International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Estimation and Concealment Deep Fake Detection in Images using Hybrid LSTM

Sunil Kumar Sharma[1, *], Waseem Ahmad Khan[2], Manoj Kumar[3,4,5*], and Rekha Bali[6]

Submitted: 17/09/2023 **Revised**: 16/11/2023 **Accepted**: 28/11/2023

Abstract: The use of deepfake techniques—wherein artificial intelligence (AI) creates influential films of actors acting out fictitious scenarios—could significantly alter how internet users evaluate the veracity of content they encounter. Given that deepfakes may be used maliciously as a source of disinformation, manipulation, harassment, and persuasion, the quality of public discourse and the protection of human rights may be impacted by content creation and modification technology. Detecting falsified media is an ever-evolving, technically complex problem that calls for teamwork from throughout the IT sector and beyond. In the Existing works, DFDC database achieve worse results and more Error Rate occurs, so to overcome this Proposed work is introduced. The proposed work aims to build innovative new technologies that can help detect deepfakes and manipulated media. The deep fake videos were generated using Generative Adversarial Network (GAN) and classified using Hybrid Long Short-Term Memory with Extreme Machine Learning Techniques (HLSTM-ELM). GAN replaces the actual image or video of the person with fake data. The suggested HLSTM-ELM brings out better classification accuracy at a lower computational cost. The comparison of the proposed technique with several Deepfake datasets that obtained results rapidly and with a performance that is better than Existing methods, including an accuracy of 93.84% on the FaceForecics++ dataset, 93.85% on the DFDC dataset, 93.66% on the VDFD dataset, and 93.43% on the Celeb-DF dataset. Our findings suggest that Proposed HLSTM-ELM techniques may be used to construct an efficient system for identifying Deepfakes.

Keywords: *Artificial Intelligence; Deepfake Techniques; Generative Adversarial Network; Hybrid Long Short-Term Memory with Extreme Machine Learning Techniques; Faceforecics++ Dataset*

1. Introduction

The technology used to edit face recordings have developed to the point where it may be impossible for a human to detect whether they have been manipulated. It is simple to swap one person's face in a video with another's, or to edit a person's lips and facial expressions so that they say whatever you want. The potential damage that may emerge from purposely created recordings is easy to imagine: false videos of politicians making silly comments, fake news presenters delivering fake news [1], fake footage of business partners or family members crying for money, etc. These could be destructive to people, communities, and even democracy as a whole. It has been alleged that a pornographic imitation of Indian novelist Rana Ayyub is spreading online. In a piece for Huffington Post [2], she described the video's effect on her career and personal life. The movie was intended to help her calm down, after all. This explains the current spike in attention in "deepfake" video and the difficulty of recognising them. Several models and algorithms have been created, and huge businesses like Google and Facebook are investing extensively in the subject [3]. The purpose of this study is to enhance the conversation in this area. The goal of this study is to report the outcomes of our research into developing a neural network model capable of identifying whether or not a given video has been doctored. The four ways of modifying faces in videos are termed FaceSwap, Deepfakes, Face2Face, and Neural Textures. The model makes use of a network for feature extraction, which will be compared against numerous publically accessible networks, such as GoogleNet, Xception, DenseNet, and a mix of the three. The benchmark and training data from [4] are used to train and test the model.

Using altered faces in media can be hazardous to global peace and security. Facial traits have a crucial role in human interactions and in biometrics-based human identification and identity services [5]. Therefore, it is essential to be able to analyse and recognise faces in still or moving material in order to spot fakes. Numerous studies have looked at various approaches to facial recognition. Among them are BlazeFaces [8], RetinaFaces [9], and Viola-Jones face detectors [6], to mention just a few.

Since then, several techniques have been created for

[1] [1]*Department of Information System, College of Computer and Information Sciences, Majmaah University, Majmaah 11952, Saudi Arabia, Email: s.sharma@mu.edu.sa*
[2]*Department of Mathematics and Natural Sciences, Prince Mohammad Bin Fahd University, P. O. Box: 1664, AL Khobar 31952, Saudi Arabia Email: wkhan1@pmu.edu.sa*
[3]*Faculty of Engineering and Information Sciences, University of Wollongong in Dubai, Dubai Knowledge, Park, Dubai, UAE Email:*
[4]*Research Fellow, INTI International University, Malaysia*
[5]*MEU Research Unit, Middle East University, Amman, 11831, Jordan wss.manojkumar@gmail.com*
[6]*Department of Mathematics, Harcourt Butlor Technical University, Kanpur, 202102, India dr.rekhabali@rediffmail.com*
* *Correspondence: s.sharma@mu.edu.sa*

spotting deepfake movies. Some of these techniques employ recurrence networks to identify visual anomalies inside a frame, while others use convolution networks to identify temporal anomalies over several video face frames [10]. This paper introduces YOLO-InceptionResNetV2-XGBoost (YIX), a novel efficient architecture for determining if a video is genuine or a deepfake. Justification for using all three approaches together In object identification and face recognition systems, the YOLO detector has been proven to be superior than state-of-the-art detectors [11] owing to its attractive trade-off between performance and speed [12,13]. The detection approach is more reliable due to its reduced background noise [14]. Dave et al. [15] investigate whether smart traffic management systems might benefit from using the YOLO detector.

(a) (b) (c) (d)

Fig 1 (a),(b) Deepfake images, (c),(d) Original images (FaceForecics++) [16]

The faces in the WiderFace dataset are categorised using a YOLO-based face recognition method [16-17]. This approach outperforms prior face detectors and is tailored for use in real-time on mobile or embedded devices. It is suggested that YOLO be utilised as a face detector in order to extract people's faces from movies. CNN also claims it will be able to automatically glean the most relevant information from visual media. To this end, we propose a tweaked version of the Convolutional Neural Network (CNN) InceptionResNetV2 as a feature extractor to expose the spatial information discrepancies between different versions of a face in a movie that has been manipulated. Furthermore, the XGBoost model generates results that are on par with the best in the business. It is a machine learning method that is both scalable and versatile, and it does not rely on overfitting.

Chest X-ray images are assessed for COVID-19 and pneumonia using a deep learning-based feature extraction approach using the XGBoost model [18]. Performance-wise, the XGBoost-based technique excels in comparison to other machine learning algorithms. The Softmax activation function [19] is often used in the top densely linked layer of a CNN. In this method, the XGBoost is used to determine if a video is real or a deepfake. This tries to enhance deepfake video identification by combining the benefits of the CNN and XGBoost models, since it is probable that a single model would not be able to identify deepfakes with the requisite accuracy. Furthermore, we will investigate several cutting-edge machine learning techniques including convolutional neural network (CNN) models for face identification. The new proposed hybrid approach, YIX, beats all prior methods across the board on the CelebDF-FaceForencics++ (c23) [20] dataset.

The study's most important findings are as follows: We introduce InceptionResNetV2-XGBoost, a novel model built specifically for learning spatial information features and genuine video detection. Technical problems, such as visual glitches and frame-to-frame differences, make deepfake videos unconvincing. Together, the Generative Adversarial Network (GAN) and the Hybrid Long Short-Term Memory with Extreme Machine Learning Techniques (HLSTM-ELM) classifier at the top of the network, and the trainable extractor InceptionResNetV2 that automatically extracts the important features from video frames, form the proposed model, which improves accuracy. This unique two-stage method ensures precise feature extraction and identification. To boost the precision of deepfake video detection, we use a Viola Jones face detector, especially a tweaked version of Viola Jones v3. AUC, accuracy, specificity, sensitivity, recall, precision, and the F-measure are all used in this article to compare deep-learning and classification algorithms with the purpose of identifying deepfakes.

The remaining sections of the paper are as follows: In Section 2, the authors of the proposal outline the various deepfake datasets now available, as well as the various techniques for creating and spotting deepfake movies. Section 3 of the proposal details an enhanced infrastructure for identifying deepfakes in video. In Section 4, we examine and provide the experimental findings. In Section 5, we discuss our findings and our plans for the future.

2. Literature Survey

It is still exceedingly difficult to distinguish and differentiate these hyper-realistic pictures [23], videos, and audio signals from legitimate unmodified audio-visual, despite the fast development of CNNs [21], Generative Adversarial Networks (GANs) [22], and its variations. A call to action has been issued because to concerns that enemies may utilise technologies like the capacity to generate convincing fake audio, video, and other material to slow down the spread of the threat. As a result, the academic community is always working on new techniques to identify Deepfakes.

2.1 Using Deep Learning to Create Deepfake Videos

The proposed study utilises deep generative models like GANs and Autoencoders (AEs) to construct and synthesis Deepfake [24]. Changing the names of people in media is an essential step in creating a deepfake. One or more of the following techniques may be employed to construct a deepfake: face swapping; puppeteering; lip-syncing; face reenacting; creating synthetic images or videos; speaking in

a computer-generated voice; and so on. FakeAPP [28], the first Deepfake method, relied on a pair of AE networks. Data fed into an AE is reconstructed using a feedforward neural network (FFNN) [29] with an encoder-decoder architecture. The encoder in FakeApp is responsible for encoding the latent facial features, while the decoder is responsible for reconstructing the face pictures. While both AE networks share a single encoder during training, they employ separate decoders when it comes to identifying faces as the source or the target. Since it is very simple to make minute adjustments to the face using techniques like face swapping [30], this is where most Deepfake systems focus. Changing the subject's face in a photograph is one common method of editing. The director is the "puppet master," orchestrating every move of the subject being filmed. Face re-enactment comprises changing a person's appearance, whereas lip-syncing relies on the source individual relocating the mouse in the target video. When creating a Deepfake, it is usual practise to employ feature maps to represent both the real and fake images. Face Action Coding System (FACS) representations, picture segmentation, face landmarks, and facial boundary representations are all examples of feature maps [31]. The FACS taxonomy of facial expression is based on Action Units (AU) and Action Descriptors to accurately capture the full range of human emotion (AD). The eyes, nose, and mouth are among the most recognisable parts of a person's face.

2.1.1 Generation of a Face

Synthesizing new pictures by the manipulation of old ones is the goal of image synthesis [31]. The processes of facial ageing, fractalization, and stance-directed generation all make use of face image synthesis methods. GANs are very effective in synthesising facial features. To generate new data models from existing data samples, generative models like Generic Adversarial Networks (GANs) have been developed [32]. A GAN consists of two adversarial networks, the generating model G and the discriminative model D. Realistic samples can only be produced when the generator and discriminator are in a constant state of competition. The intended output of the generator will have the same data distribution. The discriminator's job is to tell whether a sample is drawn from the model distribution or the data distribution. Generative a priori networks (GANs) for fractalization allow for a 90-degree rotation of faces in either direction. Using the original as a guide, you may replicate an image's expression on a different one. Style GAN [33] and FSGAN [34] are two examples of GAN architectures that produce very lifelike pictures.

2.1.2 Face-Off Deepfake video

The process of replacing a person's face in one photograph with another in which they do not appear is known as "face switching." Digital insertion of famous actors into existing film sequences is commonplace [35]. FaceSwap (itself a face-swapping programme) and ZAO (a Chinese smartphone app that can superimpose any user's face onto any video clip) are two examples of face-swapping synthesisers. In order to accomplish goals like face swapping, face re-enactment, attribute modification, and face component synthesis, researchers often turn to methods like Face Swapping GAN (FSGAN) [36] and Region-Separative GAN (RSGAN) [37]. Training pictures of the source and target faces are recreated using two AEs that share an encoder in Deepfake FaceSwap. Using a face detector and landmark data from the face to do trimming and alignment [38]. The features of the source face may be transferred to the target face using a trained pair of encoder and decoder for the source face. The output of the autoencoder is then combined with the remainder of the picture via Poisson editing [38]. When one person "re-enacts" another's facial expressions, they may experience a shift in their own feelings. A person's true nature takes on the appearance of a puppet when they maintain the same look throughout time [39]. One may "facial expression exchange" with another individual to make them feel the same way as they do [40].

The Face2Face technology allows for the live projection of one person's facial expression onto another. To mimic face shots collected in varying lighting conditions, Face2Face performs a detailed reconstruction between the source and target images.

2.2 Detection of Deepfake Videos Using Deep Learning Methods

There are primarily three types of deepfake detection techniques. The first kind of video analysis incorporates approaches that centre on the actions of the characters (actual or fictional), such as eye-tracking and facial-expression analysis. In the second group, we also include biological markers like blood flow that may be detected in photos alongside the GAN imprint. The third and last group consists of items whose primary function is visual.

Training methods that are interested in visual artefacts need a big data set. Our model fits within the third group. In this piece, we'll take a look at a few of the numerous early prototypes created to spot the telltale signs of Deepfakes. To automatically identify highly realistic fake films made using Deepfake [42] and Face2Face [43], Darius et al. [41] introduced a convolutional neural network (CNN) model called the MesoNet network. Two network topologies (Meso-4 and MesoInception-4) were utilised by the scientists, both of which zoom in on very fine details in images. To exploit the differences introduced by Deepfakes' picture alterations, Yuezun and Siwei [44] designed a CNN architecture (i.e., scaling, rotation, and shearing). Their method for spotting fakes relies on being able to see affine face warping artefacts. The resolution differences caused by face warping may be detected by comparing the Deepfake

face area to the surrounding pixels.

Table 1 Comparison of Existing Methodology with proposed work

Ref	Database used	Accuracy	Method	Disadvantage
Dolhansky et al., (2020) [45]	FaceForencics++	92	The video deepfake detection using a deep learning-based methodology was proposed.	Detection accuracy is low compared to proposed work
Liu et al., (2018) [46]	DFDC dataset	89.23	Automatic deepfake video classification using deep learning was proposed.	The dataset have lesser real and fake data when compared to other methodology
Almutairi et al., (2022) [47]	CelebA	91.50	The deepfake detection based on spectral, spatial, and temporal inconsistencies using multimodal deep learning techniques was proposed.	Consists of one dataset onverification and accuracy achieved is lower
Armanious et al., (2020) [48]	Facebook deepfake challenge dataset	65.27	Medical deepfake image detection based on machine learning and deep learning was proposed.	Accuracy is too low and time duration is more.
Kohli et al., (2021) [49]	annotated CT-GAN	86.34	The deepfake recognition based on human eyewhile blinking pattern using deep learning was proposed.	Time consumption is too high training and testing data.
Pu et al., (2022) [50]	CelebDF and FaceForencics ++	92.34	Machine learning approaches had been analyzed	Machine learning approaches have high rate of False Negative
Proposed	FaceForecics+ +, DFDC, VDFD, Celeb-DF dataset.	93.85	Deep learning hybridization -based classification	Reduced time consumption and increase accuracy rate.

Their main objective was to compare the efficacy of three different deep learning models over a spectrum of classification problems. Therefore, it can be inferred from the literature review that the current works' deep learning accuracy is subpar since it is obtained by a combination of ELM and LSTM approach.

3. Materials and Methods

Deep learning is an AI function that mimics the human brain in many ways, including its ability to recognise and categorise objects in images, translate across languages, and make decisions. Data that has been manually labelled as organised or unstructured may both be used by deep learning AI. Convolutional neural networks, as those used in a Generative Adversarial Network (GAN), are regarded as a subset of deep learning (CNN).

New data generated by a generative amplification network (GAN) is statistically indistinguishable from the original data used in the training process. For example, an image-trained GAN may produce new pictures that, at first glance, seem plausible to humans. The term "deepfakes" was used to describe these fabricated datasets. It is CNN's ability to take an input image, apply learnable weights and biases to a large number of features about the object, and then output a single value that allows for the differentiation of otherwise identical objects that has made it so popular. GAN similarly builds two neural networks, one to create an input and another to distinguish between the sample input and the produced input (deepfakes).

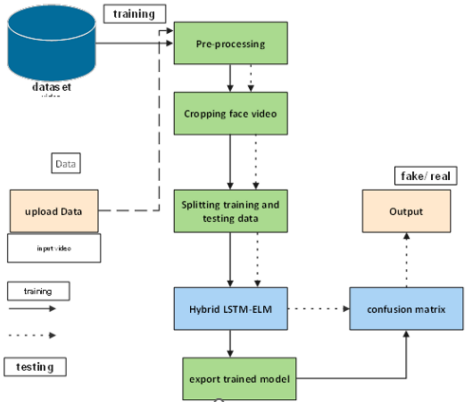


Fig 2 Block Diagram of Proposed work

These Deepfakes are instances of the synthetic media produced by deep learning AI, in which one person's likeness is replaced by that of another in a preexisting image or video. Today's society has a serious issue with those who cannot distinguish between the two. Bad actors have used them to distribute disinformation and distort images. Machines may now produce images so lifelike that human eyes have a hard time distinguishing them from the actual thing. Deepfakes include the video of Barack Obama mocking Donald Trump, Mark Zuckerberg's brag that he controls the stolen data of billions of people, and Jon Snow's tearful apology for the tragic end of Game of Thrones.

Using a GAN to Create Images from a Video Source 3.1

The proposed effort would use Generative Adversarial Network technology to construct deepfakes (GAN). For creating deepfakes using an existing dataset, GAN is currently the preferred method. The encoder, or generator (G), and the decoder, or Discriminator (D), are the two networks that make up a GAN (D). The generator in an adversarial game tries to fool the discriminator by producing fresh data that is statistically very similar to the originals in the training set. The Discriminator is someone who actively seeks for indicators that help them distinguish between genuine and fake sources of data. They work together to learn and train on datasets with several modalities. The generator model is trained to generate compelling images by first generating them using random noise (z). A generator takes a slice of the input random noise and distributes it normally or uniformly in order to produce an image. The generator sends its fake image output to the discriminator, which compares it to the training set's authentic images and learns to tell the difference. Input x is a probability, and D is the likelihood that it is true (x). If x is real, D(x) is 1, and if it's fake, D(x) is 0.

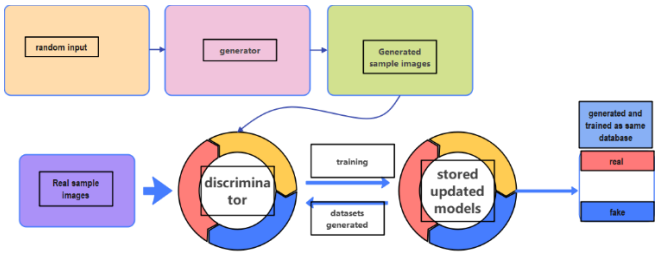


Figure 3 GAN Architecture for deepfake Generation

The core of any image generator is the GAN. A typical GAN network's block diagram is seen in Figure 2. A GAN's generator and discriminator are its fundamental components. The training phase of the proposed work makes use of a data collection x comprising numerous real photographs, x dispersed according to p "data." If x is a real image and z are noise signals with a certain distribution (p z), the generator G will try to produce G(x) that is visually similar to x. While this is going on, the discriminator D tries to distinguish between real and fraudulent images provided by G. D stands for the probability that an input photo is an actual photo rather than a phone-generated photo supplied by G. (input). Input D can definitely be a number between 0 and 1, however. While D is taught to increase the chance that it properly identifies both genuine and fraudulent images, G is trained to decrease the likelihood that its outputs are identified by D to be false photographs. In a two-player minimax game, the value that D and G place on each other's movements may be represented as:

$$\min_G \max_D \mathcal{V}(D,G) \quad = \mathbb{E}_{x\sim p_{\text{data}}(x)}[\log\, D(x)] + \mathbb{E}_{x\prime\sim p_x(x)}[\log\,(1-D(G(x)))] \tag{1}$$

The employment of z-scaling noise by a well trained generator should result in convincing graphics. Simultaneously, the discriminator will get better at distinguishing fake from real photographs.

A GAN consists of four GANs: two generators and two discriminators. The submitted work proposes training GANs using data consisting of multiple images from two different sources. The two groups under discussion will be denoted as X and Y throughout the duration of the planned task. One of the GANs differs from the norm in that it uses examples from domain y rather than domain x for training images. For simplicity, we'll refer to both the GAN's generator (which we'll name G) and the images it generates using the (x) as "suggested work." This GAN is designed to work with domain y. As with the first GAN, images y from domain x are fed into the second GAN, and its generator (denoted as F) is tasked with creating images (denoted as F(y)) that are indistinguishable from those in domain X. The focus of this GAN is on the X-domain. The total GAN loss is indicated by,L-D. Losses from discriminators in a GAN and the proposed work are added together to produce the overall network loss, denoted by,L-D. Deepfake pictures are evaluated based on the Eqn 2 trained input.:

$$\mathcal{L}_{\text{GAN}} = \mathbb{E}_{x\sim \text{ actual }(x)}\left[\left(1 - D_Y\left(G(x)\right)\right)^2\right] + \mathbb{E}_{y\sim \text{ predicted }}(y)[(1 - D_X(F(y)))^2] \quad (2)$$

$$\mathcal{L}_D = \mathbb{E}_{x\sim ptan\,(x)}[D_Y(G(x))^2] + \mathbb{E}_{y\sim \text{ p }}(y)|D_X(F(y))^2| + \mathbb{E}_{x\sim \text{ptan }(x)}[(1 - D_X(x))^2] + \mathbb{E}_{y\sim p\,(y)}[(1 - D_Y(y))^2] \quad (3)$$

$$\mathcal{L}_{\text{identity}} = \mathbb{E}_{y\sim \text{ p }(y)}[\| G(y) - y \|_1] + \mathbb{E}_{x\sim \text{ puana }}(x)[||F(x) - x \|_1] \quad (4)$$

$$\mathcal{L}_{\text{cye}} = \mathbb{E}_{x\sim \text{ p }(x)}[\| F(G(x)) - x \|_1] + \mathrm{E}_{y\rightarrow \text{ plain }(v)}[\| G(F(y)) - y \|_1] \quad (5)$$

For these four losses, $\mathcal{L}_{\text{cyc}}$ and $\mathcal{L}_{\text{identity}}$ are ℓ_1-norm losses, meanwhile $\mathcal{L}_{GAN}$ and $\mathcal{L}_D$ are MSE losses. For simplicity, proposed work combines $\mathcal{L}_{\text{cye}}$, $\mathcal{L}_{\text{identity}}$ and $\mathcal{L}_{\text{GAN}}$ together as a total generator loss $\mathcal{L}_G =$

$$\mathcal{L}_G = \lambda_1\mathcal{L}_{\text{GAN}} + \lambda_2\mathcal{L}_{\text{identity}} + \lambda_3\mathcal{L}_{\text{cye}} \quad (6)$$

where λ_1, λ_2 and λ_3 are coefficients that control relative importances of the three losses. Proposed work also divide $\mathcal{L}_D$ into two sections, i.e. $\mathcal{L}_{D_X}$ and $\mathcal{L}_{D_y}$, to express the losses of the two discriminators respectively:

$$\mathcal{L}_{D_X} = \mathbb{E}_{\text{x }\vee \text{p }(x)}[(1 - D_X(x))^2] + \mathbb{E}_{y\sim \text{p cos }(y)}[D_X(F(y))^2] \quad (7)$$

$$\mathcal{L}_{D_Y} = \mathbb{E}_{y\sim p\,(y)}[(1 - D_Y(y))^2] + \mathbb{E}_{x\sim \text{ p }(x)}[D_Y(G(x))^2] \quad (8)$$

According to Algorithm 1, proposed work first compute and back-propagate L G for each training epoch, then compute and back-propagate $\mathcal{L}_{D_X}$ for each epoch, and finally compute and back-propagate $\mathcal{L}_{D_Y}$ for each epoch, where E stands for the number of training epochs, N stands for the number of training images, and learnRateinit stands for the initial learning rate. The rate of learning will decrease linearly from the first E 0 epochs until it reaches zero in the last epoch which is represented in Algorithm 1. It is not hard to believe a picture created by a GAN. A growing number of neural networks are capable of producing remarkably lifelike recreations of human faces. Since GANs may be used to fabricate dating profiles, "catfish" individuals, and disseminate false information, this poses a serious threat.

Algorithm 1 Algorithm to Train a GAN

1: learnRate ← learnRate init

2. for $e = 0 \rightarrow E - 1$ do

3: for $i = 0 \rightarrow N - 1$ do

4: Calculate $\mathcal{L}_{\text{GAN}}$

5: Back-propagate $\mathcal{L}_G$

6: Calculate $\mathcal{C}_{D_X}$

7: Back-propagate $\mathcal{L}_{D_X}$

8: Calculate $\mathcal{C}_{D_Y}$

9: Back-propagate $\mathcal{L}_{Dv}$

10: end for

11: if $e \geq E_0$ then

12: learnRate ← learnRate $E_{\text{init}} = (1 - (e -$

13: end if

14: end for

The ability of the proposed work and the public at large to distinguish between fake and authentic images is critical for avoiding societal disturbance. GANs excel in their field because they can evaluate their own performance. The network generates new faces independently and then compares them to the training data. If the generator can distinguish between the two, it is given feedback on how to improve its process. Online and social media misinformation concerning terrorist attacks, including the fabrication of purported perpetrators, has the potential to have far-reaching and damaging effects on society.

Because of this, research into ways to detect deepfakes has accelerated. There are now a number of telltale signals of deepfakes that may be spotted with diligence. The examples show that when GANs are used for face training, any background may be utilised. Asymmetry is a significant problem with deepfakes and may take many forms, such as mismatched jewellery (like earrings), misaligned or protruding eyes, and unevenly shaped or coloured ears. Sometimes GANs may misalign the teeth by unnaturally shrinking or stretching each tooth. Messy hair or an unusual hair texture is a quick giveaway for a deepfake. It's feasible that GANs might create wild, unmanageable eyebrow and forehead hair that reaches to the shoulders. Because of the wide variety and complexity of hairdos, representing them with a GAN is one of the most difficult tasks. It's possible to get a hairy effect with even non-hair materials. The progress made by these AI technologies, however, is exponential.

3.2 Pre-Processing Stage

The pictures are made from video recordings. It is crucial that proposed work have a trustworthy method for extracting facial features in light of the popularity of facial modification methods. In order to determine who is in a video, the proposed work employs the tried and true Viola-Jones face detector. Proposed methods increase the detected bounding box for the face by 22 percent relative to its area because the Viola-Jones detector struggles to recognise faces with large bounding boxes. Therefore, there is more space around the face, making it easier to detect deepfakes. After cropping to a size of 224 by 224 pixels, these facial images are restored to their proper proportions.

Although many CNN-based detectors have been developed since their inception, Viola-Jones [46] is the first to use a single neural network to predict bounding boxes and class probabilities from input images simultaneously. To accomplish this, it first divides the image into a grid with M by M by M cells and then looks for the object in each of those cells. The box's predicted coordinate values, confidence scores, and classification outcomes are then predicted for each grid cell. Darknet-53, a hybrid of darknet-19 and ResNet-34, serves as the foundation for the next iteration of viola-jones, dubbed "v3." This network is made up of three 33 convolutional layers followed by one 11 convolutional layer and skip connections. Superior in effectiveness and strength than both ResNet and darknet-19. The viola-jones v3 architecture is the foundation of the face detector used in [47]. In order to acquire adequate small-scale facial features, the darknet-53 backbone network was upgraded by adding further layers to the first two residual blocks. Since the ratios and sizes of anchor boxes are important hyperparameters in object identification, they are also properly enhanced along with the loss function for face recognition.

One of the pre-trained CNN models, InceptionResNetV2, is used to extract the discriminant spatial characteristics for each face shot. Instead of filter concatenation, the InceptionResNetV2 makes use of Inception-style networks with residual connections. Multi-sized convolution layers are combined in the InceptionResNetV2 building block using residual connections[48].

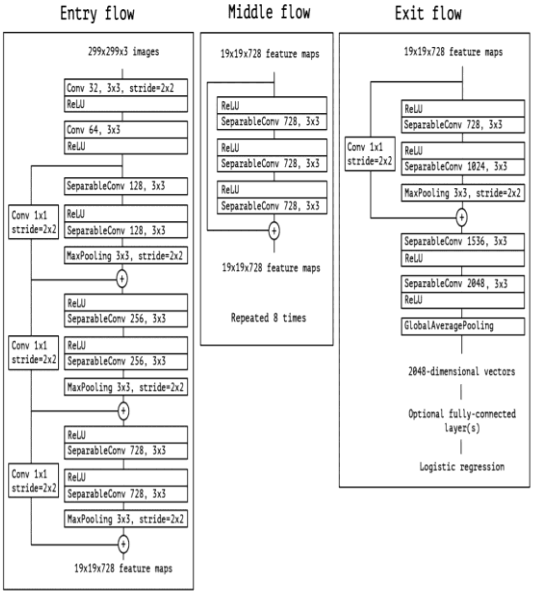


Fig 4 Flowchart of Feature Extraction and Selection using InceptionResNetV2

Pre-trained with ImageNet weights, the InceptionResNetV2 network is used as a baseline model with its final dense layer removed. A global maximum pool layer is then used to fine-tune the initial model by discarding unnecessary information. After that, the network is stacked with a dropout layer between each successive layer, including a few fully connected layers and a rectified linear activation function (ReLU). Using this dropout layer in training can help prevent overfitting [49]. A fully linked output layer is also included as an extra step. With over a thousand categories in the ImageNet dataset, retraining the foundational model with face data enables the first layers to zero down on facial characteristics.

3.3 Hybrid Long Short-Term Memory with Extreme Machine Learning Techniques

Neural networks, if implemented correctly, may be able to make decisions with minimal or no human involvement. The system's permanent memory is made up of experience weights. Since RNNs don't have a way to represent memory explicitly, LSTM networks were created to fill the need. These models are a variant of RNNs that performs well with sequential data, and their memory unit is called a "cell." The proposed work uses an evolutionary approach to investigate LSTM's effectiveness in social media sentiment classification, with a focus on condensed messages and a decentralized representation. Figure 3 depicts an illustration of the algorithm's inner workings.

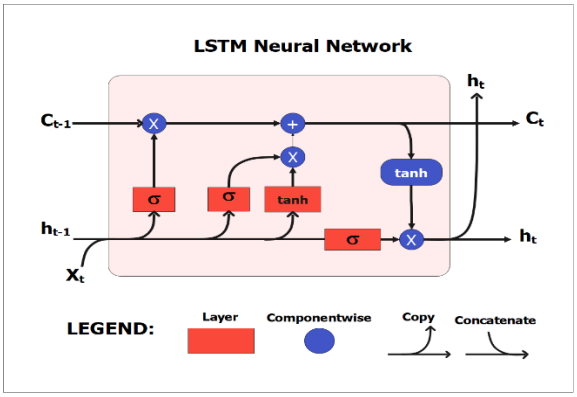


Fig 5 LSTM Neural Network

Figure 3 shows that the LSTM network accepts inputs at nodes C_{t-1}, h_{t-1}, C_t The output vector for this time step is denoted by the letter h_t The h_t output or concealed state is what was passed on from the preceding LSTM unit. And the h_{t-1}, prior unit's memory element or cell state. At is the output of the current unit, and Pt is the memory element of the current unit; the device has two outputs like these. All choices are made after taking into account the current input, the prior output, and the information stored in memory. The memory is refreshed whenever the current output is acquired.

Multiplication includes a "Forget" step, denoted by the letter "σ." Assuming a forget value of '0' will cause 90% of previously stored information to be forgotten. The unit permits the use of some already allocated memory for all other values, including 1, 2, and 3. When summing together both old and new information, piecewise summation makes use of the plus operator. The ' σ ' symbol determines how much of the past is stored in memory. C_{t-1}, becomes C_t, as a consequence of two procedures. Figure 5 depicts the sigmoid and tanh activation functions with forget valves as their output.

Since the second switch incorporates previously-learned information even as it processes fresh data, it is referred to as a "new memory element." The quantity of information stored in memory that is passed on to the next unit is determined by its current input, its previous output, and a bias vector. The Input gate, the Forget gate, and the Output gate are the three gates that make up an LSTM. The activation function "sigmoid" defines these in the interval "0" to "1," where "0" prevents any incoming data and "1" allows for the complete opposite. A successful run of the function should provide a result that agrees with reality.

LSTM is constructed from three individual gates: the input, forget, and output gates. The activation function 'sigmoid' defines these in the range from '0' to'1 ', where" completely prevents all incoming data and'1'does the inverse. It is assumed that the function will return a result that is true.

$$h_t = \sigma(c_t[C_{t-1}, h_t] + x_t). \quad (9)$$

$$f_t = \sigma(w_r[C_{t-1}, h_t] + x_r) \quad (10)$$

$$o_t = \sigma(w_t[C_{t-1}, h_{t-1}] + x_t) \quad (11)$$

The variables w_t , w_r, represent the input, forget, and output gates on the left-hand side of the LHS. The sigmoid activation function is represented by the symbol sigma (. Neuron weights in each gate are represented by the symbol 'w'. Assign the preceding unit's secret state at time t-1 to the variable "C (t-1)". "S t" denotes the input for the current time step. The numbers 0, 1, 2, and 3 in Fig.3's "b" represent the desired biases for the three gates. In addition, the kind of data that the input gate will pass on is defined by Equation 3. The quantity of data from the prior unit that will be forgotten by the current unit is represented by the fourth equation. Time-step-specific activation of the output gate is provided by equation 5.

$$\tilde{C}_t = \tanh(w_c[C_{t-1}, H_t] + x_c) \quad (12)$$

$$C_t = f_t * C_{t-1} + i_t * \bar{C}_t \quad (13)$$

$$A_t = o_t * \tanh(C^t) \quad (14)$$

C_t is the memory information for the current time step and $\bar{C}_t$ denotes the candidate for the present cell state. The symbol '*' indicates the multiplication of given vectors element wise. These LSTM will train the Deep fake features with the combining ELM network which is explained below.

3.3.1 **Extreme Learning Machine**

ELM is a simple and effective Deep learning technique. Input layer, single-hidden layer, and output layer made up the m. Figure 1 shows the basic framework of the ELM model, which consists of j input layer nodes, n hidden layer nodes, m output layer nodes, and an activation function g(x).

The outputs of the hidden layer may be written as (15), and the numerical connection between the output of the hidden layer and the output of the output layer can be stated as (16), for N different samples ($\vec{x}_1, \vec{x}_2, \vec{x}_3 \ldots\ldots \vec{x}_n$):

$$h = g(ax + b) \tag{15}$$

$$h(x_i)V = y_i,\ i = 1,2, \ldots, N. \tag{16}$$

The above equation can be written compactly as

$$HV = Y \tag{17}$$

where,

$$H = \begin{bmatrix} g(\vec{a}_1, b_1, \vec{x}_1) & g(\vec{a}_1, b_1, \vec{x}_2) & \cdots & g(\vec{a}_n, b_n, \vec{x}_N) \\ g(\vec{a}_2, b_2, \vec{x}_1) & g(\vec{a}_2, b_2, \vec{x}_2) & \cdots & g(\vec{a}_n, b_n, \vec{x}_N) \\ \vdots & \vdots & \ddots & \vdots \\ g(\vec{a}_n, b_n, \vec{x}_1) & g(\vec{a}_n, b_n, \vec{x}_2) & \cdots & g(\vec{a}_n, b_n, \vec{x}_N) \end{bmatrix}^T, \tag{18}$$

$$V = \begin{bmatrix} v_1^T \\ v_2^T \\ \vdots \\ v_n^T \end{bmatrix}_{n \times m}, \ Y = \begin{bmatrix} y_1^T \\ y_2^T \\ \vdots \\ y_N^T \end{bmatrix}_{N \times m} \tag{19}$$

where $(\vec{a}_n, b_n, \vec{x}_1)$T are the weights linking the ith input node to the hidden layer, b j is the bias of the ith hidden node, and V_j=[v_1^T, v_2^T, v_3^T,…… v_n^T] are the output values of the ith hidden node. In this case, T represents the weights between the jth hidden node and the output layer. H is the neural network's output matrix. The output weights V may be derived using a sequence of linear equation transformations, given the input weights $\vec{a}_n$ and the bias of the hidden layer b_n.

3.3.2 Hybridization for classification

When making predictions, proposed work may use transfer learning, a strategy in which it constructs models that have already been trained. With transfer learning, it may newly acquire knowledge to use in a predictive capacity where it will do you proud. Using a previously trained network, transfer learning-based fine-tuning approaches re-train a subset of the network using the new dataset. In this part, proposed work examine the inner workings of transfer learning methods used for deepfake detection. Evaluation of deep learning architecture and setup parameters is performed.

HLSTM-ELM Technique

The HLSTM-ELM is a transfer learning-based deep learning network approach often used in the image recognition problem. HLSTM-ELM is an "extreme inception" model. To further on the Recurrent Neural Network framework, we have the HLSTM-ELM model. For its model architecture, HLSTM-ELM makes use of depth-separable convolution layers. The weight serialisation in the HLSTM-ELM model is the smallest. In the HLSTM-ELM model, the convolutional layers total 36.

According to the results, the input layer has a form and the first layer is the output (100, 256, 256, 3). The HLSTM-ELM layers then get engaged, with a total of 20,861,480 parameters and 2048 individual units. The dropout layer is included into the design to avoid model overfitting. The design for translating pixel data into a sequence of one-dimensional arrays makes use of the flattened layers. Prediction of deepfakes requires the use of the architecture's family of dense layers. There are 64 relu-activated units in the dense layer. The sigmoid-activated output layer of the architecture is responsible for the deepfake identification.

Table 2 Parameters used in layers for Training

Parameters	Values
embedding	64000 (None, None, 64)
lstm (HLSTM-ELM)	98816 (None, 128)
dense (Dense)	1290 (None, 10)
encoder (HLSTM-ELM)	33024[(None, 64), (None, 33024)]
decoder (HLSTM-ELM)	33024 (None, 64)

The proposed work suggests a new method for LSTM and ELM deep learning network architectures. In this investigation, we use a novel deep learning network architecture for deepfake detection that combines transfer learning with traditional deep learning techniques. The suggested model architecture combines the ELM and LSTM network layers. It is common practise to use the artificial deep learning network family known as convolutional neural networks for image identification tasks. It's pixel data that convolutional neural networks excel at processing because of their unique architecture. In convolutional neural networks, the images are stored in multidimensional arrays. The main goal of an artificial neural network is to recognise trends in previously unknown data and extrapolate those findings to the future.

The architecture and configuration parameters of the proposed model layer are evaluated. The units and parameters that were considered and considered throughout the development of the suggested model were specified by the configuration parameters.

An overview of the architecture reveals how the suggested method for deepfake detection handles the transition of picture data from input to prediction layers. VGG16 and convolutional neural network layers are combined to form the architecture. The innovative suggested model architecture is a combination of pooling, dropout, flatten, and fully-connected layers.

The combinatorial functionality achieved by the multiplication of data by zero or one, where the multiplier is dynamically determined dependent on input. Then, the neuron multiplied the input signal, x, by the Bernoulli function, (x), where (x)=P(Xx),X-N(0,1) was always the transfer characteristic of the typical normal distribution. Given that cell characteristics tend to exhibit some dispersion, especially when Batch Normalization is used, this probability was chosen. Machine learning, text classification, and speech recognition efficiency were all improved by the GELU variant compared to the ELM and LSTM trained images.

$$\text{HLSTM-ELM: } f(x) = 0.5x\left(1 + \tanh\left[\sqrt{2/\pi}(0.5x + 0.044715x^3)\right]\right) \quad (20)$$

An N-tree ensemble is used in the XGBoost for classification and regression (CARIS). The sum of the scores for each tree's predictions constitutes the final prediction result. Here is the XGBoost model's formula:

$$\widehat{y_i} = \sum_{n=1}^{N} f_n(x_i) \quad (21)$$

where x_i,i=1,...,m, denotes training dataset items, y_i denotes class labels associated with these members, f nF denotes the leaf score for the "n^{th} " tree, and f is the set of all Deep fake. The following is the optimization objective function (obj) formula:

$$\text{obj} = \left(\sum_{i=1}^{m} m(\widehat{y_i}, y_i) + \Omega(f_m)\right) \quad (22)$$

Training differentiable loss function assessing discrepancies between predicted and desired (y i) values is denoted by the term 1. Through the application of the regularization term, the complexity of the model may be kept in check, which in turn aids in the prevention of overfitting. The following is the formula for it:

$$\Omega(f) = \gamma T + \frac{1}{2}\lambda \sum_{h=1}^{T} w_h^2 \quad (23)$$

where T is the total number of leaves in the tree, and are the constants that determine the level of regularization, and w_h is the weight score of leaf h.

Real and deepfake films may be distinguished by feeding the spatial-visual information into the XGBoost recognizer. In order to find the best tree model, the XGBoost is a more efficient and scalable variant of the gradient boosting approach. It was designed to be easily adaptable and very effective. It introduces a parallel tree boosting that efficiently addresses a wide range of issues in the field of data science

4. Experimental Analysis and Results

4.1 Dataset

For this reason, we have used the FaceForecics++ dataset, the DFDC dataset, the VDFD dataset, and the Celeb-DF dataset to test the robustness of the proposed model, since their combination yields a more diverse videos collection, reflecting challenges that may be encountered in practise. The goal here is to make the algorithm used to identify deepfake videos more broadly applicable.

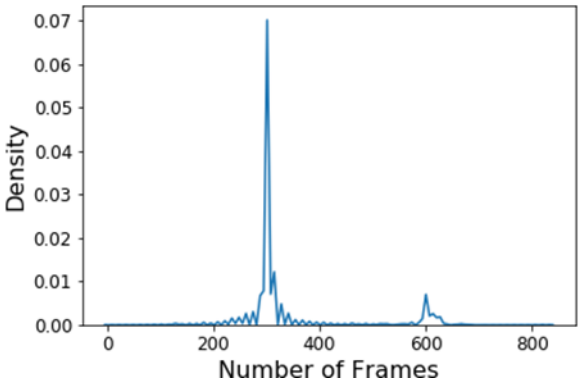


Fig 6 Density of Frames in videos

712 real Celeb-DF videos and 712 false films are used to train the proposed model. The fake videos are from the Celeb-DF fake videos and were chosen at random. All of the actual and false video from the Celeb-DF dataset are combined with 712 videos chosen at random from the FaceForencics++ (c23) dataset. The Celeb-DF test set is often used for evaluation since it contains synthetic films produced with a more advanced version of the deepfake algorithm. This algorithm can create visually stunning videos that are almost indistinguishable from the genuine thing.

The tests were run on an HP OMEN 15-dh0xxx laptop running Windows 10 and equipped with an Intel (R) Core (TM) i7-9750H CPU with 16 GB of RAM and an RTX 2060 GPU with 6 GB of RAM. The suggested model was written in Python and implemented in Python 3.7.4. The proposed model was accomplished with the help of several Python libraries, including Keras, Tensorflow, OpenCV, Sklearn, Xgboost, Numpy, Random, OS, and PIL.

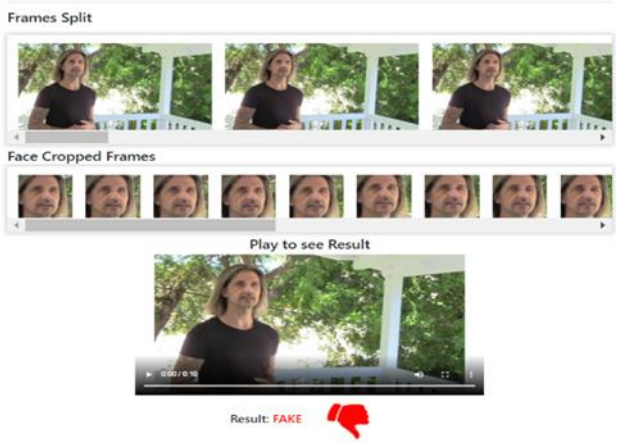


Fig 7 Testing using sample 1 from Dataset

There are four modification methods used to construct the FaceForencics++ dataset, one of which is Deepfakes, which is designed to generate fake faces in video automatically. With respect to each kind of manipulation, it includes 1,000 authentic and 1,000 fictitious videos. Produced in raw, light, and high compression levels. The 890 authentic films in the Celeb-DF dataset were hand-picked from YouTube interviews, while the 5639 deepfake videos were created using a modified deepfake synthesis algorithm. The original data set for deepfake and genuine videos is split into a training set of 5299 and a testing set of 518 (340/178). Due of its complex nature and realistic nature, this dataset is more realistic and difficult to manipulate.

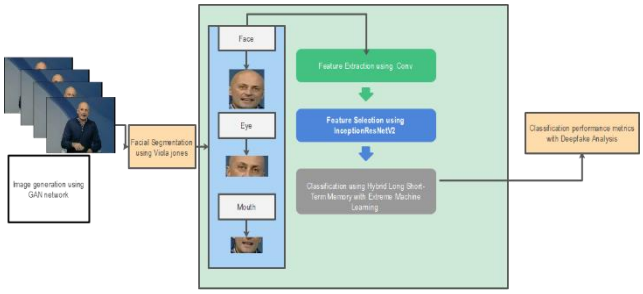


Fig 8 Testing using sample 2 from Dataset

There are four different automated face alteration algorithms used in the FaceForensics++ dataset: Deepfakes, Face2Face, FaceSwap, and NeuralTextures. The dataset contains a total of 1000 original video sequences.

More than a hundred thousand movies make up the DFDC (Deepfake Detection Challenge) dataset, which is used for deepface detection. There are two iterations of the DFDC dataset: 5k video preview dataset. Including not one, but two algorithms for altering one's face. Extensive data collection with 124k videos.

The VFD dataset includes (1) first-person photos of agents, (2) human speaker utterances, (3) speaker eye-gaze locations, and (4) agent verbal and nonverbal replies, all of which have been carefully annotated.

The Celeb-DF dataset consists of 590 authentic films culled from YouTube featuring people of varying ages, ethnicities, and sexes, and 5639 fake videos with similar content.

4.2 Evaluation Measures

An open-source Computer vision library, more specifically Open CV, is responsible for the execution of the face locating module in this particular system. The face location estimate that was worked on in the Open CV library is an implicit viewpoint of a classifier training programme. For the purpose of face locating, the data casing is converted into a diminishing scale. Due to the fact that face recognition is dependent on assisted classifiers with Haar-like components, it is possible that a shape that is similar to a human face will also be recognised, despite the fact that the shape may not be made up of human skin shading pixels. This is because face recognition is dependent on these components. The Eigen face approach is applied in the development of the face coordinating module, which allows for the information picture to be differentiated. Each customer who signs up for the service has six hundred and fourteen different photographs of their face saved in the database with their preset eigen qualities and eigen vectors. At the phase of the technique known as preparation, the standard recorded is referred to in order to determine the appropriate amount of eigen vectors to use when recreating the pictures. Conversation about face course and the turn inspiration driving the face, both of which effect the organizing of the recognizing verification outcome.

For instance, if the customer bends his head to the side with a broad point, the face disclosure may remove the ability to view the face, despite the fact that the skin shading region may still reveal that there is a significant amount of skin shading pixels.

Due to the fact that this structure provides the sensitive biometric arrangement if the customer's face is not visible, it has been discussed before. A sensitive biometric take into account a broad variety of factors, including the shade of the individual's clothes and hair, as well as their body height and weight. Due to the fact that increasing the number of components used as a touch of the structure results in an increase in the check load, only a single sensitive biometric highlight has been selected for the insistence in the system. The part chosen in the sensitive biometric is the shading information from the customer's stomach broaden, for instance, the bits of articles of clothing shade of the customer, because it can be flexible paying little mind to whether or not the customer is wearing a close bit of clothing on the day that was chosen to the structure. The ricocheting box, which is just 2/4 tall and 2/2 wide, was designed to swim on the bottom of the data plot when it rose up out of the data bundle. The decision of the attestation result with the sensitive biometric is made by locating the regular complexity of the histogram on the red, green, and

blue channels between the present bundling and the set away bundling, which contains the most recent saw confront. This is done in order to compare the two bundles.

Commonly used to measure the effectiveness of the proposed deepfake video detection technique is the area under the curve (AUC). How well a binary classifier does its job may be measured by a single number.

Fig 9: Recognized Image

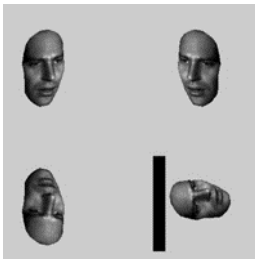


Fig 10: Images at Different Angles

Fig 9 and 10 indicates the recognized image and images at different angle at the key-point detectors. Which not only eliminate the need for annotated data sets but also saves time during the pre-processing stage of the recognition process.

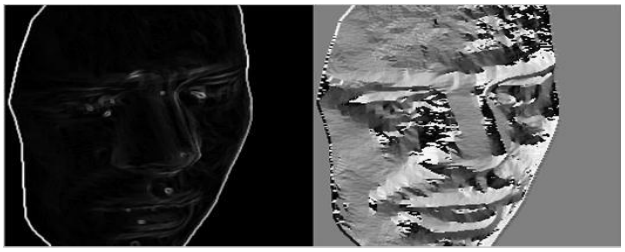


Fig 11: Image of Enhanced Robust Face Matching Process

Fig 12: Final Recognized Face Image

Fig 11 and 12 shows the performance of proposed system using enhanced algorithm And after that real part person face is recognized and the matching process is done.

Table 3 Comparison of existing and proposed system

Parameter	**Chi square matching**	**Co-occurrence Matrix method**	**Enhanced Viola Jones**
Accuracy	70%	64%	89%
Successfully recognized	45%	31%	93%
Matched images	72%	78%	97%
Error in matching	28%	52%	7%

The above table 3 shows the Accuracy value of the proposed enhanced algorithm with existing method of Chi square matching using genetic optimization and Co-occurrence Matrix method.

The AUC is a reliable metric since it is derived from the whole ROC curve for all possible classification cutoffs. Area under the ROC curve is measured in a two-dimensional space, from (0,0) to (1,1). As shown by the ROC curve, there is a cost/benefit relationship between achieving high levels of accuracy and producing many false positives. It is made by graphing the proportion of false positives against the proportion of real positives. The AUC indicates how well the model can determine whether or not a video is genuine. Accuracy, specificity, sensitivity, recall, precision, and the F-measure are the additional metrics used to assess the quality of the suggested model. The formulas for these standards of assessment are as follows.

$$\text{accuracy} = \frac{\text{Total Numberr of TN} + \text{Total Number of TP}}{\text{Sample available in dataset}} \quad (22)$$

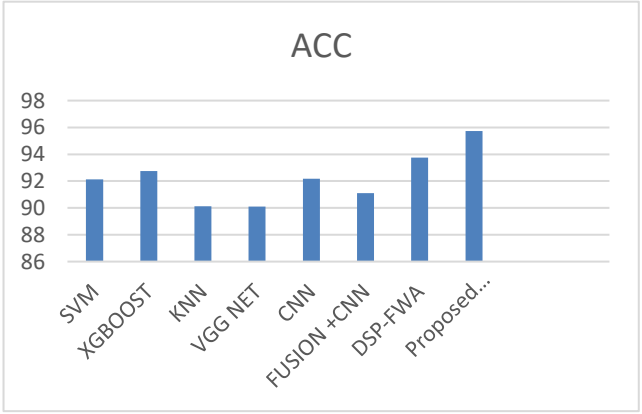
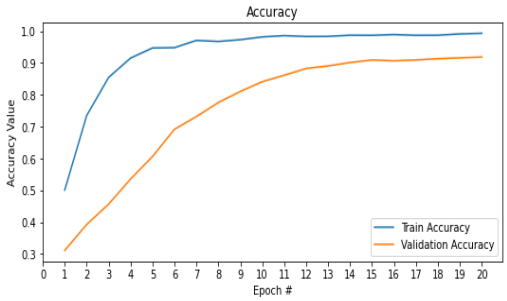
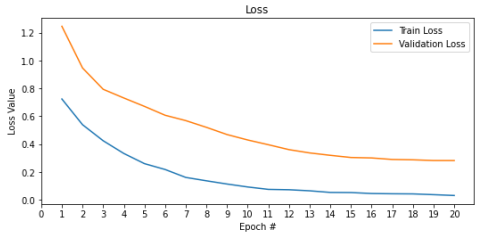


Fig 13 Accuracy of Testing Phase

Figure 13 shows the accuracy of testing phase. The first method uses the HLSTM-ELM classifier to distinguish real from false video based on attributes collected from the proposed model. Learning rate, M estimators, Max depth, Min child weight, Gamma, Subsample, Colsample bytree, Objective, Num class, and Nthread are all adjusted to various degrees. In addition, the validation set correctness of the HLSTM-ELM model is measured using the multiclass log loss (mlogloss) as the evaluation metric.



(a)



(b)

Fig 14 (a) Accuracy Rate (b) Loss Rate

In the comparison, video characteristics are fed into the Support Vector Machine classifiers is shown in figure 14 (a) and figure 14 (b). An SVM takes a data vector as input and maps it to a higher-dimensional feature space, where a hyperplane with the largest possible margin is built. The XGBoost kernel and regularisation C of 10,000 are employed in SVM. The third directly applies the KNN fully linked (dense) layer's Softmax activation function to the task of distinguishing between authentic and spoofed footage. In the fourth case, the central video characteristics recovered by the CNN-VGG NET are fed into a ELM classifier. The ELM is a kind of ensemble learning that takes into account the average prediction score from each individual tree within a multi-tree ensemble. In the ELM, the parameters used are 100 for n estimators and 42 for random state. It adds the RF classifier to the CNN features, making it the fifth. As a boosting ensemble sequential learning approach, AdaBoost adjusts the parameters of each weak classifier based on the misclassified examples of all preceding classifiers. It makes a call based on a combination of the final classifiers' weighted outcome scores. Decision trees are employed as the basic classifier in AdaBoost, and the n estimators' parameter is set at 50.

$$\text{specificity} = \frac{\text{Total Number of TN}}{\text{Total Number of TN} + \text{Total Number of FN}} \quad (23)$$

Fig 15 Specificity Comparison with Existing Algorithm

$$\text{sensitivity} = \frac{\text{Total Number of TP}}{\text{Total Number of FN} + \text{Total Number of Tp}} \quad (24)$$

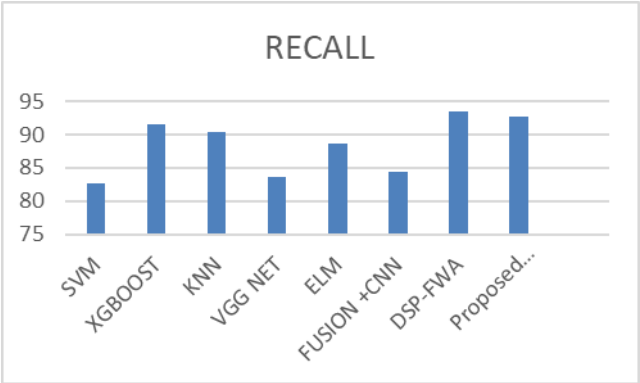


Fig 16 Sensitivity Comparison with Existing Algorithm

$$\text{recall} = \frac{\text{Total Number of TP}}{\text{Total Number of FN} + \text{Total Number of TP}} \quad (25)$$

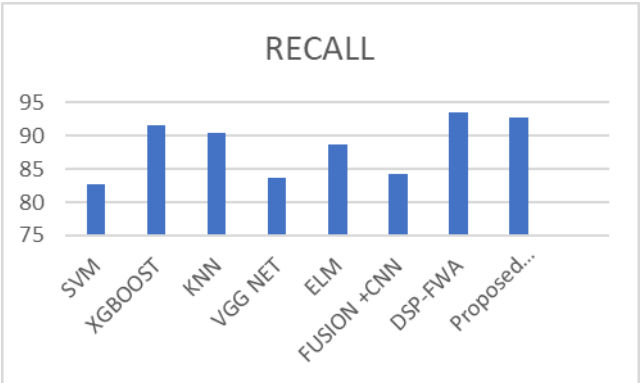


Fig 17 Recall Comparison with Existing Algorithm

$$\text{precision} = \frac{\text{Total Number of TP}}{\text{Total Number of FP} + \text{Total Number of TP}} \quad (26)$$

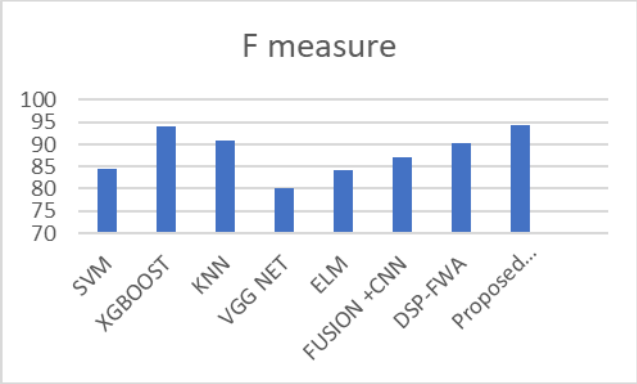


Fig 18 F-measure Comparison with Existing Algorithm

$$F1 - \text{measure} = 2 \times \frac{\text{Recall Rate} \times \text{Precision Rate}}{\text{Recall Rate} + \text{Precision Rate}} \quad (27)$$

Figure 15 to Figure 18 shows the performance metrics of the proposed work such as accuracy, sensitivity, specificity, Recall and F-measure.

As seen from Table 4, the proposed method recorded the highest performance.

Table 4 Performance metrics Comparison based on Dataset

Database	Block size	**HLSTM-ELM**	CNN [51]	Fusion+ CNN [51]	DSP-FWA [52]	SVM	XGBOOST
FaceForecics++	8	**94.1**	90.4	87.67	90.4	92.14	93.14
	12	**90.4**	87.6	89.04	89.0	87.12	89.11
	16	**91.7**	84.9	86.32	89.0	87.48	89.32
DFDC	8	**98.9**	95.3	94.94	91.4	90.14	91.33
	12	**97.8**	95.3	95.14	88.2	91.00	90.07
	16	**95.8**	93.3	93.80	86.4	83.3	83.80
VDFD	8	**93.8**	82.7	82.40	82.5	82.1	81.40
	12	**92.1**	76.0	76.07	70.3	79.9	75.02
	16	**93.0**	75.3	74.64	67.1	77.3	79.53
Celeb-DF	8	**97.0**	85.8	85.04	84.5	89.7	84.02
	12	**92.6**	86.0	84.07	80.1	85.0	83.06
	16	**92.8**	88.4	87.34	77.3	87.3	86.23

Whilst the proposed hybrid method registered an accuracy of 93.84% on the FaceForecics++ dataset, 93.85% on the DFDC dataset, 93.66% on the VDFD dataset, and 93.43% on the Celeb-DF dataset.

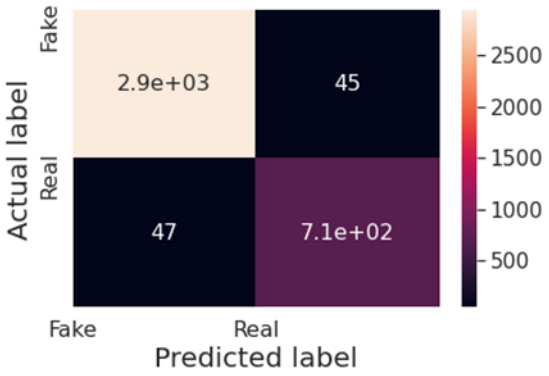
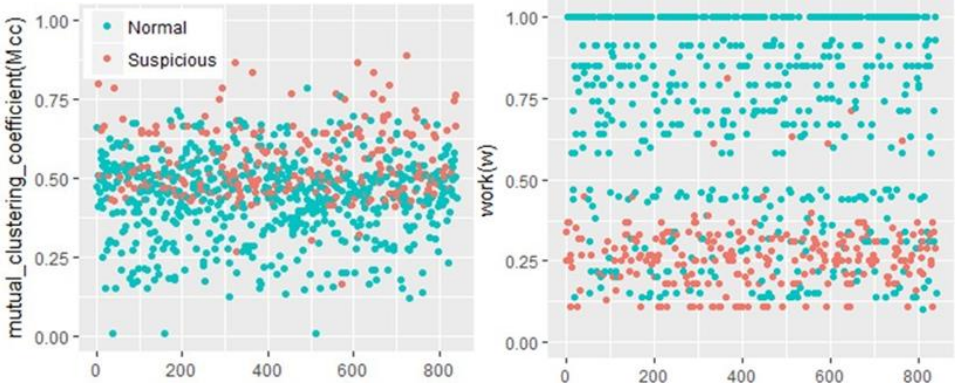


Fig 19 Confusion Matrix of the Proposed work

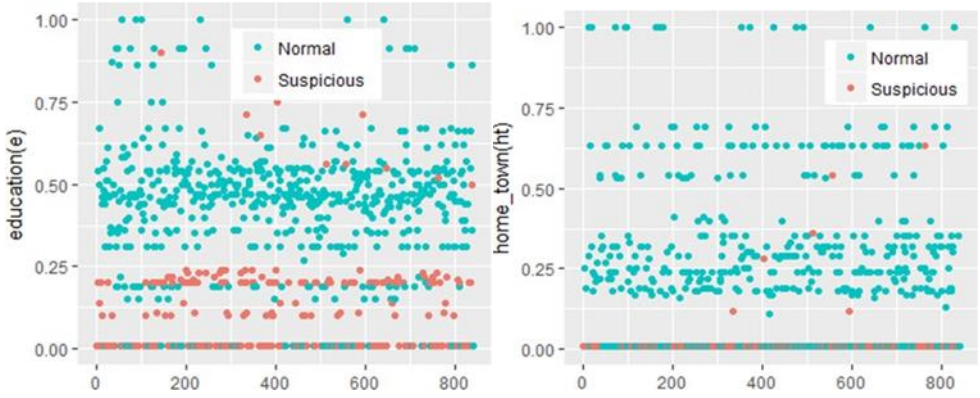
Figure 19 shows that the suggested approach, which is based on XGBoost, outperforms alternatives based on either the dense layer classifier with Softmax function or more conventional machine learning techniques like SVM, RF, and AdaBoost. In addition, Table 3 demonstrates that throughout all tests, the XGBoost classifier outperforms the SVM classifier in terms of AUC on the CelebDF-FaceForencies++ (c23) dataset. This is because when the training dataset size is sufficiently enough, SVM's benefits diminish. To add to this, XGBoost is an ensemble learning technique that uses many decision trees to reach a

conclusion. So, it gets its strength by iterating M estimators times on itself. With thus many decision trees, XGBoost can more easily adapt to the training data and get a deeper understanding of the information contained within. To prevent overfitting, the XGBoost technique limits the size of each tree and the relative importance of its leaves by use of a regularisation term. This broadens the scope in which the model may be used.

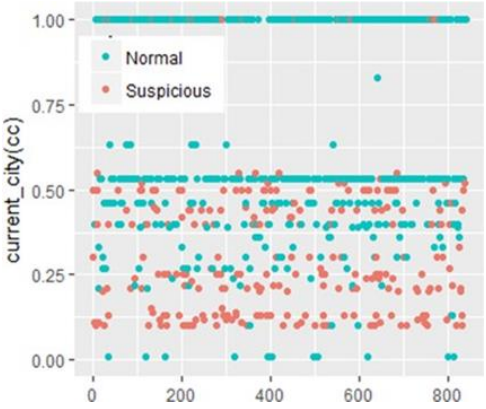
Figure 16 shows that the work similarities between normal (real) links are mostly high, in contrast to suspicions ones. This is because the majority of connections are being established on a network if the two persons know each other offline (either at some educational institute or as a result of this, it has been noted that the users who were linked via the suspicious connection had a job similarity of not more than 0.45, and an educational similarity of not more than 0.25.



a) Mutual Clustering Coefficient (MCC) (b) w*ork* (



c) education (e) *d) current_city (cc)*



e) *home_town (ht)*

Fig 20: Statistical Results

Figure 20 shows the Statistical examination of characteristics for both actual and suspect connections, with the true links indicated in teal and the suspicious links highlighted in orange. In the graph, the similarity score for each characteristic is displayed along the y-axis, and the number of connections is presented along the x-axis

5. Conclusion

In this paper, we provide a novel approach to identifying deepfakes. The Viola Jones face detector with GAN is used in this approach to isolate the visible faces in each frame of video. To aid in the identification of such visual artefacts inside video frames, the InceptionResNetV2 CNN is employed to extract the discriminant spatial features of these faces. In order to differentiate between authentic and deepfake films, these visual characteristics are fed into the Hybrid LSTM-ELM classifier. The models are tested on a combined dataset called CelebDF-FaceForencics++ (c23). Four well-known datasets, FaceForecics++ dataset, DFDC dataset, VDFD dataset, and the Celeb-DF dataset were experimentally analyzed and accuracy is calculated. According to the assessment criteria, the proposed approach has a high detection score. The results show an AUC of 90.62%, a precision of 87.36%, a sensitivity of 85.39%, a recall of 85.39%, a sensitivity of 87.36%, a sensitivity of 86.36%, and an F1-measure of 86.36%. The results of the performance comparisons showed that the suggested strategy is superior to the current best practises. As new ways for creating deepfake videos become available, it is imperative that additional work be done to enhance current detection tools. We plan on adapting several detectors that have proven successful in object identification and applying them to the task of facial recognition. We also want to develop a more robust deep-learning-based deepfake detection approach to keep up with developments in the deepfake-generation process.

Acknowledgements

The authors are thankful to Prince Mohammed bin Fahad Centre for Futuristic studies (Third Futuristic Grant) for this work.

We would like to thank you for following the instructions above very closely in advance.

Data Availability: FaceForecics++: https://www.kaggle.com/datasets/sorokin/faceforensics

DFDC dataset: The DFDC (Deepfake Detection Challenge) https://paperswithcode.com/dataset/dfdc#:~:text=The%20DFDC%20(Deepfake%20Detection%20Challenge,Full%20dataset%2C%20with%20124k%20videos.

VDFD dataset: https://github.com/yahoojapan/VFD-Dataset

Celeb-DF dataset: https://github.com/yuezunli/celeb-deepfakeforensics

Author contributions

Sunil Kumar Sharma: Conceptualization, Methodology Writing-Reviewing and Editing, **Waseem Ahmad Khan:** Data curation, Writing-Original draft preparation, Software, Validation., Writing-Reviewing and Editing **Manoj Kumar:** Visualization, Investigation, Writing-Reviewing and Editing**. Rekha Bali:** Modelling, Data curation, Writing-Reviewing and Editing

Conflicts of interest

The authors declare no conflicts of interest.

References

[1] Mustak, M., Salminen, J., Mäntymäki, M., Rahman, A., & Dwivedi, Y. K., Deepfakes: Deceptions, mitigations, and opportunities. Journal of Business Research, Volume 154, January 2023, 113368, https://doi.org/10.1016/j.jbusres.2022.113368.

[2] A. Mehra, A. Agarwal, M. Vatsa, and R. Singh, Motion Magnified 3D Residual-in-Dense Network for DeepFake Detection, IEEE Transactions on Biometrics, Behavior, and Identity Science (IEEE T-BIOM), 2023, volume 5, pp 39-52. https://doi.org/10.1109/tbiom.2022.3201887

[3] Mohiuddin, S., Sheikh, K. H., Malakar, S., Velásquez, J. D., & Sarkar, R. (2023). A hierarchical feature selection strategy for deepfake video detection. Neural Computing and Applications, 1-18.

[4] Zhao, Cairong, Chutian Wang, Guosheng Hu, Haonan Chen, Chun Liu, and Jinhui Tang. "ISTVT: Interpretable Spatial-Temporal Video Transformer for Deepfake Detection." IEEE Transactions on Information Forensics and Security (2023).

[5] Ke, Jianpeng, and Lina Wang. "DF-UDetector: An effective method towards robust deepfake detection via feature restoration." Neural Networks (2023).

[6] Rana, Md Shohel, Mohammad Nur Nobi, Beddhu Murali, and Andrew H. Sung. "Deepfake detection: A systematic literature review." IEEE Access (2022).

[7] Almutairi, Zaynab, and Hebah Elgibreen. "A Review of Modern Audio Deepfake Detection Methods: Challenges and Future Directions." Algorithms 15, no. 5 (2022): 155.

[8] Malik, Asad, Minoru Kuribayashi, Sani M. Abdullahi, and Ahmad Neyaz Khan. "DeepFake detection for human face images and videos: A survey." IEEE Access 10 (2022): 18757-18775.

[9] Nirkin, Yuval, Lior Wolf, Yosi Keller, and Tal Hassner. "Deepfake detection based on discrepancies between

faces and their context." IEEE Transactions on Pattern Analysis and Machine Intelligence 44, no. 10 (2021): 6111-6121.

[10] Yu, Peipeng, Zhihua Xia, Jianwei Fei, and Yujiang Lu. "A survey on deepfake video detection." Iet Biometrics 10, no. 6 (2021): 607-624.

[11] Chen, Tianxiang, Avrosh Kumar, Parav Nagarsheth, Ganesh Sivaraman, and Elie Khoury. "Generalization of Audio Deepfake Detection." In Odyssey, pp. 132-137. 2020.

[12] Chintha, Akash, Bao Thai, Saniat Javid Sohrawardi, Kartavya Bhatt, Andrea Hickerson, Matthew Wright, and Raymond Ptucha. "Recurrent convolutional structures for audio spoof and video deepfake detection." IEEE Journal of Selected Topics in Signal Processing 14, no. 5 (2020): 1024-1037.

[13] Malik, Asad, Minoru Kuribayashi, Sani M. Abdullahi, and Ahmad Neyaz Khan. "DeepFake detection for human face images and videos: A survey." IEEE Access 10 (2022): 18757-18775.

[14] Hussain, Shehzeen, Paarth Neekhara, Malhar Jere, Farinaz Koushanfar, and Julian McAuley. "Adversarial deepfakes: Evaluating vulnerability of deepfake detectors to adversarial examples." In Proceedings of the IEEE/CVF winter conference on applications of computer vision, pp. 3348-3357. 2021.

[15] Anay Dave, Sethi, Lastika, Raj Bhagwani, and Ameyaa Biwalkar. "Video security against deepfakes and other forgeries." Journal of Discrete Mathematical Sciences and Cryptography 23, no. 2 (2020): 349-363.

[16] Ahmed, Saadaldeen Rashid Ahmed, and Emrullah Sonuç. "Deepfake detection using rationale-augmented convolutional neural network." Applied Nanoscience (2021): 1-9.

[17] Hui, K.; Wang, J.; He, H.; Ip, W.H. A multilevel single stage network for face detection. Wirel. Commun. Mob. Comput. 2021, 2021.

[18] Zhang, Tao. "Deepfake generation and detection, a survey." Multimedia Tools and Applications 81, no. 5 (2022): 6259-6276.

[19] Ismail, Aya, Marwa Elpeltagy, Mervat S. Zaki, and Kamal Eldahshan. "A new deep learning-based methodology for video deepfake detection using XGBoost." Sensors 21, no. 16 (2021): 5413.

[20] Abdulreda, Ahmed S., and Ahmed J. Obaid. "A landscape view of deepfake techniques and detection methods." International Journal of Nonlinear Analysis and Applications 13, no. 1 (2022): 745-755 .

[21] Caldelli, Roberto, Leonardo Galteri, Irene Amerini, and Alberto Del Bimbo. "Optical Flow based CNN for detection of unlearnt deepfake manipulations." Pattern Recognition Letters 146 (2021): 31-37.

[22] Jung, Tackhyun, Sangwon Kim, and Keecheon Kim. "Deepvision: Deepfakes detection using human eye blinking pattern." IEEE Access 8 (2020): 83144-83154.

[23] Liu, Jiarui, Kaiman Zhu, Wei Lu, Xiangyang Luo, and Xianfeng Zhao. "A lightweight 3D convolutional neural network for deepfake detection." International Journal of Intelligent Systems 36, no. 9 (2021): 4990-5004..

[24] Katarya, Rahul, and Anushka Lal. "A study on combating emerging threat of deepfake weaponization." In 2020 Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC), pp. 485-490. IEEE, 2020.

[25] Hancock, Jeffrey T., and Jeremy N. Bailenson. "The social impact of deepfakes." Cyberpsychology, behavior, and social networking 24, no. 3 (2021): 149-152.

[26] Nickabadi, Ahmad, Maryam Saeedi Fard, Nastaran Moradzadeh Farid, and Najmeh Mohammadbagheri. "A comprehensive survey on semantic facial attribute editing using generative adversarial networks." arXiv preprint arXiv:2205.10587 (2022).

[27] Chadha, Anupama, Vaibhav Kumar, Sonu Kashyap, and Mayank Gupta. "Deepfake: An Overview." In Proceedings of Second International Conference on Computing, Communications, and Cyber-Security: IC4S 2020, pp. 557-566. Springer Singapore, 2021.

[28] Mahmud, Bahar Uddin, and Afsana Sharmin. "Deep insights of deepfake technology: A review." arXiv preprint arXiv:2105.00192 (2021).

[29] Abu-Ein, Ashraf A., Obaida M. Al-Hazaimeh, Alaa M. Dawood, and Andraws I. Swidan. "Analysis of the current state of deepfake techniques-creation and detection methods." Indonesian Journal of Electrical Engineering and Computer Science 28, no. 3 (2022): 1659-1667.

[30] Solaiyappan, Siddharth, and Yuxin Wen. "Machine learning based medical image deepfake detection: A comparative study." Machine Learning with Applications 8 (2022): 100298..

[31] Taeb, Maryam, and Hongmei Chi. "Comparison of Deepfake Detection Techniques through Deep Learning." Journal of Cybersecurity and Privacy 2, no. 1 (2022): 89-106..

[32] Elhassan, Ammar, Mohammad Al-Fawa'reh, Mousa Tayseer Jafar, Mohammad Ababneh, and Shifaa Tayseer Jafar. "DFT-MF: Enhanced deepfake detection using mouth movement and transfer learning." SoftwareX 19 (2022): 101115.

[33] Khormali, Aminollah, and Jiann-Shiun Yuan. "Add: attention-based deepfake detection approach." Big Data and Cognitive Computing 5, no. 4 (2021): 49. Nirkin, Yuval, Lior Wolf, Yosi Keller, and Tal Hassner. "Deepfake detection based on discrepancies between faces and their context." IEEE Transactions on Pattern Analysis and Machine Intelligence 44, no. 10 (2021): 6111-6121.

[34] Westerlund, Mika. "The emergence of deepfake technology: A review." Technology innovation management review 9, no. 11 (2019).

[35] Fagni, Tiziano, Fabrizio Falchi, Margherita Gambini, Antonio Martella, and Maurizio Tesconi. "TweepFake: About detecting deepfake tweets." Plos one 16, no. 5 (2021): e0251415.

[36] Nguyen, T.T.; Nguyen, C.M.; Nguyen, D.T.; Nguyen, D.T.; Nahavandi, S. Deep learning for deepfakes creation and detection. arXiv 2019, arXiv:1909.11573

[37] Laishram, Lamyanba, Md Maklachur Rahman, and Soon Ki Jung. "Challenges and Applications of Face Deepfake." In Frontiers of Computer Vision: 27th International Workshop, IW-FCV 2021, Daegu, South Korea, February 22–23, 2021, Revised Selected Papers 27, pp. 131-156. Springer International Publishing, 2021.

[38] Abu-Ein, Ashraf A., Obaida M. Al-Hazaimeh, Alaa M. Dawood, and Andraws I. Swidan. "Analysis of the current state of deepfake techniques-creation and detection methods." Indonesian Journal of Electrical Engineering and Computer Science 28, no. 3 (2022): 1659-1667.

[39] Fletcher, John. "Deepfakes, artificial intelligence, and some kind of dystopia: The new faces of online post-fact performance." Theatre Journal 70, no. 4 (2018): 455-471.

[40] Darius Afchar, Vincent Nozick, Junichi Yamagishi, and Isao Echizen. MesoNet: a Compact Facial Video Forgery Detection Network. pages 1–7, 2018

[41] Shruti Agarwal, Hany Farid, Yuming Gu, Mingming He, Koki Nagano, and Hao Li. Protecting World Leaders Against Deep Fakes. In CVPR Workshops, 2019.

[42] Charu C. Aggarwal. Neural Networks and Deep Learning: A Textbook. Springer International Publishing, Switzerland, 2020

[43] Yuezun Li and Siwei Lyu. Exposing DeepFake Videos By Detecting Face Warping Artifacts. arXiv preprint arXiv:1811.00656v3, 2019

[44] Rössler, Andreas, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, and Matthias Nießner. "FaceForensics++: Learning to Detect Manipulated Facial Images. arXiv (2019)." (2019).

[45] Dolhansky, Brian, Joanna Bitton, Ben Pflaum, Jikuo Lu, Russ Howes, Menglin Wang, and Cristian Canton Ferrer. "The deepfake detection challenge (dfdc) dataset." arXiv preprint arXiv:2006.07397 (2020).

[46] Liu, Ziwei, Ping Luo, Xiaogang Wang, and Xiaoou Tang. "Large-scale celebfaces attributes (celeba) dataset." Retrieved August 15, no. 2018 (2018): 11.

[47] Almutairi, Zaynab, and Hebah Elgibreen. "A Review of Modern Audio Deepfake Detection Methods: Challenges and Future Directions." Algorithms 15, no. 5 (2022): 155.

[48] Armanious, Karim, Tobias Hepp, Thomas Küstner, Helmut Dittmann, Konstantin Nikolaou, Christian La Fougère, Bin Yang, and Sergios Gatidis. "Independent attenuation correction of whole body [18F] FDG-PET using a deep learning approach with Generative Adversarial Networks." EJNMMI research 10, no. 1 (2020): 1-9.

[49] Kohli, Aditi, and Abhinav Gupta. "Detecting DeepFake, FaceSwap and Face2Face facial forgeries using frequency CNN." Multimedia Tools and Applications 80 (2021): 18461-18478.

[50] Pu, Wenbo, Jing Hu, Xin Wang, Yuezun Li, Shu Hu, Bin Zhu, Rui Song, Qi Song, Xi Wu, and Siwei Lyu. "Learning a deep dual-level network for robust DeepFake detection." Pattern Recognition 130 (2022): 108832.

[51] Tolosana, R., Romero-Tapiador, S., Vera-Rodriguez, R., Gonzalez-Sosa, E., & Fierrez, J. (2022). DeepFakes detection across generations: Analysis of facial regions, fusion, and performance evaluation. Engineering Applications of Artificial Intelligence, 110, 104673