Detecting Deepfake Videos Using Euler Video Magnification

Rashmiranjan Das #19210554
Gaurav Negi #19210869
School of Computing, Dublin City University, Ireland
Email: rashmiranjan.das2@mail.dcu.ie, gaurav.negi2@mail.dcu.ie

***Abstract*—Recent advances in machine learning and artificial intelligence make it progressively hard to distinguish between genuine and counterfeit media, especially images and videos. One of the recent developments is the rise of deepfake videos, which are based on manipulating videos using advanced machine learning techniques. This involves replacing the face of an individual from a source video with the face of a second person in the destination video. This innovative idea is getting progressively refined which implies that deepfakes are getting progressively seamless and simpler to compute. Combined with the outreach and speed of social media, deepfakes could easily fool a huge number of individuals when depicting someone saying things that never happened and thus could persuade the masses in believing fictional scenarios, creating distress, and spreading fake news. In this paper, we examine a technique for possible identification of deepfake videos. Our approach uses Euler video magnification which applies spatial decomposition and temporal filtering on video data to highlight and magnify hidden features like pulsation and subtle motions. Our approach uses features extracted from the Euler technique to train a model to classify counterfeit and unaltered videos and compare the results with existing techniques.**

Key words: Deepfakes, Euler video magnification, Neural network, Generative Adversarial Network, Autoencoders.

I. Introduction

Deepfakes is a term that originated from Reddit at the end of 2017 and describes a technique for artificially manipulating video. Deepfakes immediately picked up a reputation in the media because they were applied to swap celebrities faces into pornography recordings which were shared on sites like Reddit [16]. Deepfakes can be recognised as a bigger threat in comparison to other techniques, as it permits users with relatively little computing experience in machine learning or programming to generate seamless fake videos. The availability of state of the art deep learning libraries such as TensorFlow [2] and Keras [8], with enough accessible training data of celebrities, helps in generating fake recordings whose quality is so good they can be very persuading [14].

The initial implementations of deepfakes relied on convolutional autoencoders [29], [30]. Images of both subjects are reduced to lower dimensions using an encoder and reconstructed using a decoder. This training is performed for both source and destination facial expressions. In order to perform a face swap, a trained encoder of the source is mapped with a decoder trained on the target subject's face. An upgrade to this technique is by adding a generative adversarial network (GAN) in the decoder [3], [12]. GANs consist of two modules, a generator and a discriminator. The task of the generator is to develop images resembling the source while the discriminator determines if the image is counterfeited. It is an iterative process, which makes deepfakes seamless as they are constantly learning.

The availability of such sophisticated techniques for deepfake generation in the hands of ordinary researchers and their possible exploitation by other persons have escalated concerns about their possible misuse. Applications such as Deepfacelab [27], FakeApp and OpenFaceSwap are GUI based tools made accessible to relatively untrained researchers to create deepfake videos. With these tools, it becomes progressively possible for video evidence to be altered for political tension, false video evidence and fake news. Hence, this poses a challenge for society and demands an effective technique for the detection of such counterfeit video.

In order to adequately advance research into the detection of deepfake videos, we start by looking at features achieved by a technique called Euler Video Magnification (EVM). We perform three examinations using a structural similarity index, InceptionV3, and LSTM-based convolutional neural networks with heartbeat analysis to check whether our method can perform better than existing techniques.

II. Related work

One of the first papers on digital face-swapping was by Dmitri Bitouk [6], where the authors sought to find a face similar in appearance to an input face in a database and then concentrated on perfecting the mixture of the found face in the input image. The principal reason for this work was to de-identify an input face and protect its anonymity. Therefore, the approach did not require any two given faces to swap seamlessly. Before the current generation of neural networks, most of the techniques used for face swapping or facial reenactment were based on similarity searchers in target and in source video between faces or face patches and various blending techniques.

In 2017, Korshunova *et al.* proposed the first method that utilized a generative adversarial network to train a model between two pre-selected faces [18]. Another similar work with an even more ambitious concept was to use architecture based on long term memory (LSTM) to synthesise a mouth function solely from an audio speech [28]. These publications attracted a lot of attention and open-source approaches started to replicate these techniques which gave rise to the deepfake phenomenon.

With new work appearing regularly there is a stream of new approaches to generating deepfakes. In [32], the authors propose a CNN-based network for recognising altered facial pictures. The suggested solution includes a network that uses a pre-trained autoencoder to reconstruct images input on the original files. The restored one and the image input are then processed by the SRM filter which can delete the distribution of image noise. The minus result of two processed results is then inputted into a CNN architecture to predict whether the input image is original or tampered with.

Another interesting approach to deepfake detection focuses on psychological signals in the video [22]. The paper proposes a method of detecting deep fakes by observing eye blinking in videos, a psychological signal not well presented in synthesised videos. This method is based on a novel deep learning model combining a convolutional neural network (CNN) with a recursive neural network (RNN) that captures phenomenological and temporal regularities in the eye blinking process. Current methods employ Convolutional Neural Networks (CNN) as a binary classifier to distinguish the open and closed eye state of each frame. CNN'ss, however, produce predictions based on a single frame, which does not exploit the temporal domain information. Because human eye blinking has a strong temporal correlation with previous states, they used Long-term Recurrent Convolutional Neural Networks (LRCN) to distinguish between open and closed eye status, considering past temporal knowledge. This is a good observation as the training images used to generate deep fakes are usually obtained over the internet and do not usually include photographs of the subject with eyes closed. This identification can however, be circumvented by intentionally integrating photos into the training data with eyes closed.

The work in [20], exploits the colour disparity between GAN-generated images and real images in the non-RGB colour space to classify them. Again in the work reported in [25], authors analysed the colour difference between GAN images and real images. However, it is not clear if this approach can be applied to inspecting local areas, as in the deepFake case.

The paper which is most similar to our work is by Steven Fernandes [11], where the author investigates the heart rates of people in deepfake videos and real videos using Neural ODE. However, the objective of Fernandes' paper is to generate a heart rate from deepfake videos which are assumed to have no heartbeat. However, this approach may not perform very well in case of detecting deepfakes as deepfakes created online are usually not stable and are not under perfect lighting conditions, so it is very difficult to obtain a stable heartbeat.

At the time of writing the outputs from the Deepfake Detection Challenge Dataset (DFDC) introduced later in Section V, are not yet available but it is expected that a further set of new techniques will have been developed and submitted by DFDC participants.

III. Generative Adversarial Networks

Before going in-depth into a description of deepfake videos, we should understand the basic functioning of the components that contribute to the creation of such counterfeit videos. A Generative adversarial network (GAN) is the essential component that achieves a level of realism using a generator that creates data similar to the training data set and a discriminator that tries to identify if the generated data is real or fake. GANs were first introduced by Ian Goodfellow in 2014 [13]. A generative model is pitted against an adversary, a discriminator where the model learns to determine whether a sample is from a model distribution or the data distribution. This is shown in Figure 1.

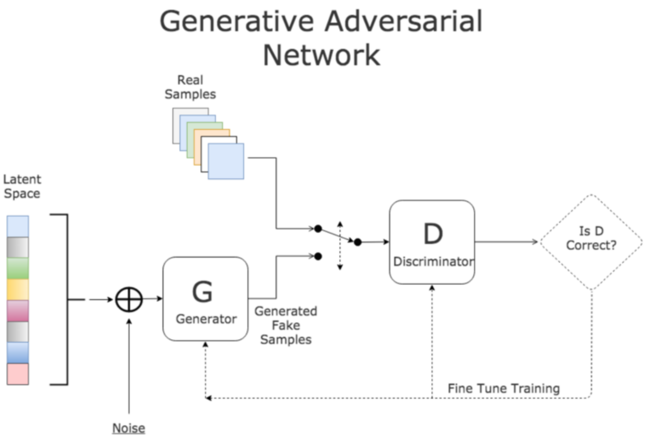


Figure 1. GAN Architecture

GANs are a generative modelling approach which means that they create new data from given data, while most of the other machine learning models that we see are discriminative models. When training a generative model on a dataset which is a fully unsupervised process as there are no labels on the data, the models discover the underlying structure of the dataset. Once the model has discovered these structures, this information can be exploited for other applications such as generating new data.

In essence, there are two neural networks at play in a GAN, a generator and a discriminator. The generator receives a random sample noise vector as input. In most of the cases, the noise is extracted from a Gaussian distribution. These noise vectors follow through a set of convolutional layers and output an image. These images are then fed to a second neural network called the discriminator. The discriminator's task is to classify the received image as real or fake. As the training process is controlled, the images are labelled, that it is from the real image or the fake image from the generator. The next step is to propagate the training loss into the network to make it better. This generator network itself is a fully differential neural network so if these two signals are stuck back to back, the learning signal can be backpropagated through the entire model pipeline and this way one can update with the same single loss function. The discriminator and the generator network can be updated until both the networks perform well. The most critical aspect of the pipeline is to ensure balance in both the networks.

After sufficient training, the generator will learn from the feedback of its discriminator network and eventually manage to generate images that look very similar to the data set which was used for training.

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))].$$

Figure 2. Min Max function [13]

The generator and the discriminator are trained on a min-max objective function shown in Figure 2. In this function, we are minimising for the generator but maximising for the discriminator. The discriminator network outputs a single scalar value of x called $D(x)$ per image which indicates how likely it is that the image X is, in fact, a real image coming from the dataset. Similarly, for the generated images coming from the generator called $G(z)$ where z is the noise vector and G is the generator network. It outputs a score $D(G(z))$ of this fake image. While training it is expected that the discriminator recognises a real image x as real so a high value as output i.e. close to 1, is expected. At the same time, we want to recognise fake images G of Z as being fake and so outputting a low value close to 0. The generator and discriminator functions are shown in Figure 3.

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log D\left(\boldsymbol{x}^{(i)}\right) + \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right) \right].$$

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right).$$

Figure 3. Generator and Discriminator functions [13]

The first part of the objective function does not rely on the parameters of the generator. So when we are optimising the generator, then the second part of the objective function is optimised. Every time we calculate the objective function, it is necessary to minimise it with respect to the generator parameter and to maximise it with respect to the parameters of our discriminator.

So at every step of the algorithm, there are two inputs, a sample batch of random noise vectors and sample batch of images. The objective function is used to update the parameters of the discriminator by performing stochastic gradient descent with respect to its parameters. While updating the discriminator, the generator network is fixed.

Once the discriminator has been updated for several time steps, then the discriminator is frozen and we move to the next step of training the generator. Here a new batch of noise vectors is sampled. An image is generated from them and gradient descent is applied in the second part of the objective function to update the parameters of the generator. There have been multiple iterations and changes since this algorithm was initially proposed to make sure that the objective function converges nicely and smoothly.

Being a generative model, GAN applications are widespread and is rapidly being implemented in many domains. Some of the popular applications are prediction of the next frame in a video using a dual video discriminative GAN [4], text to image synthesis with the help of object driven attentive GAN [21], enhancing the resolution of an image using Super-resolution GAN [36] and image to image translation.

IV. Euler Video Magnification

Eulerian Video Magnification (EVM) [34] is used to uncover fleeting and hidden details in videos that will, in general, be hard to see. It is a method to magnify and visualise temporal and/or colour variations in videos that are difficult to see with the naked eye. The method combines spatial and temporal processing to emphasise subtle temporal and spatial changes which occur naturally are encoded within the video but are never actually seen when viewed. For instance, one can enhance the slight colour changes in videos which include exposed human skin such as around the face where the capillaries in the skin show pulsing due to the bloodstream and blood flow changes caused, in turn from heart rate. EVM can also magnify low degrees of movement without having to perform image segmentation.

So how does EVM intensify video features and visualise them given it is difficult to see with the human eye? The basis is that a portion of these difficult-to-see changes happen at specific fleeting frequencies that we can expand utilising a static window in the frequency space. For instance, in a plucked guitar, each guitar string resonates at a different frequency, so to amplify the vibration one can look at the pixel variation in respective frequencies of the note being played. Similarly to amplifying human pulse visualisation in exposed skin like around the face, one can consider pixel changes in frequencies somewhere in the range 1.0 to 3Hz and to process this as it resonates to between 60 and 180 beats per minute.

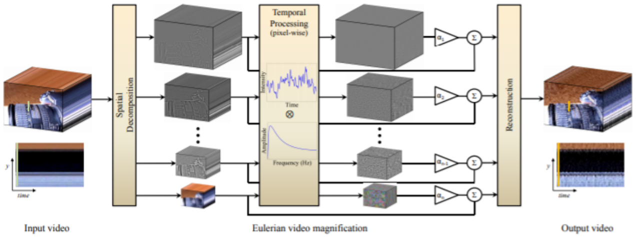


Figure 4. Euler magnification architecture [34]

The Eulerian amplification process consists of several key steps shown in Figure 4. A video is decomposed into images and this sequence of images is broken down using a Laplacian pyramid into various frequency ranges. The temporal changes concerning pixels in all frequency ranges of the Laplacian pyramid are bandpass filtered to select important and relevant frequency bands. These selected frequency bands are amplified by a magnification factor and this outcome is added to the respective signal. These amplified signals which belong to different frequency bands in the Laplacian pyramid are flattened to generate the last yield. The key attribute is the temporal frequency band which can be specified by adjusting the high and low cutoff frequencies for the filter.

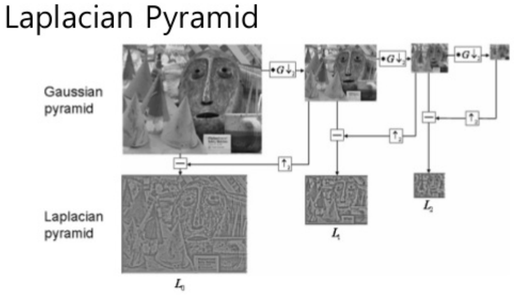
The initial step is to take a standard video sequence as an input and decompose it into different spatial frequency bands using a Laplacian pyramid. The "pyramid" is built by over and over computing a weighted normal of the neighbouring pixels of a source picture and downsizing the picture. It may be pictured by continuously stacking small variants of the picture on top of each other. This procedure makes a pyramid shape with the base as the first picture and gradually compressing it as the pyramid rises.

This pyramid is developed by taking the distinction between adjoining levels of a Gaussian pyramid and take the second derivative to enhance areas with frequent changes.

The Gaussian pyramid is a model through which an image is decomposed into compressed images with a small size of pixels. The Gaussian pyramid is a blurring technique used as a pre-processing step in the field of computer vision. Johann Carl Friedrich Gauss was a German mathematician and physicist who was first to come up with this technique [1].

Gaussian pyramids are used in the field of computer vision as the primary key of the multi-layer portrayal of image features.

A Laplacian pyramid is shown in Figure 5 and is fundamentally the same as a Gaussian pyramid, the only difference is that it saves the transition image between two levels. [7] Each level in the pyramid is a unique spatial frequency. The original image can be constructed by merging all the distinction pictures on each elevated level. Changing the temporal value of each pixel across all levels of the Laplacian pyramid is considered. The sequence of pixel values is considered overtime and applies a temporal bandpass filter to extract the frequency band of interest. We then perform a Fast Fourier Transform on the data followed by applying a bandpass filter. Then we extract the frequency band of interest and amplify the signal and add it back to the source data. The key is to figure out the appropriate parameter to extract ideal amplification. Adjusting the amplification factor of the bandpass signal results in a larger boost to the temporal bandpass. Changing these parameters can make variations in the scene more apparent but large amplification can add artefacts to the result. The following picture shows how this technique can be used to visualise imperceptible variations in videos, for example, a human pulse using the colour amplification technique.



http://cs.brown.edu/courses/csci1430/2011/results/proj1/georgem/explained.jpg

Figure 5. Gaussian and Laplacian pyramid [7]

By flattening the Laplacian pyramid by for each frame, the final amplified video will be received. The Euler magnification technique selects and amplifies a narrow band of temporal frequencies around the human heart rate and reveals the variation of redness as blood flows through capillaries in exposed areas of the face, such as the cheeks. This process can be robust to different skin tones and small motion of the subject. The colour amplified video can be used to extract vital signs from standard videos, for example, extracting heart rate measurement of a patient admitted to a hospital through contactless video.

Interestingly temporal colour amplification can also be used to amplify spatial motion considering, for example, a 1-D illustration in Figure 6. This shows a translated wave in two-time instants for a smooth image in small motion one can filter the signal temporally and get a good approximation to the translated wave. By amplifying the temporal signal, a larger translation can be approximated, essentially producing a motion amplified sequence.

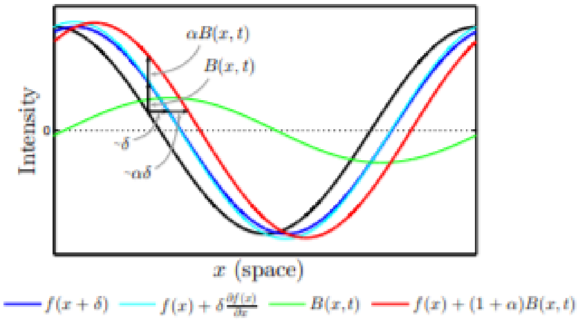


Figure 6. 1-D Signal undergoing translation motion [34]

Motion amplification can be useful for example to visualise artery pulsation or for contact-less tracking of a baby breathing in an intensive care unit or it can be used to simply make changes in an everyday video such as buildings swaying in the wind, more apparent.

V. Data Used in this Work

An important component of our experiments is a data set ideally comprising of a huge amount of recordings containing human faces with respective labels associated with them describing if the video recordings were deepfake or real. Various data sets including counterfeited faces are available such as Celeb-DF [23], UADFV [35] and DeepFake-TIMIT [17]. We selected Kaggle's Deepfake Detection Challenge Dataset (DFDC) [10][1] [9] for our experiments as it consists of videos with varied lighting conditions and head poses, and participants were able to record their videos with any background they desired. The DFDC dataset was created as part of the Facebook AI-sponsored challenge during 2020 also the participants in the videos are from varied ethnicity and have different skin tones, genders and techniques used for video manipulation. This variance in the attributes will be essential to determine the efficiency of our experiment.

One key factor which differentiates DFDC from other datasets is that actors have agreed to participate in the creation of the dataset which uses and modifies their likeness. The approximation of the general distribution of gender and race across this data set is 74% female and 26% male; 68% Caucasian, 20% African-American, 9% East-Asian, and 3% South-Asian. The development of this dataset does not use any publicly accessible data or data from social media sites.

Many state-of-the-art methods were applied to generate videos in the DFDC dataset [10], using the whole range of manipulation techniques to generate such tampering with the intent of representing the real adversarial space of facial manipulation, no further details of used methods were provided to the participants in the challenge. All the clips that comprised the training set were left at their original resolution and quality, so deriving appropriate augmentations of the training set is left as an exercise to researchers. The DFDC dataset includes 400 videos in the training dataset and 400 videos in the test dataset [10], divided into multiple sets of 10GB for downloading

Along with the above DFDC data set, we generated our own set of deepfakes created using the Deepfacelab application. The participants' consent was taken and participants signed an agreement approved by the DCU School of Computing Research Ethics Committee. A set of 30 participants were chosen and a video of 10 seconds was submitted by each of the participants. These videos were recorded in a controlled environment with suitable lighting at 1080p resolution and the participants remained quite still unlike some of the DFDC videos. Several face swaps were performed using the H64, H128 and SAE techniques [24].

These are autoencoder techniques that reduce the data to smaller dimensions, for example, the H64 model compresses the data into 64x64 pixels. Each face swap video was trained for 30,000 epochs with a mean loss of 0.0630. All these videos are of the same dimension as in DFDC dataset with 30 frames per second. The duration of the videos is 10 secs. A labelled data set was then created by merging the above data sets and this was used in our experiments. All the processing was done in IdeaPad L340 with an NVidia graphic card GTX 1650 and 8 cores, Cuda enabled version 10.1.

VI. Methodology

Given the challenges with deepfake videos, this paper explores the use of Euler video magnification as a way to pre-process video and use the result as an indicator of whether a video is a deepfake or not and we set out to evaluate the techniques' precision in identifying such videos. The essential objective of this research is to consider a deepfake video and its corresponding original video and to perform colour and movement amplification on the videos to see if the resulting differences can distinguish between them. These set of data are treated as the source to our model to evaluate if EVM can be exploited for deepfake detection.

A related technique to our work is photoplethysmography (PPG) [26]. This is a process which can be utilised to identify fluctuations in blood volume. The process of obtaining PPG is to shine the light of a given wavelength onto the skin and to measure changes in light assimilation. The pumping of heart drives blood to the peripheries in an oscillation cycle and it is the differences in the colour of oxygenated blood that causes these changes in light assimilation. To calculate the blood volume change caused by pulsation,the skin is enlightened using a light source like an LED and afterwards, we estimate the quantity of light reflected from the surface. Blood perfusion helps in detecting the components of a heart rate, due to richness in the blood flow through the skin of the face, and it becomes easy to artificially enhance and capture such subtle changes. Along with pulsation, the Euler

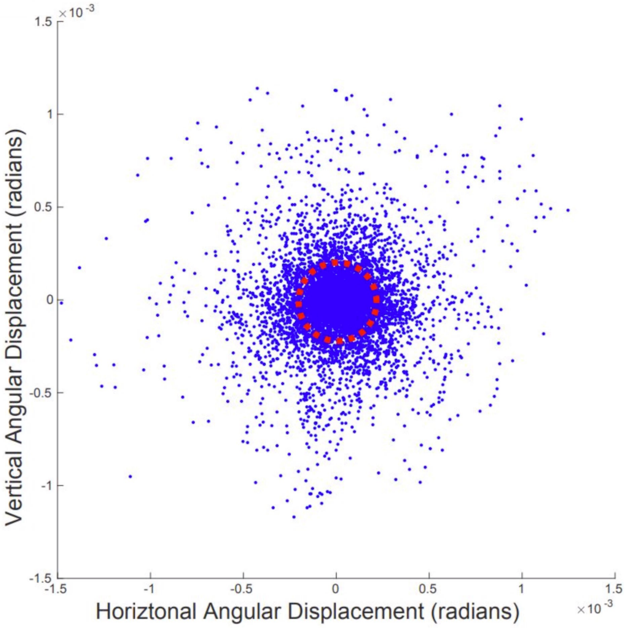


Figure 7. Horizontal and vertical displacement of camera shots taken in burst mode and caused by natural tremors, taken from [33]

[1] https://ai.facebook.com/datasets/dfdc/

magnification technique can also be used to detect and to magnify tiny motions by the subject in the video. One of the characteristics that are present with people while clicking a video or a photograph is natural tremor. These tremors are a naturally occurring oscillatory motion which is frequent but not observable to the naked eye due to the very small amplitude of these motions. Their recurrence is within the scope of 8 to 12 Hz. These periodic motions have been observed to stay with age and are caused by the compressions that are caused in muscles of the limbs.
Figure 7 taken from [33] is a graph that displays vertical and horizontal displacements that occurred naturally to a participant while taking a series of burst shot images with a smartphone camera where the displacements were caused by natural human tremors. The graph was based on 86 burst shots taken and the circle formed with red dots marks one standard deviation of movement in any direction. The observation that can be extracted from the graph is that human tremors are symmetrically distributed across all directions. Using Euler magnification, these small motions observed due to natural tremors might be magnified by Euler magnification so that they may be detectable as a differentiator between real and deepfake videos. The complete progression of our implementation can be divided into stages as shown in Figure 8. The initial step of the research is to accumulate a dataset. Data was gathered from the Kaggle dataset to represent standard video format. The videos were recorded on 30fps for a duration of 10 seconds. The videos were shot on varied backgrounds and in different lighting conditions.

The key advance of our task is to perform Euler amplification on deepfake videos, which are composed of superimposed appearances of source video onto a destination video. We converted the Euler magnification model which was developed in MatLab by an MIT research team into a Python function for our implementation with various key functions like the Gaussian kernel, bandpass filter and amplification factor. A Gaussian kernel is defined as a technique to break down frames of each video into a compressed hierarchy of pixels. Each level of the pyramid is of different frequencies. The number of levels in the pyramid is characterised by the user. Later a bandpass filter is applied which takes a signal as input along with parameters like temporal sampling interval and frequency range. This produces a band passed signal with frequency components within the range specified. The bandpass signal is magnified with an amplification factor α and is added back into the original signal.

These processed videos are then projected to three different models to extract the features which are used for video classification.

The first model to differentiate authentic and counterfeit videos is to compare the Euler magnified versions of the respective videos, extract features and create a classification model. One of these features crated is an SSIM plot. SSIM is a structural similarity index that compares two frames and shares a similarity probability between 1 and 0 where 1 means two frames are exactly similar. SSIM is calculated based

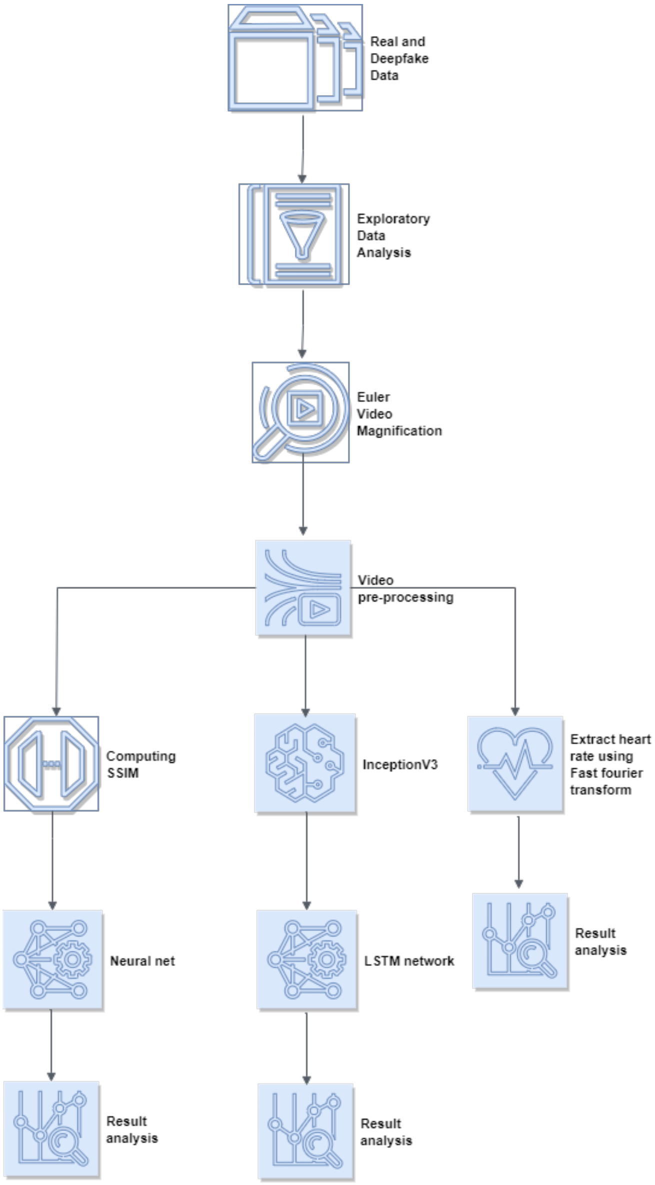


Figure 8. Flow graph of our implementation

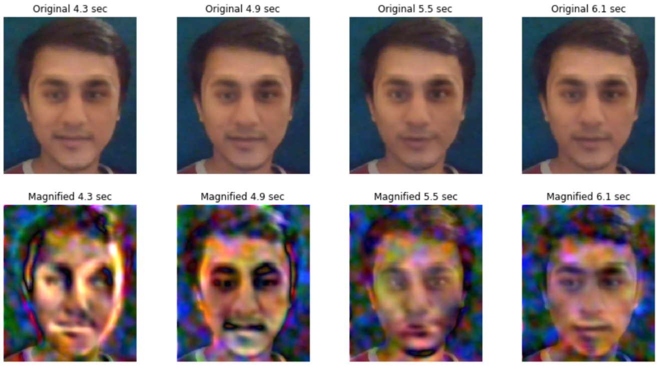


Figure 9. EVM on deepfake video

on three major components namely luminance, contrast and structure. After comparing the plot on the original video and the Euler magnified video it can be seen in Figure 9 that Euler magnification enhances the inconsistency in the frames of deepfake videos. SSIM is defined as Figure 10 [31].

$$\mathrm{SSIM}(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}$$

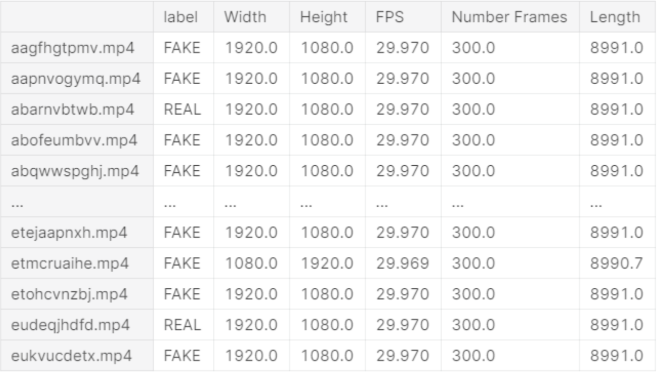
Figure 10. structural similarity index of two windows x and y of common size N×N. μ is the average value of (x and y). σ^2 is the variance. σ is the covariance of x and y. c1 and c2 two variables to stabilise the division with weak denominator.

Secondly, as seen by the amount of success achieved by CNN models in video analysis, adding a Long Short Term Memory (LSTM) network, a type of Recurrent Neural Network (RNN) into a neural network architecture can be used to learn any long term dependencies in the data sequence. The LSTM is coupled with the inception module to learn discriminative features from video frames [19].

When computing Euler magnification for a deepfake video, it was observed that deepfake videos also exhibit pulsation as seen in Figure 9. Our third technique was to calculate the heartbeats of both videos and analyse if any changes could be observed between them. A comparison of heartbeat based on pulsation observed in the Euler magnified video of the original and the deepfake revealed that beats per minute (BPM) for both the videos were very similar, with slight changes only in the first decimal position.

VII. Exploratory Data Analysis

We carried out an exploratory data analysis for video files which was required for extracting video features. One of the most well-known and powerful libraries for this is OpenCV which provides functions for video pre-processing. Video features like FPS, width, height,the number of frames and length are analysed as shown for some of our sample videos in Figure 11.



	label	Width	Height	FPS	Number Frames	Length
aagfhgtpmv.mp4	FAKE	1920.0	1080.0	29.970	300.0	8991.0
aapnvogymq.mp4	FAKE	1920.0	1080.0	29.970	300.0	8991.0
abarnvbtwb.mp4	REAL	1920.0	1080.0	29.970	300.0	8991.0
abofeumbvv.mp4	FAKE	1920.0	1080.0	29.970	300.0	8991.0
abqwwspghj.mp4	FAKE	1920.0	1080.0	29.970	300.0	8991.0
...	...	...	...	...	...	...
etejaapnxh.mp4	FAKE	1920.0	1080.0	29.970	300.0	8991.0
etmcruaihe.mp4	FAKE	1080.0	1920.0	29.969	300.0	8990.7
etohcvnzbj.mp4	FAKE	1920.0	1080.0	29.970	300.0	8991.0
eudeqjhdfd.mp4	REAL	1920.0	1080.0	29.970	300.0	8991.0
eukvucdetx.mp4	FAKE	1920.0	1080.0	29.970	300.0	8991.0

Figure 11. Video analysis

The most common observation is that the length of the video is either 299 or 300 frames and the fps varies slightly. The variation in width and height has stemmed from flipped axes. The size of the video is large and hence a generator is required to concurrently load data for training. The library used for reading the data is Keras. In Keras, the custom ImageGenerator is used to subclass sequence. In the dataset, around 19.25 % or 77 of the samples have missing original file names. For most of the missing data, the videos which are labelled as "REAL" data have missing original files.

We observe that the original label has the same pattern for unique values. There are 323 original labels for data and we observe that we have 209 unique examples.

We see that the most frequent label in the metadata is of FAKE videos which account for 80.75% of the total collection. The video file "meawmsgiti.mp4" is labelled as an original video and for most of the counterfeit videos, the number of deepfakes generated from it is around 6 videos. The data set consists of 19.25% of REAL videos which have not been adulterated, With the FAKEs accounting for 80.75% of the samples. There is a huge imbalance between the categories as shown in Figure 12 due to which a model might be biased to categorise videos as counterfeits hence the data needs to be up-scaled to balance it.

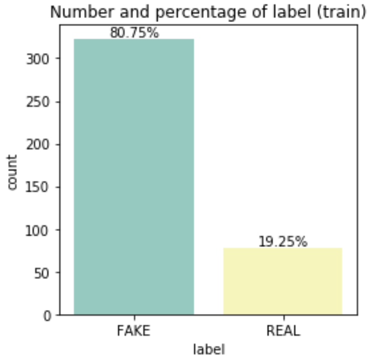


Figure 12. Video labels

We observe that in some cases, when the subject is not looking frontally at the camera or when the luminosity is low, the algorithm for face detection does not detect the face or eyes correctly. Due to a large number of false positives, we deactivated, for now, the smile detector.

OpenCV is used to detect the locations of faces in videos using the face recognition package. The video is divided into small frames of the face only and then re-merged into a new video. As seen in the illustration frames in Figure 13 there are no faces in these frames which will make it difficult for a classifier to learn whether it is REAL or FAKE. Hence, such frames are removed from the training set.

For the initial run, there were glitches in frames observed for videos from a particular source. On analysing such videos, we discovered that faces were not being detected in some of the frames or throughout the video. This was due to the absence of light or because there was a side profile of the subject. We had to remove such videos as data with noise and allow the model to extract better features.

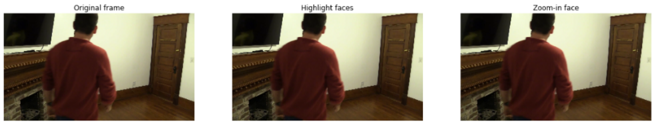


Figure 13. Video with no faces in the frames

VIII. Experiments

We now describe three complimentary techniques we used to pre-process videos before determining if they are real or deepfakes.

A. Experiment 1

The first of the technique we used is SSIM which is a comparison technique used to compare two frames and evaluate their likeness and calculate a similarity index based on visual structures. The Structural Similarity Index (SSIM) is a perceptual metric that quantifies image quality degradation caused by processing such as data compression or by losses in data transmission, and for deepfake videos, a quality degradation in frames are due to a less well-trained neural net. It is a full reference metric that requires two images from similar image capture.

SSIM is based on visible structures within the image. We extracted an SSIM index graph on videos to plot their inter-frame similarity. The idea is to compare the current frame with the previous frame so that it can highlight structural faults and inconsistencies. These inconsistencies in frames are due to pixelated faces and the irregularities have been intensified and magnified as a result of Euler magnification where the spatial amplification factor of EVM magnifies the irregularity and hence there are drops in the similarity index.

The graph in Figure 14 shows the inter-frame dissimilarities created from an original, a deepfake, an Euler magnified deepfake and an Euler magnified original video.

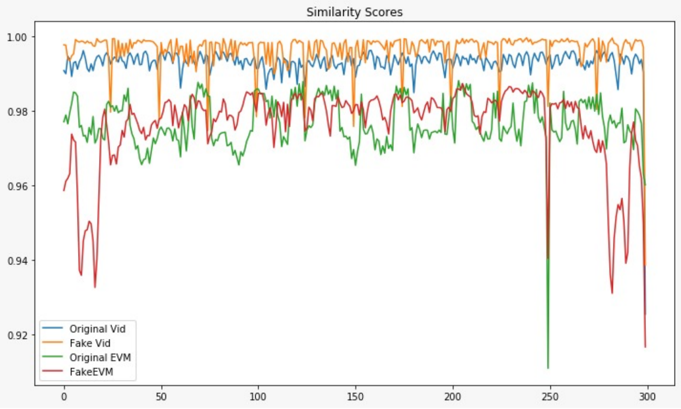


Figure 14. Inter frame dissimilarity for original, deepfake, Euler magnified deepfake and Euler magnified original videos

B. Experiment 2

The idea of using a long short term memory (LSTM) network within a neural network architecture is to help the model learn long term dependencies across the data series. LSTM networks were first presented by Hochreiter & Schmidhuber in 1997 [15] and their original idea has been upgraded numerous times. The LSTM model's primary objective is to recollect information over an extensive stretch of time. Unlike an RNN with its single tanh layer, LSTMs have four strategically arranged modules. Hence the LSTM has been used by us for our classification task.

Inception V3 includes an Inception module where there are changes in the spatial convolutions to depth-wise separable convolutions. Our model is build using CNN network layers for feature extraction followed by an LSTM layer for temporal analysis of Euler magnified videos. Our network has fully connected layers and a dropout layer to make sure that there is no over-fitting. The total number of trainable parameters used is 5,500,898 and these are used as input for the LSTM network and 2 node network working as a detector for deepfake videos from original videos.

To obtain the ground truth, the neural network was trained on videos which were not modified by the Euler magnified method. The hidden layer had a 'relu' activation function, while the last layer had 'softmax' as the activation function. We calculated the loss and accuracy of the model on both the training and test sets. We performed the same sequential steps on the same sets of the video but on the Euler magnified form of the video for comparison on how the technique compares to the standard classification model. We used an LSTM network with 512 widths and dropout of 0.5 to randomly set values of outgoing edges of hidden layers to zero. The last layer is constructed using a softmax activation layer to predict video class.

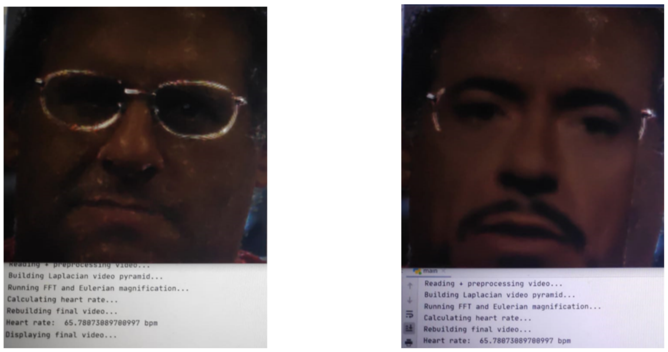
C. Experiment 3

The main hypothesis behind our work is to Euler magnify both the source and counterfeit videos in order to extract features which would highlight skin pulsation by a subject in the videos. Through pulsation, we can easily visualise and thus are able to extract the heartbeat of the subject in the video by calculating the number of colour change peaks and counting each one as a heartbeat.

The Euler video magnification technique can amplify spatial as well as the temporal aspects of a video. Spatial magnifies the motion while temporal magnifies colour changes on skin tone. We use the temporal aspect to visualise the pulse on exposed facial skin. The features of this that are customisable are filter type, magnification factor and range of frequency. The videos were subjected to a frequency range of 1Hz (60 BPM) to 1.33 Hz (80 BPM). The amplification factor was set to 50. To fetch heartbeat, a Fast Fourier transform algorithm was used. The temporal signal was transformed into a frequency domain to fetch the signal measured in hertz.

As seen in Figure 15, the heart rate of both the original and deepfake videos came to 65.78 beats per minute (1.096 Hz).

Thus the temporal variance calculated through Euler video magnification is insufficient to differentiate deepfake from original videos [5].



```
Reading + preprocessing video...
Building Laplacian video pyramid...
Running FFT and Eulerian magnification...
Calculating heart rate...
Rebuilding final video...
Heart rate:  65.78073089700997 bpm
Displaying final video...
```

```
Reading + preprocessing video...
Building Laplacian video pyramid...
Running FFT and Eulerian magnification...
Calculating heart rate...
Rebuilding final video...
Heart rate:  65.78073089700997 bpm
```

Figure 15. Heartbeat calculation using Fast Fourier transform on EVM of Original and Deepfake video

IX. Results

The results of our experiments reported below were produced by using python 3 code on a device with 8 GB RAM and 4 core Ryzen 5 AMD processor. We used 400 sets of videos from the DFDC Kaggle dataset and 30 assembled directly by us. We generated multiple datasets by changing parameters of Euler magnification, notably amplification factors, and generated 5 datasets. The complete research sequence can be divided into the following steps

1) Create metadata of the videos extracted from multiple sources;
2) Detect faces and crop video to leave only the face in the video;
3) Euler magnify the video with a specific frequency range and amplification factor and store it separately;
4) Train a model on Euler magnified video.

A. Experiment 1:

As observed from multiple graphs of SSIM scores for videos, there were more similarity score drops for counterfeit recordings when compared to their real video counterpart which was magnified by the Euler magnification process. After magnifying the data using EVM, we calculated similarity scores for all videos in the dataset. Deepfake video detection is particularly difficult to train as the manipulation can be observed only on a few frames and it is restricted to certain areas of the face. When there is much movement in the video, there are inconsistencies and important areas in the frame appear only briefly. Below are the results we obtained on various machine learning models:

Technique	L regression	Decision tree	NNet
Original Data	68.7%	65%	77%
EVM Data	53.7%	62%	70%

In the table below, we tested video with Euler magnification with multiple amplification factors (15, 20, 30, 40 and 50). The frequency band was restricted to between 0.8hz and 1.0 Hz. As we can see the lower amplification factor performed better than the higher amplification factor. In these sets of videos, a higher amplification factor led to additional noise which blocked some of the features in the videos. This indicates that Euler magnification is introducing noise as the amplification factor increases. The accuracy score for classification using the original videos is better than when using Euler magnified video.

Amplification factor	Accuracy	Loss
10	70.24%	0.6036
20	68.36%	0.6043
30	65.77%	0.6051
40	63.54%	0.6243
50	60.494%	0.6189

B. Experiment 2:

The Inception v3 inspired LSTM model ran for 100 epochs to give the following results on the test set. Once again we found that classification using the original videos outperformed classification using EVM videos on 30 amplification factor thus refuting the idea of EVM as a promising feature.

Video set	Accuracy	Loss
Original	77.24%	0.88
EVM	61.79%	2.52

C. Experiment 3:

Subjects appearing in both deepfake videos and original videos have a heartbeat which refutes the hypothesis of difference in pulsation and heart rate. As seen in experiment 1, the beats per minute for both the video types, real and deepfake remain the same. This reveals that the GAN used for creating the deepfake does not simply superimpose a new image of the subject on top of the real image which would have concealed facial pulsation colour changes, but the GAN manages to faithfully model the true data distribution of the real data at the pixel level and in this way it keeps the spatial-temporal changes from the genuine video, intact.

X. Conclusion

In this paper, we tested the effect and impact of Euler magnification as a technique for possible detection of deepfake videos. Both the temporal and the spatial aspects of the Euler magnification were tested as possible features for a number of classification models we built for discriminating between real and deepfake. Our experimental results with a large variety of balanced datasets including a dataset we created ourselves have not been able to demonstrate that the Euler magnification technique can not accurately predict a counterfeit video of time duration around 10 sec and at 30 fps.

We believe that our experimental results help to understand how deepfake videos incorporate pulsation information and tremor motion into the generated videos. Hence, for now, these features cannot be used as attributes for a deepfake detection technique. In future work, we would like to explore the most difficult facial objects to alter like lips and eyes, for fake detection.

References

[1] carl friedrich gauss.

[2] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: A system for large-scale machine learning. In *12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16)*, pages 265–283, 2016.

[3] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017.

[4] M. Beaulieu, S. Foucher, D. Haberman, and C. Stewart. Deep image- to-image transfer applied to resolution enhancement of sentinel-2 images. In *IGARSS 2018 - 2018 IEEE International Geoscience and Remote Sensing Symposium*, pages 2611–2614, 2018.

[5] Stephanie L Bennett, Rafik Goubran, and Frank Knoefel. Adaptive eulerian video magnification methods to extract heart rate from thermal video. In *2016 IEEE International Symposium on Medical Measurements and Applications (MeMeA)*, pages 1–5. IEEE, 2016.

[6] Dmitri Bitouk, Neeraj Kumar, Samreen Dhillon, Peter Belhumeur, and Shree K Nayar. Face swapping: automatically replacing faces in photographs. In *ACM SIGGRAPH 2008 papers*, pages 1–8. 2008.

[7] Peter Burt and Edward Adelson. The laplacian pyramid as a compact image code. *IEEE Transactions on communications*, 31(4):532–540, 1983.

[8] François Chollet et al. Keras. https://keras.io, 2015.

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

[10] Brian Dolhansky, Russ Howes, Ben Pflaum, Nicole Baram, and Cristian Canton Ferrer. The deepfake detection challenge (dfdc) preview dataset. *arXiv preprint arXiv:1910.08854*, 2019.

[11] Steven Fernandes, Sunny Raj, Eddy Ortiz, Iustina Vintila, Margaret Salter, Gordana Urosevic, and Sumit Jha. Predicting heart rate variations of deepfake videos using neural ode. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 0–0, 2019.

[12] Ian Goodfellow. NIPS 2016 tutorial: Generative adversarial networks. *arXiv preprint arXiv:1701.00160*, 2016.

[13] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative Adversarial Networks. *arXiv preprint arXiv:1406.2661*, 2014.

[14] David Güera and Edward J Delp. Deepfake video detection using recurrent neural networks. In *2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 1–6. IEEE, 2018.

[15] Sepp Hochreiter and Jürgen Schmidhuber. LSTM can solve hard long time lag problems. In *Advances in neural information processing systems*, pages 473–479, 1997.

[16] Marissa Koopman, Andrea Macarulla Rodriguez, and Zeno Geradts. Detection of deepfake video manipulation. In *The 20th Irish Machine Vision and Image Processing Conference (IMVIP)*, pages 133–136, 2018.

[17] Pavel Korshunov and Sébastien Marcel. Deepfakes: a new threat to face recognition? assessment and detection. *arXiv preprint arXiv:1812.08685*, 2018.

[18] Iryna Korshunova, Wenzhe Shi, Joni Dambre, and Lucas Theis. Fast face-swap using convolutional neural networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3677–3685, 2017.

[19] Akash Kumar, Arnav Bhavsar, and Rajesh Verma. Detecting deepfakes with metric learning. In *2020 8th International Workshop on Biometrics and Forensics (IWBF)*, pages 1–6. IEEE, 2020.

[20] Haodong Li, Bin Li, Shunquan Tan, and Jiwu Huang. Detection of deep network generated images using disparities in color components. *arXiv preprint arXiv:1808.07276*, 2018.

[21] W. Li, P. Zhang, L. Zhang, Q. Huang, X. He, S. Lyu, and J. Gao. Object-driven text-to-image synthesis via adversarial training. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12166–12174, 2019.

[22] Yuezun Li, Ming-Ching Chang, and Siwei Lyu. In ictu oculi: Exposing AI generated fake face videos by detecting eye blinking. *arXiv preprint arXiv:1806.02877*, 2018.

[23] Yuezun Li, Xin Yang, Pu Sun, Honggang Qi, and Siwei Lyu. Celeb-df: A new dataset for deepfake forensics. *arXiv preprint arXiv:1909.12962*, 2019.

[24] Artem A Maksutov, Viacheslav O Morozov, Aleksander A Lavrenov, and Alexander S Smirnov. Methods of deepfake detection based on machine learning. In *2020 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus)*, pages 408–411. IEEE, 2020.

[25] Scott McCloskey and Michael Albright. Detecting GAN-generated imagery using color cues. *arXiv preprint arXiv:1812.08247*, 2018.

[26] Seyedeh Somayyeh Mousavi, Mohammad Firouzmand, Mostafa Charmi, Mohammad Hemmati, Maryam Moghadam, and Yadollah Ghorbani. Blood pressure estimation from appropriate and inappropriate ppg signals using a whole-based method. *Biomedical Signal Processing and Control*, 47:196–206, 2019.

[27] Ivan Petrov, Daiheng Gao, Nikolay Chervoniy, Kunlin Liu, Sugasa Marangonda, Chris Umé, Jian Jiang, Luis RP, Sheng Zhang, Pingyu Wu, et al. Deepfacelab: A simple, flexible and extensible face swapping framework. *arXiv preprint arXiv:2005.05535*, 2020.

[28] Supasorn Suwajanakorn, Steven M Seitz, and Ira Kemelmacher-Shlizerman. Synthesizing Obama: learning lip sync from audio. *ACM Transactions on Graphics (TOG)*, 36(4):1–13, 2017.

[29] Ayush Tewari, Michael Zollhofer, Hyeongwoo Kim, Pablo Garrido, Florian Bernard, Patrick Perez, and Christian Theobalt. Mofa: Model-based deep convolutional face autoencoder for unsupervised monocular reconstruction. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 1274–1283, 2017.

[30] Justus Thies, Michael Zollhofer, Marc Stamminger, Christian Theobalt, and Matthias Nießner. Face2face: Real-time face capture and reenactment of rgb videos. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2387–2395, 2016.

[31] Zhou Wang, Eero P Simoncelli, and Alan C Bovik. Multiscale structural similarity for image quality assessment. In *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*, volume 2, pages 1398–1402. Ieee, 2003.

[32] Lilong Wen and Dan Xu. Face image manipulation detection. In *IOP Conference Series: Materials Science and Engineering*, volume 533, page 012054. IOP Publishing, 2019.

[33] Bartlomiej Wronski, Ignacio Garcia-Dorado, Manfred Ernst, Damien Kelly, Michael Krainin, Chia-Kai Liang, Marc Levoy, and Peyman Milanfar. Handheld multi-frame super-resolution. *ACM Transactions on Graphics (TOG)*, 38(4):1–18, 2019.

[34] Hao-Yu Wu, Michael Rubinstein, Eugene Shih, John Guttag, Frédo Durand, and William Freeman. Eulerian video magnification for revealing subtle changes in the world. *ACM transactions on graphics (TOG)*, 31(4):1–8, 2012.

[35] Xin Yang, Yuezun Li, and Siwei Lyu. Exposing deep fakes using inconsistent head poses. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8261–8265. IEEE, 2019.

[36] Yufan Zhou, Haiwei Dong, and Abdulmotaleb El Saddik. Deep learning in next-frame prediction: A benchmark review. *IEEE Access*, PP:1–1, 04 2020.