Deepfake Video Detection

Shankar Bhawani Dayal and Brett van Niekerk
University of KwaZulu-Natal, South Africa
sbdayal15@gmail.com
vanniekerkb@ukzn.ac.za
DOI: 10.34190/EWS.21.110

Abstract: Deepfakes pose a threat to many aspects of society, such as election manipulation, involuntary pornography and fraud by means of identity theft. This paper aims to determine if deepfake models which are pre-trained on older datasets are still able to accurately detect whether a video is real or a deepfake from a newer dataset. From a comprehensive literature review, two papers were selected to be tested. The first model tested was from Afchar et al. (2018), was unable to run due to an error involving Keras, to no fault of the code. The model that was successfully tested on a sub-dataset was the XceptionNet model from the FaceForensics++ paper by Rössler et al. (2019). It was shown that the XceptionNet model was not able to effectively detect deepfake videos, having a 51.31% classification accuracy on the subdataset, further analysis of the results showed that it only had a 13.16% accuracy when detecting deepfake videos and it had 89.47% accuracy when detecting real videos. As the methods which are used to create deepfake material improve, the previous work which has been done will need to be tested on the material created by the newer methods to determine if they are still effective at detecting deepfakes.

Keywords: deepfake, FaceForensics++, generative adversarial networks, Xceptionnet, deepfake detection challenge dataset

1. Introduction

Deepfake media, such as videos and images, are the greatest form of crime created from artificial intelligence (Smith, 2020). The idea of deepfakes was first proposed on Reddit by a user called “deepfakes” (Fink and Diamond, 2020; Sample, 2020), who uploaded videos of his work which happened to be deepfake pornographic material of famous Hollywood actresses. This eventually resulted in Reddit banning the subreddit “r/deepfakes'' due to a policy change concerning involuntary pornography (u/landoflobsters, 2018).

Effective deepfake detection methods are required to combat the threats that deepfakes pose, and more importantly deepfake detection methods that can remain effective against newer deepfake creation methods and thus staying at least one-step ahead. Deepfakes are mostly created using Generative Adversarial Networks (GANs) (GoodFellow et al., 2020). GANs utilize two neural networks working against each other, where the “generator” network creates material trying to fool the second network, and the second network which is the “discriminator” attempts to spot the real or fake instances. This is the adversarial component of the network. The discriminator is essentially a classifier (Google Developers, 2019). The generator and discriminator learn from each other, since the generator's output is linked directly to the discriminators input, and when backpropagation is occurring, the discriminators prediction/output is used as a signal for the generator to update its weights (Google Developers, 2019).

1.1 Threats of deepfakes

Deepfakes pose a threat to many fields, such as politics/election manipulation, pornography, and financial fraud by means of identity theft. The use of deepfakes in election manipulation has already been seen in India in February 2020, where a deepfake video shows a politician criticizing the Delhi government at the time (Christopher, 2020; Jee, 2020). This deepfake video was distributed on WhatsApp and allegedly reached 15 million people (Christopher, 2020). Deepfake pornographic material consists of 96% of all online deepfake material, 99% of which consists of women who work in the entertainment industry (Paul, 2019). Deepfake pornographic material can be used for defamation and/or blackmail, and can be damaging to the victim. The creation and distribution of deepfake pornographic is already a crime in California, Virginia and Texas (Paul, 2019; Ruiz, 2020) and on 20th December 2019, President Trump signed the USA’s first federal law in relation to deepfakes (Fink and Diamond, 2020). Deepfakes have already been used for financial fraud, which occurred in 2019 where a UK energy firm was fooled into sending $240,000 to another company, who they believed was a legitimate Hungarian supplier (Damiani, 2019; Stupp, 2019).

1.2 Aims and objectives

The aim of this project is to validate the claimed accuracy of Rössler et al. (2019) and Afchar et al. (2018) by testing their pretrained models on a subdataset from the DFDC dataset (Dolhansky et al., 2020) available on Kaggle (2019). The belief is that the quality of the deepfakes in their datasets are not of high quality and therefore their results are high since the deepfakes are easy to detect. There is no doubt whether the models achieved the accuracy that is claimed (Afchar et al., 2018; Rössler et al., 2019); however, will these models achieve similar accuracies when tested on another deepfake dataset? Since these papers propose models that are able to detect deepfakes with a high accuracy, they should work on all types of deepfakes, which is what this project aims to validate.

The chosen evaluation method is classification accuracy, as in a commercial environment, the performance of an implemented prediction system will be measured by its classification accuracy. Prediction systems will do a tally of how many frames in the video were classified as real, and if that tally divided by the total number of frames is over a certain percentage, then the prediction will be real. This project aims to investigate how effective the deepfake models by Rössler et al. (2019) and Afchar et al. (2018) are when using different datasets. Rössler et al. (2019) and Afchar et al. (2018) used public datasets and are cited by many other papers (293 and 255 citations on Google Scholar, respectively); hence, this is a major factor in choosing to validate their results.

1.3 Paper structure

The paper is structured as follows; a literature review is presented, which gives a detailed explanation of existing deepfake models as well as existing datasets. The methodology section outlines the work done by this project. This is followed by a presentation of the results and discussion, and lastly conclusion and future work.

2. Literature review

There exists a variety of Deepfake Detection methods such as Faceforensics++ (Rössler et al., 2019) and MesoNet (Afchar et al., 2018). The summarised method of Faceforensics++ and MesoNet is using a face detection method to capture the faces from a picture or each frame of a video and then using a pre-trained Convolutional Neural Network (CNN) model to predict if that picture/frame contains a deepfake face/altered face. There are new novel detection methods such as eye blinking patterns (Jung, Kim and Kim, 2020).

Rössler et al. (2019) created a dataset called Faceforensics++ as well as another dataset in partnership with Google and Jigsaw (https://jigsaw.google.com/), which is a unit of Google that works on solutions to tackle threats to society. The FaceForencsics++ dataset was created with four types of methods, namely Face2Face (Thies et al., 2016), FaceSwap (MarekKowalski, 2018), Deepfakes (Deepfakes, 2020) and Neural Textures (Thies et al., 2019). Face2Face is a system that transfers the expressions of source video (source face) to a target video (target face) while maintaining the target's face. Essentially Face2Face puts the expressions from one person onto another person's face. FaceSwap is a "graphic-based approach" which transfers the face region from a source video to a target video (MarekKowalski, 2018). The deepfake method used in Rössler et al. (2019) is a method available on GitHub (Deepfakes, 2020). Thies et al. (2019) propose a method utilising neural textures, which are learned feature maps that are trained during the scene capturing process. They performed testing on a variety of existing models (Afchar et al., 2018; Bayar and Stamm, 2016; Chollet, 2017; Deepfakes, 2020; Rahmouni et al., 2017). The best performing model is XceptionNet (Chollet, 2017), they fine-tuned the model and trained and tested it on their dataset and achieved an accuracy of 99.26% of raw video footage cropped on the face, and 82.01% accuracy when tested on the raw full video footage (not cropped on the face).

In Afchar et al. (2018), two CNN models to detect Deepfakes were created. The first network, Meso4, has four layers, and the second network is MesoInception-4. MesoInception-4 differs from Meso4 by replacing the first two layers with a variant of the Inception model (Szegedy et al., 2017). In Afchar et al. (2018), they created their own dataset and they used an existing dataset. The deepfake dataset they created and the existing dataset is Face2Face (Thies et al., 2016). They tested both models on the Face2Face dataset at different compression levels. The first network, Meso4, achieved 89.1% accuracy on the Deepfake dataset and 94.6% accuracy for the Face2Face classification score at 0 Compression level, 92.4% accuracy for the Face2Face classification score at 20 Compression level and 83.2% accuracy for the Face2Face classification score at 40 Compression level. The second network MesoInception-4 achieved 91.7% accuracy on the Deepfake dataset and 96.8% accuracy for the Face2Face classification score at 0 Compression level, 93.4% accuracy for the Face2Face classification score at

20 Compression level and 81.3% accuracy for the Face2Face classification score at 40 Compression level (Afchar et al., 2018).

In Jung, Kim and Kim (2020), they propose a method, called DeepVision, to detect if a video is a deepfake or not by analyzing the eye blinking patterns, while taking into account other variables such as age, gender, time of day, type of activity (static or dynamic). Currently they do not have an automated system to gather these variables, they have to manually gather these variables by watching the video. In Jung, Kim and Kim (2020), they utilize two algorithms to capture the eye blinking, namely Fast-HyperFace (Ranjan, Patel, and Chellappa, 2019) algorithm and EAR (Eye-Aspect-Ratio) algorithm (Soukupova and Cech, 2016). The Fast-HyperFace algorithm is good at detecting faces but not adequate at detecting eye blinking, and the EAR algorithm is good at detecting eye blinking but not good at detecting faces. In Jung, Kim and Kim (2020), they tested DeepVision on eight videos, and it achieved an accuracy of 87.5%, seven out of eight videos were identified correctly.

Güera and Delp (2018) created a recurrent neural network to detect deepfake videos. The model uses a CNN for frame feature extraction and the convolutional long short-term memory (LSTM) network to perform sequence analysis on the extracted features. They assembled their dataset, consisting of 600 videos, where 300 real/pristine videos came from HOHA dataset (Hollywood Human Actions) (Laptev, 2020) and the 300 deepfake videos came from undisclosed sources/websites. Güera and Delp (2018) tested their network against 20, 40 and 80 frames from each video. They state that they have 96.7%, 97.1% and 97.1% accuracy respectively when they tested it.

Table 1: Results of previous deepfake detection models

Paper	Model	Dataset	Accuracy/PerformanceMeasure
Rössler, et al (2019)	XceptionNet	FaceForensics++	82.01% accuracy on rawuncropped video footage
Afchar, et al (2018)	Meso4	Deepfakes	89.1% accuracy at 0compression,
		Face2Face (Thies et al., 2016)	94.6% accuracy at 0 compression level, 92.4% accuracy at 20 compressionlevel and 83.2% accuracy at40 compression level.
	MesoInception-4	Deepfakes	91.7% accuracy at 0compression,
		Face2Face (Thies et al., 2016)	96.8% accuracy at 0 compression level, 93.4% accuracy at 20 compressionlevel and 81.3% accuracy at40 compression level.
Jung, S. Kim and K. Kim(2020)	DeepVision	DeepVision Dataset (8 videos)	87.5% accuracy
Güera and Delp (2018)	RNN	HOHA (Laptev, 2020) + undisclosedsources	96.7% accuracy with 20 frames per video, 97.1% accuracy with 40 frames pervideo and 97.1% accuracy with 80 frames per video

A recent challenge, Deepfake Detection Challenge (DFDC), on Kaggle (2019) finished recently which was hosted by AWS, Facebook, Microsoft, the Partnership on AI's Media Integrity Steering Committee (Lyons, 2019), the winner receives $500,000 and the four runner-ups receive a total of $500,000. The winner of the competition achieved a log loss score of 0.42798 or rather he achieved an accuracy of 65.18% (Ferrer et al., 2020; Seferbekov, 2020). The discrepancy between the accuracy of published models (Afchar et al., 2018; Güera and Delp, 2018; Jung, Kim and Kim, 2020; Rössler et al., 2019) and a winner of a worldwide competition with 2265 teams, begs the question of how accurate these systems truly are. The caveat is that these models did not have that many datasets available to them at the time considering that the concepts of deepfakes are still fairly new in the academia world. As a result, they had to create their own datasets, as seen in Rössler et al. (2019) and Afchar et al. (2018), and the ways that deepfakes are being created are only growing in performance and quality.

The DFDC dataset (Dolhansky et al., 2020) uses five methods to create the dataset, namely DFAE (Deepfake Autoencoder), MM/NN (Morphable Mask/Neural Network) face swap, Neural Talking Heads (NTH), Face Swap GAN (FSGAN) and StyleGAN. DFAE is how most of the deepfakes seen on the internet are made, they are seen

in off-the-shelf products/software such as the DeepFaceLab (iperov, 2020). MM/NN face swap performs face swaps with a custom frame-based morphable-mask model. MM/NN face swap works by computing the facial landmarks in the source image and the target image and the pixels from the source face/image are morphed to match the landmarks in the target image, the method was adapted from Huang and De La Torre (2012). This technique works best when both faces (target and source) have similar expressions, otherwise the resultant video will have obvious discontinuities in the face, seeing as how most deepfake detection methods detect on a frame-by-frame basis, this would not affect the prediction, but this would be evident to a human observer. To overcome the discontinuities that occur they used a nearest-neighbours approach on the frame landmarks in order to find the best source/target face pair, this means that not every frame is created with the same two faces. Zakharov et al. (2019) create deepfakes using a GAN architecture, it utilizes two training stages, the first training stage is a meta-learning stage and the second training stage is a fine-tuning stage. FSGAN is another model that uses a GAN architecture to create deepfakes for face swapping and re-enactment from a source video/face to a target video/face, while accounting for facial expressions and the pose of the individuals (Nirkin, Keller, and Hassner, 2019). They modified the StyleGAN model (Karras, T., Laine, S. and Aila, 2019) to fit their purpose to produce face swaps between a given fixed identity descriptor onto a video by projecting this descriptor on the latent face space, they did this for every frame of the video. Dolhansky et al. (2020) use the most methods to create the dataset, compared to the four methods used by Rössler et al. (2019) and the two methods by Afchar et al. (2018).

3. Methodology

The computer used for this project has:

- Operating System: Windows 10
- Processor: Intel(R) Core (TM) i5-7300HQ CPU @ 2.5GHz
- Graphics Card: Nvidia GeForce GTX 1050
- Installed RAM: 8GB
- System Type: 64bit Operating System, x64-based processor

3.1 Creating the dataset

The first step is creating the sub-dataset from the DFDC dataset. Since the DFDC is fairly large in size (471.84 GB zipped), the dataset is also segmented into smaller files (≈ 10GB zipped) to allow for smaller downloads. The first four segmented datasets were chosen, “00.zip”,”01.zip”, “02.zip” and “03.zip”. Each zipped folder contains a set of videos, real and fake, as well as a metadata.json file which lists the name of the video, the label of the video (“REAL” or “FAKE”), and if the video is fake, the source/original video. The “03.zip” folder got corrupted and due to data issues, the decision was made not to redownload the folder.

Since each video in the file is not labelled real or fake in the title, the next step is to separate them into two folders, a “real” folder and a “fake” folder. Two folders are created for each segmented dataset, i.e there are six total folders. A python program is created to open the metadata.json file that is in each segmented folder, create a list of the fake videos present in the folder by selecting each video where label equals “FAKE” in the metadata.json file. This list of fake videos now has the name of each video present in the segmented folder, in a for-loop for each video in the list:

- Add “\\” and the name of the video to the end of the file path of the segmented folder, this gives you the file path of the video (source path)
- Create a new destination path with the the file path of the output folder +”\\”+video name (destination path)
- Check if the source path exists, and if the source path exists, move the file from the source path to the destination path.

This is done for real videos in the same manner, and we repeat this for each segmented folder. A new folder was created called “Test” folder, which contains 150 videos total, 50 videos from each segmented folder, where 25 videos are real and 25 videos are fake. They are stored in either a “real” folder or “fake” folder. The reason for only testing 150 videos is due to computation time. Each frame in a video is checked for a face, and if a face is

found, the frame is then sent to the model where a prediction is returned, this is the reason for why the computation time is high.

The testing for Rössler et al. (2019) and Afchar et al. (2018) was implemented in Spyder (https://www.spyder-ide.org/) on Anaconda (https://www.anaconda.com).

3.2 Setting up the environments and adapting the code to the dataset

To run Rössler et al. (2019) and Afchar et al. (2018), a separate environment has to be created for each implementation. This is because they both have different requirements, such as Rössler et al. (2019) runs on Python 3.6 and Afchar et al. (2018) runs on Python 3.5.

The environment created for Afchar et al. (2018) is called "Meso". The requirements are listed on Github (Afchar, 2018). Where possible, conda -install was used to install the requirements otherwise pip install was used. There are only 6 listed requirements for Afchar et al. (2018):

- Python 3.5 (https://www.python.org/)
- Numpy 1.14.2 (https://pypi.org/project/numpy/1.14.2/)
- Keras 2.1.5 (https://faroit.com/keras-docs/2.1.5/)
- Imageio (https://pypi.org/project/imageio/)
- FFMPEG (https://www.ffmpeg.org/download.html)
- face_recognition (Geitgey, 2018)

The recommended installation method for the face_recognition method is to use pip install (Geitgey, 2018). The installation was not successful due to certain packages not being compatible. The next step taken to try and install face_recognition was to use conda install (https://anaconda.org/conda-forge/face_recognition), the installation was successful with no errors. The package version of face_recognition was not specified on the requirements list (Afchar, 2018).

The first attempt at launching Spyder in the Meso environment with the model and attempting to run the code resulted in a crash of Spyder. The only fix was to do a full reset of Spyder, thereafter the Meso environment was deleted, and a new environment was created, also called Meso, with the same list of requirements. Spyder successfully launched with no crashes or bug errors.

The first environment created for Rössler et al. (2019) was called "FaceForen", and the requirements are quite extensive and can be found on Github along with the code (Rössler, 2020). The first attempt to install the requirements, involved using the pip command "pip install -r requirements.txt", it stopped installing the requirements once it could not successfully install a package, thereafter manually installing the requirements occurred and there were issues with installing 3 packages:

- mkl-fft==1.0.10
- mkl-random==1.0.2
- torch==1.0.1.post2

The first two packages did not appear to exist. The error happened to be a naming error; it should have been:

- mkl_fft==1.0.10
- mkl_random==1.0.2

This is an error with the requirements list on Github (Rössler, 2020). The torch package version was not available on the official pytorch website (https://pytorch.org/get-started/previous-versions/), a pip installation of torch==1.1.0 was successful.

After all packages were installed, the next step was to launch Spyder to test the model, and subsequently there were errors with the installed packages. The decision was made to create a new environment called "ffv2" and to try to find the closest compatible versions for every listed requirement.

In the new environment, ffv2, each requirement was manually installed; the closest torch version found is pytorch==1.0.1. The difference in the package name is because "torch" is the package name when using pip install, and "pytorch" is the package name when using conda install. The package installation instructions for older versions of pytorch is from the official pytorch website (https://pytorch.org/get-started/previous-versions/). After installing each package on ffv2, a test launch of Spyder shows that the installation of each package is successful.

On testing the Afchar et al. (2018) model on a test video, an unsolvable error occurred, "ERROR (theano.gof.opt) : Optimization failure due to: local_abstactconv_check", which is caused by the package Theano (https://github.com/Theano/Theano) and the fix for this problem is to update to a later version of keras (Lamblin, 2017). Theano is a required package that is installed when installing keras. Therefore, this project did not get Afchar et al. (2018) model to work. The testing for Rössler et al. (2019) went without any problems. The code needed slight adaptations; it had no evaluation metric for the output/prediction video.

When predicting videos from the "real" folder from the Test folder:

- A counter is kept for each frame predicted as REAL, r
- A counter is kept for the number of frames n
- $Classification\ Accuracy\ for\ video$ = (r/n) × 10
- A counter is kept for every video that is predicted REAL, tR, i.e. where the $classification\ accuracy\ for\ video \geq 50$
- $Classification\ Accuracy\ for\ real\ videos$ = (tR/75) × 100 there are 75 real videos in the "real" folder.

When predicting videos from the "fake" folder from the Test folder:

- A counter is kept for each frame predicted as FAKE, f
- A counter is kept for the number of frames n
- $Classification\ Accuracy$ = (f/n) × 100
- A counter is kept for every video that is predicted as FAKE, tF, ie where the $classification\ accuracy\ for\ video \geq 50$
- $Classification\ Accuracy\ for\ fake\ videos$ = (tF/75) × 100, there are 75 fake videos in the "fake" folder.

The classification accuracy for each video and the subdatasets are written to a file. To test Rössler et al. (2019), their modified XceptionNet model is being used with the pretrained weights, trained on the FaceForensics++ dataset. The pretrained weights are available on the Github page (Rössler, 2020), the pretrained weights used is "full_raw.p" for the XceptionNet model.

4. Results and discussion

Afchar et al. (2018) was not able to be tested, to no fault of the code or the model, this is an error with Keras, as such, there is no way we can validate or verify the effectiveness of the model on a portion of the DFDC dataset.

Rössler et al. (2019) with pretrained weights achieved 13.1578947368% accuracy for detecting deep fakes and 89.4736842105% for detecting real videos, with an overall accuracy of 51.3157894736%.

Table 2: Table showing results achieved for Rössler et al. (2019)

	Pretrained XceptionNet tested on DFDC dataset (this papers results)	**XceptionNet results from Rössler et al. (2019)**
Classification accuracy for deepfake videos:	13.1578947368%	Unknown
Classification accuracy for realvideos:	89.4736842105%	Unknown
Average classification accuracy:	51.3157894736%	82.01% accuracy on raw uncropped video footage

Table 3: Table showing the worst 10 detections of deepfake videos by XceptionNet tested on the DFDC subdataset

Name of the video:	Classification accuracy:
aaknzywids.mp4	0.0% fake
aarsmohwrt.mp4	0.0% fake
aayrffkzxn.mp4	0.0% fake
abxtkdjyru.mp4	0.0% fake
afnxnrrqsj.mp4	0.0% fake
aheocfkxjx.mp4	0.0% fake
aijlttdlrj.mp4	0.0% fake
akfjqoantp.mp4	0.0% fake
akpuczgfpk.mp4	0.0% fake
alqiqhnrza.mp4	0.0% fake

Table 4: Table showing the best 10 detections of deepfake videos by XceptionNet tested on DFDC subdataset

Name of the video:	Classification accuracy:
aejvkfbtxs.mp4	97.56944444444444% fake
aimkjacvip.mp4	94.66192170818505% fake
ablzpwqhcc.mp4	88.62068965517241% fake
ambabjrwbt.mp4	87.58389261744966% fake
ajeegjgyzk.mp4	82.0% fake
alxodlppci.mp4	78.76712328767124% fake
adckadazdl.mp4	67.90123456790124% fake
ahofrimoni.mp4	59.51417004048582% fake
acdkfksyev.mp4	57.89473684210527% fake
agdivudslh.mp4	54.666666666666664% fake

As we can see a portion of the results, based on only 150 videos, their modified XceptionNet model, was not able to detect the deepfake videos in the DFDC dataset. The believed reason for the classification accuracy for the deepfakes being so low is that the FaceForensics++ dataset, is not a true generalization of deepfakes (Dolhansky et al., 2020), the FaceForensics++ dataset contains only 1000 videos or about about half a million edited images (Rössler, 2020). In relation, the DFDC dataset has 128,154 videos. These results shows that models trained on the first generation of deepfakes, is not as accurate as they claim to be. The model architecture is not in question, but rather the quality of the dataset.

Table 5: Table showing worst 10 detections of real videos by XceptionNet tested on DFDC subdataset

Name of the video:	Classification accuracy:
chfkrpvgnz.mp4	19.014084507042252% real
chqqxfuuzi.mp4	27.666666666666668% real
apedduehoy.mp4	30.333333333333336% real
awkvatcshx.mp4	31.103678929765888% real
bwtyeeplxj.mp4	31.1787072243346% real
eppyqpgewp.mp4	47.0% real
bvsnqubtjc.mp4	49.831649831649834% real
fopjiyxiqd.mp4	50.0% real

Name of the video:	Classification accuracy:
fsaronfupy.mp4	51.162790697674424% real
eyguqfmgzh.mp4	54.333333333333336% real

Table 6: Table showing the best 10 detection of real videos by XceptionNet on DFDC subdataset

Name of the video:	Classification accuracy:
aayrffkzxn.mp4	100.0% real
almnilnfyu.mp4	100.0% real
bvpeerislp.mp4	100.0% real
cxsvvnxpyz.mp4	100.0% real
cxwcpgspni.mp4	100.0% real
dvwpvqdflx.mp4	100.0% real
dzrrklwrgn.mp4	100.0% real
exseruhiuk.mp4	100.0% real
extbidooov.mp4	100.0% real
exxqlfpnbz.mp4	100.0% real

5. Conclusion

This project has successfully tested a popular and well respected paper by Rössler et al. (2019) and found the accuracy is potentially not as high as it is claimed to be, the believed reason for the accuracy being so low is that the deepfakes in the Faceforensics++ dataset are firstly, not big enough in terms of the amount of videos present in the dataset, secondly, does not accurately represent each scenario in which a deepfake video may occur. For example, the videos in the DFDC dataset had a variety of physical scenarios which involves the target person in the video (both real and deepfake videos), such as a person pacing back and forth in a room, a person sitting upright and looking straight at a camera and so on. The Faceforensics++ dataset chose videos where the targets face is front facing, thus it would be expected for any model trained on only front facing deepfakes to not be able to accurately or effectively detect deepfake videos when the target is not front facing. The Faceforensics++ dataset is considered as a first-generation dataset (Dolhansky et al., 2020). The methods used to create the deepfakes in the Faceforensics++ dataset are also seen in other papers such as in Afchar et al. (2018), the premise is that they will have deepfakes of the same quality which would not adequately train them to detect deepfakes that are created with methods seen today.

This project shows the evident need for constant testing of previous work to determine if and when they are no longer sufficiently adequate at detecting deepfake material, which indicates that the fight against deepfakes may be an unending task.

This project can be improved by testing more models, which are trained on datasets other than Faceforensics++, which this project did not do. This will illustrate if deepfake models which are trained on older datasets are in fact unable to reliably and effectively determine whether a video/picture is real or a deepfake. The Faceforensics++ dataset is approximately 3.5TB in size, and therefore it was not downloaded due to lack of storage space. This project can be trained on the DFDC dataset and tested on the Faceforensics++ dataset to determine if the accuracy achieved is greater than trained on the Faceforensics++ dataset and tested on the DFDC dataset, as this project accomplished. The effectiveness of a deepfake detection model seems to be highly dependent on the quality of the dataset, this might suggest that deepfake detection models may always be proved to be ineffective once a newer deepfake creation method is used, since methods which utilize the GAN architecture may utilize a deepfake detection model as the discriminator to create the new generation of deepfake material.

References

Afchar, D. (2018) MesoNet, Github, [online], https://github.com/DariusAf/MesoNet.

Afchar, D., Nozick, V., Yamagishi, J. and Echizen, I., (2018) “Mesonet: a compact facial video forgery detection network”, *2018 IEEE International Workshop on Information Forensics and Security (WIFS)*, Hong Kong, China, 1-7.

Bayar, B. and Stamm, M.C., (2016) “A deep learning approach to universal image manipulation detection using a new convolutional layer”, *Proceedings of the 4th ACM Workshop on Information Hiding and Multimedia Security*, 5-10.

Chollet, F., (2017) "Xception: Deep Learning with Depthwise Separable Convolutions," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, 1800-1807.

Christopher, N., (2020) “We've Just Seen the First Use of Deepfakes in an Indian Election Campaign”, *Vice*, 18 February, [online], accessed 4 November 2020, https://www.vice.com/en/article/jgedjb/the-first-use-of-deepfakes-in-indian-election-by-bjp.

Damiani, J., (2019) “A Voice Deepfake Was Used to Scam a CEO Out of $243,000”, *Forbes*, 3 September, [online], accessed 4 November 2020, https://www.forbes.com/sites/jessedamiani/2019/09/03/a-voice-deepfake-was-used-to-scam-a-ceo-out-of-243000/?sh=74bb412d2241.

Deepfakes (2020) Faceswap, Github, [online], accessed 5 November 2020, https://github.com/deepfakes/faceswap.

Dolhansky, B., Bitton, J., Pflaum, B., Lu, J., Howes, R., Wang, M. and Ferrer, C.C., (2020) “The deepfake detection challenge dataset”, arXiv, [online], https://arxiv.org/abs/2006.07397.

Ferrer, C., Dolhansky, B., Pflaum, B., Bitton, J., Pan, J. and Lu, J., (2020) “Deepfake Detection Challenge Results: An Open Initiative to Advance AI”, Facebook AI, [online], https://ai.facebook.com/blog/deepfake-detection-challenge-results-an-open-initiative-to-advance-ai/.

Fink, D. and Diamond, S., (2020) “Deepfakes: 2020 and Beyond”, *The Recorder*, 3 September [online], accessed 4 November 2020, https://www.law.com/therecorder/2020/09/03/deepfakes-2020-and-beyond/?slreturn=20201116162814.

Geitgey, A. (2018) face_recognition, Github, [online], https://github.com/ageitgey/face_recognition.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. and Bengio, Y., (2020) “Generative adversarial nets”, *Communications of the ACM* 63(11), 139-144.

Google Developers. (2019) “Overview of GAN Structure”, *Generative Adversarial Networks*, [online], accessed 4 November 2020, https://developers.google.com/machine-learning/gan/gan_structure.

Güera, D. and Delp, E.J., (2018) "Deepfake Video Detection Using Recurrent Neural Networks," *15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, Auckland, New Zealand, 1-6.

Huang, D. and De La Torre, F., (2012) “Facial action transfer with personalized bilinear regression”, *12th European Conference on Computer Vision*, Proceedings Part II Florence, Italy, 144-158.

iperov (2020) Deepfacelab, Github, [online], https://github.com/iperov/DeepFaceLab.

Jee, C., (2020) “An Indian Politician is Using Deepfake Technology to Win New Voters”, *MIT Technology Review*, 19 February, [online], accessed 4 November 2020, https://www.technologyreview.com/2020/02/19/868173/an-indian-politician-is-using-deepfakes-to-try-and- win-voters/.

Jung, T., Kim, S. and Kim, K., (2020) “DeepVision: Deepfakes Detection Using Human Eye Blinking Pattern”, *IEEE Access* 8 83144-83154.

Karras, T., Laine, S. and Aila, T., (2019). “A style-based generator architecture for generative adversarial networks”, *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, 4396-4405.

Kaggle. (2019) Deepfake Detection Challenge, [online], https://www.kaggle.com/c/deepfake-detection-challenge.

Lamblin, P., (2017) “MILA and the future of Theano”, theano-users Google Groups, 28 September, [online], accessed 5 November 2020, https://groups.google.com/g/theano-users/c/7Poq8BZutbY/m/rNCIfvAEAwAJ.

Laptev, I., (2020), Learning Human Actions from Movies, Département d'Informatique, ENS, [online], https://www.di.ens.fr/~laptev/actions/.

Lyons, T., (2019) “The Partnership on AI Steering Committee on AI and Media Integrity”, The Partnership on AI, 5 September, [online], https://www.partnershiponai.org/the-partnership-on-ai-steering-committee-on-ai-and-media-integrity/.

MarekKowalski (2018) Faceswap, GitHub, [online], accessed 5 November 2020, https://github.com/MarekKowalski/FaceSwap/.

Nirkin, Y., Keller, Y. and Hassner, T., (2019) “FSGAN: Subject agnostic face swapping and re-enactment”, *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, Seoul, South Korea, 7183-7192.

Paul, K., (2019) “California Makes ‘Deepfake’ Videos Illegal, But Law May be Hard to Enforce”, *The Guardian*, 7 October, [online], accessed 4 November 2020, https://www.theguardian.com/us-news/2019/oct/07/california-makes-deepfake-videos-illegal-but-law-may-be-hard-to-enforce.

Rahmouni, N., Nozick, V., Yamagishi, J. and Echizen, I., (2017) “Distinguishing computer graphics from natural images using convolution neural networks”, *2017 IEEE Workshop on Information Forensics and Security*, Rennes, France, 1-6.

Ranjan, R., Patel, V.M., and Chellappa, R., (2019) "HyperFace: A Deep Multi-Task Learning Framework for Face Detection, Landmark Localization, Pose Estimation, and Gender Recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence* 41(1), 121-135.

Rössler, A., (2020) FaceForensics++: Learning to Detect Manipulated Facial Images, Github, [online], https://github.com/ondyari/FaceForensics.

Rössler, A., Cozzolino, D., Verdoliva, L., Riess, C., Thies, J. and Niessner, M., (2019) “Faceforensics++: Learning to detect manipulated facial images”, *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, Seoul, South Korea, 1-11.

Ruiz, D., (2020) “Deepfakes Laws And Proposals Flood US”, Malwarebytes Labs, 23 January, [online], accessed 4 November 2020, https://blog.malwarebytes.com/artificial-intelligence/2020/01/deepfakes-laws-and-proposals-flood-us/.

Sample, I., (2020) “What Are Deepfakes – And How Can You Spot Them?”, *The Guardian*, 13 January, [online], accessed 2 November 2020, https://www.theguardian.com/technology/2020/jan/13/what-are-deepfakes-and-how-can-you-spot-them.

Seferbekov, S., (2020) dfdc_deepfake_challenge, Github, [online], https://github.com/selimsef/dfdc_deepfake_challenge.

Smith, A., (2020). “Deepfakes Are The Most Dangerous Crime of the Future, Researchers Say”, *The Independent*, 5 August, [online], accessed 2 November 2020, https://www.independent.co.uk/life-style/gadgets-and-tech/news/deepfakes-dangerous-crime-artificial- intelligence-a9655821.html.

Soukupova, T., and Cech, J. (2016) "Real-time eye blink detection using facial landmarks", *21st Computer Vision Winter Workshop*, Rimske Toplice, Slovenia, 42-50.

Stupp, C., (2019) “Fraudsters Used AI to Mimic CEO’S Voice in Unusual Cybercrime Case”, *The Wall Street Journal*, 30 August, [online], accessed 4 November 2020, https://www.wsj.com/articles/fraudsters-use-ai-to-mimic-ceos-voice-in-unusual-cybercrime-case-11567157402.

Szegedy, C., Ioffe, S., Vanhoucke, V. and Alemi, A., (2017) “Inception-v4, inception-resnet and the impact of residual connections on learning”, Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI'17), 4278–4284.

Thies, J., Zollhofer, M., Stamminger, M., Theobalt, C. and Niessner, M., (2016) “Face2face: Real-time Face Capture and Reenactment of RGB videos”, *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, 2387-2395.

Thies, J., Zollhöfer, M. and Niessner, M., (2019) “Deferred neural rendering: Image synthesis using neural textures”, *ACM Transactions on Graphics* 38(4), 1-12.

u/landoflobsters, (2018) Update On Site-Wide Rules Regarding Involuntary Pornography And The Sexualization Of Minors, [online], accessed 2 November 2020, https://www.reddit.com/r/announcements/comments/7vxzrb/update_on_sitewide_rules_regarding_involuntary/.

Zakharov, E., Shysheya, A., Burkov, E. and Lempitsky, V., (2019) “Few-shot adversarial learning of realistic neural talking head models”, *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, Seoul, South Korea, 9458-9467.