DEEPFAKE IN PICTURE USING CONVOLUTIONAL NEURAL NETWORK

A Report for the ETE Review of Project - III

S. No	Enrolment Number	Admission Number	Student Name	Degree / Branch	Sem
1	19021011513	19SCSE1010328	ASHUTOSH GAUTAM	B.Tech /CSE	V
2	19021011241	19SCSE1010030	HARSH TOMAR	B.Tech / CSE	V

Under the Supervision of

Prof. SURENDRA SINGH CHAUHAN

GALGOTIAS UNIVERSITY

School of Computing Science and Engineering
Greater Noida, Uttar Pradesh
Fall 2020 - 2021

Table of Content

Title		Page No.
Candidates Declaration		3
Acknowledgement		4
Abstract		5
List of Table		6
List of Figures		7
Acronyms		
Chapter 1	Introduction	8
	1.1 REQUIREMENT OF DEEPFAKE IN PICTURE	8-9
	1.2 DISADVANTAGE OF CURRENT SYSTEM	10-11
	1.3 MERITS OF PROPOSED SYSTEM	11-13
	1.4 TOOLS AND TECHNOLOGY USED	15
Chapter 2	Literature Survey/Project Design	16-21
Chapter 3	Functionality/Working of Project	22-26
Chapter 4	Results and Discussion	27
Chapter 5	Conclusion and Future Scope	28-30
	5.1 Conclusion	
	5.2 Future Scope	
	Reference	31

CANDIDATE’S DECLARATION

We hereby certify that the work which is being presented in the thesis/project/dissertation, entitled **“Deepfake in picture using convolutional neural network”** in partial fulfillment of the requirements for the award of the Project review submitted in the School of Computing Science and Engineering of Galgotias University, Greater Noida, is an original work carried out during the period of month, Year to Month and Year, under the supervision of Mr. Surendra Singh Chauhan Department of Computer Science and Engineering/Computer Application and Information and Science, of School of Computing Science and Engineering , Galgotias University, Greater Noida

The matter presented in the thesis/project/dissertation has not been submitted by me/us for the award of any other degree of this or any other places.

Ashutosh Gautam

19SCSE1010328

This is to certify that the above statement made by the candidates is correct to the best of my knowledge.

Mr. Surendra Singh Chauhan

AP/SCSE

Galgotias University

ACKNOWLEDGMENT

I am sincerely thankful to Galgotias University, Greater Noida for providing me with the opportunity to write a research paper on the topic **“Deepfake in the picture using convolutional neural network”.**

We would like to thank our guide **Mr. Surendra Singh Chauhan** for guiding us in every stage of this project. Without his support, it would have been very difficult for me to prepare so meaningful and interesting project.

This paper has helped me a lot learn about machine learning and deep learning and I hope it helps people to have a basic understanding of deepfake in pictures and its detection.

Abstract

Deepfake is the combination of fake images and videos and deep learning technology. Deepfake has been derived from deep learning. The neoteric advancement in the field of Artificial Intelligence and Machine Learning allows the immoral practice of the generation of forged images and videos. Nowadays it is very easy to create photo-realistic images by using generative adversarial networks. These forges images and videos are roaming on the internet and social media. It is very difficult to find which of them is real or fake. Most of the time these images are created with the purpose to create a social disturbance, political disturbance, or conveying false information to the public. These images will look real so people will get easily conveyed by them. According to recent surveys, it has been found that deepfake is increasingly affecting individuals, society communities, organizations, security, religion, and democracy. The quality of these videos and images is remarkable and their reach is enormous on social media. Its repercussions are devastating beyond one's imagination. This paper lists a comprehensive overview of the various deep-fake techniques. Therefore, the purpose of this paper is to detect these fake images using a conventional neural network. This paper aims to solve this problem in that we train a model for some datasets and create deep-fakes and that model tries to detect the deep fake. In the training procedure, it needs an original and fake image to train the model.

LIST OF TABLES

S.NO	CAPTION	PAGE NO.
1	TABLE 1 (SUMMARY OF DEEPFAKE TOOLS)	13
2	TABLE 2 (SUMMARY OF PROMINENT DEEPFAKE DETECTION METHOD)	19

LIST OF FIGURES

S.NO.	TITLE	PAGE NO.
1	Deep Fake Impact	14
2	Image Manipulation	15
3	Flow Chart	22
4	Face Augmentation sample	20
5	Pictorial Model	22

ACRONYMS

SVM	Support Vector Machine
ML	Machine Learning
DT	Decision Tree
KNN	K-Nearest Neighbour
LR	Logistic Regression
RF	Random Forrest
CNN	Convolutional Neural Network
RNN	Recurrent Neural Networks
CONVNET	Convolutional Neural Network

1. Introduction

DeepFake is a type of artificial intelligence which is used to produce convincing pictures, audio, and video forgeries. The main methods used to construct DeepFakes are based on deep learning and correlate training Generative Neural Networks (GANs) [1] architectures. Generative Adversarial Networks (GAN) are deep learning techniques for training generative models, which are most commonly used for the generation of synthetic images. The GAN model architecture involves two sub-models: a generator model for generating new examples and a discriminator model for classifying whether the generated examples are real or fake. The growth of GAN lead to the development of a series of applications and sophisticated techniques, such as face swapping, face manipulation, and face synthesis, resulting in a rapidly increasing number of fake videos with accurate quality and more complexity. The results of the DeepFake generation have become increasingly realistic in recent years, making it harder to isolate the real from the fake for the normal eyes. Multimedia content that has been tampered with is increasingly being utilized in a variety of cybercrime operations (also mentioned in Ferreira et al. research [2]). Fake news, disinformation, digital kidnapping, and ransomware-related crimes are only a few

of the most common crimes perpetrated and disseminated using altered digital pictures and videos.

1.1 REQUIREMENT OF DEEPFAKE IN PICTURE

DeepFake detection solutions usually use multimodal detection approaches to evalu- ate whether target material has been altered or created synthetically. Existing detection approaches often focus on developing AI-based algorithms in algorithmic detection meth- ods such as Vision Transformer , two-stream neural network [5], MesoNet (which is proposed by Afchar et al.), etc. However, less attention is paid to manual image process- ing to focus on highlighting the important regions of an image. This often results in the model having to process all of the videos, making the model heavier. In order to improve the DeepFake detection approach, we used both a manual processing and an AI-based algorithm in this research. The most important information, regions, and features will be carefully focused and processed before being put into deep neural networks. Concentrating on the most relevant elements to learn not only reduces the needless learning burden on these networks, but also improves the overall model's accuracy.

The major concept of our article will be to take advantage of few most popular classification models to identify fake videos and show how to transform the DeepFake detection into a simpler classification problem. As the existing classification models are designed for high accuracy, the reasonable selection of these models will also increase the ability to solve the problem of DeepFake detection. We also propose processing methods to convert the input sequences of DeepFake detection into the inputs of a basic classification model with two classes (class “1” for real video and class “0” for fake video). Our proposal also accommodates the processing steps to avoid losing essential features and supports synthesizing afterwards.

Face Forgery Generation

Face forgery generation is one of the fields of image synthesis. The objective is to create new faces using generative adversarial networks (GANs) [1]. The most popular approach is StyleGAN [7], which makes it possible to control the image synthesis via scale-specific modifications to the styles. Even with the growth of StyleGAN2 [8], which is based on data-driven unconditional generative image modeling, non-existent lifelike faces can be made with near-real sophistication and are often indistinguishable fake people who don’t actually exist in real life. An application to create non-existent lifelike face of peoples using StyleGAN2 is mentioned in this tool [9]. The algorithms underpinning the AI are trained on publicly available pictures before being asked to generate fresh variants that satisfy the requisite level of realism. In addition, many synthetic programs [10] are available as open source and may be used by anyone.

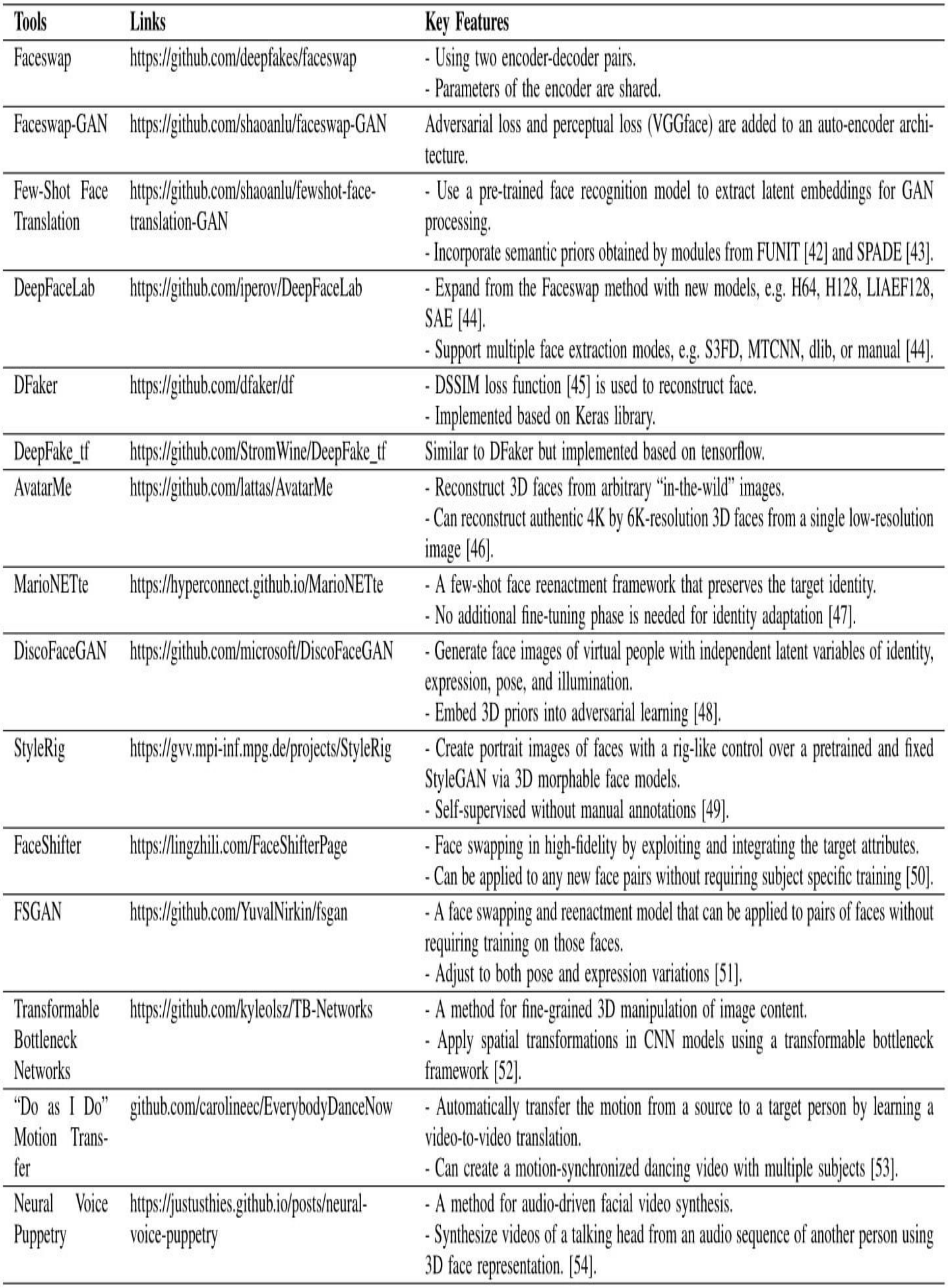
Face swapping is the most common type of face modification currently. The DeepFake face swap method is built on two Auto-encoders, where one common encoder is used in training and rebuilding sources and another one to target face training pictures, respectively. The aim of face swapping is to generate a new fake image or video of a person after swapping its face. Presently, a variety of approaches have been proposed. Some prominent methods can be mentioned such as RSGAN, FSGAN, DCGAN, PGGAN, FSNet, High Fidelity Identity Swapping, and StarGAN v2 [18]. A lot of datasets were created based on face swaps, of which the standout is the DeepFake Detection Challenge (DFDC) dataset. In addition, face manipulation was also used in face forgery generation such as MulGAN, MaskGAN, PuppetGAN, and HistoGAN.

1.2 DISADVANTAGE OF THE CURRENT SYSTEM

Rather than benefiting anyone, this AI-based technology has disadvantages affecting different groups of our society. Apart from creating fake news and propaganda, deepfake is majorly used for revenge porn to defame notable celebrities. As soon as fake videos go viral people believe initially, and keep sharing with others makes the targeted person become embarrassed watching such unusual acts. Until and unless an official statement of the targeted personality not comes, many people start believing making their life difficult, especially when they are criticized by their fans.

TABLE I

SUMMARY OF NOTABLE DEEPFAKE TOOLS



Tools	Links	Key Features
Faceswap	https://github.com/deepfakes/faceswap	- Using two encoder-decoder pairs. - Parameters of the encoder are shared.
Faceswap-GAN	https://github.com/shaoanlu/faceswap-GAN	Adversarial loss and perceptual loss (VGGface) are added to an auto-encoder architecture.
Few-Shot Face Translation	https://github.com/shaoanlu/fewshot-face-translation-GAN	- Use a pre-trained face recognition model to extract latent embeddings for GAN processing. - Incorporate semantic priors obtained by modules from FUNIT [42] and SPADE [43].
DeepFaceLab	https://github.com/iperov/DeepFaceLab	- Expand from the Faceswap method with new models, e.g. H64, H128, LIAEF128, SAE [44]. - Support multiple face extraction modes, e.g. S3FD, MTCNN, dlib, or manual [44].
DFaker	https://github.com/dfaker/df	- DSSIM loss function [45] is used to reconstruct face. - Implemented based on Keras library.
DeepFake_tf	https://github.com/StromWine/DeepFake_tf	Similar to DFaker but implemented based on tensorflow.
AvatarMe	https://github.com/lattas/AvatarMe	- Reconstruct 3D faces from arbitrary "in-the-wild" images. - Can reconstruct authentic 4K by 6K-resolution 3D faces from a single low-resolution image [46].
MarioNETte	https://hyperconnect.github.io/MarioNETte	- A few-shot face reenactment framework that preserves the target identity. - No additional fine-tuning phase is needed for identity adaptation [47].
DiscoFaceGAN	https://github.com/microsoft/DiscoFaceGAN	- Generate face images of virtual people with independent latent variables of identity, expression, pose, and illumination. - Embed 3D priors into adversarial learning [48].
StyleRig	https://gvv.mpi-inf.mpg.de/projects/StyleRig	- Create portrait images of faces with a rig-like control over a pretrained and fixed StyleGAN via 3D morphable face models. - Self-supervised without manual annotations [49].
FaceShifter	https://lingzhili.com/FaceShifterPage	- Face swapping in high-fidelity by exploiting and integrating the target attributes. - Can be applied to any new face pairs without requiring subject specific training [50].
FSGAN	https://github.com/YuvalNirkin/fsgan	- A face swapping and reenactment model that can be applied to pairs of faces without requiring training on those faces. - Adjust to both pose and expression variations [51].
Transformable Bottleneck Networks	https://github.com/kyleolsz/TB-Networks	- A method for fine-grained 3D manipulation of image content. - Apply spatial transformations in CNN models using a transformable bottleneck framework [52].
"Do as I Do" Motion Transfer	github.com/carolineec/EverybodyDanceNow	- Automatically transfer the motion from a source to a target person by learning a video-to-video translation. - Can create a motion-synchronized dancing video with multiple subjects [53].
Neural Voice Puppetry	https://justusthies.github.io/posts/neural-voice-puppetry	- A method for audio-driven facial video synthesis. - Synthesize videos of a talking head from an audio sequence of another person using 3D face representation. [54].

1.3 MERITS OF PROPOSED SYSTEM

In this section, we describe the architecture for DeepFake detection based on the classifier network with manual attention target-specific regions to create distillation set, which not only can improve the accuracy of classification using neural networks, but also allows the use of a lighter backbone. We introduce some steps in image processing to manually create a set of important data which is called manually distillation set in this paper and focus on special regions in Section 3.1. In addition, we also provide a model structure that we have used as a normal image classification model, which can generate features for each domain, to facilitate synthesis in the next step. In Section 3.3, we discuss on how to merge several frames and multi-regions in images before deciding on the final result.

I. Image Pre-processing

The objective of this section is to pre-process the picture before it is fed into the next stages. This part is critical because it influences the quality of the entire process moving ahead. It also improves the quality of the entire process through data processing. Image pre-processing are the steps taken to format images before they are used by model training and inference. This includes, but is not limited to, resizing, orienting, and color corrections.

Image augmentation are manipulations applied to images to create different versions of similar content in order to expose the model to a wider array of training examples. For example, randomly altering rotation, brightness, or scale of an input image requires that a model consider what an image subject looks like in a variety of situations.

Image augmentation manipulations are forms of image pre-processing, but there is a critical difference: while image pre-processing steps are applied to training and test sets, image augmentation is only applied to the training data**.** Thus, a transformation that could be an augmentation in some situations may best be a pre-processing step in others.

Consider altering image contrast. A given dataset could contain images that are generally low contrast. If the model will be used in production on only low contrast in all situations, requiring that every image undergo a constant amount of contrast adjustment may improve model performance. This pre-processing step would be applied to images in training and in testing. However, if the collected training data is not representative of the levels of contrast the model may see in production, there is less certainty that a constant contrast adjustment is appropriate. Instead, randomly altering image contrast during training may generalize better. This would be augmentation.

II. Introduction to deep learning

Artificial Intelligence has been witnessing a monumental growth in bridging the gap between the capabilities of humans and machines. Researchers and enthusiasts alike, work on numerous aspects of the field to make amazing things happen. One of many such areas is the domain of Computer Vision.

The agenda for this field is to enable machines to view the world as humans do, perceive it in a similar manner and even use the knowledge for a multitude of tasks such as Image & Video recognition, Image Analysis & Classification, Media Recreation, Recommendation Systems, Natural Language Processing, etc. The advancements in Computer Vision with Deep Learning has been constructed and perfected with time, primarily over one particular algorithm **Convolutional Neural Network**.

A **Convolutional Neural Network (ConvNet/CNN)** is a Deep Learning algorithm which can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image and be able to differentiate one from the other. The pre-processing required in a ConvNet is much lower as compared to other classification algorithms. While in primitive methods filters are hand-engineered, with enough training, ConvNets have the ability to learn these filters/characteristics.

The architecture of a ConvNet is analogous to that of the connectivity pattern of Neurons in the Human Brain and was inspired by the organization of the Visual Cortex. Individual neurons respond to stimuli only in a restricted region of the visual field known as the Receptive Field. A collection of such fields overlap to cover the entire visual area.

At its simplest, deep learning can be thought of as a way to automate predictive analytics. While traditional machine learning algorithms are linear, deep learning algorithms are stacked in a hierarchy of increasing complexity and abstraction. To understand deep learning, imagine a toddler whose first word is *dog*. The toddler learns what a dog is -- and is not -- by pointing to objects and saying the word *dog*. The parent says, "Yes, that is a dog," or, "No, that is not a dog." As the toddler continues to point to objects, he becomes more aware of the features that all dogs possess. What the toddler does, without knowing it, is clarify a complex abstraction -- the concept of dog -- by building a hierarchy in which each level of abstraction is created with knowledge that was gained from the preceding layer of the hierarchy. Deep learning is the branch of machine learning that is part of a neural network with three or more layers. These neural networks work like a human brain. They learn from a large amount of data. Using deep learning and artificial intelligence, engineers have created many applications and services that are used to enhance automation and perform analytical and physical tasks without human intervention. Today, deep learning technology drives daily profits and services (such as smart assistants, voice-activated TV, and remote controls.

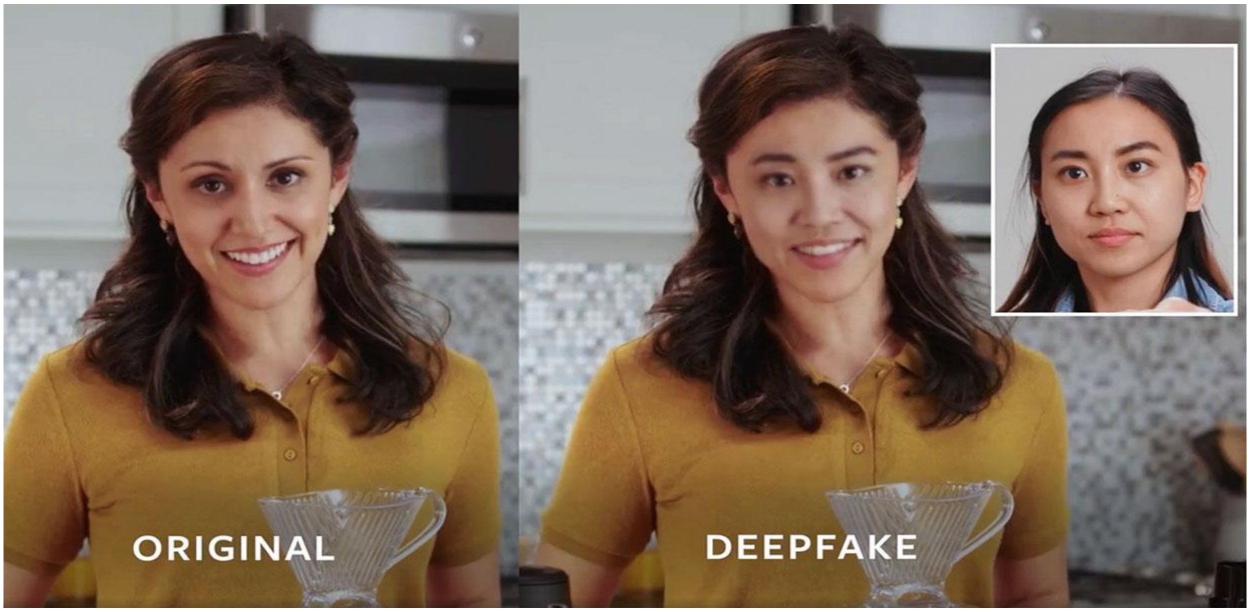


Figure 1: Deepfake impact on person image[5]

III. Deep learning versus machine learning.

Machine learning algorithms have a certain leverage effect. This means that these algorithms need labeled data to make a prediction. It just means that it undergoes pre-processing to organize it into structured data.

IV. What is machine learning?

Machine learning is the general term for when computers learn from data. It describes the intersect of computer science and statistics where algorithms are used to perform a specific task without being explicitly programmed; instead, they recognize patterns in the data and make predictions once new data arrives.

In general, the learning process of these algorithms can either be supervised or unsupervised, depending on the data being used to feed the algorithms. If you want to dive in a little bit deeper into the differences between supervised and unsupervised learning have a read
through this article.

A traditional machine learning algorithm can be something as simple as linear regression. For instance, imagine you want to predict your income given your years of higher education. In a first step, you have to define a function, e.g. income = y + x * years of education. Then, give your algorithm a set of training data. This could be a simple table with data on some people's years of higher education and their associated income. Next, let your algorithm draw the line, e.g. through an ordinary least squares (OLS) regression. Now, you can give the algorithm some test data, e.g. your personal years of higher education, and let it predict your income.

While this example sounds simple it does count as machine learning – and yes, the driving force behind machine learning is ordinary statistics. The algorithm learned to make a prediction without being explicitly programmed, only based on patterns and inference.

So much about machine learning in general – to summarize:

- Machine learning is at the intersection of computer science and statistics through which computers receive the ability to learn without being explicitly programmed.
- There are two broad categories of machine learning problems: supervised and unsupervised learning.
- A machine learning algorithm can be something as simple as an OLS regression.

Let's now examine how the term deep learning relates to all of this.

V. What is a deep fake?

The 21st century is the age of fake videos and images and this technology and has generated a lot of propaganda in recent years. Deep fake is like a weapon that can be used for both good and bad reasons. The Deep Fake is the combination of artificial intelligence and deep learning used to create faked images and videos. Fake content in the form of images, videos and audio is created using artificial intelligence and deep learning by feeding data to the computer and creating new images or videos. Convolutional neural network.

A convolutional neuron is mainly used to recognize an image and to process images specially designed to process pixel data. CNN consists of several convolutional neural networks.

Deepfakes are so-named because they create a fake video using deep learning technology, a branch of machine learning that applies neural net simulation to massive data sets. Artificial intelligence effectively learns what a source face looks like at different angles in order to transpose the face onto a target, usually an actor, as if it were a mask. Huge advances came through the application of generative adversarial networks (GANS) to pit two AI algorithms against each other, one creating the fakes and the other grading its efforts, teaching the synthesis engine to make better forgeries.

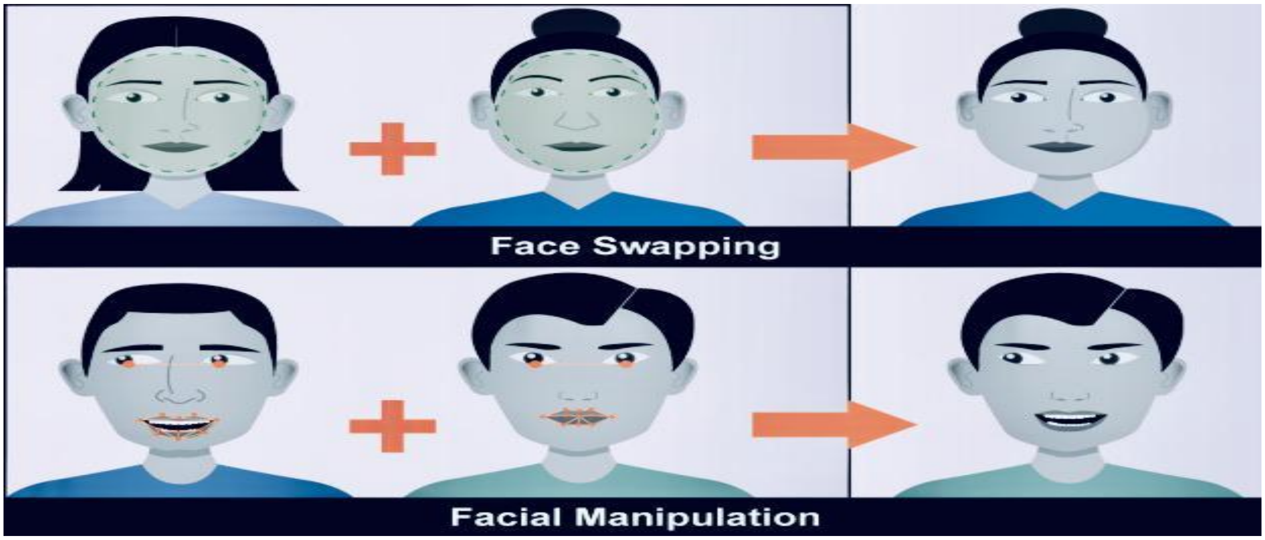


Figure 2: Image Manipulation on Deepfake

VI. How does CNN work?

Convolutional neural networks are made up of several layers of artificial neural neutrons and arrival neutrons. A rough indication of this biological counterpart, a mathematical function calculates the weighted sum of the multiple inputs and generates a trigger value.

Tools and Technology used:

System requirements :

- Ram : 4gb
- Solid State drive: 128gb
- Operating System: Windows/Mac/Linux

Tools:

- Image Dataset from Kaggle.

Language:

- Python

Algorithm:

- Convolutional Neural Network.

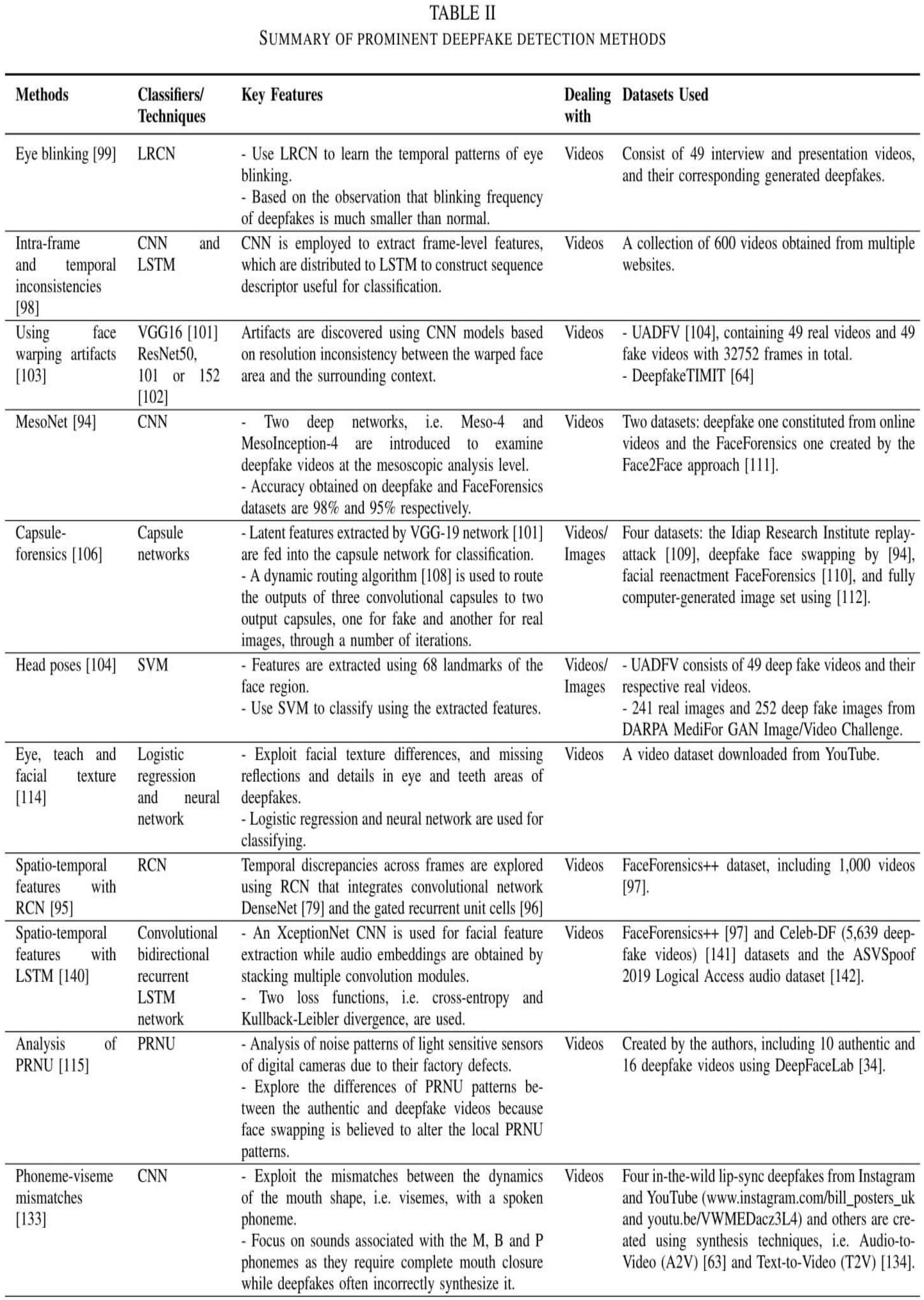


TABLE II
SUMMARY OF PROMINENT DEEPFAKE DETECTION METHODS

Methods	Classifiers/ Techniques	Key Features	Dealing with	Datasets Used
Eye blinking [99]	LRCN	- Use LRCN to learn the temporal patterns of eye blinking. - Based on the observation that blinking frequency of deepfakes is much smaller than normal.	Videos	Consist of 49 interview and presentation videos, and their corresponding generated deepfakes.
Intra-frame and temporal inconsistencies [98]	CNN and LSTM	CNN is employed to extract frame-level features, which are distributed to LSTM to construct sequence descriptor useful for classification.	Videos	A collection of 600 videos obtained from multiple websites.
Using face warping artifacts [103]	VGG16 [101] ResNet50, 101 or 152 [102]	Artifacts are discovered using CNN models based on resolution inconsistency between the warped face area and the surrounding context.	Videos	- UADFV [104], containing 49 real videos and 49 fake videos with 32752 frames in total. - DeepfakeTIMIT [64]
MesoNet [94]	CNN	- Two deep networks, i.e. Meso-4 and MesoInception-4 are introduced to examine deepfake videos at the mesoscopic analysis level. - Accuracy obtained on deepfake and FaceForensics datasets are 98% and 95% respectively.	Videos	Two datasets: deepfake one constituted from online videos and the FaceForensics one created by the Face2Face approach [111].
Capsule-forensics [106]	Capsule networks	- Latent features extracted by VGG-19 network [101] are fed into the capsule network for classification. - A dynamic routing algorithm [108] is used to route the outputs of three convolutional capsules to two output capsules, one for fake and another for real images, through a number of iterations.	Videos/ Images	Four datasets: the Idiap Research Institute replay-attack [109], deepfake face swapping by [94], facial reenactment FaceForensics [110], and fully computer-generated image set using [112].
Head poses [104]	SVM	- Features are extracted using 68 landmarks of the face region. - Use SVM to classify using the extracted features.	Videos/ Images	- UADFV consists of 49 deep fake videos and their respective real videos. - 241 real images and 252 deep fake images from DARPA MediFor GAN Image/Video Challenge.
Eye, teach and facial texture [114]	Logistic regression and neural network	- Exploit facial texture differences, and missing reflections and details in eye and teeth areas of deepfakes. - Logistic regression and neural network are used for classifying.	Videos	A video dataset downloaded from YouTube.
Spatio-temporal features with RCN [95]	RCN	Temporal discrepancies across frames are explored using RCN that integrates convolutional network DenseNet [79] and the gated recurrent unit cells [96]	Videos	FaceForensics++ dataset, including 1,000 videos [97].
Spatio-temporal features with LSTM [140]	Convolutional bidirectional recurrent LSTM network	- An XceptionNet CNN is used for facial feature extraction while audio embeddings are obtained by stacking multiple convolution modules. - Two loss functions, i.e. cross-entropy and Kullback-Leibler divergence, are used.	Videos	FaceForensics++ [97] and Celeb-DF (5,639 deepfake videos) [141] datasets and the ASVSpoof 2019 Logical Access audio dataset [142].
Analysis of PRNU [115]	PRNU	- Analysis of noise patterns of light sensitive sensors of digital cameras due to their factory defects. - Explore the differences of PRNU patterns between the authentic and deepfake videos because face swapping is believed to alter the local PRNU patterns.	Videos	Created by the authors, including 10 authentic and 16 deepfake videos using DeepFaceLab [34].
Phoneme-viseme mismatches [133]	CNN	- Exploit the mismatches between the dynamics of the mouth shape, i.e. visemes, with a spoken phoneme. - Focus on sounds associated with the M, B and P phonemes as they require complete mouth closure while deepfakes often incorrectly synthesize it.	Videos	Four in-the-wild lip-sync deepfakes from Instagram and YouTube (www.instagram.com/bill_posters_uk and youtu.be/VWMEDacz3L4) and others are created using synthesis techniques, i.e. Audio-to-Video (A2V) [63] and Text-to-Video (T2V) [134].

2.Literature Survey

This section aims to debate and analyse varied techniques that are used for sleuthing deepfake. varied tries are created to sight deepfakes that use deep learning at their core. These approaches work either on sleuthing faults in video or in separate frames of the video. varied different approaches have used CNN in conjunction with different learning models like perennial Neural Network (RNN), Long immediate memory Networks (LSTM) and Capsule Network to any improve the accuracy by sleuthing temporal discrepancies and have shown smart results on dataset containing videos generated by Face Swap and deepfake.

A convolutional neural network (CNN) is that the most ordinarily used deep neural network model. CNN, like neural networks, has Associate in Nursing input and output layer, also joined or additional hidden layers. In CNN, the hidden layers 1st scans the inputs from the primary layer and so apply a convolution calculation on the input values.

Recurrent Neural Network (RNN) is another application of artificial neural network that is capable to find out options from sequence information. almost like neural networks, RNN is formed of many invisible layers, every of that contains a weight and a bias.

LSTM could be a variety of artificial continual neural network (RNN) that handles long-run dependencies. LSMT contains feedback connections to find out the whole sequence of knowledge. LSTM has been applied to several fields that supported statistic information like classifying, process and creating predictions. Although varied enhancements are created, additional powerful models are required to sight deepfakes of lower quality and maintaining wide accuracy remains a challenge with perpetually up ways in deepfake creation.

DEEPFAKE CREATION

Deepfakes have become popular due to the quality of tampered videos and also the easy-to-use ability of their applications to a wide range of users with various computer skills from professional to novice. These applications are mostly developed based on deep learning techniques. Deep learning is well known for its capability of representing complex and high-dimensional data. One variant of the deep networks with that capability is deep autoencoders, which have been widely applied for dimensionality reduction and image compression. The first attempt of deepfake creation was FakeApp, developed by a Reddit user using autoencoder-decoder pairing structure. In that method, the autoencoder extracts latent features of face images and the decoder is used to reconstruct the face images. To swap faces between source images and target images, there is a need of two encoder- decoder pairs where each pair is used to train on an image set, and the encoder’s parameters are shared between two network pairs. In other words, two pairs have the same encoder network. This strategy enables the common encoder to find and learn the similarity between two sets of face images, which are relatively unchallenging because faces normally have similar features such as eyes, nose, mouth positions. Fig. 2 shows a deepfake creation process where the feature set of face A is connected with the decoder B to reconstruct face B from the original face A. This approach is applied in several works such as DeepFaceLab, DFaker, DeepFake tf (tensorflow- based deepfakes). By adding adversarial loss and perceptual loss implemented in VGGFace to the encoder-decoder architecture, an improved version of deepfakes based on the generative adver- sarial network (GAN), i.e. faceswap-GAN, was proposed in. The VGGFace perceptual loss is added to make eye movements to be more realistic and consistent with input faces and help to smooth out artifacts in segmentation mask, leading to higher quality output videos. This model facilitates the creation of outputs with 64x64, 128x128, and 256x256 resolutions. In addition, the multi-task convolutional neural network (CNN) from the Face-Net implementation is introduced to make face detection more stable and face alignment more reliable. The Cycle GAN is utilized for generative network implementation. Popular deepfake tools and their features are summarized in table.[1] Nowadays, deepfake become very popular because of the quality of tempered photos and videos .There are many applications in market which make deep fake so easier. These applications are mostly based on deep learning technology[1]. The first attempt of deepfake creation was Fake App, developed by a Reddit user using an autoencoder-decoder pairing structure .In this method encoder is used to extract the features of image and decoder is used to construct the image. To exchange face between source image and target image we need two pair of encoder and decoder where each pair is used to train on image.

FACE AUGMENTATION

The overfitting problem is always considered carefully in DFDC datasets. When a the overfitting problem is always considered carefully in DFDC datasets. When a model learns the information and noise in the training data to the point then it degrades model learns the information and noise in the training data to the point then it degrades the model's performance on new data, this is known as overfitting. This means that the model's performance on new data, this is known as overfitting. This means that the model learns too well with the training data for the DFDC dataset, but the outcomes with model learns too well with the training data for the DFDC dataset, but the outcomes with the test data are not as good as expected. One method for resolving this issue is to use the test data are not as good as expected. One method for resolving this issue is to use augmentation. In terms of increasing this quality, previous research has also found augmentation. In terms of increasing this quality, previous research has also found that data augmentation can help to mitigate this negative effect. This is a crucial ap- that data augmentation can help to mitigate this negative effect. This is a crucial approach for generating more usable data and improving the model's quality during training approach for generating more usable data and improving the model's quality during training - the methods of augmentation used in this proposal are mostly focused on information.

The methods of augmentation used in this proposal are mostly focused on infor- dropping, illustrated in Figure 2. It mainly focuses on the meaningful regions of mation dropping, illustrated in Figure 2. It mainly focuses on the meaningful the face to distinguish the real from the fake, such as the eyes, nose and mouth. These regions of the face to distinguish the real from the fake, such as the eyes, nose and mouth. Appl. Sci. 2021, 11, x FOR PEER REVIEW 5 of 15 important regions are illustrated in Figure 3. Important regions are dropped randomly to These important regions are illustrated in Figure 3. Important regions are dropped increase data diversity.

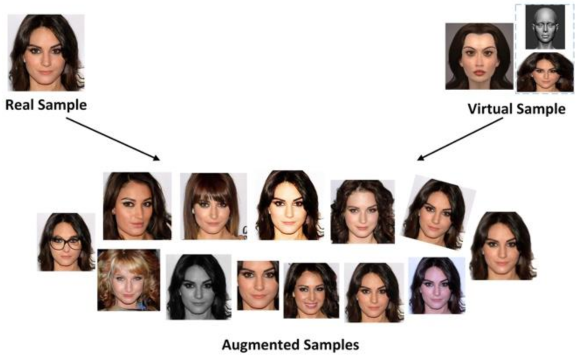


Figure 3: Face Augmentation.

DEEPFAKE DETECTION

Deepfake detection is a relatively young research space, which started in early 2018. There are two types of techniques.

- **Bio-logical signals[1]**

  Several works have observed unusual behaviours in deepfake videos, such as no blinking, facial abnormalities and movement abnormalities. These methods suffer from the potential for immediate improvements in video generation techniques, such as adding flicker.

- **Pixel-level irregularities. [1]**

  There is a greater body of research that extracts faces and uses various forms of deep learning to address inconsistencies between images and images. While many of these methods work well with specific types of manipulations, they fail to generalize to multiple and unknown types of deepfakes, which is essential for open-world detection. None of the deepfake detection techniques have so far been developed to become a real tool for real-world detection. Also, to our knowledge, no studies have been done on how to design such a tool for effective use by journalists.

Deepfake detection is normally deemed a binary classification problem where classifiers are used to classify between authentic videos and tampered ones. This kind of methods requires a large database of real and fake videos to train classification models. The number of fake videos is increasingly available, but it is still limited in terms of setting a benchmark for validating various detection methods.[1]

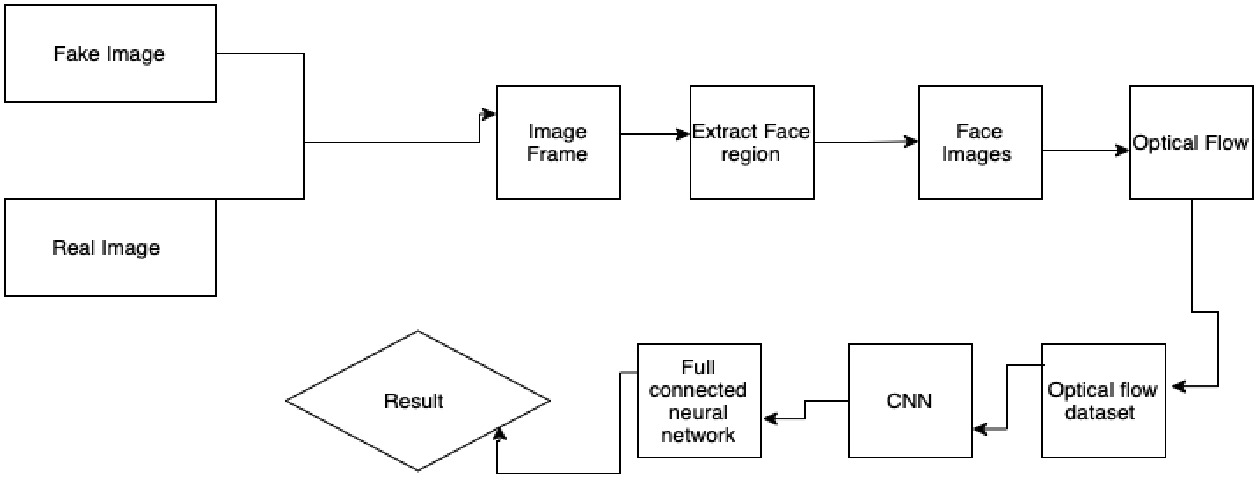


Figure 4: Flow chart of deepfake of images using convolutional neural network

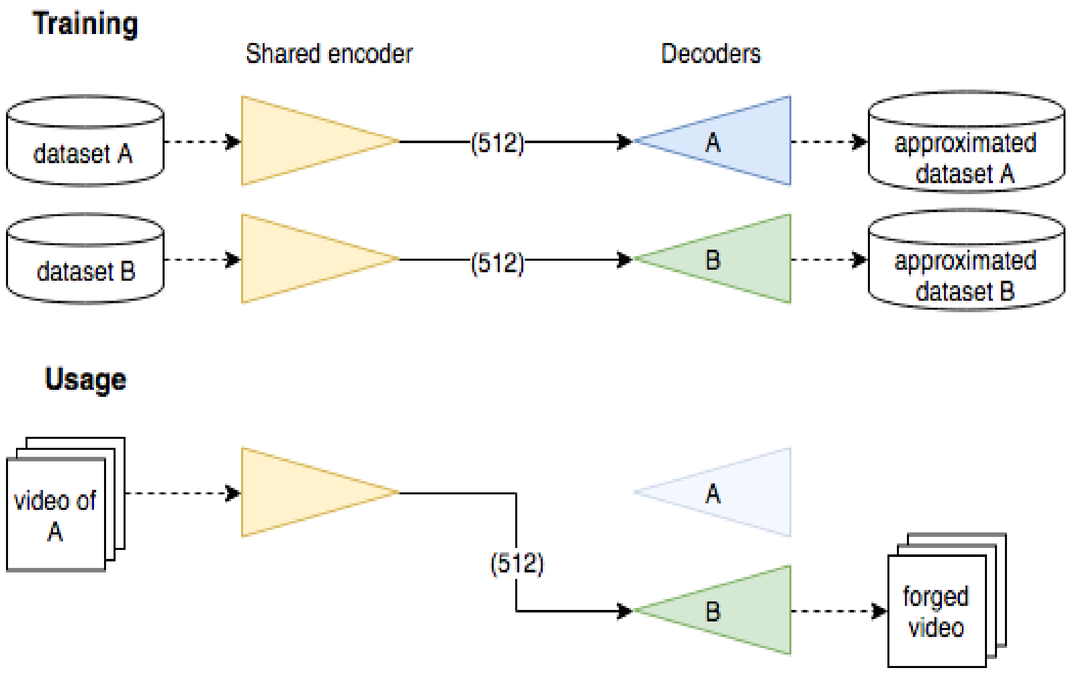


Figure 5: Pictorial representation of model [4]

Fake Image Detection

Face swapping has a number of compelling applications in video compositing, transfiguration in portraits, and especially in identity protection as it can replace faces in photographs by ones from a collection of stock images. However, it is also one of the techniques that cyber attackers employ to penetrate identification or authentication systems to gain illegitimate access. The use of deep learning such as CNN and GAN has made swapped face images more challenging for forensics models as it can preserve pose, facial expression and lighting of the photographs Zhang et al.used the bag of words method to extract a set of compact features and fed it into various classifiers such as SVM , random forest (RF) and multi-layer perceptrons (MLP) for discriminating swapped face images from the genuine. Among deep learning- generated images, those synthesised by GAN models are probably most difficult to detect as they are realistic and high- quality based on GAN's capability to learn distribution of the complex input data and generate new outputs with similar input distribution.

Most works on detection of GAN generated images however do not consider the generalization capability of the detection models although the development of GAN is ongoing, and many new extensions of GAN are frequently introduced. Xuan et al. used an image pre-processing step, e.g. Gaussian blur and Gaussian noise, to remove low level high frequency clues of GAN images. This increases the pixel level statistical similarity between real images and fake images and requires the forensic classifier to learn more intrinsic and meaningful features, which has better generalization capability than previous image forensics methods or image steganalysis networks.

On the other hand, Agarwal and Varshney cast the GAN-based deepfake detection as a hypothesis testing problem where a statistical framework was introduced using the information-theoretic study of authentication. The mini- mum distance between distributions of legitimate images and images generated by a particular GAN is defined, namely the oracle error. The analytic results show that this distance

increases when the GAN is less accurate, and in this case, it is easier to detect deepfakes. In case of high-resolution image inputs, an extremely accurate GAN is required to generate fake images that are hard to detect.

Recently, Hsu et al. introduced a two-phase deep learning method for detection of deepfake images. The first phase is a feature extractor based on the common fake feature network (CFFN) where the Siamese network architecture presented in is used. The CFFN encompasses several dense units with each unit including different numbers of dense blocks to improve the representative capability for the fake images. The number of dense units is three or five depending on the validation data being face or general images, and the number of channels in each unit is varied up to a few hundreds. Discriminative features between the fake and real images, i.e. pairwise information, are extracted through CFFN learning process. These features are then fed into the second phase, which is a small CNN concatenated to the last convolutional layer of CFFN to distinguish deceptive images from genuine. The proposed method is validated for both fake face and fake general image detection. On the one hand, the face data set is obtained from CelebA, containing 10,177 identities and 202,599 aligned face images of various poses and background clutter. Five GAN variants are used to generate fake images with size of 64x64, including deep convolutional GAN (DC- GAN), Wasserstein GAN (WGAN), WGAN with gradient penalty (WGAN-GP), least squares GAN, and progressive growth of GAN (PGGAN). A total of 385,198 training images and 10,000 test images of both real and fake ones are obtained for validating the proposed method. On the other hand, the general data set is extracted from the ILSVRC12. The large-scale GAN training model for high fidelity natural image synthesis (BIGGAN), self-attention GAN and spectral normalization GAN are used to generate fake images with size of 128x128. The training set consists of 600,000 fake and real images whilst the test set includes 10,000 images of both types. Experimental results show the superior performance of the proposed method against its competing methods such as those introduced in.

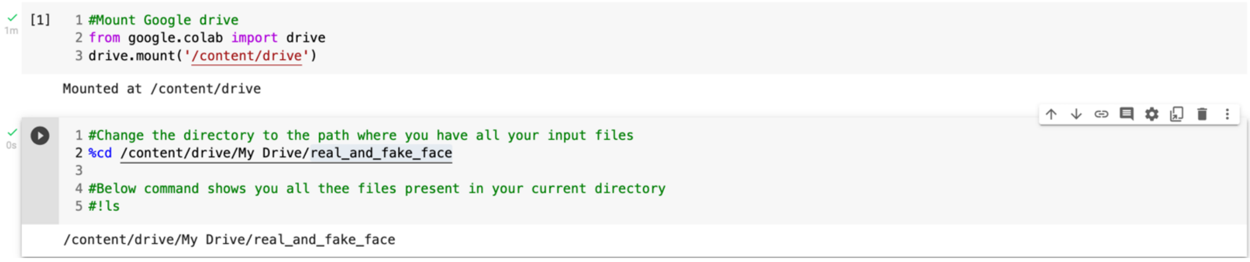
Fake Video Detection

Most image detection methods cannot be used for videos because of the strong degradation of the frame data after video compression. Furthermore, videos have temporal characteristics that are varied among sets of frames and thus challenging for methods designed to detect only still fake images. This subsection focuses on deepfake video detection methods and categorizes them into two groups: methods that employ temporal features and those that explore visual artifacts within frames.

1) Temporal Features across Video Frames: Based on the observation that temporal coherence is not enforced effectively in the synthesis process of deepfakes, Sabir et al. leveraged the use of spatio-temporal features of video streams to detect deepfakes. Video manipulation is carried out on a frame-by-frame basis so that low level artifacts produced by face manipulations are believed to further manifest themselves as temporal artifacts with inconsistencies across frames. A recurrent convolutional model (RCN) was proposed based on the integration of the convolutional network DenseNet and the gated recurrent unit cells to exploit temporal discrepancies across frames (see Fig. 4). The proposed method is tested on the FaceForensics++ data set, which includes 1,000 videos, and shows promising results.

Functionality/Working of project

1. Mounting Google drive



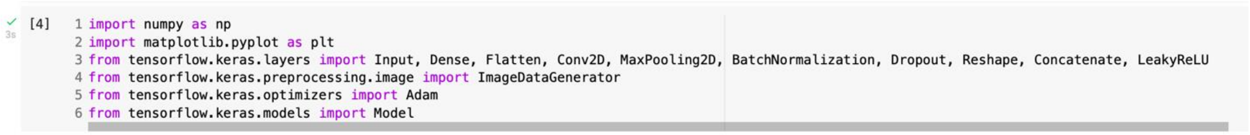
```
#Mount Google drive
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```
#Change the directory to the path where you have all your input files
%cd /content/drive/My Drive/real_and_fake_face

#Below command shows you all thee files present in your current directory
#!ls
```

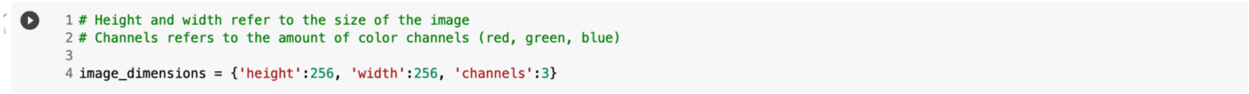
```
/content/drive/My Drive/real_and_fake_face
```

2. Importing all the necessary libraries.

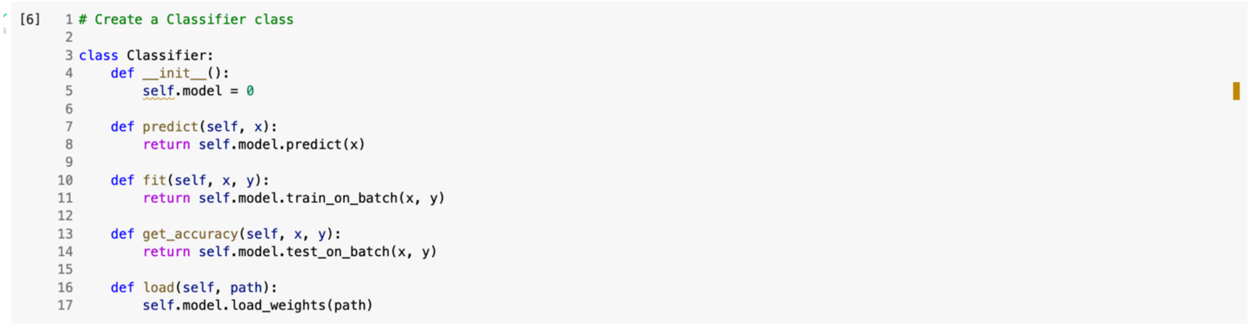


```
import numpy as np
import matplotlib.pyplot as plt
from tensorflow.keras.layers import Input, Dense, Flatten, Conv2D, MaxPooling2D, BatchNormalization, Dropout, Reshape, Concatenate, LeakyReLU
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.models import Model
```

3. Setting Image height and width and creating a classifier class

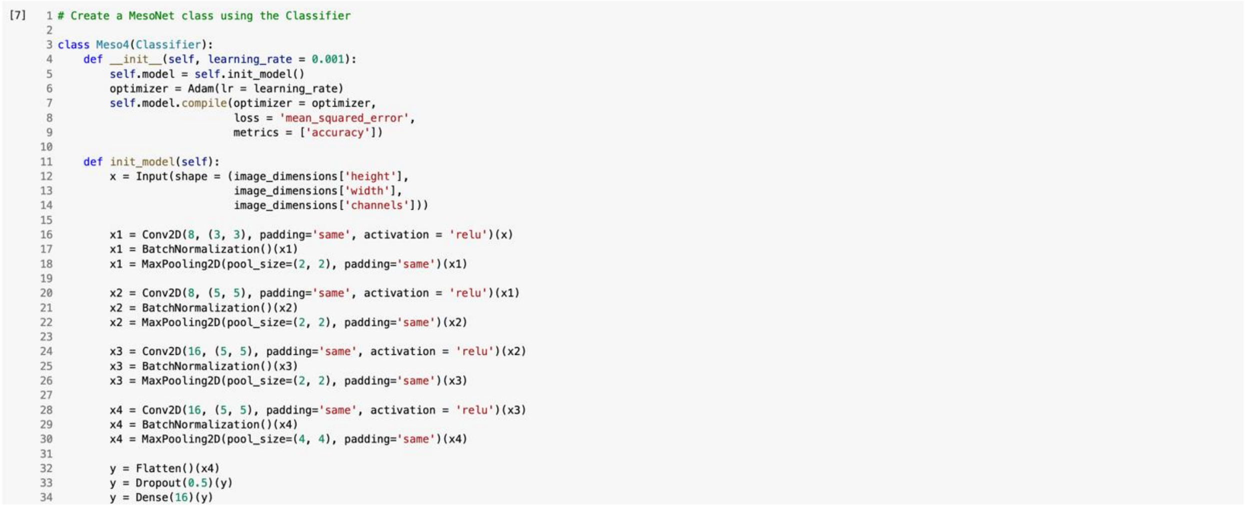


```
# Height and width refer to the size of the image
# Channels refers to the amount of color channels (red, green, blue)

image_dimensions = {'height':256, 'width':256, 'channels':3}
```



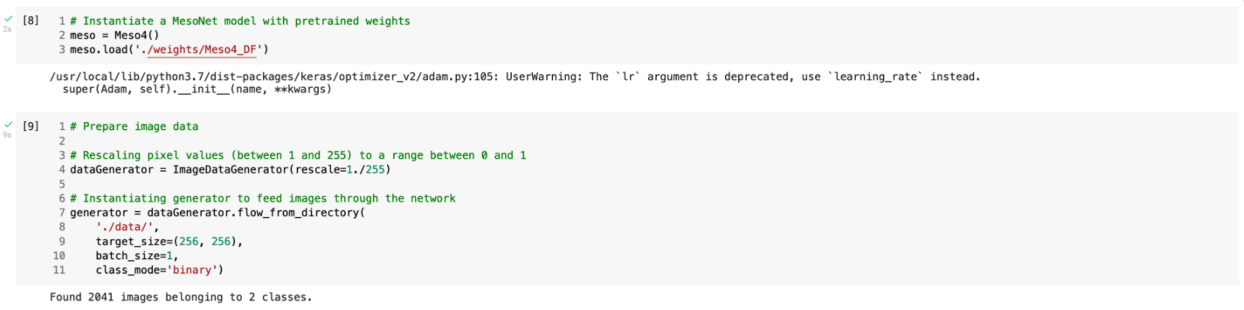
```
# Create a Classifier class

class Classifier:
    def __init__():
        self.model = 0

    def predict(self, x):
        return self.model.predict(x)

    def fit(self, x, y):
        return self.model.train_on_batch(x, y)

    def get_accuracy(self, x, y):
        return self.model.test_on_batch(x, y)

    def load(self, path):
        self.model.load_weights(path)
```

4. Creating a mesonet class.



```
[7]
# Create a MesoNet class using the Classifier

class Meso4(Classifier):
    def __init__(self, learning_rate = 0.001):
        self.model = self.init_model()
        optimizer = Adam(lr = learning_rate)
        self.model.compile(optimizer = optimizer,
                           loss = 'mean_squared_error',
                           metrics = ['accuracy'])

    def init_model(self):
        x = Input(shape = (image_dimensions['height'],
                           image_dimensions['width'],
                           image_dimensions['channels']))

        x1 = Conv2D(8, (3, 3), padding='same', activation = 'relu')(x)
        x1 = BatchNormalization()(x1)
        x1 = MaxPooling2D(pool_size=(2, 2), padding='same')(x1)

        x2 = Conv2D(8, (5, 5), padding='same', activation = 'relu')(x1)
        x2 = BatchNormalization()(x2)
        x2 = MaxPooling2D(pool_size=(2, 2), padding='same')(x2)

        x3 = Conv2D(16, (5, 5), padding='same', activation = 'relu')(x2)
        x3 = BatchNormalization()(x3)
        x3 = MaxPooling2D(pool_size=(2, 2), padding='same')(x3)

        x4 = Conv2D(16, (5, 5), padding='same', activation = 'relu')(x3)
        x4 = BatchNormalization()(x4)
        x4 = MaxPooling2D(pool_size=(4, 4), padding='same')(x4)

        y = Flatten()(x4)
        y = Dropout(0.5)(y)
        y = Dense(16)(y)
```

5. Preparing image data.

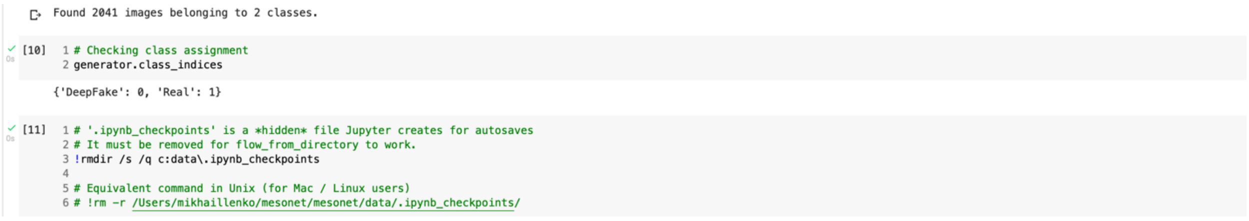


```
[8]
# Instantiate a MesoNet model with pretrained weights
meso = Meso4()
meso.load('./weights/Meso4_DF')
```

```
/usr/local/lib/python3.7/dist-packages/keras/optimizer_v2/adam.py:105: UserWarning: The `lr` argument is deprecated, use `learning_rate` instead.
  super(Adam, self).__init__(name, **kwargs)
```

```
[9]
# Prepare image data

# Rescaling pixel values (between 1 and 255) to a range between 0 and 1
dataGenerator = ImageDataGenerator(rescale=1./255)

# Instantiating generator to feed images through the network
generator = dataGenerator.flow_from_directory(
    './data/',
    target_size=(256, 256),
    batch_size=1,
    class_mode='binary')
```

```
Found 2041 images belonging to 2 classes.
```

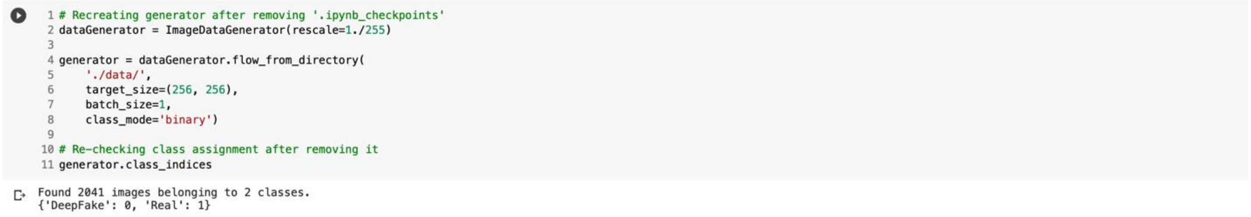


```
Found 2041 images belonging to 2 classes.
```

```
[10]
# Checking class assignment
generator.class_indices
```

```
{'DeepFake': 0, 'Real': 1}
```

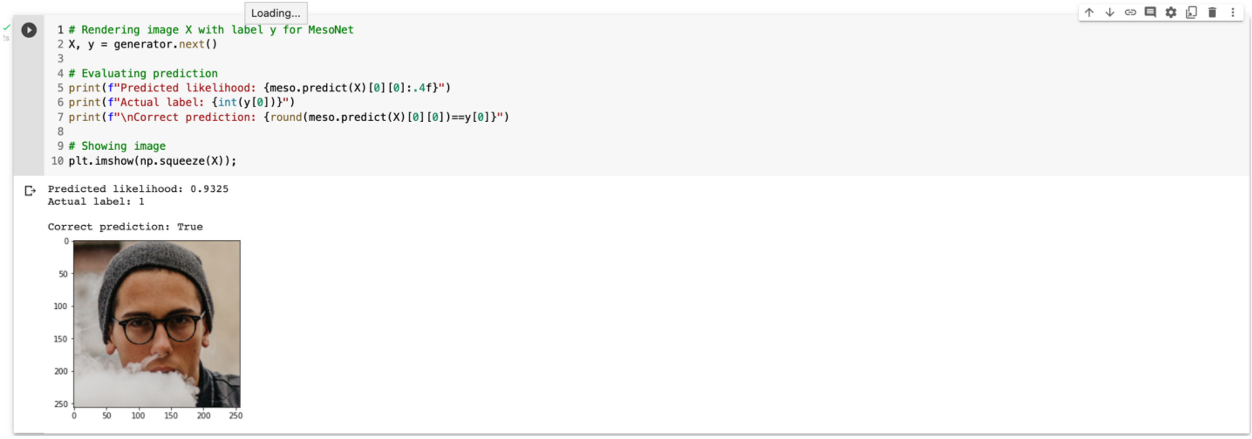
```
[11]
# '.ipynb_checkpoints' is a *hidden* file Jupyter creates for autosaves
# It must be removed for flow_from_directory to work.
!rmdir /s /q c:data\.ipynb_checkpoints

# Equivalent command in Unix (for Mac / Linux users)
# !rm -r /Users/mikhaillenko/mesonet/mesonet/data/.ipynb_checkpoints/
```



```
# Recreating generator after removing '.ipynb_checkpoints'
dataGenerator = ImageDataGenerator(rescale=1./255)

generator = dataGenerator.flow_from_directory(
    './data/',
    target_size=(256, 256),
    batch_size=1,
    class_mode='binary')

# Re-checking class assignment after removing it
generator.class_indices
```

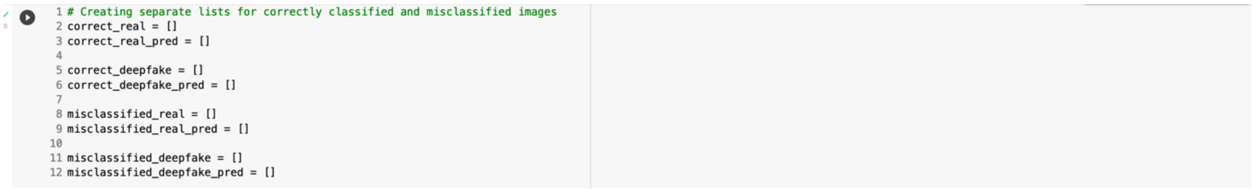
```
Found 2041 images belonging to 2 classes.
{'DeepFake': 0, 'Real': 1}
```

6. Training our data.

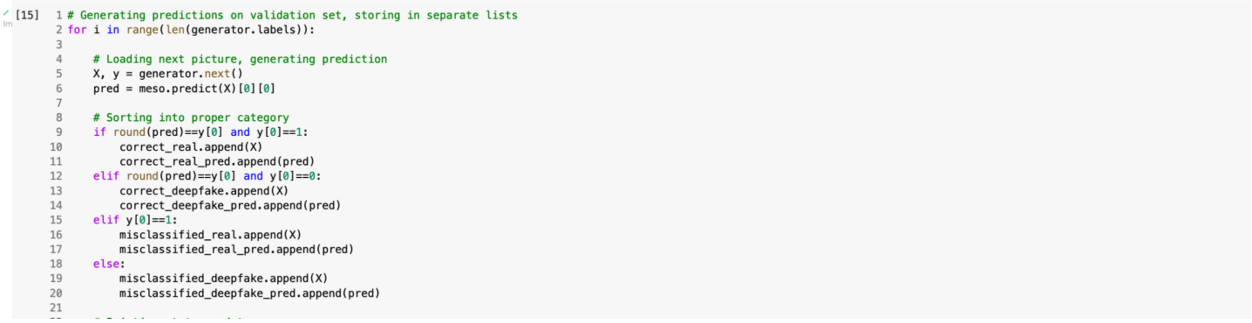


```python
# Rendering image X with label y for MesoNet
X, y = generator.next()

# Evaluating prediction
print(f"Predicted likelihood: {meso.predict(X)[0][0]:.4f}")
print(f"Actual label: {int(y[0])}")
print(f"\nCorrect prediction: {round(meso.predict(X)[0][0])==y[0]}")

# Showing image
plt.imshow(np.squeeze(X));
```

```
Predicted likelihood: 0.9325
Actual label: 1

Correct prediction: True
```

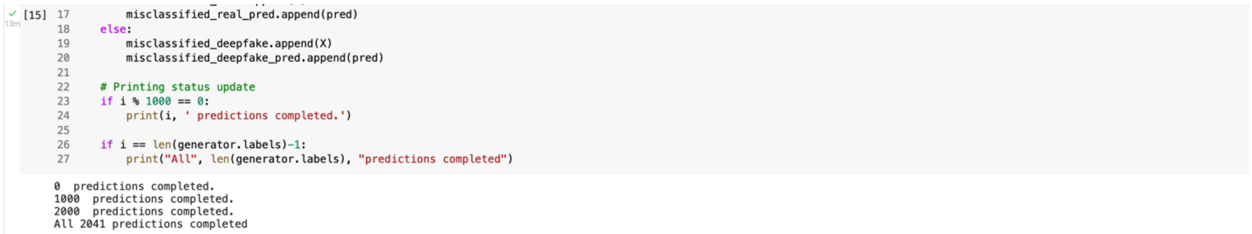


```python
# Creating separate lists for correctly classified and misclassified images
correct_real = []
correct_real_pred = []

correct_deepfake = []
correct_deepfake_pred = []

misclassified_real = []
misclassified_real_pred = []

misclassified_deepfake = []
misclassified_deepfake_pred = []
```



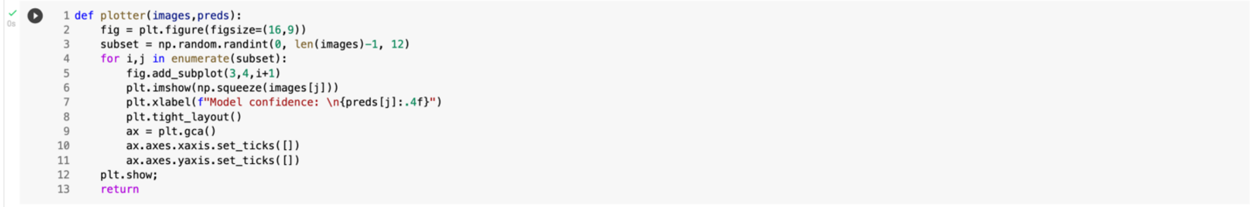
```python
[15]
# Generating predictions on validation set, storing in separate lists
for i in range(len(generator.labels)):

    # Loading next picture, generating prediction
    X, y = generator.next()
    pred = meso.predict(X)[0][0]

    # Sorting into proper category
    if round(pred)==y[0] and y[0]==1:
        correct_real.append(X)
        correct_real_pred.append(pred)
    elif round(pred)==y[0] and y[0]==0:
        correct_deepfake.append(X)
        correct_deepfake_pred.append(pred)
    elif y[0]==1:
        misclassified_real.append(X)
        misclassified_real_pred.append(pred)
    else:
        misclassified_deepfake.append(X)
        misclassified_deepfake_pred.append(pred)

```

8. Prediction on complete data.



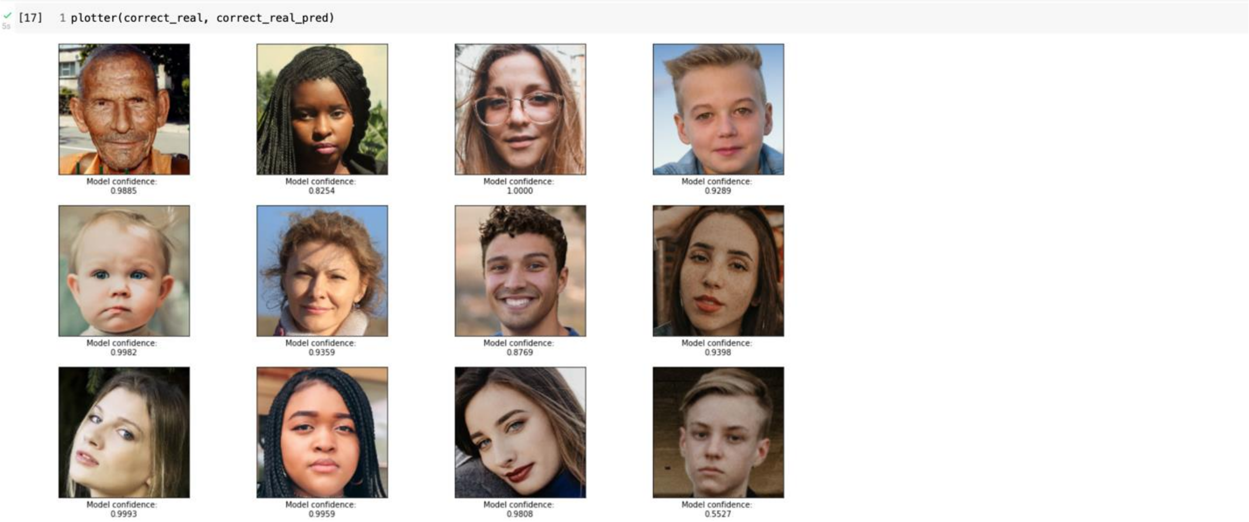
```python
[15]
        misclassified_real_pred.append(pred)
    else:
        misclassified_deepfake.append(X)
        misclassified_deepfake_pred.append(pred)

    # Printing status update
    if i % 1000 == 0:
        print(i, ' predictions completed.')

    if i == len(generator.labels)-1:
        print("All", len(generator.labels), "predictions completed")
```

```
0  predictions completed.
1000  predictions completed.
2000  predictions completed.
All 2041 predictions completed
```



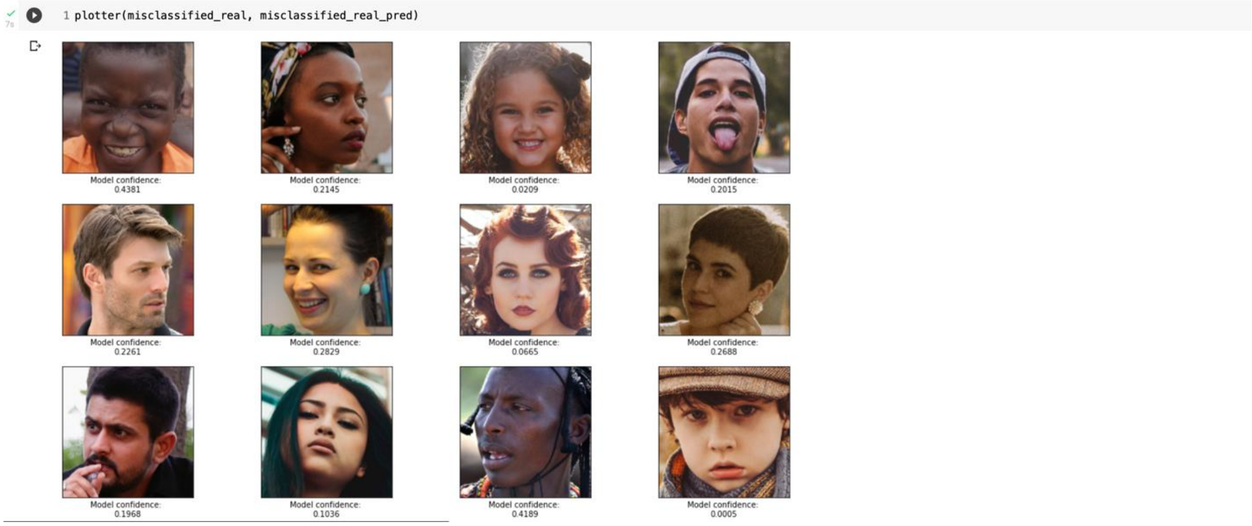
```python
def plotter(images,preds):
    fig = plt.figure(figsize=(16,9))
    subset = np.random.randint(0, len(images)-1, 12)
    for i,j in enumerate(subset):
        fig.add_subplot(3,4,i+1)
        plt.imshow(np.squeeze(images[j]))
        plt.xlabel(f"Model confidence: \n{preds[j]:.4f}")
        plt.tight_layout()
        ax = plt.gca()
        ax.axes.xaxis.set_ticks([])
        ax.axes.yaxis.set_ticks([])
    plt.show;
    return
```

9. Plotting real images.



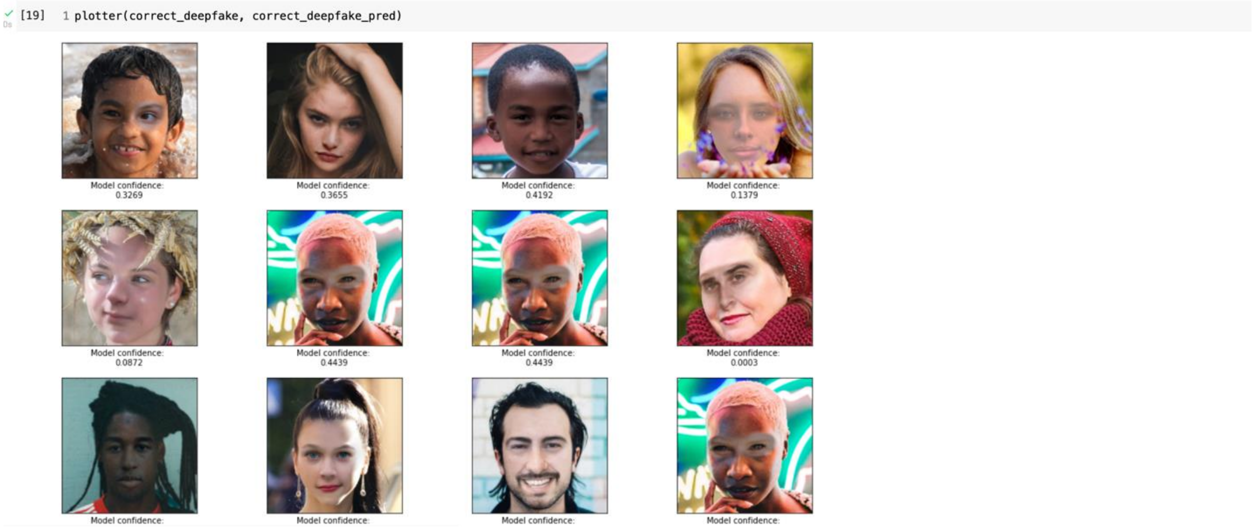
```
[17] plotter(correct_real, correct_real_pred)
```

10. Plotting misclassified real images



```
plotter(misclassified_real, misclassified_real_pred)
```

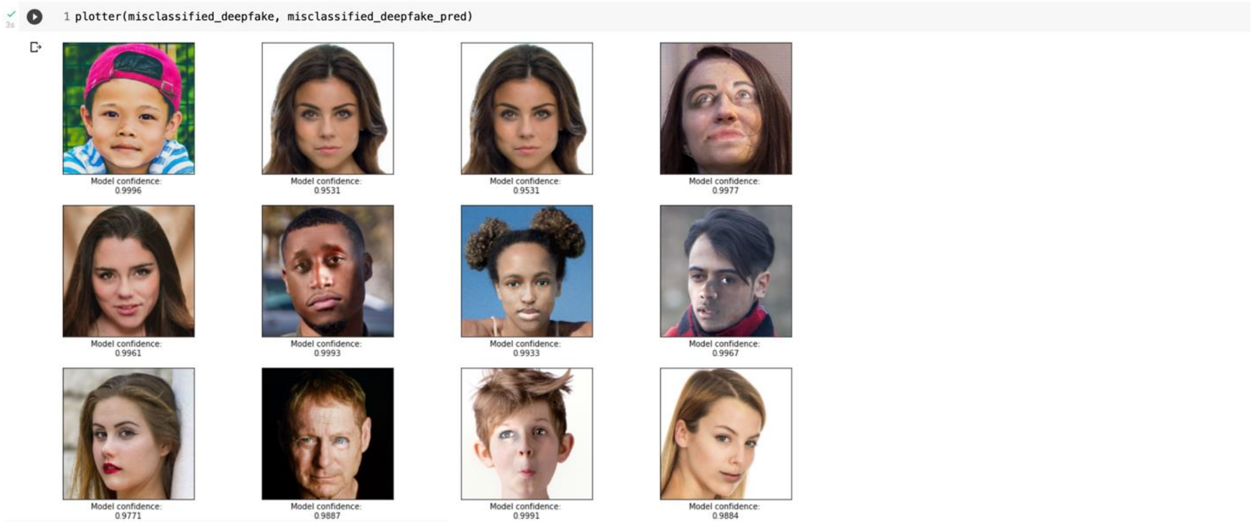
11. Plotting fake images.



```
[19]
plotter(correct_deepfake, correct_deepfake_pred)
```

```
Model confidence:   Model confidence:   Model confidence:   Model confidence:
0.3269              0.3655              0.4192              0.1379
Model confidence:   Model confidence:   Model confidence:   Model confidence:
0.0872              0.4439              0.4439              0.0003
Model confidence:   Model confidence:   Model confidence:   Model confidence:
```

12. Plotting misclassified fake images.



```
plotter(misclassified_deepfake, misclassified_deepfake_pred)
```

```
Model confidence:   Model confidence:   Model confidence:   Model confidence:
0.9996              0.9531              0.9531              0.9977
Model confidence:   Model confidence:   Model confidence:   Model confidence:
0.9961              0.9993              0.9933              0.9967
Model confidence:   Model confidence:   Model confidence:   Model confidence:
0.9771              0.9887              0.9991              0.9884
```

Result and Discussion

It is not very easy to find deep fake videos where manipulation is present in a small portion of the video. ie in a few frames.to tackle this issue we need to train, test and validate every image or every frame of video. We believe that our program has a powerful architecture to spot the fake images and video using tools and described in our paper.we have shown how our system can achieve results.In future work we have planned to study how to increase the robustness of our model against manipulated images and forged videos.

Deepfakes have begun to erode trust of people in media contents as seeing them is no longer commensurate with believing in them. They could cause distress and negative effects to those targeted, heighten disinformation and hate speech, and even could stimulate political tension, inflame the public, violence or war. This is especially critical nowadays as the technologies for creating deepfakes are increasingly approachable and social media platforms can spread those fake contents quickly. Sometimes deepfakes do not need to be spread to massive audience to cause detrimental effects. People who create deepfakes with malicious purpose only need to de- liver them to target audiences as part of their sabotage strategy without using social media. For example, this approach can be utilized by intelligence services trying to influence decisions made by important people such as politicians, leading to national and international security threats. Catching the deepfake alarming problem, research community has focused on developing deepfake detection algorithms and numerous results have been reported. This paper has reviewed the state- of-the-art methods and a summary of typical approaches is provided in Table II. It is noticeable that a battle between those who use advanced machine learning to create deepfakes with those who make effort to detect deepfakes is growing.

CONCLUSION AND FUTURE SCOPE

CONCLUSION

Deepfakes' quality has been increasing and the performance of detection methods needs to be improved accordingly. The inspiration is that what AI has broken can be fixed by AI as well. Detection methods are still in their early stage and various methods have been proposed and evaluated but using fragmented data sets. An approach to improve performance of detection methods is to create a growing updated benchmark data set of deepfakes to validate the ongoing development of detection methods. This will facilitate the training process of detection models, especially those based on deep learning, which requires a large training set.

On the other hand, current detection methods mostly focus on drawbacks of the deepfake generation pipelines, i.e. finding weakness of the competitors to attack them. This kind of information and knowledge is not always available in adver- sarial environments where attackers commonly attempt not to

reveal such deepfake creation technologies. Recent works on adversarial perturbation attacks to fool DNN-based detectors make the deepfake detection task more difficult. These are real challenges for detection method development and a future research needs to focus on introducing more robust, scalable and generalizable methods.

Another research direction is to integrate detection methods into distribution platforms such as social media to increase its effectiveness in dealing with the widespread impact of deep- fakes. The screening or filtering mechanism using effective detection methods can be implemented on these platforms to ease the deepfakes detection. Legal requirements can be made for tech companies who own these platforms to remove deepfakes quickly to reduce its impacts. In addition, watermarking tools can also be integrated into devices that people use to make digital contents to create immutable metadata for storing originality details such as time and location of multimedia contents as well as their untampered attachment. This integration is difficult to implement but a solution for this could be the use of the disruptive blockchain technology. The blockchain has been used effectively in many areas and there are very few studies so far addressing the deepfake detection problems based on this technology. As it can create a chain of unique unchangeable blocks of metadata, it is a great tool for digital provenance solution. The integration of blockchain technologies to this problem has demonstrated certain results but this research direction is far from mature.

FUTURE SCOPE

Using detection methods to spot deepfakes is crucial, but understanding the real intent of people publishing deepfakes is even more important. This requires the judgement of users based on social context in which deepfake is discovered, e.g. who distributed it and what they said about it .This is critical as deepfakes are getting more and more photorealistic and it is highly anticipated that detection software will be lagging behind deepfake creation technology. A study on social context of deepfakes to assist users in such judgement is thus worth performing.

Videos and photo graphics have been widely used as evidences in police investigation and justice cases. They may be introduced as evidences in a court of law by digital media forensics experts who have background in computer or law enforcement and experience in collecting, examining and analysing digital information. The development of machine learning and AI technologies might have been used to modify these digital contents and thus the experts' opinions may not be enough to authenticate these evidences because even experts are unable to discern manipulated contents. This aspect needs to take into account in courtrooms nowadays when images and videos are used as evidences to convict perpetrators because of the existence of a wide range of digital manipulation methods. The digital media forensics results therefore must be proved to be valid and reliable before they can be used in courts. This requires careful documentation for each step of the forensics process and how the results are reached. Machine learning and AI algorithms can be used to support the determination of the authenticity of digital media and have obtained accurate and reliable results, e.g. ,but most of these algorithms are unexplainable. This creates a huge hurdle for the applications of AI in forensics problems because not only the forensics experts oftentimes do not have expertise in computer algorithms, but the computer professionals also cannot explain the results properly as most of these algorithms are black box models. This is more critical as the most recent models with the most accurate results are based on deep learning methods consisting of many neural network parameters. Explainable AI in computer vision therefore is a research direction that is needed to promote and utilize the advances and advantages of AI and machine learning in digital media forensics.

REFERENCES

[1] Deep Learning for Deepfakes Creation and Detection.
Thanh Thi Nguyen, Cuong M. Nguyen, Dung Tien Nguyen, Duc Thanh Nguyen, Saeid Nahavandi, *Fellow, IEEE (2020)*
[2] Faceswap: Deepfakes software for all. Available at https://github.com/deepfakes/faceswap
[3] FakeApp 2.2.0. Available at https://www.malavida.com/en/soft/fakeapp/
[4] A Survey on Deepfake Detection Techniques.
Bismi Fathima Nasar1*, Sajini. T 2 , Elizabeth Rose Lalson
[5] Deepfake Video Detection Using Recurrent Neural Networks
David Guera Edward J. Delp (Purdue University)
[6] https://www.researchgate.net/figure/Deepfake-principle-Top-the-training-parts-with-the-shared-encoder-in-yellow-Bottom_fig1_330791801 (Uploaded by Vincent Nozick)
[7] https://www.extremetech.com/extreme/298831-deepfake-tech-can-now-anonymize-your-face-to-protect-privacy(uploaded by Adam Dachis)
[8] Deep Fake Detection using Neural Networks

Anuj Badale, Chaitanya Darekar, Lionel Castelino, Joanne Gomes

[9] Deepfake Video Detection Using Convolutional Neural Network Aarti Karandikar, Vedita Deshpande, Sanjana Singh, Sayali Nagbhidkar, Saurabh Agrawal Shri Ramdeobaba College of Engineering and Management.

[10] Deepfakes Detection Techniques Using Deep Learning
Abdulqader M. Almars College of Computer Science and Engineering, Taibah University, Yanbu, Saudi Arabia.

[9] Deep Learning for Deepfakes Creation and Detection: A Survey Thanh Thi Nguyen, Quoc Viet Hung Nguyen, Cuong M. Nguyen, Dung Nguyen, Duc Thanh Nguyen, Saeid Nahavandi, Fellow, IEEE

[10] DEEPFAKE DETECTION TECHNIQUES: A REVIEW