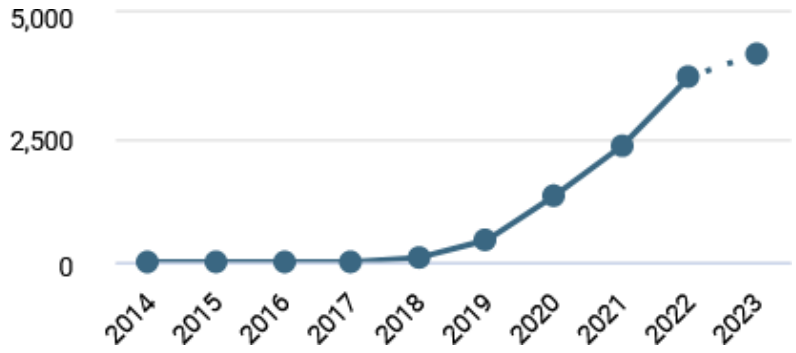

Fig. 1.1: Number of papers with deep fakes in years from 2014 to 2023.

The Data (Figure 1) obtained from https://app.dimensions.ai indicates a notable and significant rise in the number of academic papers focusing on deep fake technology in recent years,

which how deep fake creation technologies have improved. Such forms of manipulation create a significant threat to privacy and identity violations, impacting various facets of human existence.

CHAPTER 2: DEEPFAKE CREATION

Deep Fakes are made using algorithms of deep learning to replace the visage of a targeted individual in a video or picture with the visage of another individual. This technology was enhanced by developers and online communities, resulting in user-friendly programs like Fake App and FaceSwap, which are widely available online.

Goodfellow, et al. [11] proposed a novel approach for estimating generative models using an adversarial method. In this approach, we concurrently train two models i.e., generative model G which interprets the data distribution, and discriminative model D, which estimates the likelihood that a sample originates from training data rather than G. In the proposed adversarial network approach, the generative model and discriminator model compete among themselves. Through this process, both models continue to improve to an extent where the image generated by the generator model is indistinguishable from actual input data. MNIST, Toronto Face Database (TFD) and CIFAR-10 datasets were used to train the adversarial networks [11].

Coupled Generative Adversarial Networks (CoGAN) were presented by Liu and Tuzel [12], which is an extension of Generative adversarial networks or GANs customized for studying joint distributions of multi-domain images in two distinct domains. Coupled Generative Adversarial Networks (CoGAN) is comprised of two GANs - G1 and G2, both independently responsible for image synthesis in a single domain. MNIST digits, image faces, NYU dataset, and RGBD dataset were applied on CoGAN while experimenting which demonstrated its ability to produce corresponding images even without particular training for the same [12].

In the last few years, there has been an increasing trend of using generative deep neural networks for facial manipulation. To create completely non-existent faces Karras et al. [13] used a generative adversarial network called styleGAN. CycleGAN a Generative Adversarial Network based face-swapping method was introduced by Zhu et al. [14]. Additionally for replacing the face of one person in an image or video with the visage of another person tools such as DeepFakes [15] and FaceSwap [16]can be used.

2.1 Image Generation using GANs

GANs are exciting and rapidly advancing generative models that promise to generate realistic examples across an exceptional range of problems, one of the most notable is in translation tasks of image-to-image, such as turning the images of winter or summer to day and night, as well as in creating the photorealistic images of scenes, people that are AI-generated, and objects that do not exist. GANs automatically train the generative model by treating an unsupervised problem as a supervised problem and using discriminator and generator models both. GANs have the capacity for enhanced domain-specific data augmentation as well as solving generative tasks like image-to-image translation.

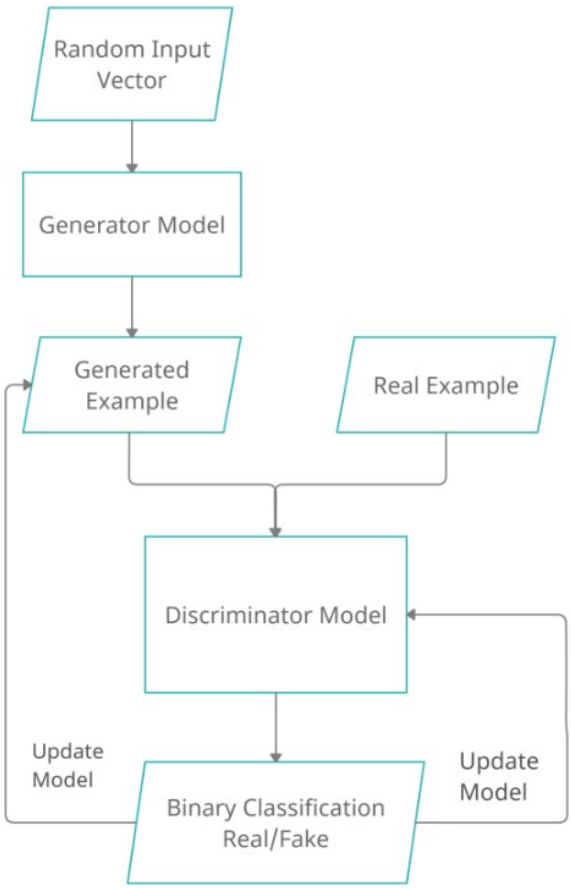

Fig. 2.1 Working of Generative Adversarial Network Model

The Architecture of GAN has two sub-models: a discriminator model for identifying whether the created examples are fake,

generated by the generator model, or real, generated by the domain, and a generator model for producing new instances. The Generator Model is used for the generation of new believable instances from the issue set, while the Discriminator Model is used for determining if instances are real (from the domain) or not (fabricated).

Generator Model of GANs - In the context of GANs, the generator model uses a fixed length random vector, generated from a Gaussian distribution, as input to create samples within the target domain. During training, points in this multidimensional vector space correspond to points in the domain, offering a condensed representation of the data distribution, this vector space is called as latent space. The generator assigns significance to points in the latent space, enabling the generation of diverse output examples by drawing new points from the latent space as input [10]. Discriminator takes image as input and attempts to classify the image as real or fake, it is similar to any other neural network in this aspect. The convolutional neural network outputs one value for each image [9].

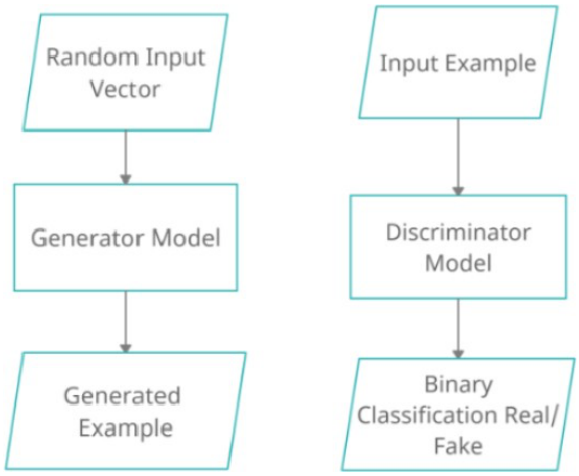

Fig. 2.2 Generator Model and Discriminator Model of GAN

Discriminator Model of GANs - The discriminator model decides whether the image generated by the generator is phony fake or real, by taking a domain sample as an input. The training data set was used to generate the real-world example. The generator model produces the generated examples. The discriminator is a common (and well-understood) classification paradigm. Because we are only considering the generator, we mainly destroy the discriminator model after training. The generator can sometimes be reused since it has developed the capacity to extract features from examples in

the issue area. Using the same or comparable input data most of the feature extraction layers can be used in a transfer learning application.

2.2 Image Generation using Autoencoders

Deepfakes, a common method for swapping faces in photos and movies, have frequently been linked to Generative Adversarial Networks (GANs). Recent advancements, however, indicate that GANs may not be the most effective method for making deep fakes. Instead, developers are turning to a more dependable option: Autoencoders, a form of deep learning algorithm.

In the realm of unsupervised machine learning Autoencoders are of the highest importance. It may used to reduce the dimensions of the input and compress the data. The key distinction between Autoencoders and Principal Component Analysis is that whereas PCA discovers the directions with the least variance along which you may project the data, Autoencoders recreate our original input from a compressed version of the input. An Autoencoder is a special type of neural network which can learn to rebuild pictures, text, and other instances from compressed copies of itself.

An Autoencoder works by encoding the input picture into increasingly smaller layers, until reaching the bottleneck layer, using an artificial neural network. It then compares the data to a certain number of variables and decodes it to its original size, resulting in the final picture.

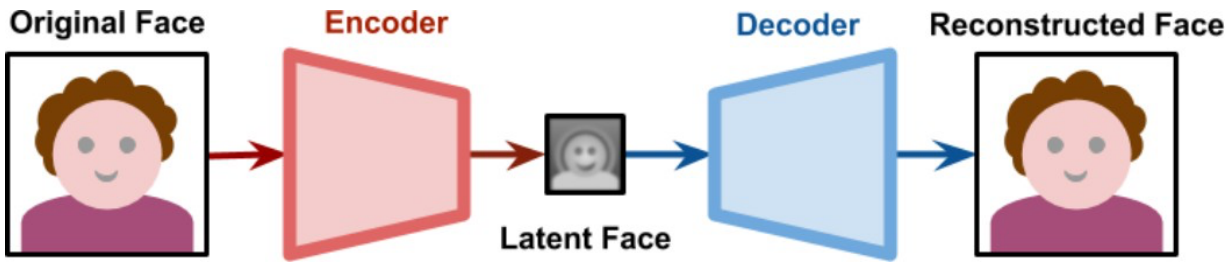

Fig. 2.3 The figure above depicts a picture being fed into an encoder. It produces a lower-dimensional representation of the same face, which is frequently referred to as the **base vector** or **latent face**.

Encoder, Code, and Decoder are the three levels of an autoencoder. The Encoder layer is responsible for compressing the input picture into a representation in the latent space. It compresses and reduces the dimension of the input image. The original picture is disfigured in the compressed image. The Code layer does the representation of the compressed input for the decoder layer. The

decoder layer restores the original dimension to the encoded picture. The decoded picture is re-generated from latent space representation, and it is a lossy reconstruction of the original image.

To improve the process, the Autoencoder is trained with varied data, encoding, decoding, computing the loss, and adjusting the model repeatedly until the desired results are obtained. One of the primary advantages of Autoencoders for creating deep fakes is that they focus solely on recreating the information provided to them. Unlike GANs, which employ imagination to fill in data gaps and often lead to unrealistic results, Autoencoders deliver more consistent and accurate face swaps. For instance, if the original image does not include sunglasses, an Autoencoder will not introduce them into the deep fake, ensuring a truer representation of the subject.

It is critical to understand that if we train two autoencoders individually, they will be of no use. During the training, every connection tries to determine the necessary features, resulting in very different latent regions. Face-swapping technology becomes possible when both latent images encode the same traits. Deepfakes overcame the problem by having both networks use the same encoder but very different specific to need-based decoders.

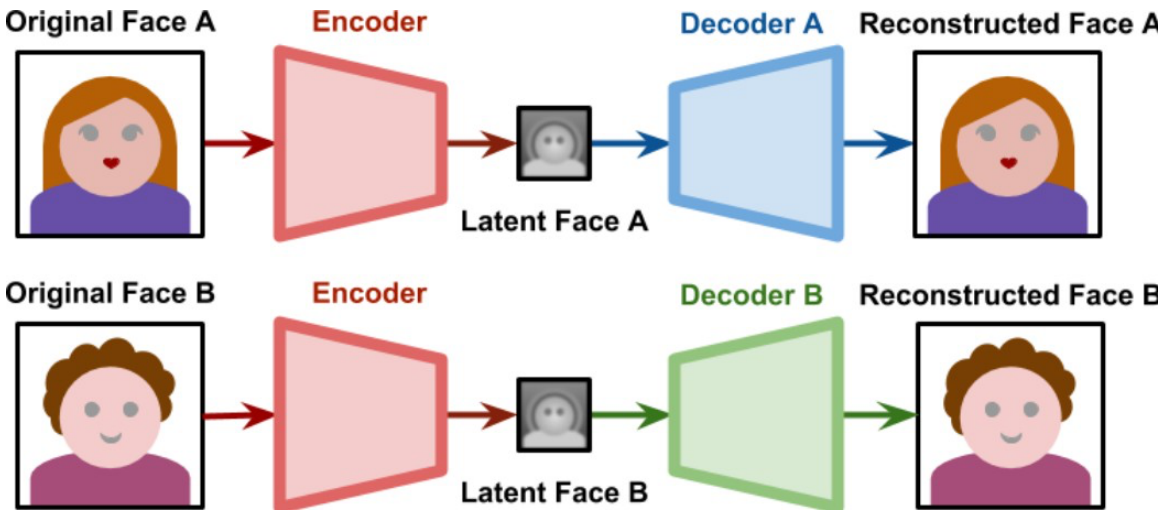

Fig. 2.4 Depicts the two encoder-decoder networks being treated separately

In the training process, two distinct networks, Decoder A and Decoder B, are trained separately. Decoder A exclusively learns from faces of type A, while Decoder B focuses solely on faces of type B. Despite this, a shared Encoder generates latent representations for all faces. Consequently, the

Encoder is compelled to identify shared features in both A and B faces. Given the inherent structural similarities in all faces, the Encoder is likely to develop a generalized understanding of the concept of a "face." After the completion of training, it becomes possible to input a latent face generated from Subject A into Decoder B. In this scenario, Decoder B endeavors to reconstruct a face resembling Subject B, leveraging the knowledge acquired from Subject A's latent representation.

The image below illustrates this process, showcasing how Decoder B applies its learned information to recreate a representation of Subject B using the input from Subject A

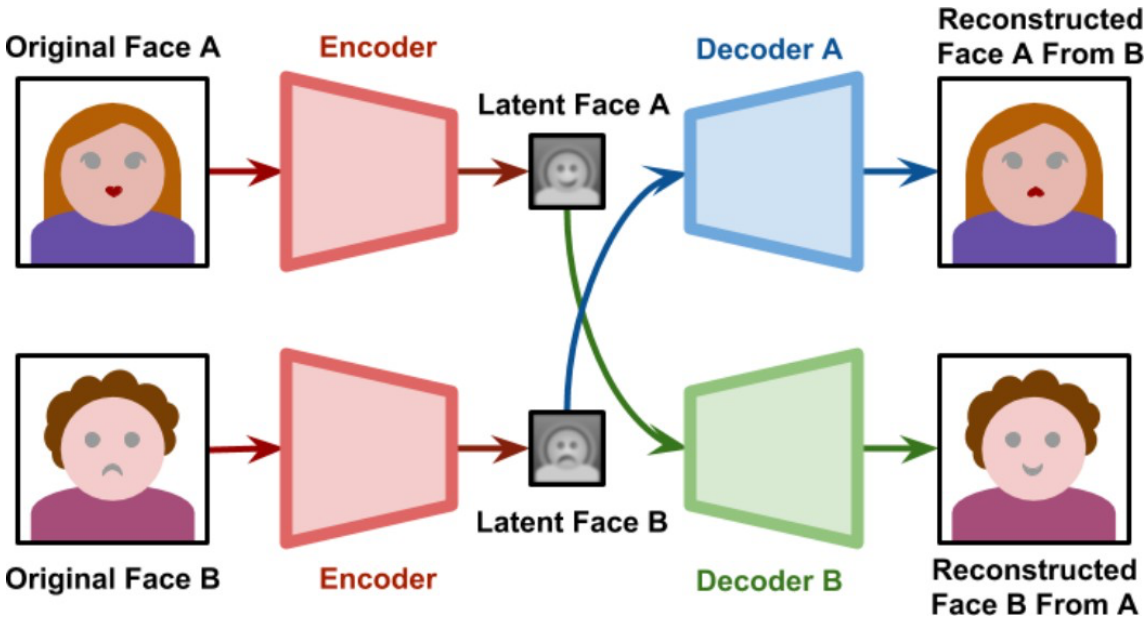

Fig. 2.5 Depicts the latent face being generated from A to B

If the network has effectively generalized the essential components of a face, the latent space will contain the facial expressions and orientations. Consequently, when generating a face for another subject B based on the latent representation of Subject A, the outcome would entail producing a face with identical expression and position as observed in Subject A

CHAPTER 3: DEEPFAKE DETECTION

As technology has become more widely available, many deepfake videos have proliferated on social media. Deepfake is the term for digital media manipulation, such as when someone else's visage appears in lieu of the original person in a picture or video. Deepfake is, in reality, one of the more significant problems facing contemporary society. Popular Hollywood celebrities' faces have regularly been swiped using Deepfake over pornographic images and video content. Deepfake was also utilized to generate rumors and false information for politicians.

A spoof video including statements that Barack Obama never said was made in 2018. Additionally, deepfakes have been used to rig Joe Biden's tongue-out videos during the 2020 United States election. These damaging applications of deepfakes have the potential to propagate false information, particularly on social media, and to have a significant negative influence on our society. Deepfake photos and videos are now widely shared on social media, according to recent studies. As a result, it is now more crucial than ever to detect deepfake photos and videos.

In an effort to support researchers, numerous firms, including Google, Facebook Inc., and the US Defense Advanced Research Projects Agency (DARPA), started a research project aimed at detecting and preventing deepfakes. The term "deep fake detection" describes how challenging it is to identify fraudulent images or videos created using deep learning techniques. Machine Learning Algorithms are used to produce the deepfakes, so as to replace or change certain elements of an original image or video, such as a person's visage. Detection of these deepfakes is used to identify these alterations and separate them from authentic films or images

At the mesoscopic level of analysis, we plan to apply our technology to the detection of fake faces in photos. When phony images involve a person's face, they are particularly difficult for the human eye to discern at higher semantic levels. As a result, we suggest taking a middle-ground strategy and deploying a deep neural network with a fixed number of layers. With a low degree of representation and an unexpectedly small number of parameters, the architecture that we will talk about produced the best classification results of all our experiments. They are built on robust image classification

networks that switch between a dense network for image classification and layers of convolutions and pooling for feature extraction.

There is a difference between deepfake image detection and deepfake video detection. Deepfake image detection leverages the image pixels [17] and the analysis of noise level [18] for detecting the manipulation/AI generation in the image. The authors of [19] mentioned that deepfake detection methods can be categorized into holistic and feature-based matching techniques. Holistic techniques, includes PCA, SVM [21], Goal is to reduce data dimensionality forming set of linear combinations of image pixels which are fed to a binary classifier. This technique detects localized characteristics of deep fake images like (eyes, mouth, nose) [20]. The most Successful Techniques to identify and detect deep fakes are Deep Learning techniques which are primarily based on CNN [22].There are various CNN architectures that have been proposed for example VGG16 [24], XceptionNet [25], InCeptionV3 [23].

The combination of CNN and LSTM architecture is used in paper [26]. In which CNN helps to detect whether the eye is closed or open and LSTM helps to find temporal information, as blinking of an eye is a good correlation between the nearby frames. X.Yang et al in paper [27] suggests a way to detect deep fakes which are generated by superimposing the target person's face on the source face whereas detection is done by calculating the face key point difference and then feeding to the SVM for the final decision

CHAPTER 4 : CONVOLUTIONAL NEURAL NETWORK

ConvNets, short for Convolutional Neural Networks, are a particular kind of deep-learning algorithm that is mostly used for tasks requiring object recognition, such as picture categorization, detection, and segmentation. CNNs are used in many real-world applications, including security camera systems and driverless cars, among others.

4.1 CNN & Human Vision

The layered structure of the human visual cortex served as the model for convolutional neural networks. Some notable similarities and distinctions between the two are shown below.

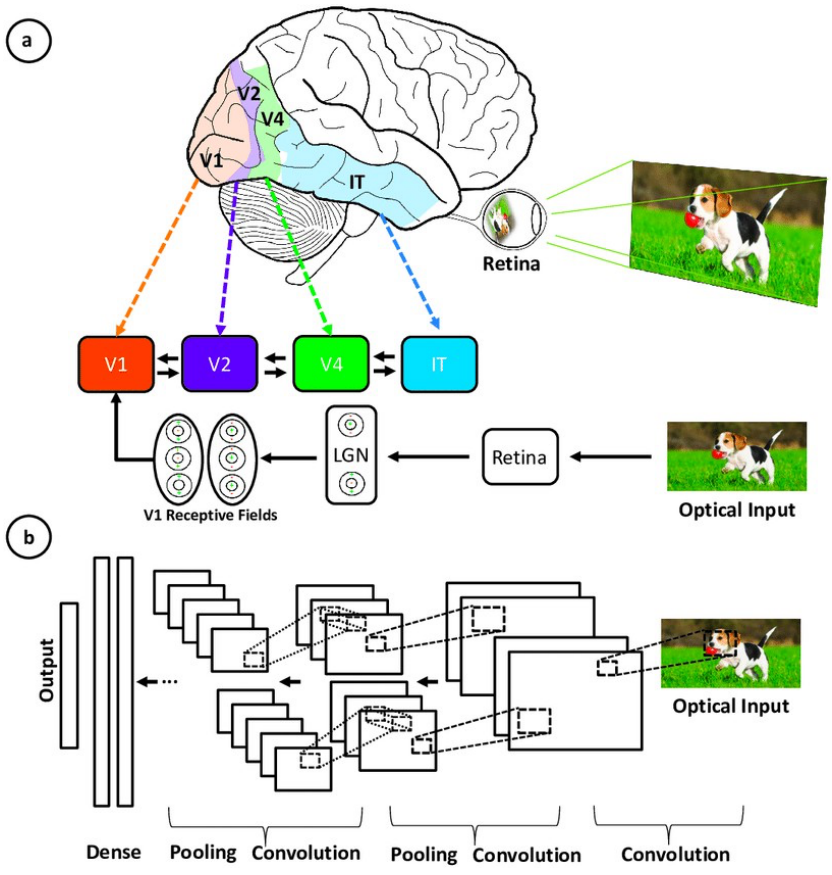

Fig 4.1 - An illustration of how the layers of a convolutional neural network correspond to the regions linked to the primary visual cortex

Hierarchical Structure - Both CNNs and the visual cortex exhibit a hierarchical organization,

progressing from simple to complex features through layers. Local Connectivity - Neurons in both systems establish connections locally, enabling efficient processing of visual information. Translation Invariance - CNNs, like the visual cortex, possess mechanisms to detect features regardless of their location, aided by pooling layers. Multiple Feature Maps - Both systems employ multiple feature maps at different processing stages to extract diverse visual information, Non-linearity - Neurons in the visual cortex and CNNs demonstrate non-linear response properties, crucial for capturing complex visual patterns

4.2 The Convolutional Layer: Core of CNNs

The convolutional layer plays a crucial role in CNNs, where much of the processing occurs. It requires a feature map, a filter, and input data, among other components. In the case of a color image, depicted by a 3-dimensional matrix of pixels (height, width, and depth for RGB values), these dimensions serve as input. The feature detector, akin to a kernel or filter, scans the image's receptive fields to identify features. Convolution, the underlying process, applies a filter represented by a 2D array of weights onto segments of the image. Typically, filters are 3x3 matrices that control the size of the receptive field. The resulting feature map or convolved feature arises from the dot product of the input pixels and the filter, generating an array output. After the filter traverses the entire image, it progresses by one step, continuing the process. The outcome of the input and filter interaction is a convolved feature or a feature map. Following each convolution, a Rectified Linear Unit (ReLU) correction introduces nonlinearity to the model.

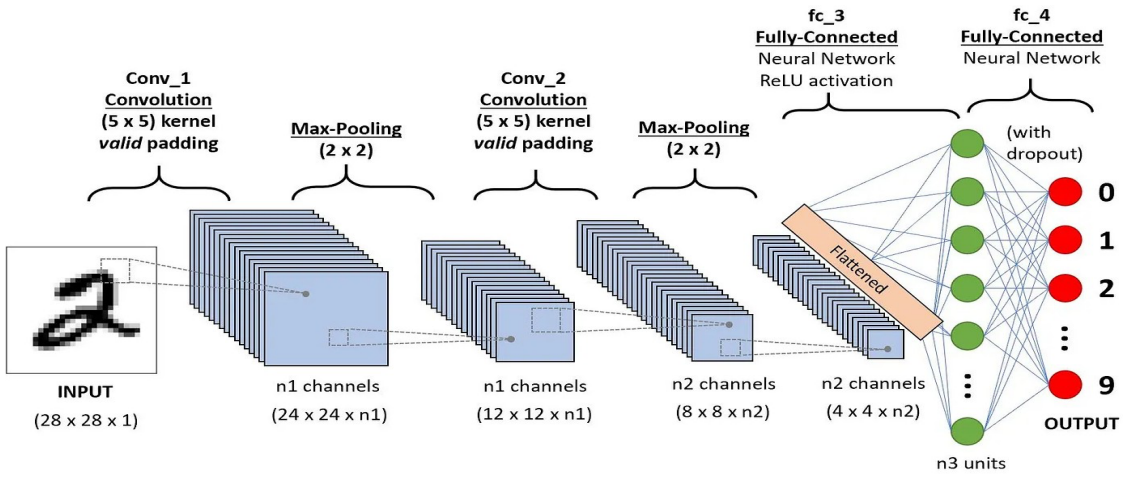

Fig. 4.2 - CNNs Architecture applied to digit recognition

Following the initial convolution, there may be subsequent convolutional layers. This hierarchical arrangement of CNNs allows successive layers to recognize pixels within the receptive fields of earlier levels. For example, in the task of identifying a bicycle in a photograph, each component of the bicycle represents a pattern of a lower level, while the amalgamation of its parts signifies a pattern of a higher level, providing a feature hierarchy within the CNN.

4.3 Pooling Layers

Extracting the most important features from the complex matrix is the aim of the pooling layer. This is accomplished by using a few aggregation processes, which decrease the feature map's (convoluted matrix) dimension and, as a result, the amount of memory needed for network training. Pooling is important for reducing overfitting as well. The most often used aggregating functions that are available for use are, Max Pooling - As it goes through the input, the filter chooses the pixel with the greatest value from the input array to deliver to the output array. This approach is more often used than classic pooling. Average Pooling, on the other hand, estimates the average value within the receptive field as the filter goes through the input and transfers it to the output array.

4.4 Activation Function

Following every convolution operation, a ReLU activation function is performed. By teaching the network non-linear correlations between the image's characteristics, this function strengthens the network's ability to recognize various patterns. It also aids in lessening the issues with fading gradients.

CHAPTER 5: DEEPFAKE DETECTION MODEL

There are ample of methods that have been proposed to detect Deepfake images or videos. Machine Learning based models includes Support Vector Machine[33], Naive Bayes(NB), Logistic Regression.It creates a feature vector using a feature selection algorithm, and then the vector is fed as an input to train the classifier to predict whether the media is manipulated. M. S.Rana, B. Murali, and A. H. Sung explains that feature extraction and selection are significant problems in machine learning models.

Based on how deepfake detection methods analyze facial alteration in photos and videos, they can be broadly categorized. Examining attributes that are individually retrieved from every frame is one method of looking for indications of tampering. An alternative method is to look for any discrepancies in the temporal characteristics of the video stream. These models look for the presence of deepfakes by examining the connection between the elements of successive frames across time.

Deep learning based models are mostly used due to their selection mechanism ability and features extraction property. Most of the Deep learning models are Built on Convolutional Neural Networks (CNN).In this study we mainly focus on Two CNN models namely ResNet And DenseNet. We focus on Developing a hybrid model leveraging the strength of Both ResNet and DenseNet for deep fake Image Detection.

5.1 ResNet

Residual Network is referred to as ResNet. In 2015 computer vision research paper titled "Deep Residual Learning for Image Recognition," Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun initially presented this novel neural network[28]. Stacked layers help deep convolutional neural networks solve complicated issues effectively. However,adding more layers may result in a decrease in accuracy, because of problems like vanishing or bursting gradients. This degradation can happen as the network gets deeper and is different from overfitting.

In order to mitigate this degradation, factors like optimization functions and initialization strategies are essential. This very dilemma was the motivation behind the creation of ResNet. Residual blocks are

used by deep residual networks to increase model accuracy.This type of neural network leverages its power using the concept of "skip connections", which is also a building block for understanding the residual blocks[28].

5.1.1 - Skip Connection

Skip connections in ResNet help to solve the vanishing gradient problem. The training of much deeper networks with lower error rates is made possible by skip connections, which add outputs from earlier layers to stacked layers. In general, ResNet's architecture reduces gradient vanishing and facilitates identity learning, which improves the performance of deep neural networks.

There are many versions of ResNet , ResNet-34, RestNet-50, ResNet-101. Our Main Focus is on RestNet-50.

5.1.2 - ResNet-50 Architecture

This Architecture consist of 4 parts:

Convolutional layers - They are essential to the process of extracting features. Convolution is the process of adding filters to input images so that the model can identify different textures, edges, and patterns in the data. Convolution blocks - Usually consisting of several convolution layers, these blocks are preceded by activation and normalization functions. High-level features can be more easily extracted from the input data thanks to them. Residual blocks - The model can bypass one or more layers by using residual blocks as skip links or shortcuts. This facilitates the efficient flow of information and helps to mitigate the vanishing gradient issue during training. Fully connected layers - These layers use the retrieved features as a basis for their predictions. The fully connected layers in the context of ResNet associate the acquired features with the final output classes.

5.2 - DenseNet

The next development in deep convolutional network technology is called Densely Connected Convolutional Networks, or DenseNets[29]. DenseNets streamline the layer-to-layer connectivity pattern that is present in other systems. The problem is solved by the authors in a way that maximizes gradient and information flow. They just connect each layer directly to the others to do this. It may seem counterintuitive, but DenseNets with this connection require less parameters than a typical CNN counterpart since redundant feature maps do not need to be learned.

Due to the previously described gradients and information flow, training very deep networks also presented challenges. Since each layer of DenseNets has direct access to the gradients from both the

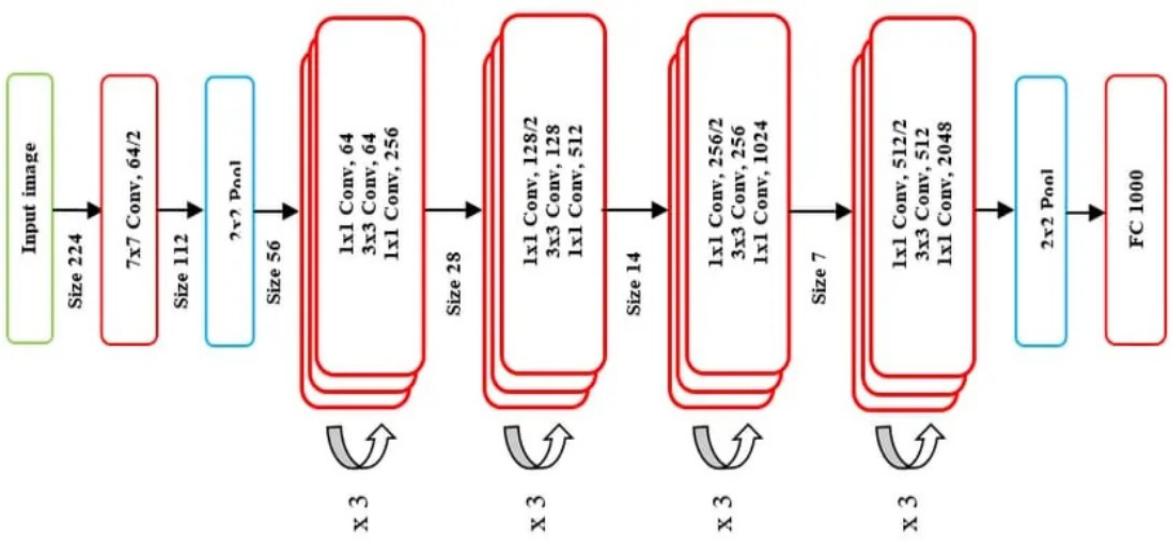

Fig. 5.1 - ResNet-50 Architecture

original input image and the loss function, Dense Nets address this problem.Using this type of Neural network has boosted the feature reuse encouragement, propagation, and resolved the vanishing-gradient problem, and also have reduced the number of parameters remarkably.

5.2.1 DenseNet Architecture

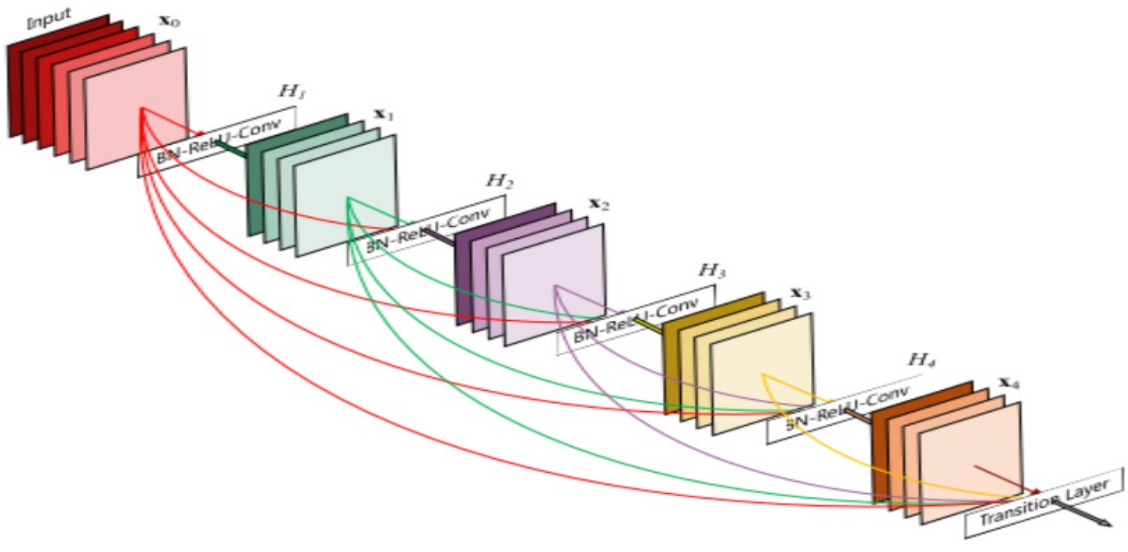

Fig. 5.2 DenseNet Architecture

Every layer in a Dense Net is connected to every other layer. There are n(n+1)/2 direct connections for n layers. Each layer uses the feature maps of all the layers that come before it as inputs, and each layer that comes after it uses its own feature maps as inputs[29].

However, we cannot simply maintain the same feature map size across the network; instead, down sampling layers, which alter feature map size, are a crucial component of convolutional networks. The network was split up into several densely connected blocks by the authors in order to aid with feature concatenation and architectural down sampling. The feature map size doesn't change inside the dense blocks. Transition layers - The layers between the dense block which reduces the feature map size a.k.a convolution and pooling are the transition layers[29].Transition layers in Dense Net architecture consist of batch norm layers, 1x1 conv and followed by 2x2 average pooling layers. Dense Block - Inside dense-block feature map size remains the same and each layer is connected to every other layer.

5.2.2 Different Dense Net Architecture

Every architecture has 4 Dense Blocks, each with a different number of layers. In the 4 Dense Blocks, for instance, DenseNet-121 has {6,12,24,16} layers respectively, but DenseNet-169 has {6, 12, 32, 32} layers.The first component comprises 1 Conv Layer having stride of size 2 and a filter of 7x7 size, Succeeded by a Max Pooling layer with a stride of size 2 and a filter of 3x3 size.

Layers	Output Size	DenseNet-121	DenseNet-169	DenseNet-201	DenseNet-264
Convolution	112 × 112	7 × 7 conv, stride 2			
Pooling	56 × 56	3 × 3 max pool, stride 2			
Dense Block (1)	56 × 56	[1 × 1 conv; 3 × 3 conv] × 6	[1 × 1 conv; 3 × 3 conv] × 6	[1 × 1 conv; 3 × 3 conv] × 6	[1 × 1 conv; 3 × 3 conv] × 6
Transition Layer (1)	56 × 56	1 × 1 conv			
	28 × 28	2 × 2 average pool, stride 2			
Dense Block (2)	28 × 28	[1 × 1 conv; 3 × 3 conv] × 12	[1 × 1 conv; 3 × 3 conv] × 12	[1 × 1 conv; 3 × 3 conv] × 12	[1 × 1 conv; 3 × 3 conv] × 12
Transition Layer (2)	28 × 28	1 × 1 conv			
	14 × 14	2 × 2 average pool, stride 2			
Dense Block (3)	14 × 14	[1 × 1 conv; 3 × 3 conv] × 24	[1 × 1 conv; 3 × 3 conv] × 32	[1 × 1 conv; 3 × 3 conv] × 48	[1 × 1 conv; 3 × 3 conv] × 64
Transition Layer (3)	14 × 14	1 × 1 conv			
	7 × 7	2 × 2 average pool, stride 2			
Dense Block (4)	7 × 7	[1 × 1 conv; 3 × 3 conv] × 16	[1 × 1 conv; 3 × 3 conv] × 32	[1 × 1 conv; 3 × 3 conv] × 32	[1 × 1 conv; 3 × 3 conv] × 48
Classification Layer	1 × 1	7 × 7 global average pool			
		1000D fully-connected, softmax			

Fig. 5.3 Comparison of Different Dense Net Architecture

Subsequently, the last Dense block is followed by a Classification Layer which takes in the feature maps from all levels of the network in order to carry out the classification process.

Additionally, the convolution processes inside each of the topologies serve as the Bottle Neck layers. This implies that the 1x1 convolution decreases the number of channels in the input, while the 3x3 convolution performs the convolution action on the altered input version with the decreased number of channels, then the initial input.

5.3 Hybrid Model

In this proposed model, we have endeavored to integrate two prominent convolutional neural network (CNN) architectures, specifically ResNet50 and DenseNet121, with the aim of enhancing the accuracy in detecting deepfake content**.**

5.3.1 Architecture

The idea to combine them arises from their complementary strength. ResNet known for its effective residual connections which facilitate the training of very deep networks, while DenseNet focuses more on feature reuse and dense connectivity helping in representation learning and feature propagation. The Hybrid model incorporates a mechanism of feature fusion, where features are extracted from ResNet50 and DenseNet121 which are concatenated to create a rich and informative feature representation. This model re-cognises diverse aspects of the input images and learns discriminative representation of real and fake images.

5.3.2 Training Strategy

Data Preprocessing **-** All images are uniformly resized to (224 x 224)px to ensure consistency. Techniques for augmenting data, like random rotation, arbitrary horizontal flip, and color modifications (hue, brightness, saturation ,and contrast), are utilized to boost the diversity of training data and enhance the model's capacity for generalization.

Loss Function **-** The loss function is called log loss, or cross-entropy loss. It penalizes inaccurate classifications and works effectively for multi-class classification jobs. Optimizer **-** The optimizer selected for model training is the Adam optimizer. AdaGrad and RMSProp's benefits are combined in Adam, an adaptive learning rate optimization technique. Learning Rate Scheduler - During training, the

learning rate is dynamically modified with the help of learning rate scheduler (ReduceLROnPlateau). When the validation loss reaches a certain number of epochs (patience), it reduces the learning rate by a factor of 0.1 and keeps an eye on validation loss as well.

5.3.3 HyperParameter Tuning

Hyperparameters are crucial settings that affect the performance of the model during training. In our model, several hyperparameters are tuned for optimal training and performance. Some of the key hyperparameters include

Learning rate - The learning rate is set to 0.0001 in this code and can be adjusted based on experimentation and model performance. Batch Size - Refer to the frequency of samples analyzed prior to changing the model parameters.The batch size is set to 32 in our model. Dropout Probability - Dropout helps prevent overfitting by regularizing the model.Its value is set to 0.3 which indicates each neuron has a probability of 30% of being dropped during training. Weight Decay - To include penalty for excessive weights term added to loss function.Weight decay is set to 0.001.

CHAPTER 6: RESULTS

The analysis of various deep learning architectures for image generation shows us some valuable insights. StyleSwin demonstrated much better performance as compared to StyleGAN, with a lower FID score of 3.25 [30]. VAEBM showed mixed results, it achieved an FID score of 20.38 on CelebHQ and 5.31 on CelebA 64 datasets, which tells us about room for further optimization [31]. FaderNet achieved a decent accuracy of 88.6% on CelebA, indicating its effectiveness in maintaining attribute invariance [32]. These findings show us the importance of exploring and refining deep learning architectures for improving image generation tasks.

We worked with CNN-based models for the classification of the fake and real images. Our first model which was based on ResNet-50 achieved an accuracy of 75.24%, with precision and recall scores of 0.80 and 0.68 respectively, and a test F1 score of 0.73. The second model based on DenseNet-121 gave a test accuracy of 79.13%, a precision of 0.77, a recall of 0.71, and a test F1 score of 0.74. We have done the hyperparameter tuning for both models to achieve the optimized results.

The performance of the Hybrid model on both training and validation sets, the model achieves a final validation accuracy of 85.4% after 20 epochs. Evaluation on the test set yields an accuracy of 83.7%, precision of 0.81, recall of 0.84, F1 score of 0.82, and a confusion matrix demonstrating the model's ability to effectively distinguish between real and fake faces. Visualization of training and validation metrics reveals convergence dynamics, with the model demonstrating robust classification capabilities, thus showcasing the potential of hybrid CNN architectures in facial image analysis tasks.

Model	Test Accuracy	Precision	Recall	F1 Score
ResNet-50	75.24%	0.80	0.68	0.73
DenseNet-121	79.13%	0.77	0.71	0.74
Hybrid Model	83.7%	0.81	0.84	0.82

Table 1. Performance metrics of CNN Models

CHAPTER 7 : CONCLUSION

In conclusion, this study talks about the hybrid CNN model for classifying real and fake faces. By integrating the ResNet-50 and DenseNet-121 architectures and applying data preprocessing techniques, we were able to obtain test accuracy of 83.7% and validation accuracy of 85.4%.This was achieved by combining the features we get from both the models. The model shows classification capabilities, as evidenced by strong precision, recall, and F1 score metrics. Our findings found the potential of hybrid CNNs in facial image analysis tasks.