



## Detecting Deepfakes: A Novel Framework Employing XceptionNet-Based Convolutional Neural Networks

Akash Saxena<sup>1</sup>, Dharmendra Yadav<sup>2</sup>, Manish Gupta<sup>3\*</sup>, Sunil Phulre<sup>4</sup>, Tripti Arjariya<sup>5</sup>, Varshali Jaiswal<sup>6</sup>, Rakesh Kumar Bhujade<sup>7</sup>

<sup>1</sup> Department of CSE, Compucom Institute of Information Technology and Management, Jaipur 302022, India

<sup>2</sup> Department of CSE, University College of Engineering and Technology, Bikaner Technical University, Bikaner 334004, India

<sup>3</sup> Department of Electronics and Communication Engineering, GLA University, Mathura 281406, India

<sup>4</sup> Department of CSE, Lakshmi Narain College of Technology, Bhopal 462022, India

<sup>5</sup> Department of CSE, Bhabha Engineering Research Institute, Bhabha University, Bhopal 462026, India

<sup>6</sup> Department of CSE, School of Engineering, Avantika University, Ujjain 456066, India

<sup>7</sup> Department of IT, Government Polytechnic, Daman 396210, India

Corresponding Author Email: [manish.gupta@gla.ac.in](mailto:manish.gupta@gla.ac.in)

<https://doi.org/10.18280/ts.400301>

### ABSTRACT

**Received:** 8 November 2022

**Accepted:** 10 April 2023

#### Keywords:

*deepfake faces and videos, facial landmarks, image processing, machine learning, deep learning, convolutional neural networks, Xception neural networks, preprocessing, classification*

Social networking sites have become primary sources of information for web users, making the rapid dissemination of deepfakes a cause for concern. Deepfakes are digitally manipulated images or videos that contain the computer-generated face of another person. Advancements in hardware and computational technologies have made the creation of deepfakes increasingly accessible, even to individuals without technical expertise. The potential harm posed by deepfakes necessitates urgent efforts to improve the detection of these manipulated media. Deep learning (DL) models have experienced rapid growth, enabling the synthesis and generation of hyper-realistic videos, often referred to as "deepfakes." DL algorithms can now create faces, swap faces between 2 individuals in video, and modify facial expressions, gender, also other features. These video manipulation techniques have applications in numerous fields, but deepfakes specifically exploit DL to synthesize and alter images in a manner that makes it difficult to discern between fake and genuine media. In this study, we present novel deepfake detection framework using DL and pre-trained XceptionNet model depends upon deep CNNs (Convolutional Neural Networks). We employ facial landmark recognition to extract information related to several facial characteristics from videos. This data is then used to facilitate the deep learning model's differentiation between genuine and deepfake videos. Features extracted from videos are utilized to train CNN concurrently. Our deepfake detection system is built on a multi-input Xception Neural Network model, which leverages CNNs. The system is trained using the Dassa Dataset and subset of Deepfake Detection Challenge Dataset. Proposed model demonstrates strong performance, achieving 96% classification accuracy and an AUC of 0.97, offering a promising solution for detecting deepfake videos.

## 1. INTRODUCTION

The significance of social media in disseminating news has grown exponentially in recent years, leading to increased attention on online information operations. Advancements in computer vision methods have facilitated the creation of convincing fake graphics, which, coupled with the rapid dissemination of false news via social media, have exacerbated this issue. Contemporary data-driven approaches have simplified the generation of fabricated images from scratch, in stark contrast to the past, when image manipulation required extensive expertise in rendering and image manipulation tools. Consequently, deep fake videos or images have the potential to inflict unparalleled damage on political climates and individuals' private lives [1].

In recent years, technological advancements have enabled even non-experts to create deep fakes effortlessly [2]. These deep fakes involve digitally altering real people's faces in

videos or photographs, often generated by generative adversarial networks (GANs), and can deceive audiences into believing that prominent figures are genuinely speaking when they are not [3]. As society becomes increasingly reliant on social media for critical information, the threat of misinformation from deep fakes intensifies. The need for software capable of detecting deep fakes—which can deceive the human eye—has become essential in light of the significant risks posed by false information. Given the rapid advancements in artificial intelligence (AI) in related domains [4, 5], it is evident that AI-driven solutions are necessary to counteract the detrimental effects of deep fake misuse [6].

In September 2020, prior to the U.S. elections, Microsoft introduced advanced technology for deep fake detection. To train or evaluate the deep fake detector, the Face Forensics++ dataset and the DFDC dataset were used by the Video Authenticator program. This program detects quiet or grayscale elements within the blending borders of deep fakes.

However, most research in this area is still evolving to keep pace with the rapid development of deep fake detection and production, resulting in a largely concealed subject.

The term "Deepfake" is a portmanteau of "Deep Learning" and "Fake" and refers to content that has been technologically altered to appear photorealistic. This term was coined by an anonymous Reddit user in 2017 to describe the use of deep learning algorithms for replacing one person's face with another's in sexually explicit videos, producing highly convincing results [7]. Deep fake videos, created using deep learning algorithms, have gained significant attention recently. Rapid advancements in AI, ML, and DL have led to the development of novel methods and tools for manipulating multimedia content, paving the way for deepfake applications powered by deep learning [8].

Convolutional neural networks (CNNs) are used in various forms in all deep fake detection methods, as demonstrated by recent deep learning research [2, 9-12]. CNNs, which employ deep-learning, multi-layer classification neural network architectures, are capable of accurately identifying the visual attributes most relevant to a target category. Deep learning-based models often utilize CNNs due to their ability to detect areas of the frame with strong spatial and temporal connections. Extracting a face or facial features, such as the eyes or mouth, may help identify inconsistencies between real and deep-faked versions of an image. Several deep fake models employ CNNs [13, 14], albeit with differing implementations.

This study investigates the use of face recognition-extracted features within a classifier for the aforementioned data. As deep fake techniques alter a person's identity, they may modify their features. Current detection approaches typically rely on handcrafted low-level features, pre-trained object recognition embeddings, or CNNs, but rarely on face recognition networks [15]. A detailed examination of identity-separated extracted features is needed to improve deep fake detection.

The primary objective of this research is to implement a deep learning-based pre-trained CNN model, XceptionNet, using the DFDC dataset. A deep learning-based Xception Neural Network is proposed for detecting deep fake videos. XceptionNet is well-suited for image classification tasks for several reasons:

- **Extreme depth:** With 71 hidden layers, XceptionNet is a profoundly deep neural network, allowing the model to learn complex features and patterns from input data, which is especially beneficial for image recognition tasks.
- **Separable convolutions:** XceptionNet utilizes separable convolutions, a type of convolutional operation that separates spatial and channel-wise components of the convolution. This technique helps limit the number of model parameters, addressing the overfitting issue commonly encountered in deep learning.
- **Better performance:** XceptionNet has demonstrated superior performance on image classification tasks compared to many other state-of-the-art deep learning models.
- **Lower computational cost:** XceptionNet requires less computing power than other deep learning models due to the utilization of separable convolutions, which reduces the total number of training parameters.

The goal is to enhance the deep learning model's detection rate of deep fake videos. This research provides the following contributions:

- A comprehensive review of existing literature on deep fake detection, discussing current research tools, methodologies, and datasets while addressing specific research challenges.
- A proposal for a deep learning approach using Xception Neural Networks to detect fake videos.
- Improvement of the deep fake detection model by incorporating face recognition-extracted features, which can potentially identify inconsistencies between real and deep-faked versions of an image.

The remainder of the paper is organized as follows: Section 2 reviews deep fake generation methods and related deep fake detection techniques. Section 3 details the proposed deep learning-based approach for deep fake detection using Xception Neural Networks. Section 4 presents the experimental setup, including the dataset used, data preprocessing, and model evaluation. Section 5 discusses the experimental results, highlighting the performance of the proposed model. Finally, Section 6 concludes the paper and provides recommendations for future research.

## 2. RELATED WORK

The development of deep fakes has increased dramatically in recent years. These days, it's easy to make a deep fake with the help of one of the several methods available. Privacy, democracy, and confidence are all at risk as a result. As a result, there is a growing need for a deep fake investigation [16, 17]. While the principles behind face-swapping have been known for quite some time, the arrival of AI and its availability has greatly increased the gravity of deep fakes. Only lately have ideas been proposed to identify deep fakes, as well as we go through a number of them in depth [18]. In this part, we will discuss how deep fake methods came to be, the research done, and the development in the field of deep fake video generation and detection.

### 2.1 DL techniques for deepfake video generation

AEs (Autoencoders) and GANs (Generational Adversarial Networks) are two types of generative models that are used to generate and synthesize deep fake content [19, 20]. Deepfake is made by switching the identity of the people in a photo or video [21]. Deepfake may be made using a variety of approaches, with but not limited to face swapping, puppetry, lip-syncing, face-reenacting, computer-generated image/video creation, and synthetic voice production. Translating between still images and moving videos (both supervised and uncontrolled) [22, 23] Authentic-looking Deepfakes may be produced with the help of the study [22].

The first Deepfake method is FakeAPP [24], which relies on a pair of AE networks. An AE is encoder-decoder-style FFNN (Feed Forward Neural Network) which is trained to reconstruct its input data. The encoder in FakeApp is responsible for revealing the concealed face characteristics, while the decoder is responsible for reassembling the facial pictures. To switch here between target and source confronts, both AE networks share the same encoder but are provided with training on distinct sets of decoders.

Most Deepfake generation techniques are centered on the 'facial' area, where alterations like 'face swapping' and 'pixel-by-pixel' editing are often implemented [25]. The face swap process involves switching the facial features of two images. The video's director acts as a "puppet master," manipulating the filmed subject's performance. Lip-sync and face reenactment include manipulating a target video's mouse movement under the source's direction [26]. Feature maps that stand in for the original and fake images are used by the vast majority of Deepfake generation algorithms. FACS (Facial Action Coding System), segmentation methods, landmarks, & facial borders are all examples of representations of facial features using convolutions [20]. The FACS taxonomy classifies human facial expressions according to 32 atomic facial muscle activities (AU) and 14 ad hoc actions (AD). The eyes, nose, and mouth locations are all examples of facial landmarks [27].

## 2.2 DL techniques for deepfake video detection

There are three main types of deep fake detection techniques [14, 28]. Approaches in the first category analyze the videos, for instance, by monitoring the participants' visual stimuli or the orientation of their heads. The second class focuses on GAN fingerprinting and image-based biological signals like blood flow. Artifacts of visual nature make up the third group. Techniques that emphasize visual artifacts are also data-driven, so they need a sizable quantity of data for training. The third option describes the recommended model. Here we will go through many different architectures created to detect

the classic symptoms of a Deepfake's visual manipulations.

Darius et al. introduced a CNN model called the Mesonet network to identify highly realistic Deepfake and Face2Face-generated fake videos [29]. Two network topologies were used by the researchers (Meso-4 and MesoInception-4) that are particularly well-suited for analyzing mesoscopic features of a picture. Li and Lyu [14] suggested a CNN architecture that uses imperfections introduced by image transforms (i.e., scaling, rotation, and shearing) to generate Deepfakes. Their method uses affine face-warping artifacts as a fingerprint for authentic and false pictures. To detect resolution discrepancies caused by face warping, their technique compares the Deepfake face area to the surrounding pixels.

Nguyen et al. have suggested a revolutionary deep-learning method for identifying fake media. Research concentrates mostly on replay assaults, facial reenactments, face swapping, & completely computer-generated photo spoofing [30].

Md. Shohel Rana and Andrew H. Sung introduced DeepfakeStack, ensemble technique (stack of several DL models) for detecting such fakes. Open-source DL models InceptionV3, XceptionNet, MobileNet, InceptionResNetV2, DenseNet121, ResNet101, as well as DenseNet169 are included in this compilation [31]. Kim et al. [32] have developed a classifier that uses pre-trained DL models (ShallowNet, VGG-16, and Xception) to identify target persons from a group of people with similar characteristics. Their major motivation for developing this system was to evaluate efficacy of three distinct DL models for classification task.

**Table 1.** Comparative analysis of various techniques for deep fake detection

Refer.	Author/Year	Model	Dataset	Performance	Limitation
[33]	Pokroy and Egorov (2021)	EfficientNets	DFDC dataset	70% to 74% accuracy	While no. of variables may be drastically decreased by switching to a B6 or B7 model, the precision takes a major hit.
[34]	Korshunov and Marcel (2021)	Xception and EfficientNet	Google (from FaceForensics++) and Celeb-DF deep fake datasets	AUC value of 87.47%.	Many videos that people may easily see as false are difficult for computers to identify.
[35]	Jaiswal (2021)	Multilayer hybrid recurrent deep learning models	Deepfake videos data	The experimental results show that these models outperform stacked recurrent DL models.	-
[36]	El Rai et al. (2020)	pre-trained deep network InceptionResNetV2	DFDC	accuracy of 85% on the available dataset	The work drawback is to incorporate the temporal consistency between frames
[37]	Kharbat et al. (2019)	Support Vector Machine (SVM)	Videos dataset	High detection rate	
[38]	Ismail et al. (2021)	YOLO-CNN-XGBoost	CelebDF-FaceForencics++(c23) dataset	Measured at 90% (AUC), 93% (sensitivity), 93% (specificity), 85% (recall), 86% (precision), and 86% (F1)	The current methods of detection need to be improved further.
[39]	Kaddar et al. (2021)	HciT hybrid architecture	Faceforensics++ and DeepFake Detection Challenge preview datasets	According to the results, the suggested strategy produces far better results than the current gold-standard approaches.	-

The fast development of CNNs, GANs, and their derivatives has made it feasible to generate hyper-realistic pictures, films, and sounds that are far more difficult to tell apart from authentic, unaltered recordings. Concerned parties have issued an appeal to action to prevent the development of technologies that enable the creation of fake but convincing audio, video, and other media from being employed by adversaries for malevolent ends [11]. Thus, there is a pressing need for the scientific community to create measures for detecting Deepfakes. Table 1 displays a selection of ancillary methods and software for identifying deep fake films.

While several existing deep fake detection methods show promise, they are still hampered by several serious problems. Under novel or novel deep fake generation approaches, poor generalization outcomes have been achieved. While progress has been made, this is still a difficult problem to solve since Deepfake technology has been advancing rapidly and can now deep train detectors.

Otsu suggested a global thresholding technique, which he calls his approach [39]. This approach, which relies on the histogram's global characteristic to determine the right threshold, is both automated and consistent.

### 3. RESEARCH METHODOLOGY

#### 3.1 Problem statement

Face recognition is one of the most challenging things in image processing due to the difficulties posed by developing systems for facial classification. Face recognition is the first concern to be solved. Face Detection issues may be attributed to the following two sources. Included are facial emotions and other facial characteristics. Facial expressions are one of the most powerful and immediate indicators of a person's personality since they reveal their emotions and intentions. Facial expressions of emotion, such as rage and happiness, may cause noticeable changes in a person's facial features. Quite a few people have specs, while some are sporting facial

hair or scars from accidents or earlier injuries. Facial traits are the correct phrase for these features. New methods of detecting deep fakes have to be created because of their improving quality. Deep and shallow classifiers are the two most important classifiers for deep fake identification. Using DL models, fake photos and videos may be differentiated from authentic ones based on their distinctive characteristics. As an illustration, reflections in the eyes and other features may be lost. Similar differences in the teeth may be employed in the same manner.

#### 3.2 Proposed methodology

To tackle the challenges above with deep fake, we will construct a deep learning-based Xception NN model to recognize deep fake movies using the Deepfake Detection Challenge Dataset in this research. Using facial landmark detection, we gather data relevant to different facial features from the videos. The raw video feed was assembled from the still images. The location of the eyes, nose, & lips has been determined using facial landmark detector. Eye blinks & other facial characteristics have also been identified using this data. There is some preprocessing work involved in supplying this information to the model. However, the images have been converted to their numerical representation via a preprocessing procedure. The first process involves resizing the input images and cropping them to focus on the faces. Now ensures that almost all pictures use the RGB channel. After finishing the preprocessing step, the training, validation, and testing sets were divided. The deep learning model undertakes feature extraction and train on the extracted features. To use this Xception Neural Network model, the classification step has determined whether or not a particular video is deep fake. Deep transfer learning is advantageous since it improves the model's interpretability and comprehension while reducing its computing cost. We will discuss findings on relatively quickly acquired Deepfake datasets with similar or better performance than existing DL-based algorithms.

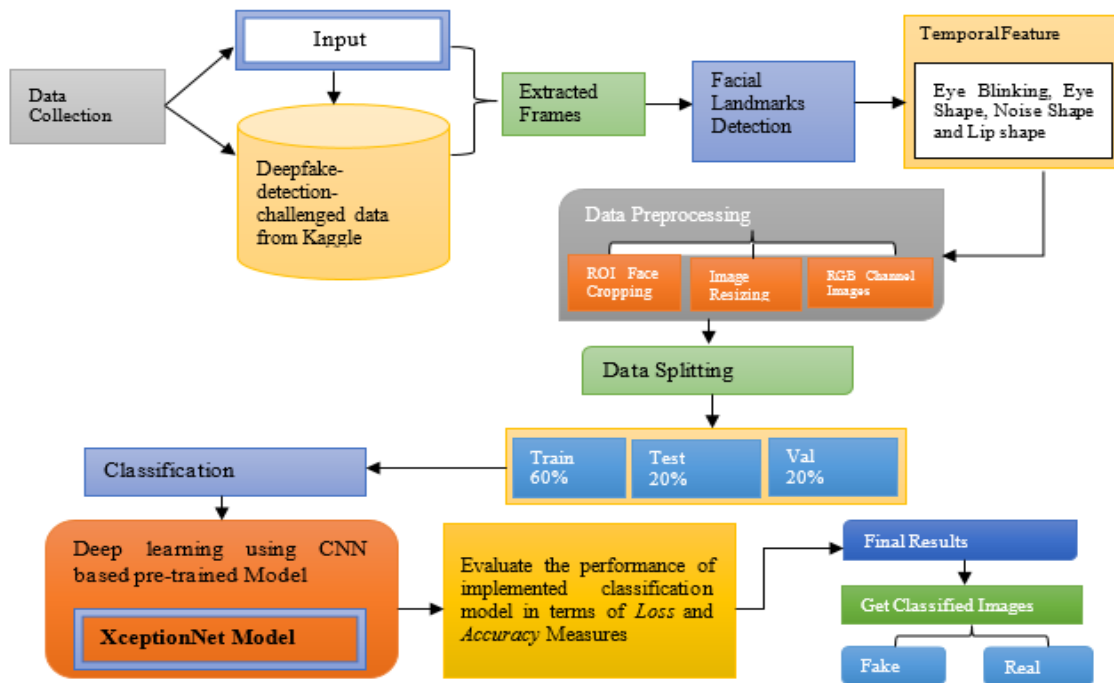


Figure 1. Block diagram of the research methodology

Figure 1 depicts planned work's flowchart. In Figure 1, we see that this work starts with the data collection process, so we have used a deep fake dataset from Kaggle. The input dataset extracts the number of frames and then detects facial landmarks with the help of temporal features like eye, nose, lip, etc. After this, preprocessed the extracted images for image ROI face cropping, resizing, and converting images in RGB channels. The dataset has been divided into train, test, and Validation in the following proportions: 60%, 30%, and 20%. Here implements the deep learning using CNN-based pre-trained XceptionNet model for classification of deep fake. In conclusion, assess how well the suggested model performs in terms of its accuracy and loss. This complete method is being offered to determine if a video is fake or real.

### 3.2.1 Face landmark detection, and frame extraction

We have collected the 68 facial landmarks features from face images those landmarks features belong to eyes, nose, mouth, and edge of the face. We begin by splitting the input video into individual frames. The facial area is extracted from every image. Obtaining the face in X and Y coordinates for face recognition has various steps:

- **Face Detection:** Then, we use the Face detection technique from the lib package to identify any faces inside the picture.
- **Face alignment:** Once the face is detected, it is necessary to align the face using a landmark detection algorithm.
- **Face encoding:** After the face is aligned, it returns the set of features of landmark value such as the shape of face, eyes, nose, and mouth.
- **Recognition area:** The face is in image's recognition area, which is the area around it. Bounding boxes are often used to indicate the region of interest (ROI), which is the area around a detected face.

This landmark detector employs a group of regression trees to make estimations of face landmark coordinates using only the relative intensity of neighboring frames.

### 3.2.2 Eye blink detection

The vast majority of deep fakes by performers never blink nor blink abnormally. So, this nuance calls for a greater degree of skill to implement well. Next, we analyze the photographed person for any discernible blinking patterns.

The blink detector takes on facial landmarks as input, namely eye coordinates (points 37-46 in list of retrieved facial features). Soukupová and tech's formula for calculating the EAR (Eye Aspect Ratio) are used in the eye blink detector (2016). The six (x, y) coordinates that make up each eye are labeled from top left (p1) to bottom right (p6) in a clockwise fashion (p2, p3, p4, p5, p6). To calculate the EAR using this information, just use the formula in Step 1 below [40].

$$EAR = \frac{||p_2 - p_6|| + ||p_3 - p_5||}{2||p_1 - p_4||} \quad (1)$$

While an eye is open, the EAR is 1, but it drops to 0 when the eye is closed. An increase in blink rate is seen if the median EAR value of one eye goes below a threshold (EYE AR THRESH) for a consistent no. of frames (EYE AR CONSEC FRAMES). In this study, we have used values of 0.3 & 3 for EYE AR THRESH and EYE AR CONSEC FRAMES parameters, respectively.

### 3.2.3 Extraction of shape features

The facial landmark detector may be used to get the locations of the eyes, mouth, and nose. While face swapping is so common in deep fake films, adapting the shape of the eyes is a crucial aspect of the feature extraction process. During a real-world video, we found that a person's eye shape remained quite constant. When the film has been doctored countless times, it is no longer reliable. And similar to how most cases of facial warping occur close to the mouth, most cases of inconsistent facial features also occur there. Deepfake modification may also result in altered lip contours. As a result, we want to train our classifier using the differences in the shape of the facial features in various photos.

Coordinates (points 37-46) for the eye are provided into the eye form detector, which was determined using the facial landmarks. The furthest locations of the eyes are measured to calculate the left-right Euclidean distance (d1) and the right-left Euclidean distance (d2). Lip coordinates extracted from face landmarks are sent into the lip form detector (points 49-68). Euclidean distance between centers of the inner lips is used to determine their length (d3). As with the inner lip, the length of the outer lip is supposed to be determined by the inverse square of the distance between its coordinates as calculated using the Euclidean distance (d4).

The Nose Shape Detector gathers data about the nose from the facial landmarks (points 28-32). You can figure out how wide your nose is at its base by measuring the distance (in Euclidean units, d5) between your nostrils. You may approximate the nose's width by measuring the Euclidean distance (d6) between its extremities.

Hence, we can calculate the distance between inner & outer lip coordinates (d1, d2), width of nose (d3, d4), as well as width of eyes (d1, d2) (d5, d6).

### 3.2.4 Data preprocessing

Deep fake image classification requires pre-processing that improves the picture quality for further processing. Figure 1 depicts a flowchart of the preprocessing phase. Figure 1 depicts the three pre-processing steps: picture resizing, cropping the facial area of interest, and ensuring all photos are in the RGB color space.

#### Resize Image

All the images in the dataset were different sizes, and it would be impossible to get a reliable result from processing data of varying sizes. For the sake of processing, all photos were scaled down to 196×196. Downsampling, as well as upsampling techniques, were used to adjust the input image's resolution.

#### Region of Interest

ROI stands for "Region of Interest", It refers to a subset of data inside a larger collection that has been singled out for analysis. The idea of a profit from an investment is often used. For instance, data size may be determined by drawing its borders on an image or in a volume to analyze deep fake data. For face recognition, ROI is region within face's bounding box, which is determined by the face detection method.

#### Images Convert in RGB Channel

Red, green, and blue are the three colors that make up an RGB picture. The human visual system is made up of three-color receptors, as well as RGB channels are utilized in computer monitors and image scanners because they closely correspond to these.

### 3.2.5 Data splitting

Splitting data is a common practice in machine learning, and it's usually done to prevent overfitting. That's an instance of a machine learning model being faulty because it overfits its training data. In a machine learning model, the initial data is often divided into three distinct subsets labeled "train," "test," as well as "validation." Here we split the dataset into in ratio of 60:20:20.

**Training Set:** A dataset is used to teach a model to identify anomalies and new patterns in the training data. 60 percent of data is in training set.

**Validation Set:** It's a different collection of data than the training set we use to check how well our model is doing while it's being trained. The validation set contains 20% of the dataset.

**Test Set:** The model is then tested on a different dataset when the training phase is complete. The training set is comprised of 20% of the whole dataset.

**Table 2.** Validation set size for training and testing, in terms of the number of photos

Split data	Images
Training data	2399
Validation data	750
Testing data	600

Above Table 2 shows the different number of images for each section of data i.e., training, testing, and validation. There are 2399 images for training data, 750 images for validation data, and 600 images for testing data given in this Table 2.

As was previously noted, the dataset was split as follows: 60% for training, 40% for validation, and 20% for testing, and the performance results on these ratios is given in Table 3. This is the conclusion we have reached, or so it seems:

**Table 3.** Performance results for different splitting ratio in terms of Training (60%), Validation (40%), and Testing (20%)

Training_Loss	Training_Acc	Validation_Loss	Validation_Acc	AUC_Score
0.0113	0.9987	0.1597	0.9693	0.97

These variables would affect the outcomes if dataset was separated into a training (40%), validation (30%), and testing (30%) ratio. The Table 4 below shows the AUC score change and the validation accuracy.

**Table 4.** Performance results for different splitting ratio in terms of Training (40%), Validation (30%), and Testing (30%)

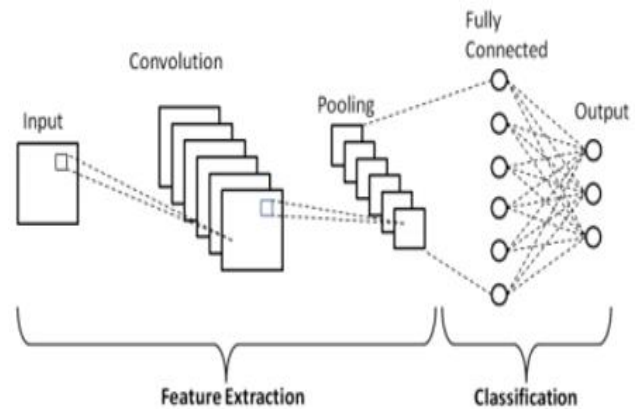
Training_Loss	Training_Acc	Validation_Loss	Validation_Acc	AUC_Score
0.0131	0.9989	0.2500	0.9511	0.95

### 3.2.6 Classification using the DL model

Deep Learning [41, 42] is a well-known subfield within AI. It performs very well on a wide variety of data formats. When a DL model is fused with CNN, an image classification network may be produced. Figure 2 shows the skeleton of a typical CNN system. Digital pictures are stored in computers as collections of pixels that have common characteristics.

Some groups of pixels in the picture may stand for edges, while others may depict shadows or other patterns. The use of convolution is one strategy for spotting such patterns. During processing, a matrix is used to describe the image's pixels. A "filter" matrix is increased by the matrix of picture pixels to identify the patterns. One may start convolution from the first pixel in the picture pixel matrix and choose pixels depending on the filter size for multiplication, where filter size determines the range of possible filter sizes. Convolution is performed for each pixel in the matrix until the whole picture has been processed. Numerous parameters are shared when convolving. A feature map is the layer-two output from a convolution. The pooling layer follows the activation layer in the CNN model. In order to prevent overfitting, this layer reduces size of final product, the feature map. Finally, on the topmost layer, all connections are made. A single vector representing the output of the current layer is "flattened" so that it may be utilized as an input by the subsequent layer. The feature maps obtained at each stage can be normalized using batch normalizations. Because of this, we employ dropouts to speed up the computation.

A DL strategy that uses CNN based on a pre-trained Xception Neural Network model is proposed in this article as a method for identifying false videos.



**Figure 2.** Classification using a fundamental CNN architecture

### Pre-trained Xception architecture

The name Xception comes from the phrase "Extreme Inception." Google was the one that suggested using this approach. It utilizes the same parameters found in Inception V3 and comprises those parameters. The higher performance of Xception may be attributed to the effective use of the model's parameters and the system's expanded capacity. Under the inception framework, the output maps consist of cross-channel and spatial correlation mappings. The Xception architecture partitioned these mappings [43]. To extract features from the data using the network, 36 convolutional layers of the design were employed. These 36 levels are broken up into 14 different modules altogether. Linear residual connections encompass each module. Modules created first and last do not have any examples of this kind. The final FC tier will now consist of six courses instead of the previous seven. The proposed model contains some hyperparameters described below in Table 5 and a detailed function description.

The no. of epochs, batch size, learning rate, as well as loss function are all parameters that we set. Epochs are nothing more than the number of iterations, that is, 20, and each epoch lasts 20 years. By maintaining the batch size at its default value

of 32, the no. of input samples sent to the network may be defined by the batch size some portions of the weights are adjusted when the neural network is being trained. These numbers are known as learning rates. This is a major CNN model hyperparameter, with values between 0.0 and 1.0. In our model, we used 0.0001 learning rates and investigated how different learning rates affected accuracy. Also, we used binary Crosse entropy loss function for classification models, which output a probability. A neural network's ability to accurately represent the training data is evaluated using something called a loss function, which is a function that contrasts the target as well as forecasted output values. During training, one of our goals is to reduce the deviation between the expected and target outputs.

**Table 5.** Setting of hyperparameters

Parameters	Set values
Epochs	20
Batch_size	32
Learning_rate	0.0001
Loss function	Binary Crossentropy

The model's probabilities are compared to the actual class output, which may be either zero or one, using the binary cross-entropy method. Next, it assigns a value to each likelihood depending on how much it deviates from the expected value, and finally, it generates a score. It shows how near or distant we are from true value. Binary cross-entropy is described in this way:

$$\prod_{c=1}^c y_c(x, w_c)^{t_c} (1 - y_c(x, w_c))^{(1-t_c)} \quad (2)$$

Model: "model"

Layer (type)	Output Shape	Param #
image_input_lr (InputLayer)	[(None, 196, 196, 3)]	0
xception (Functional)	(None, None, None, 2048)	20861480
flatten (Flatten)	(None, 73728)	0
dense_layer_1 (Dense)	(None, 1024)	75498496
dropout (Dropout)	(None, 1024)	0
dense_layer_2 (Dense)	(None, 1024)	1049600
dropout_1 (Dropout)	(None, 1024)	0
output_layer (Dense)	(None, 2)	2050

=====  
 Total params: 97,411,626  
 Trainable params: 97,357,098  
 Non-trainable params: 54,528

**Figure 3.** Model summary of proposed Xception model

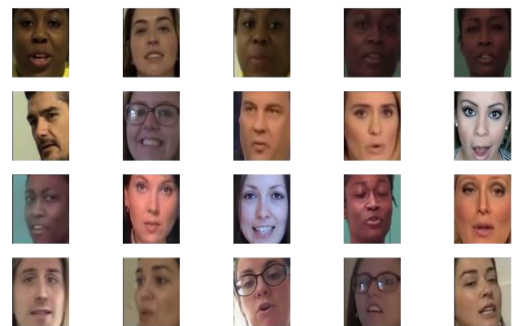
A description of suggested Xception model, which consists of various layers & parameters, is demonstrated in Figure 3. The total parameters of model are 97411,626, with trainable parameters of 97,357,098 & non-trainable parameters are 54,528, respectively.

## 4. RESULTS ANALYSIS

The simulation outcomes of the experimental work for the deep fake identification using the deep learning algorithm are provided in this part. We used multiple pieces of gear to run the simulation, including a Windows 10 PC with 2.6GHz Intel Core i7-9750H Processor & 16.0GB of physical memory. Python was the language of choice for the software engineering toolset. Pandas is a module for Python that enables arrays to have multiple dimensions. This module may be used to do data analysis. A simulation system was included with the Jupyter notebook and several Python libraries. These libraries included NumPy, Pandas, Matplotlib, Keras, TensorFlow, and seaborn. A large range of different specialized performance matrices was applied (described below). The conclusions of this research were only possible because of the usage of a dataset, which will be discussed further in this section. The experiment results are shown in several different graphs, metrics, and tables that are summarised.

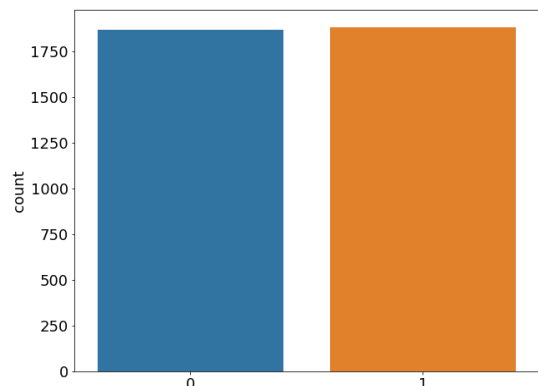
### 4.1 Dataset description

As for the numbers, the Deepfake Detection Challenge dataset has a total of 400 movies, 323 of which are fake and 77 of which are real. We selected 199 fake videos and 53 real ones using the criteria laid forth in the study. These short clips last for just 10 seconds each. We utilized 66 films from a FaceForensics++-obtained YouTube dataset of actual videos to ensure an even distribution of false and real videos. So, out of a total of 318 videos, 199 are false and 119 are genuine. Below, Figure 4 shows sample image of deep fake dataset.



**Figure 4.** Sample images of the deep fake dataset

Figure 5 displays a distribution graph of the deep fake dataset, which is split evenly between fake and actual data.



**Figure 5.** Data distribution of a dataset

## 4.2 Performance measures

Performance metrics are essential for the computation of the experiment's outcome regarding classification models. There may be a lot of different performance indicators that might represent the outcome of the categorization. As a visual representation of a classification algorithm's performance on the test dataset, the confusion matrix. It is comprised of following four parameters: TP, FP, TN, as well as FN.

- **TP (True Positive):** a measure of how many positive examples were correctly predicted by a classifier.
- **FP (False Positive):** A measure of the frequency with which a classifier incorrectly predicts a positive outcome when a negative one occurred.
- **TN (True Negative):** Representing the total number of false-negative predictions made by a classifier.
- **FN (False Negative):** A measure of how often a classifier incorrectly predicted a good event to be a negative one.

**Confusion Matrix:** A confusion table may be used to analyze the performance of a classification model (or "classifier") on test data when the real values are known. The results of an algorithm may be shown in this way for the first time.

We discovered that the most important variables to consider for gauging our model's efficacy were accuracy, precision, recall, ROC, and F1 score:

### 1) Accuracy

The degree to which a classifier can successfully provide reliable predictions is called its accuracy. The classifier's total performance capability is improved due to this feature. The precision may be determined using Eq. (3).

$$\text{Accuracy} = \frac{TP + TN}{N} \quad (3)$$

### 2) Precision

Precision is described as the capacity to predict positive outcomes accurately. A high FPR, as well as a high accuracy, definitely connect. Calculating accuracy involves applying the continuity formula (4) to the supplied data:

$$\text{Precision} = \frac{TP}{TP + FP} \quad (4)$$

### 3) Recall

Keeping track of which samples should have been counted as positive necessitates keeping them apart by an exact positive number. To get a reliable outcome, this is essential. Achieving the desired recall rate with this method:

$$\text{Recall} = \frac{TP}{TP + FN} \quad (5)$$

### 4) F1 score

The F1 score is determined by taking an average of the weighted categories of accuracy and recall. It's a metric for measuring the classifier's current health as a statistical tool. The F1 score may be calculated by using Eq. (4), which can be found below:

$$F_1 \text{ score} = \frac{2 \times (\text{Precision} \times \text{Recall})}{(\text{Precision} + \text{Recall})} \quad (6)$$

By contrasting the relationship between False Positive Rate (FPR) & True Positive Rate (TPR), ROC curve may be used to evaluate the performance of various classifiers. FPR is shown on x-axis & TPR on y-axis of ROC diagram. ROC curve may be used to evaluate classifiers by varying the threshold.

## 4.3 Simulation results

The simulation results of the suggested model are shown below; it is an Xception Neural Network trained on the deep fake dataset, which is a tool for deep learning. The suggested model uses hyperparameters such as 20 epochs, 32 batches, a 0.0001% learning rate, and a loss function based on binary cross entropy. The below simulation outcomes show loss, accuracy, and AUC graphs. The results perform on python simulation and split dataset training and validation set.

Due to run time constraints, the developed Xception model was only evaluated for 20 epochs. Yet, it still managed to obtain a 99.87% accuracy rate on the training set and a 96.93% accuracy rate on the validation set. Graphs obtained after implementation (see Figure 6) show that validation and testing accuracy increases with epoch count.

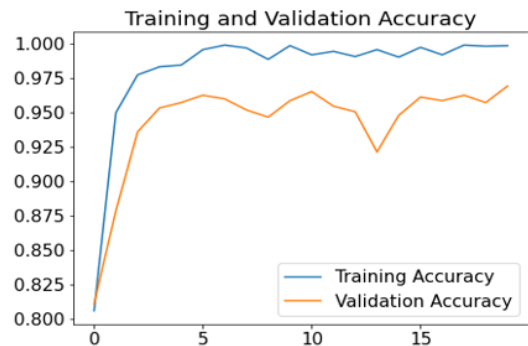


Figure 6. Accuracy plot graph of train and Val dataset using 20 epochs

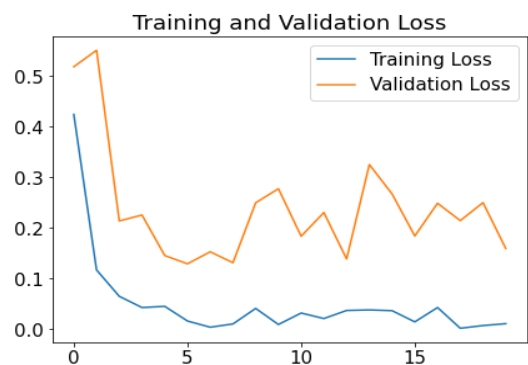


Figure 7. Loss plot graph of train and Val dataset using 20 epochs

Figure 7 demonstrated loss graph of the Xception model. The Prediction Performance of the Xception Neural Network displayed graphs of the training loss & validation loss versus neural network. The value of the loss will decrease as the epoch number grows. Nevertheless, there was a significant gap between training loss and validation loss. The value of the training loss was higher than the validation loss. The gap in the number of losses experienced during training and validation increases as the epoch values continue to rise.



The above confusion matrix obtains from the proposed Xception neural network that builds with the help of a deep fake dataset that categorized two classes, 0 as fake and one as real, shown in Figure 8. The matrix x and y-axis show predicted label and true label of the dataset. The TP value of the data is 288, the FN value of the data is 14, TN is 293 and the FP are five, respectively.

The above ROC curve depicts performance of proposed Xception neural network on the deep fake dataset. Values for FPR are shown along the x-axis and TPR values along the y-axis in the Figure 9. 0.97 is the recommended area under the ROC curve, and random guessing is not accounted for. Moreover, 0.50 is the ROC cutoff value.

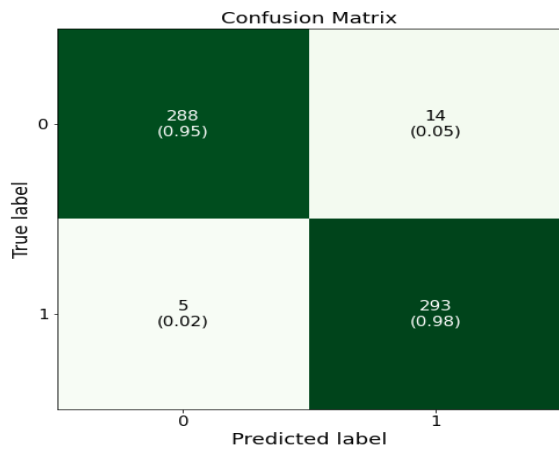


Figure 8. Proposed neural network confusion matrix using deepfake data

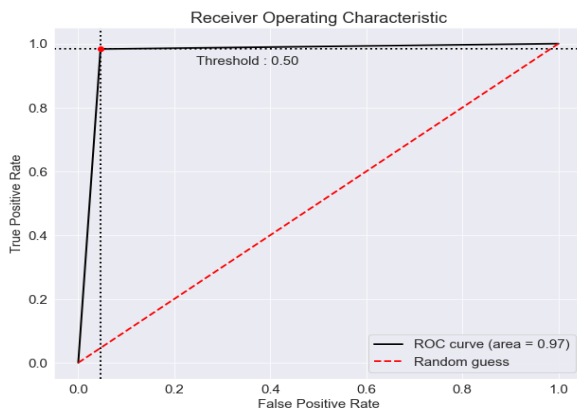


Figure 9. Proposed neural network ROC using deepfake data

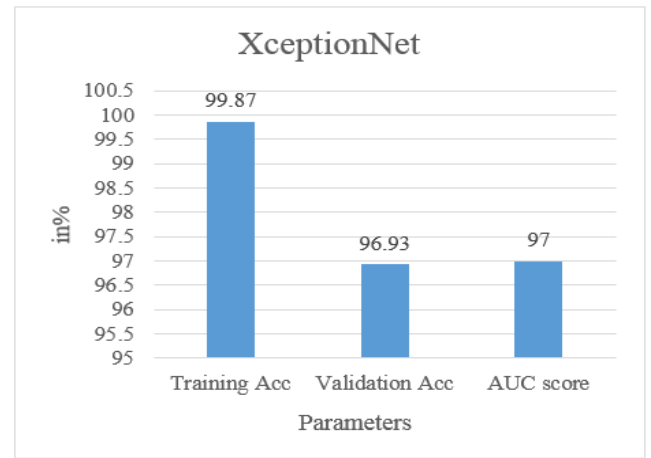


Figure 10. Bar graph of accuracy and AUC score of proposed models

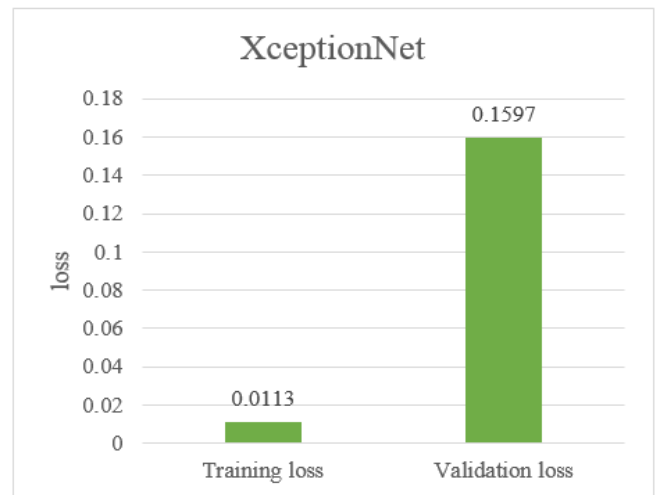


Figure 11. Bar graph of train and Val loss of proposed models

The proposed Xception neural network model's accuracy and area under the curve (AUC) using training and validation data are shown in bar graph form in Figures 10 and 11, respectively. Similarly, Table 6 shows the proposed model's performance using the deep fake dataset. The training set gets 99% accuracy and 0.0113 loss, while the validation set gets 96% accuracy and 0.1597 validation loss, and the AUC of the proposed model is 97%, respectively.

Table 6. Performance of proposed Xception model

Model	Training_loss	Training_Acc	Validation_loss	Validation_Acc	AUC_score
XceptionNet	0.0113	0.9987	0.1597	0.9693	0.97

Table 7. Comparison of base and proposed Xception model

Models	Training_loss	Training_Acc	Validation_loss	Validation_Acc	AUC_score
Base (MLP-CNN) [1]	0.1064	0.9519	0.4317	0.8795	0.87
Base (CNN) [1]	2.4227	0.8473	2.4440	0.8514	0.83
XceptionNet	0.0113	0.9987	0.1597	0.9693	0.97

#### 4.4 Comparative analysis

This section compares the base and proposed model for predicting deep fake videos. We found that our innovative

deep learning-based Xception neural network model on the same dataset (deep fake) outperformed the prior work, which suggested two models (MLP and CNN). This comparative analysis is based on training, validation accuracy/loss, and

AUC score. This accuracy/loss/AUC is described above very briefly.

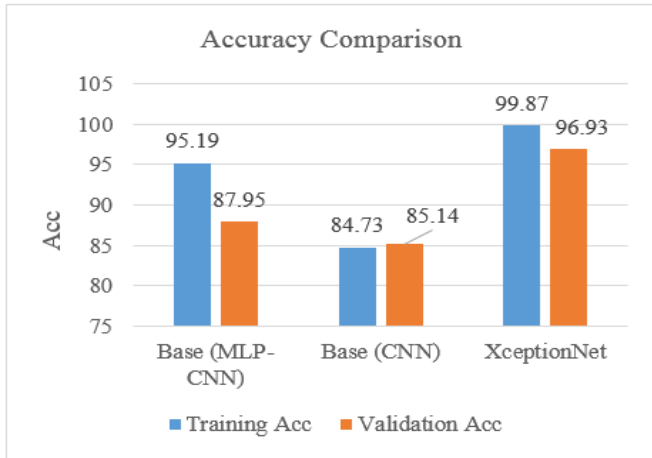


Figure 12. Accuracy comparison between the base and proposed model

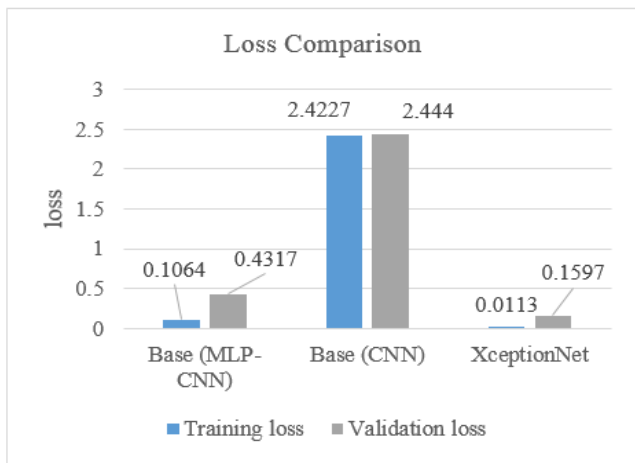


Figure 13. Comparison between base and proposed model loss

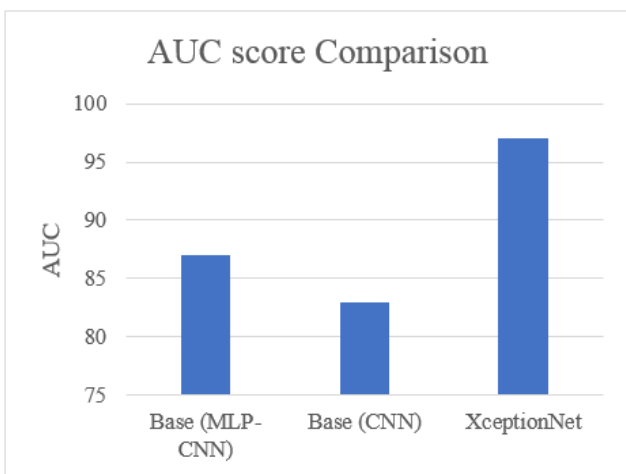


Figure 14. AUC comparison between the base and proposed model

Bar charts depicting the proposed model's accuracy, loss, and AUC relative to the baseline model on training and validation data are shown in Figures 12-14. Table 7 displays

the results of the deep fake dataset model comparison between the baseline and the suggested model. The training set gets 99% accuracy and 0.0113 loss, the validation set gets 96% accuracy and 0.1597, and the AUC of the proposed model is 97%, respectively. While base MLP+CNN model training and validation accuracy is 95% and 87% and loss is 0.1064 of the training set, and validation set loss is 0.4317, the second model CNN obtained train and Val accuracy of 84% and 86% and validation loss of 0.1597/training loss 0.0113. Also, the proposed model AUC curve is 97%, base CNN is 83%, and MLP-CNN AUC is 87%, respectively.

## 5. CONCLUSIONS

The Deep Fakes technology is used in a variety of different sectors. It is still an extremely risky method that threatens the whole world, even though it has certain beneficial applications in the sphere of the film industry, such as maximizing output while using the fewest possible resources. Using a unique approach that combines structured and unstructured data, we suggest a new way to uncover artificial intelligence-created deep fake movies in this work. Our strategy is grounded on studies that reveal that deep fakes like these are made by superimposing a computer-generated visage over a genuine image. Errors like misaligned eyes, lips, and noses, which are seldom visible in actual recordings, are injected into the picture during this process. Facial data is extracted from videos using convolutional neural networks (CNN), however, current approaches don't directly assess the changes across clips. In this study, we propose a strategy for improving outcomes by integrating the substantial feature extraction capabilities of CNN with an appreciation of these disparities. In spite of just having access to a small subset of data, the CNN-Based Xception Net detectors technique achieves a 96.93% accuracy and an AUC score of 0.97 during training. We believe the proposed approach provides an excellent starting point for rapidly reviewing deep fake films with modest computational resources. We believe that the suggested method provides this, while it is not a comprehensive detection tool.

To ensure justice and reduce bias, it may be necessary to expand the scope of this study to find methods to increase the diversity of individuals the model may reliably recognize, like people of color. It is also significant to consider possibility of including more face data that suitably combines spatial and temporal information. In addition to this, it is essential to validate better models using larger and more diverse databases.

## REFERENCES

- [1] Kolagati, S., Priyadarshini, T., Rajam, V.M.A. (2022). Exposing deepfakes using a deep multilayer perceptron-convolutional neural network model. *International Journal of Information Management Data Insights*, 2(1): 100054. <https://doi.org/10.1016/j.ijime.2021.100054>
- [2] Mallet, J., Dave, R., Seliya, N., Vanamala, M. (2022). Using deep learning to detecting deepfakes. *Computer Vision and Pattern Recognition (cs.CV)*, arXiv preprint arXiv:2207.13644. <https://doi.org/10.48550/arXiv.2207.13644>
- [3] Rajpoot, V., Mannepalli, P.K., Choubey, S.B., Sohoni, P., Chaturvedi, P. (2020). A novel approach for weighted

- average filter and guided filter based on tunnel image enhancement. *Journal of Intelligent & Fuzzy Systems*, 39(3): 4597-4616. <https://doi.org/10.3233/JIFS-200551>
- [4] Gunn, D.J., Liu, Z., Dave, R., Yuan, X., Roy, K. (2019). Touch-based active cloud authentication using traditional machine learning and LSTM on a distributed tensorflow framework. *International Journal of Computational Intelligence and Applications*, 18(4): 1950022. <https://doi.org/10.1142/S1469026819500226>
- [5] Ackerson, J.M., Dave, R., Seliya, N. (2021). Applications of recurrent neural network for biometric authentication & anomaly detection. *Information*, 12(7): 272. <https://doi.org/10.3390/info12070272>
- [6] Singh, L.K., Garg, H., Khanna, M. (2021). An artificial intelligence-based smart system for early glaucoma recognition using OCT images. *International Journal of E-Health and Medical Communications (IJEHMC)*, 12(4): 32-59. <https://doi.org/10.4018/IJEHMC.20210701.oa3>
- [7] Rana, M.S., Nobi, M.N., Murali, B., Sung, A.H. (2022). Deepfake detection: A systematic literature review. *IEEE Access*. <https://doi.org/10.1109/ACCESS.2022.3154404>
- [8] Nguyen, T.T., Nguyen, Q.V.H., Nguyen, D.T., Nguyen, D.T., Huynh-The, T., Nahavandi, S., Nguyen, T.T., Pham, Q.V., Nguyen, C.M. (2022). Deep learning for deepfakes creation and detection: A survey. *Computer Vision and Image Understanding*, 223: 103525. <https://doi.org/10.1016/j.cviu.2022.103525>
- [9] Shah, Y., Shah, P., Patel, M., Khamkar, C., Kanani, P. (2020, October). Deep Learning model-based Multimedia forgery detection. In 2020 Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), IEEE, pp. 564-572. <https://doi.org/10.1109/I-SMAC49090.2020.9243530>
- [10] Mehta, V., Gupta, P., Subramanian, R., Dhall, A. (2021). Fakebuster: A deepfakes detection tool for video conferencing scenarios. In 26th International Conference on Intelligent User Interfaces-Companion, pp. 61-63 <https://doi.org/10.1145/3397482.3450726>
- [11] Wodajo, D., Atnafu, S. (2021). Deepfake video detection using convolutional vision transformer. *Computer Vision and Pattern Recognition (cs.CV)*, arXiv preprint arXiv: 2102.11126. <https://doi.org/10.48550/arXiv.2102.11126>
- [12] Albahar, M., Almalki, J. (2019). Deepfakes: Threats and countermeasures systematic review. *Journal of Theoretical and Applied Information Technology*, 97(22): 3242-3250.
- [13] Rafique, R., Nawaz, M., Kibriya, H., Masood, M. (2021). Deepfake detection using error level analysis and deep learning. In 2021 4th International Conference on Computing & Information Sciences (ICCIS), IEEE, pp. 1-4. <https://doi.org/10.1109/ICCIS54243.2021.9676375>
- [14] Li, Y., Lyu, S. (2018). Exposing deepfake videos by detecting face warping artifacts. *Computer Vision and Pattern Recognition (cs.CV)*, arXiv preprint arXiv:1811.00656. <https://doi.org/10.48550/arXiv.1811.00656>
- [15] Yadav, P., Jaswal, I., Maravi, J., Choudhary, V., Khanna, G. DeepFake detection using InceptionResNetV2 and LSTM.
- [16] Tripathi, R.K., Jalal, A.S. (2023). Face recognition under large age gap using age face generation. *International Journal of Biometrics*, 15(2): 233-250. <https://doi.org/10.1504/IJBM.2023.129227>
- [17] Ciftci, U.A., Demir, I., Yin, L. (2020). Fakecatcher: Detection of synthetic portrait videos using biological signals. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. <https://doi.org/10.1109/TPAMI.2020.3009287>
- [18] Jalal, A.S., Sharma, D.K., Sikander, B. (2023). Suspect face retrieval system using multicriteria decision process and deep learning. *Multimedia Tools and Applications*, 1-28. <https://doi.org/10.1007/s11042-023-14968-z>
- [19] Wang, H., Wang, J., Wang, J., Zhao, M., Zhang, W., Zhang, F., Li, W., Xie, X., Guo, M. (2019). Learning graph representation with generative adversarial nets. *IEEE Transactions on Knowledge and Data Engineering*, 33(8): 3090-3103. <https://doi.org/10.1109/TKDE.2019.2961882>
- [20] Mirsky, Y., Lee, W. (2021). The creation and detection of deepfakes. *ACM Computing Surveys*, 54(1): 7. <https://doi.org/10.1145/3425780>
- [21] Tolosana, R., Vera-Rodriguez, R., Fierrez, J., Morales, A., Ortega-Garcia, J. (2020). Deepfakes and beyond: A survey of face manipulation and fake detection. *Information Fusion*, 64: 131-148. <https://doi.org/10.1016/j.inffus.2020.06.014>
- [22] Mallya, A., Wang, T.C., Sapra, K., Liu, M.Y. (2020). World-consistent video-to-video synthesis. In *Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VIII*, Springer International Publishing, 16: pp. 359-378. [https://doi.org/10.1007/978-3-030-58598-3\\_22](https://doi.org/10.1007/978-3-030-58598-3_22)
- [23] Wang, T.C., Liu, M.Y., Zhu, J.Y., Liu, G., Tao, A., Kautz, J., Catanzaro, B. (2018). Video-to-video synthesis. *Computer Vision and Pattern Recognition (cs.CV)*, arXiv preprint arXiv:1808.06601. <https://doi.org/10.48550/arXiv.1808.06601>
- [24] OpenAI. (2020). OpenAI API. OpenAI. <https://openai.com/blog/openai-api/>
- [25] Khodabakhsh, A., Ramachandra, R., Raja, K., Wasnik, P., Busch, C. (2018). Fake face detection methods: Can they be generalized?. In 2018 International Conference of the Biometrics Special Interest Group (BIOSIG), IEEE, pp. 1-6. <https://doi.org/10.23919/BIOSIG.2018.8553251>
- [26] Karras, T., Laine, S., Aila, T. (2019). A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 4217-4228. <https://doi.org/10.1109/TPAMI.2020.2970919>
- [27] Martinez, B., Valstar, M.F., Jiang, B., Pantic, M. (2017). Automatic analysis of facial actions: A survey. *IEEE Transactions on Affective Computing*, 10(3): 325-347. <https://doi.org/10.1109/TAFFC.2017.2731763>
- [28] Almars, A.M. (2021). Deepfakes detection techniques using deep learning: A survey. *Journal of Computer and Communications*, 9(5): 20-35. <https://doi.org/10.4236/jcc.2021.95003>
- [29] Afchar, D., Nozick, V., Yamagishi, J., Echizen, I. (2018). Mesonet: A compact facial video forgery detection network. In 2018 IEEE International Workshop on Information Forensics and Security (WIFS), pp. 1-7. <https://doi.org/10.1109/WIFS.2018.8630761>
- [30] Nguyen, H.H., Yamagishi, J., Echizen, I. (2019). Capsule-forensics: Using capsule networks to detect

- forged images and videos. In ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 2307-2311. <https://doi.org/10.1109/ICASSP.2019.8682602>
- [31] Rana, M.S., Sung, A.H. (2020). Deepfakestack: A deep ensemble-based learning technique for deepfake detection. In 2020 7th IEEE International Conference on Cyber Security and Cloud Computing (CSCloud)/2020 6th IEEE International Conference on Edge Computing and Scalable Cloud (EdgeCom), pp. 70-75. <https://doi.org/10.1109/CSCloud-EdgeCom49738.2020.00021>
- [32] Kim, J., Han, S., Woo, S.S. (2019). Classifying genuine face images from disguised face images. In 2019 IEEE International Conference on Big Data (Big Data), pp. 6248-6250. <https://doi.org/10.1109/BigData47090.2019.9005683>
- [33] Pokroy, A.A., Egorov, A.D. (2021). EfficientNets for deepfake detection: Comparison of pretrained models. In 2021 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (ElConRus), pp. 598-600. <https://doi.org/10.1109/ElConRus51938.2021.9396092>
- [34] Korshunov, P., Marcel, S. (2021). Subjective and objective evaluation of deepfake videos. In ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 2510-2514. <https://doi.org/10.1109/ICASSP39728.2021.9414258>
- [35] Jaiswal, G. (2021). Hybrid recurrent deep learning model for deepfake video detection. In 2021 IEEE 8th Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering (UPCON), pp. 1-5. <https://doi.org/10.1109/UPCON52273.2021.9667632>
- [36] El Rai, M.C., Al Ahmad, H., Gouda, O., Jamal, D., Talib, M.A., Nasir, Q. (2020). Fighting deepfake by residual noise using convolutional neural networks. In 2020 3rd International Conference on Signal Processing and Information Security (ICSPIS), IEEE, pp. 1-4. <https://doi.org/10.1109/ICSPIS51252.2020.9340138>
- [37] Kharbat, F.F., Elamsy, T., Mahmoud, A., Abdullah, R. (2019). Image feature detectors for deepfake video detection. In 2019 IEEE/ACS 16th International Conference on Computer Systems and Applications (AICCSA), pp. 1-4. <https://doi.org/10.1109/AICCSA47632.2019.9035360>
- [38] Ismail, A., Elpeltagy, M.S. Zaki, M., Eldahshan, K. (2021). A new deep learning-based methodology for video deepfake detection using XGBoost. *Sensors*, 21(16): 5413. <https://doi.org/10.3390/s21165413>
- [39] Kaddar, B., Fezza, S.A., Hamidouche, W., Akhtar, Z., Hadid, A. (2021). HCiT: Deepfake video detection using a hybrid model of CNN features and vision transformer. In 2021 International Conference on Visual Communications and Image Processing (VCIP), IEEE, pp. 1-5. <https://doi.org/10.1109/VCIP53242.2021.9675402>
- [40] Dheeraj, J.C., Nandakumar, K., Aditya, A.V., Chethan, B.S. Kartheek, G.C.R. (2021). Detecting deepfakes using deep learning. In 2021 International Conference on Recent Trends on Electronics, Information, Communication & Technology (RTEICT), <https://doi.org/10.1109/RTEICT52294.2021.9573740>
- [41] Tolosana, R., Romero-Tapiador, S., Fierrez, J., Vera-Rodriguez, R. (2021). Deepfakes evolution: Analysis of facial regions and fake detection performance. In Pattern Recognition. ICPR International Workshops and Challenges: Virtual Event, January 10–15, 2021, Proceedings, Part V, Cham: Springer International Publishing, 442-456. [https://doi.org/10.1007/978-3-030-68821-9\\_38](https://doi.org/10.1007/978-3-030-68821-9_38)
- [42] Chollet, F. (2017). Xception: Deep learning with depthwise separable convolutions. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1251-1258. <https://doi.org/10.1109/CVPR.2017.195>
- [43] Rajpoot, V., Dubey, R., Khan, S.S., Maheshwari, S., Dixit, A., Deo, A., Doohan, N.V. (2022). Orchard boumans algorithm and MRF approach based on full threshold segmentation for dental X-ray images. *Traitement du Signal*, 39(2): 737-744. <https://doi.org/10.18280/ts.390239>