DEEPFAKE1

INSE6610: Cybercrime Investigations

Mukul Sandeep Prabhu
MEng Information Systems Security
Montreal – QC
m_prabhu@live.concordia.ca

Nandinye Selvaraj
MEng Information Systems Security
Montreal – QC
Nandinye13@outlook.com

Rryan Jebaseelan
MEng Information Systems Security
Montreal – QC
rryan.jebaseelan@mail.concordia.ca

Tharani Santharam
MEng Information Systems Security
Montreal – QC
t_santha@mail.concordia.ca

***Abstract*— DEEPFAKE1 is a project that deals with images that are artificially generated and tries to prove that they are artificially generated. The project is based on a scenario of a client involved in a deep fake incident concerning his identity. Someone on the internet used the client's face on a social media post that negatively affected his reputation. As it was difficult to prove the legitimacy of the image simply by looking at it, our job is to try to provide proof that it is fake. Going ahead with that prompt we searched for papers that researched deep fake detection and executed their algorithm with different datasets.**

Keywords*— *Face X-ray, Forgery Detection, Deepfake.

I. INTRODUCTION

A new era of the internet began at the end of 2022 when OpenAI released ChatGPT to the public. Technology that seemed to be of the distant future is now at the general public's fingertips blowing away conventional ways of using the internet. Although deep fakes are not strictly an AI application, AI paved the path and exposed people to the limitless possibilities of machine learning. This resulted in easy access to machine learning-based applications such as the creation of deep fakes. Deep fakes are media that are digitally altered using machine learning based technology. It can be images, videos or even voice where the original content is changed either partially or completely. The media is altered to such an extent that it is impossible to distinguish between the real and modified media. The groundwork for this technology has been placed since the late 1990s and it has since been developing to what we know today. When people got the idea of deep fakes and got exposed to its technology, they used it for malicious purposes such as pornographic and false news but it was still discernable by the human eye that the media is manipulated. Today's technology has further improved it and made it almost impossible to find the difference. This project aims to find different papers or algorithms that researchers have published and check their credibility and compatibility with our problem statement. The project goes ahead with the assumption that deep fake detection does not have a one stop solution and that our tests should give us different results for the different algorithms we use.

II. BACKGROUND

This section will have a discussion of the papers used for the project along with a brief explanation of the algorithm used.

A. Unmasking Deepfakes with simple features

The use of pickle files is used to save the extracted features and use them multiple times without spending time to extracting the features every time the program is run. Pickling is not specific for just machine learning but can be used generally to save a "state" of the program and use it later. The data present in the pickle file is then de-serialised/extracted to data and label variables. The real and fake images are separated based on the labels where '0' means real and '1' means fake. The spatial frequency and power spectrum graph is important in distinguishing a real and fake images.

The paper takes advantage of the fact that deepfakes often have some sort of artifacts that are hidden to the naked eye. A classical frequency domain analysis is used followed by a basic classifier. This method showed good results compared to other systems that required large amounts of labeled data. The test images are plotted against Spatial frequency and Power spectrum.

The celebalow1000 and Faces-HQ datasets are divided into training and testing by using 80% for training and 20% for testing. The training images are then sent through 4 different classifiers and then the test images are used for testing against it. The 4 different classifiers are: Logistic Regression, Support Vector Machine with linear Kernel, Support Vector Machine with a Radial Basis Function kernel, Support Vector Machine with a polynomial kernel.

The results are provided in the form of graphs that show us how the test images compare to the data set used for training. According to the research conducted, real images tend to not have a flat region at higher frequencies, but fake images do.

B. Face X-ray for More General Face Forgery Detection

We present a novel approach for identifying counterfeit face photos in this paper, which we name face X-ray. An image in greyscale that indicates whether a facial image is a composite of two separate images is called a face X-ray. While actual photos

lack such a barrier, it draws attention to the blending boundary in phony images. Face X-ray successfully identifies these fakes because the majority of face alteration methods involve mixing a changed face with a backdrop. This method can be taught without the use of fake photos from existing face modification techniques, and it does not rely on identifying particular manipulation artifacts. under contrast to many other existing detection technologies that become ineffective under such situations, extensive testing shows that face X-ray performs effectively even against novel and unknown manipulation approaches.

C. Capsule-Forensics: Using Capsule Networks to Detect Forged Images and Videos

The topic introduces an innovative method for detecting forged media by utilizing the strengths of capsule networks. As the creation of highly realistic forged images and videos becomes increasingly accessible through advanced technologies, the demand for effective detection methods has never been greater. Conventional detection techniques are typically specific to particular types of forgeries and rapidly become outdated as new forgery techniques emerge. This paper proposes a capsule network-based approach that transcends its initial use in addressing inverse graphics problems to tackle a broad spectrum of forgeries. The suggested method achieves high accuracy in identifying various types of forgeries, such as replay attacks, face swapping, facial reenactment, and completely computer-generated images, outperforming current state-of-the-art detection methods. The findings underscore the robustness and adaptability of capsule networks in digital forensics, presenting a promising solution to the dynamic challenges of media forgery.

D. Efficient Deepfake Detection Using Binary Neural Networks and Advanced Feature Integration

Regarding the discovery of deepfakes, etc. , we would like to convey a novel methodology leveraging Binary Neural Networks that achieve exceptional performance with constrained precision, in contrast to the colossal and intricate models employed presently. We have integrated Fast Fourier Transform and Local Binary Pattern functionalities, which exhibit remarkable efficacy in tracking manipulation across spectral and textural domains. Remarkably, this strategy does not compromise accuracy, as affirmed by the COCOFake, DFFD, and CIFAKE datasets. This system offers a seamless journey to the detection realm, with a notable up to 20-fold reduction and sundry efficiency enhancements. It can be viewed as the inaugural component towards the future's pursuit of more rapid and precise deepfake identification, where the emphasis is on FLOPS curtailment. Precision alone should not be the single focal point, but rather, efficacy and computational perspectives should also be considered.

III. Summary

The summary outlines four distinct approaches to deepfake detection. The first approach, "Unmasking Deepfakes with Simple Features," utilizes pickle files to store and reuse extracted features, distinguishing between real ('0') and fake ('1') images by their labels, and applies frequency domain analysis coupled with a basic classifier. Real images typically do not exhibit flat regions at higher frequencies, unlike fake images. The second approach, "Face X-ray for More General Face Forgery Detection," detects counterfeit photos by revealing blending boundaries in greyscale images, training without relying on specific fake images, and effectively identifying new manipulation techniques. The third method, "Capsule- Forensics: Using Capsule Networks to Detect Forged Images and Videos," uses capsule networks to identify a broad range of forgeries with high accuracy, showcasing the robustness and flexibility of these networks in digital forensics. Finally, "Efficient Deepfake Detection Using Binary Neural Networks and Advanced Feature Integration" employs Binary Neural Networks for high performance with limited precision, integrating Fast Fourier Transform and Local Binary Pattern to track manipulations, significantly improving computational efficiency and reducing requirements by up to 20-fold.

IV. METRICS

A. Metrics definition for Unmasking Deepfakes with simple features

The following metrics are used for graphical representations.

Spatial Frequency: Used to measure how often the features of an image like textures and edges repeat per unit distance. High SF means fast and sharp changes in features

Power spectrum: Tells us how intense the features are present, a higher value indicates a sharper change after every unit of distance and a lower value indicates a smoother transition

The following are the different classifiers used for the paper, it has the formal definition as well as analogies to better understand the terms.

Logistic Regression - It is implemented as a linear model for classification rather than regression in terms of the scikit-learn/ML nomenclature. The logistic regression is also known in the literature as logit regression, maximum-entropy classification (MaxEnt) or the log-linear classifier.

Analogy: Imagine fake and real images are present on a table and you are trying to separate them by drawing a line in between them

Support vector machines (SVMs) are a set of supervised learning methods used for classification, regression and outliers detection. Different Kernel functions can be specified for the decision function.

SVM with linear Kernel - In the case of a linear kernel, the SVM algorithm tries to find a linear decision boundary (hyperplane) that separates the classes.

Analogy: This is like trying to find the biggest distance between a real and fake image and then drawing a line

SVM with an RBF kernel - The RBF kernel function for two points X_1 and X_2 computes the similarity or how close they are to each other.

Analogy: This is the same as the second classifier but now the line is curved between all the images, thus not "linear" anymore

SVM with a polynomial kernel - Polynomial kernel SVM is a type of SVM that uses a polynomial function to transform the input data into a higher dimensional space

Analogy: The table example is on a 2D space but this classifier deals in 3 dimentional spaces and separates accordingly.

B. Metrics definition/Implementation of Face X-ray for MoreGeneral Face Forgery Detection

- *Preprocessing*

Face Detection - To find faces in photos, use a face detection algorithm (e.g., MTCNN, dlib).
Synthetic Forgery Creation - Make matching images of genuine faces. To generate artificial forgeries, blend faces from one image onto another. Apply methods such as Poisson blending to create realistic-looking forgeries.

- *Model Training*

Network Architecture - Define a convolutional neural network (CNN) with the ability to recognize blending borders from training data. Typically, convolutional, pooling, and fully linked layers are part of the network architecture.
Training Process - Input the real and synthetic forged face images to CNN. Provide CNN with crafted faces, both real and artificial. The result is a face X-ray in grayscale that shows the borders of blending. To calculate the difference between the expected and actual blending limits, use a loss function (such as binary cross-entropy). To reduce the loss, use an optimizer (such as Adam or SGD) to tune the network's parameters.

- *Inference*

Face X-ray Generation - Preprocess a new face image to conform to the training data format by detecting the face region. Run the face image via CNN's training. Acquire the facial X-ray output, which indicates possible borders for blending.
Forgery Detection - Examine the output of the face X-ray. If there are blending borders, the picture should be considered a fake. If there are no blending boundaries found, the image should be considered real.

Explanation of Key Functions

preprocess_images: This feature blends faces from many photographs to produce synthetic forgeries and recognizes faces in actual images.
train_model: Using both artificially created and actual images, this function defines and trains a CNN. Before training the model on the ready-made dataset, it assembles the model with the proper optimizer and loss function.
detect_forgery: This function generates a face X-ray from a new image by applying the trained model to it, checks it for blending boundaries, and determines whether the image is a fake.

Key Components

Face Detection: Identifying and extracting facial regions from images using algorithms.
Synthetic Forgery Creation: Combining faces from several photos to create the appearance of authentic frauds.
CNN Model: A neural network trained to detect blending boundaries in face images.
Face X-ray: The greyscale output from the CNN that highlights potential blending boundaries.

C. Metrics definition for Capsule-Forensics

Accuracy – Evaluates the performance on various datasets, including deepfake and Face Forensics datasets.

Precision - It measures the proportion of true positive results among all positive results predicted by the model. It indicates the number of detected forgeries.

Recall (True positive rate) - Measures the proportion of true positive results with respect to all actual positive cases.

F1 Score - This is the harmonic means of precision and recall, providing a balance between the two metrics. It is useful when we need balanced precision and recall in case of imbalanced dataset.

Area Under the Receiver Operating Characteristic Curve (AUC-ROC) - measures the ability of the model to distinguish between classes. The ROC curve plots the true positive rate (recall) against the false positive rate.

False Positive Rate (FPR) & False Negative Rate (FNR) - They measure the proportion of actual negatives and positives that are incorrectly identified as positives and negatives.

Half Total Error Rate (HTER) - It is used to evaluate the detection performance on the Idiap REPLAY-ATTACK dataset. It is calculated as the average of the False Rejection Rate (FRR) and the False Acceptance Rate (FAR)

Comparison with State-of-the-Art Methods - The performance of the proposed method is compared against multiple state-of-the-art detection techniques. For example, on the Idiap REPLAY-ATTACK dataset, the proposed Capsule-Forensics-Noise method achieved an HTER of 0.00%, outperforming several other methods.

Detection Performance on Various Types of Forgeries - The method's ability to detect several types of forgeries, such as replay attacks, face swapping, facial reenactment, and fully computer-generated images, is evaluated. For instance, Capsule Forensics-Noise achieved the highest accuracy for face swapping detection at both frame and video levels.

D. Metrics Definition for Deepfake Detection with Binary Neural Networks(BNN)

Deepfake detection's efficacy was subjected to a thorough examination. Parameters -- exactness, effectiveness, and sturdiness -- were assessed via distinct criteria, each contributing to this assessment. Precision, recollection, F1-score, and Area Under the Curve of the Receiver Operating Characteristic (AUC-ROC) were incorporated into this appraisal. As regards detection's accuracy, these metrics were employed. Furthermore, computational proficiency was meticulously scrutinized through the computation of the volume of Floating-Point Operations (FLOPs) incurred during inference. The technique proved its efficacy in real-time applications, as consistently demonstrated by the outcomes. In comparison with conventional approaches, enhanced precision and diminished computational intricacy were observed.

V. Empirical Study

A. *Unmasking Deepfakes with simple features*

1) *Examined variables*

This study is conducted on the problem statement, “Someone on the internet used the client’s face on a social media post that negatively affected his reputation. As it was difficult to prove the legitimacy of the image simply by looking at it, our job is to try to provide proof that it is fake”. To determine if the image is deepfake or not we conducted this study where we trained and tested our model with two different datasets and then used test images from a different dataset to see how our model holds up. Our independent variable is our test images, datasets and our dependent variable is our classifier, algorithm. The datasets to train and test the algorithm as well as test images that act the client’s image will be changed and results of the algorithm will be checked whose output depends on these images and datasets.

2) *Data collection*

The project has used pre trained models that are present in pickle files to save time. The data is as follows:

The first one is CelebA - since this contains only real images, the authors of the paper created deepfakes of the same images and made a dataset of 50% real and 50%fake. https://github.com/cc-hpc-itwm/DeepFakeDetection/blob/master/Experiments_CelebA/dataset_celebA.7z

The second one is Faces-HQ, this has a mix of 4 different datasets of both real and fakes https://github.com/cc-hpc-itwm/DeepFakeDetection/blob/master/Experiments_Faces-HQ/dataset_freq_1000.pkl

Since the project is based on the scenario that a client comes with a complain of his image being deep faked, I used images from other datasets to test the trained model. https://www.kaggle.com/datasets/selfishgene/synthetic-faces-high-quality-sfhq-part-1
https://www.kaggle.com/competitions/deepfake-detection-challenge/discussion/121173

The individual images present here are used for testing, after it has been trained by a different dataset. This is done to test the model for a more general application.

3) *Statistical analysis*

This section is going to explain the overall design and the results obtained. The overall program has been divided into multiple functions each present in their own cells. The "MAIN CELL" that controls all the functions is present at the bottom. The program runs in an infinite loop asking for the file path of a picture to be tested. As 2 pickle files are present in the list, for every image 2 sets of results will be given. There will be graphs indicating how the test image from other datasets perform in comparison to the trained real and fake images. To terminate the program type, "exit".

```
Enter the file path of the image to be tested
Or type 'exit' to quit/SFHQ_pt1_00000008.jpg
Results for the dataset: /celeba_low_1000.pkl and algorithm: Unmasking Deepfak
```

Figure a

Figure a shows us what happens when the “MAIN CELL” is run. “/SFHQ_pt1_00000008.jpg” is the path of the image that is given as input. The algorithm goes through the list containing the pickle files and tests this image. First “/celeba_low_1000.pkl” is used.

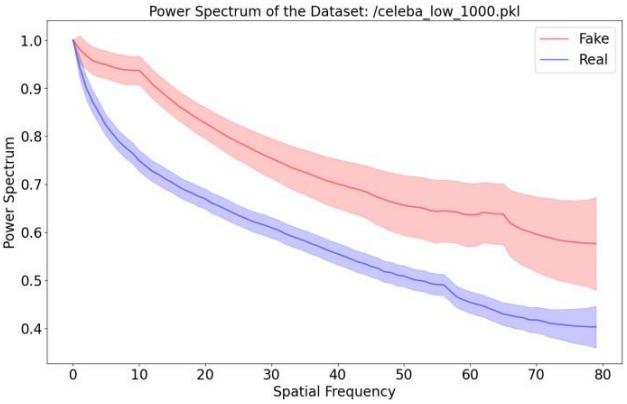

Figure b

The dataset is split in 80 to 20 ratio and figure b shows the result of the 20% test images.

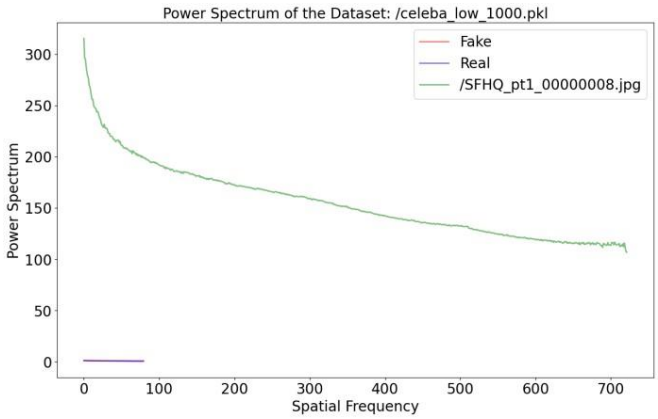

Figure c

The green graph is the test image we used sourced from a different dataset that is not used for training. The image is taken from this dataset https://www.kaggle.com/datasets/selfishgene/synthetic-faces-high-quality-sfhq-part-1

As can be seen from the graph, the result isn’t clear because the power spectrum and spatial frequency of this image is much higher than the dataset is trained on.

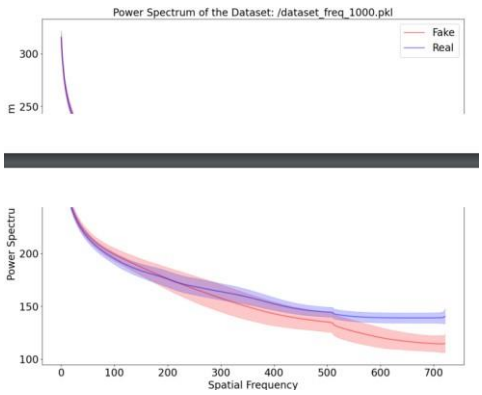

Figure d

The program is now finished with the first pickle file and will iterate to the second one which is called, "dataset_freq_1000.pkl". As we can see from figure d, the same procedure is followed again where we are now presented with the results of 20% test images. Now we test the same test image we used for the previous pickle file to this one and the result is as shown in figure e. The green graph of the test image is in-line with the red fake image output. The program can keep giving us such analysis for different images or the program can be terminated by typing "exit" as seen in figure f.

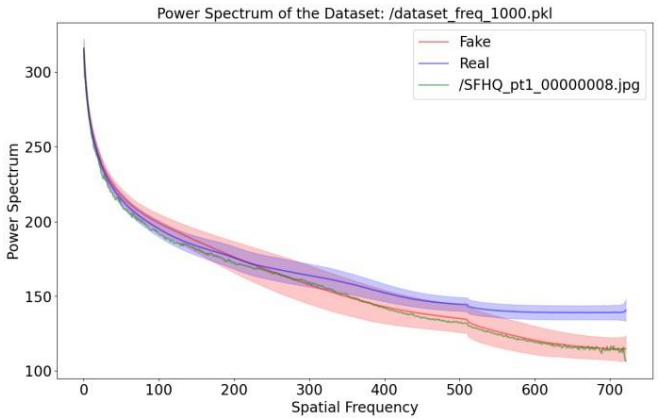

Figure e

```
Enter the file path of the image to be tested
Or type 'exit' to quitexit
```

Figure f

4) Threats to validity

As it has been seen in the Statistical analysis section, changing the dataset had a big clarity on the image being deep faked. The first data set, was created from a set of real images and the authors deep faked them and created a new dataset called "celeba_low_1000.pkl" which had 50% real and 50% fake images. The second data set, "dataset_freq_1000.pkl" is a combination of 4 different sets of images of 25% each. From our results we can see that the second data set gave us better results because of its more diverse raw data. Thus in this study the dataset proved to be the threat to validity.

B. Face X-ray for More General Face Forgery Detection

1. Dataset Collection

The FaceForensics++ (FF++) dataset served as the main source of data for this work. This is a large-scale video dataset comprising 1000 original videos that have been processed usingfour cutting-edge face alteration techniques: NeuralTextures (NT), Face2Face (F2F), DeepFakes (DF), and FaceSwap (FS).

2. *Result Analysis*

The results of this study are displayed using a variety of measures for distinct models and datasets, including Area Under the Curve (AUC), Average Precision (AP), and Equal Error Rate (EER). This is a summary of the findings and the analysis that went along with it.

Benchmark Result

The paper presents benchmark results using metrics such as AUC (Area Under the Curve), AP (Average Precision), and EER (Equal Error Rate). The following key points are observed:

- All test datasets (DFD, DFDC, and Celeb-DF) show that the Face X-ray model performs better than the Xception model when trained on Blended Images (BI).
- The Face X-ray model performs much better when more fictitious photos are included from the FF++ dataset.

Generalization and Performance

Generalization Ability: The Face X-ray model outperforms baseline detectors trained on certain modification techniques, exhibiting high generalization to unknown datasets. By using Blended Images for training, the model is compelled to understand the fundamental properties of face X-rays, which improves generalization.

Effectiveness Against Different Manipulations: The efficacy of the model is evaluated using a range of manipulation techniques, such as NeuralTextures (NT), Face2Face (F2F), DeepFakes (DF), and FaceSwap (FS). Throughout all of these manipulation kinds, the Face X-ray framework continuously demonstrates great detection accuracy.

Model	Training Set	Detection Accuracy (F2F)	Detection Accuracy (FS)
LAE	X	90.93	63.15
FT-res	4 images	94.47	72.57
MDTS	X	92.77	54.07
Face X-Ray	X	97.73	85.69

Figure f: Detection Accuracy Comparison

Effect of Data Augmentation

The generation of different samples is a critical function of data augmentation procedures, like mask deformation and color correction, which improve the performance and robustness of the model.

C. *Capsule-Forensics: Using Capsule Networks to Detect Forged Images and Videos*

1. Methodology Overview

Input Preprocessing: Images and videos undergo preprocessing to detect faces, which are subsequently standardized to a size of 128x128 pixels. Feature extraction utilizes a modified VGG-19 network, focusing on outputs from the third maxpooling layer to capture essential characteristics.

Network Architecture: The network architecture features:

- *Primary Capsules:* These capsules receive latent features from the VGG-19 network and employ statistical pooling to aggregate information before passing it to subsequent layers.
- *Output Capsules:* Responsible for final classification, these capsules distinguish between real and fake images/videos based on processed features.
- *Dynamic Routing:* Integral to the architecture, dynamic routing optimizes interactions between primary and output capsules, enhancing the model's ability to accurately discern between authentic and forged content.

2. Training and Evaluation

Loss Function: During training, the capsule network utilizes cross-entropy loss, incorporating dynamic routing to iteratively refine predictions and improve model performance.

Experimental Setup: Evaluation encompasses benchmark datasets such as Idiap REPLAY-ATTACK and FaceForensics, alongside datasets specific to deepfake detection. Performance metrics include Half Total Error Rate (HTER), accuracy, and comparisons with state-of-the-art methods like MesoNet to validate effectiveness. Utilised data set for face forgery detection from "roboflow" website.

3. Experimental Results

Performance on Different Tasks:

- *Replay Attack Detection:* Achieved zero HTER on the Idiap REPLAY-ATTACK dataset, indicating robust detection of replayed content.
- *Face Swapping Detection:* Demonstrated high accuracy in identifying manipulated faces using deepfake techniques, surpassing competitors in both frame-level and video-level analyses.
- *Facial Reenactment Detection:* Effectively detected facial reenactment on the FaceForensics dataset, performing competitively with existing methods.
- *CGI vs. Photographic Image Detection:* Attained perfect accuracy in distinguishing between computer-generated images (CGIs) and photographic images (PIs), highlighting robust discriminative capabilities of the model.

This comprehensive methodology and experimental results underscore the efficacy of Capsule-Forensics in detecting a wide range of image and video forgeries, showcasing its potential for advancing forensic analysis in digital media.

D. Efficient Deepfake Detection Using Binary Neural Networks and Advanced Feature Integration

Evaluation of deepfake detection technique was conducted proficiently in an empirical study, where we thoroughly screened the proposed method against three sets of benchmarks: COCOFake, DFFD, and CIFAkE. Use of Binary Neural Networks (BNNs) with Fast Fourier Transform and Local Binary Pattern techniques were introduced and observed to have a positive impact in the detection's accuracy, making computational activities here more enhanced. Comparison between our method and current highly developed recent methods show gradual yet remarkable advancements in both detection accuracy and computational efficiency. High computational efficiency is reached by way of at least a 20× reduction in FLOPs which makes it more fitted for real-time operations. Implications of these results altogether indicate the utility and potential of our approach for real world application, especially in addressing the issue of deepfake media. The logical connection regarding these findings are not straightforward, delay detected during the detection process is decreased over time which positioning our approach as the most effective and pragmatic in addressing deepfake media's concerns.

The results showed consistent performance across different types of deepfake manipulations and varying levels of image quality, further validating the robustness and versatility of our approach. These findings underscore the practical applicability of our method in real-world scenarios and its effectiveness in combating the proliferation of deepfake media.

VI. CONCLUSIONS

From the perspective of this study, we can conclude that deep fake detection is not a one stop solution. There cannot be one algorithm and one dataset that can work for all the people of the world. As we have seen from the results our dataset seemed to be biased to a certain demographic or type of data. When we try to use this detection system, we need to make sure the dataset we are using is suited to the client's features. We won't get a clear result if we use a dataset comprising of people from the western part of the world and use it to test on people from the east.

REFERENCES

[1] https://www.britannica.com/technology/deepfake

[2] https://medium.com/@songda/a-short-history-of-deepfakes-604ac7be6016

[3] https://blog.devgenius.io/machine-learning-algorithm-series-polynomial-kernel-svm-understanding-the-basics-and-applications-89b4b42df137

[4] https://towardsdatascience.com/radial-basis-function-rbf-kernel-the-go-to-kernel-acf0d22c798a

[5] H. H. Nguyen, J. Yamagishi, and I. Echizen, "Capsule-Forensics: Using Capsule Networks to Detect Forged Images and Videos," Proc. of the

2019 International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2019), 5 pages, (May 2019)

[6] https://arxiv.org/abs/1911.00686

[7] DeepFakes. www.github.com/deepfakes/ faceswap. Accessed: 2019-09-18. 1, 5

[8] Shruti Agarwal, Hany Farid, Yuming Gu, Mingming He, Koki Nagano, and Hao Li. Protecting world leaders against deep fakes. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pages 38– 45, 2019.

[9] https://github.com/AlgoHunt/Face-Xray?tab=readme-ov-file

[10] Darius Afchar, Vincent Nozick, Junichi Yamagishi, and Isao Echizen. Mesonet: a compact facial video forgery detection network. In 2018 IEEE International Workshop on Information Forensics and Security (WIFS), pages 1–7. IEEE, 2018.

[11] FaceSwap. www.github.com/MarekKowalski/ FaceSwap. Accessed: 2019-09-30.

[12] https://arxiv.org/abs/1810.11215

[13] https://github.com/nii-yamagishilab/Capsule-Forensics

[14] https://paperswithcode.com/paper/use-of-a-capsule-network-to-detect-fake

[15] https://github.com/nii-yamagishilab/Capsule-Forensics-v2

[16] https://universe.roboflow.com/hithyshi-kandula-6fq72/forged-images/dataset/1