EasyChair Preprint

№ 7789

Deepfake Detection

Eshit Bansal and A. Helen Victoria

April 14, 2022

Deepfake Detection

Eshit Bansal
Dept. of Networking & Communication
SRM Institute of Science & Technology
Kattankulathur, Tamil Nadu, 603203
eb5153@srmist.edu.in

Mrs. A. Helen Victoria
Dept. of Networking & Communication
SRM Institute of Science & Technology
Kattankulathur, Tamil Nadu, 603203
helenvia@srmist.edu.in

***Abstract*—Deepfakes are synthetic media that are made by digitally modifying an existing image, video or audio, so that they appear to portray someone else from what they originally did. Deepfakes are popular in spreading malicious false information across the general populace. This is because the quality of the deepfakes being developed is improving with time as a result of breakthroughs in 'Data Science' in general. It has becoming more difficult to distinguish between an original and a deepfake (well-made) media for the same reason. As a result, being able to distinguish between the original and the deepfake becomes critical, as any disinformation spreads like wildfire on social media, causing problems for everyone.**

The goal of this project was to create a model that, when fed digital media (such as video), could determine whether it was a deepfake or not. The training data consisted of videos that had been pre-processed so that only a few frames from each video were extracted. The retrieved frames are then sent to retinaface, which extracts only the section of the frame (image) that contains a person's face. Utilizing the information gathered in the previous step, the frame is cropped before being subjected to various augmentations and experiments using the XceptionNet and EfficientNet (and its variants) models. To determine the accuracy of the resulting model, a log-loss function was used. The initial model runs resulted in a score of 0.6∼0.7, which has since been improved to 0.199.

As a result, the project has successfully produced a model that can predict deepfakes (in this case, videos), with an accuracy benchmark of 0.199. This score could be improved even further with additional optimization.

***Keywords*—Deepfake; NLP, retinaface, log-loss, XceptionNet, EfficientNet.**

I. INTRODUCTION

A. About

Deepfake is a type of artificial intelligence used to create convincing pictures, audio, and video scams. The phrase is a combination of deep learning and fake, and it describes both the technique and the ensuing fraudulent content. While faking information isn't new, deepfakes use advanced machine learning and artificial intelligence techniques to edit or synthesise visual or audio content that has a high potential for deception. Deepfakes have gotten a lot of publicity because of their use in celebrity pornography, revenge porn, fake news, hoaxes, and financial fraud. This has prompted business and government efforts to detect and prohibit their use.

B. Ways of how deepfakes are being maliciously used nowadays

Disinformation and hoaxes have progressed from minor annoyances to high-stakes warfare aimed at sowing strife, increasing polarisation, and, in some cases, swaying election results. Deepfakes are a new method for disseminating computational propaganda and deception on a large scale and quickly.

The availability of low-cost cloud computers, algorithms, and large amounts of data has created the ideal storm for democratising media creation and manipulation.

Regardless of who they are, where they are, or how they listen, speak, or communicate, synthetic media can open up possibilities and chances for everyone. It may give people a voice, a sense of purpose, and the potential to make an influence on a large scale and with speed. However, as with any new breakthrough technology, it has the potential to be weaponized to cause harm.

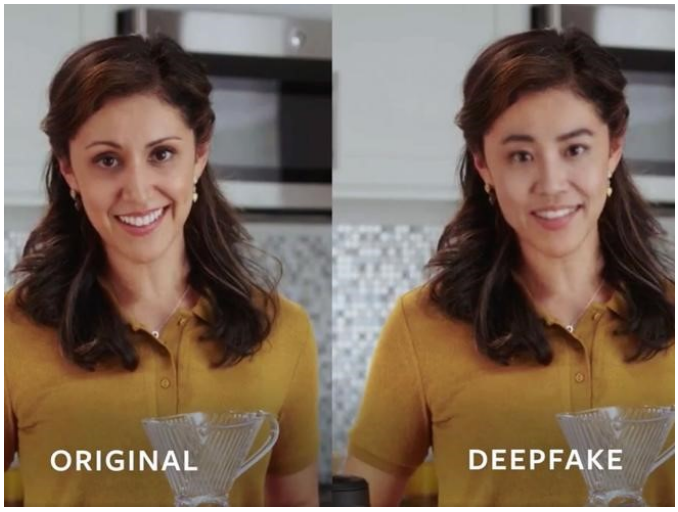

Fig. 1. (a) An original image and (b) a deepfake image of the original image

Individuals, institutions, businesses, and democracy can all be harmed by deepfakes, which are hyper-realistic digital falsifications. They allow for the fabrication of media — swapping faces, lip-syncing, and puppeteering — with little or no agreement, posing a threat to psychology, security, political stability, and corporate disruption. Deepfakes can be used by nation-state actors with geopolitical ambitions,

ideological believers, violent extremists, and commercially motivated companies to control media narratives with ease and scale never seen before. [1] [2]

C. Motivation

We need a multi-stakeholder and multi-modal approach to defend the truth and preserve freedom of expression. To tackle the threat of malevolent deepfakes, collaborative activities and communal strategies across legislative restrictions, platform policies, technical intervention, and media literacy can provide effective and ethical countermeasures.

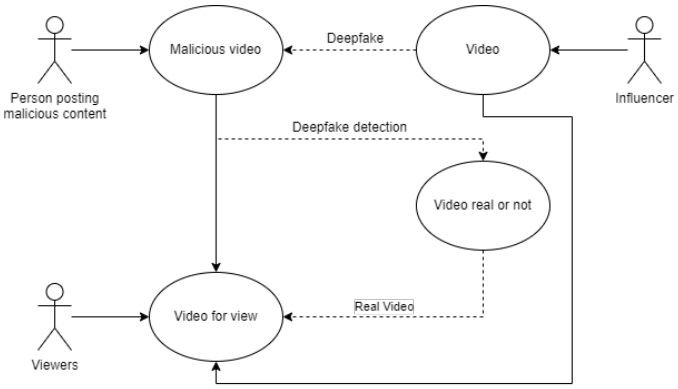

Fig. 2. Use Case Diagram

Deepfakes can be used to manipulate the stock market. Many more phoney claims regarding other brands have surfaced, such as Coca-Cola cancelling Dasani bottled water due to ”clear parasites,” an Xbox console killing a teenager, and Costco terminating its membership programme. Wilmerhale's Ferraro, Chipman, and Preston outline the legal and business risks of disinformation and deepfake in their paper, which focuses heavily on the harm to a corporation. They explicitly point out that organisations risk not only losing the value of fraudulent funds and reputational goodwill, but also facing shareholder action, regulatory investigations, and the loss of access to more financing. [3]

II. LITERATURE SURVEY

A. How deepfakes are made?

FakeApp, produced by a Reddit user utilising an autoencoder-decoder pairing structure, was the first attempt at deepfake creation. The autoencoder obtains latent features from facial images, and the decoder reconstructs the images in that fashion. Two encoder-decoder pairs are required to switch faces between source and target images; each pair is used to train on an image set, and the encoder's parameters are shared between two network pairs. In other words, the encoder networks of two pairs are identical. This technique allows the common encoder to detect and learn the similarity between two sets of face images, which is very easy because faces have comparable features like eyes, noses, and mouth positions. [4]

To improve the quality of the deepfake material, an algorithm is formed by combining two AI algorithms, one of which is known as the generator and the other as the discriminator. The discriminator is asked by the generator, which develops the phoney multimedia content, to assess whether the content is real or artificial. A generative adversarial network (GAN) is formed when the generator and discriminator operate together. Each time the discriminator correctly recognises faked content, it gives the generator vital feedback on how to enhance the next deepfake.

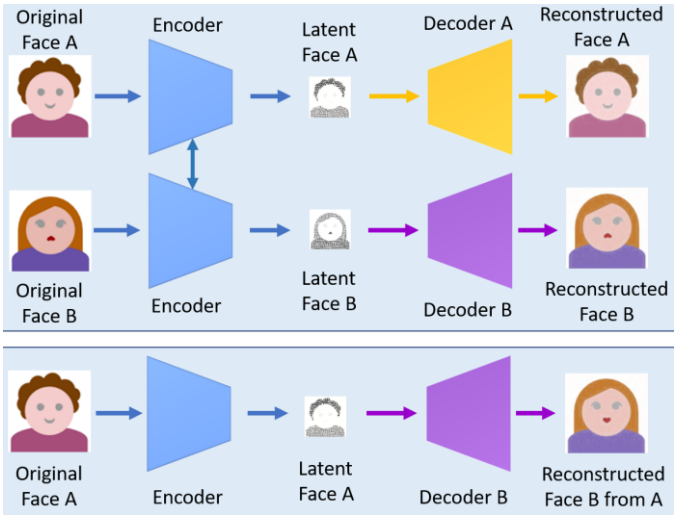

Fig. 3. A deepfake creation model using two encoder-decoder pairs

Identifying the desired output and creating a training dataset for the generator are the initial steps in setting up a GAN. Video clips can be supplied to the discriminator once the generator has reached an acceptable level of output. The discriminator improves at recognising bogus video clips as the generator improves at making them. In turn, as the discriminator improves at detecting bogus video, the generator improves at producing it. [5]

B. Deepfake Detection

Deepfake detection is typically thought of as a binary classification problem, in which classifiers are employed to distinguish between genuine and altered videos. [6] To train classification models, this type of technique necessitates a big library of real and fake videos. Although the quantity of fake videos is growing, there are still limitations in terms of establishing a benchmark for verifying various detection methods. [7] [8]

- **Rossler et al.** [9] propose an automatic facial manipulation detection benchmark to standardize the evaluation of detection methods. DeepFakes, Face2Face, FaceSwap, and Neural Textures are used as significant examples of facial modifications at random compression levels and sizes in the benchmark. They conduct a thorough analysis of data-driven forgery detectors using this information. They show that using additional domain-specific information forgery detection increases to unprecedented levels of

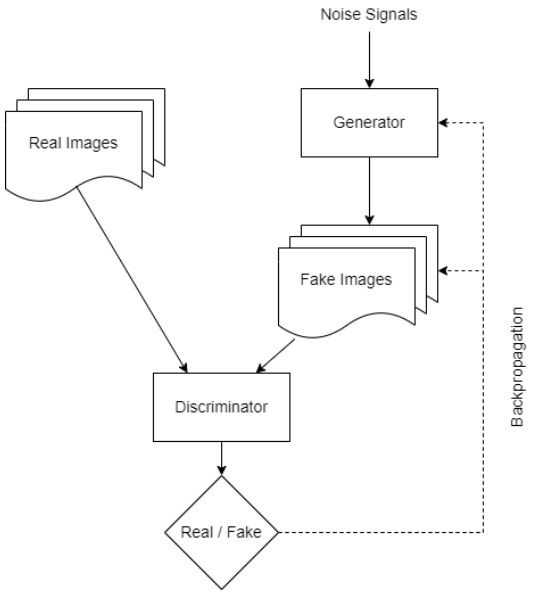

Fig. 4. The GAN architecture consisting of a generator and a discriminator

accuracy, even in the presence of significant compression, and beats human observers.
The above mentioned paper mainly focuses on facial forgeries and does not talk much about the audio forgeries.

- **Nguyen et al.** [10] propose a capsule network that can identify a variety of attacks, ranging from presentation attacks including printed images and repeated videos to deep learning-based attacks involving fake videos. With similar performance, it requires far less parameters than typical convolutional neural networks. Furthermore, they describe the theory underlying the application of capsule networks to the forensics problem through extensive analysis and visualisation for the first time in the literature (according to them).
  According to the authors, the enhancements have offered Capsule-Forensics performance that is comparable to or better than state-of-the-art approaches while requiring fewer parameters, which helps minimise the computation cost. [16] By visualising the activation of each capsule and the entire network, and analysing the agreement between the primary capsules for video input, a detailed analysis of how the Capsule-Forensics works was able to explain the mechanism that helped the Capsule-Forensics perform well on several digital forensics tasks. According to them, future work could involve applying capsule networks to time series input rather than just frame aggregation.

III. METHODOLOGY AND RESULTS

This section presents the appraoch proposed to make the model. The whole process was divided into multiple steps -

A. Data Acquisition

The dataset was downloaded from Kaggle, and contained 20,795 videos with a random file name with the mp4 extensions. The data also contains a CSV file which contains info about a video file (the column here has values of the file name), and a label about if the given training video is a **REAL** video or a **FAKE**. If a given video is marked as *FAKE* then the CSV also contains the info of the original video. An analysis was done on the given dataset so as to decide the appropriate split for the training and testing phase. The data size was approximately 97GB.

B. Selection of the initial model

The second step was to search about the baseline model using which the initial testing runs can be done. On the basis of current research done regarding deepfake detection, XceptionNet was selected as the initial model to start the training iterations. [11]

C. Testing using Log loss Score

The accuracy of the models at every stage in the program was calculated using the log loss score. This was done, as log loss is one of the major metrics to assess the performance of a classification problem.

We want the observation to be predicted with a probability as near to the real value (of 0 or 1) as possible while training a classification model. As a result, log-loss is an excellent choice for a loss function when training and optimising classification models, because the prediction probability is penalised the further away it is from its true value.

$$LogLoss = \frac{-1}{n}\Sigma[y_i.log_e(\hat{y_i}) + (1 - y_i).log_e(1 - \hat{y_i})] \quad (1)$$

D. Algorithm (s) / Implementation

1) Extract Frames from videos: Videos have to be first converted into a more accessible format so that the deepfake detection can be done. For this purpose, the videos had to be first processed so as to form individual frames (images) from a complete video. At the top of that since extracting all the frames and saving them as images would create a lot of data, the data can be trimmed such that instead of saving each and every frame, frames would now be extracted on a periodic basis. For example, one scenario can be to retrieve a frame after every 5 frames (n frames so as to say) from the video. This has to be done in the pre-processing step as the frame extraction for such large amount data when done on-fly can make the whole process even slower (the overall training process was already slow due to the large size of the training data).

Another scenario can be to extract a frame every 0.5 seconds. This would mean that for a 10 second video, the number of frames extracted would be 20.

2) Extracting a face from a given frame (image): After the frame extraction process has been completed, we are then required to extract the face from the given frame. This is required since –

- 1. Even after the frame extraction process was completed, the data size of the whole data is very large. This does not mean that the data is un-processable, it only means that the data can not be processed by a normal machine.
- 2. The frames extracted have extra information from what is required. For example, for deepfake detection the only feature we are interested in the image is the face itself, and not the background.

Thus, we searched for a good face extraction module which can help us achieve this. We checked the web and found **Retinaface** which is claimed to be a leading facial detector as its detection performance is amazing even in the crowd. [12]

Fig. 5. Single frame from a video

The frames were thus then subjected to retinaface after which data points like *height, width, coordinates* are returned alongside a confidence score, which tells about the confidence retinaface has for a given identified face, indeed being a face.

Fig. 6. Face cropped from image refered in Fig. 5

Using this data, we then crop our original frame such that now the training data has even more filtered and smaller size data.

3) Problems that occurred in the pre-processing step: Here are a few problems which we faced when working on the pre-processing step –

1) Many a times, the cropped frame didn't have equal size. To solve this problem, each face extracted from a given video was resized to match the maximum size of a face extracted from the video.
2) Some videos had 2 people. To detect this, set threshold to confidence score and if the confidence score of the second confident face detected is bigger than the threshold then confirm that these are two people.
3) In many cases the detector (retinaface) was able to find a face in a given video in some frames, but was unable to do the same in other frames of the same video. For this case, the frames which yielded no face were removed from the pre-processed set of data.
4) In some videos, no face was detected at all. Here reduce the confidence threshold for the first confident face.

E. Augmentations

Image augmentation is a method of modifying existing images in order to generate additional data for the model training process. In other words, it is the technique of artificially increasing the dataset available for deep learning model training.

So far using the baseline XceptionNet model a score of 0.6 was achieved. Thus, it was investigated if different augmentations can potentially improve the score or not. Do note that previously no augmentation was planned due to an assumption that augmentation would distort features that were introduced by manipulation. Though when the results were generated, it was found that the augmentations instead made the model more robust. Some augmentations made were - shift, scale, rotate, rgb shift, brightness, contrast, hue, saturation, value. After applying these basic augmentations, the score improved from the previous value of 0.6 to 0.4.

F. Selecting a better model

Since we got a major jump in quality after trying some basic augmentations we decided to then first finalise on the training model we are using, so that the same is not required to be changed again at a later point in time.

After we got some initial results using XceptionNet, we tried variations in the training model, for e.g. U-Net etc. While experimenting with these models, an improve in score was recorded after using EfficientNet. The EfficientNet-b0 model improved the score to 0.39. This is not a large value, but was certainly a sign that more can potentially be explored in this direction. We then tried experimenting with different variations of EfficientNet, for e.g. when switching from EfficientNet-b0 to EfficientNet-b3, the score improved to 0.36. Later when the training model was again changed to EfficientNet-b7, the score now improved to 0.33.

More variations were tested but no significant results were recorded which would surpass the currently achieved benchmark of 0.33.

G. Experimenting with the model

We tried to incorporate audio into the image model using a spectrogram as a second input in our initial experiments, but it didn't assist much and complicated the process, so we dropped it. We used the bounding box information from original videos to process matching fake videos after processing all real videos. After we finished processing all of the videos, we updated the metadata with the newly retrieved data.

Since the training model was now decided we then decided to experiment with the model itself, instead of making augmentations to the data (we will do more augmentations later). Here are some experiments which did work and hence improved the score of the model from the previous benchmark of 0.33 to now 0.3 –

1) Tune learning rate considering the batch size. Batch size – 24 and learning rate – 0.0002.
2) Tune ratio of decreasing the learning rate when plateau (0.1 to 0.3).
3) Use different frames for each video every epoch (#0 → #7 → #14 → #1 → #8 →...)
4) Increase in resolution of the image
5) Use conservative fix: multiply constant (¡1) to the logits than take sigmoid. It helps improve logloss when training and testing distributions differ.

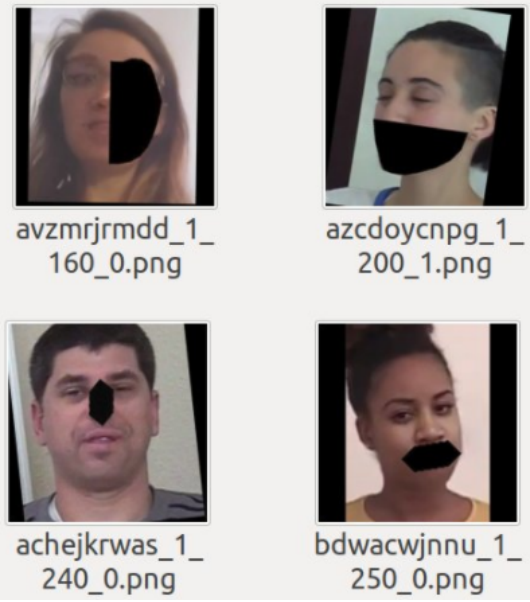

Fig. 7. An example of the cutout type augmentation to catch the blending artifacts

H. More augmentations and Experiments

Since we previously got a significant boost in score after applying some simple augmentations, we now again tried to make some augmentations like adding noise and blur, and increasing the degree of augmentations (applying a number of different kinds of augmentations, all at the same time).

We then tried to apply some domain specific augmentations, specifically the ones to catch the blending artifacts possibly created due to the deepfake. To catch the blending artefacts we black our landmarks like eyes, nose or mouth, and used MTCNN landmarks for that. [13]

I. Increase in the face margin

When we were making initial experiments, we found an improvement in the score when trying to tune the face margin parameter. Thus, as a result we now, specifically investigated into if a change in the face margin value does improve the score or not.

1) For a start, we had previously made only minute changes in the value of the face margin parameter. The original value was 0.05 which was then updated to 0.1. As already mentioned, this change had previously improved the score.
2) Now, this time around we made a large change in the value, starting from an original value of 0.1 to an updated value of 0.5. The resultant b5 model had an improved score of 0.222, from the previously attained score of 0.27 (which was not improving even after different regular experiments).
3) Thus, after this we changed the value to 1.0, after which we used binary search to pinpoint the face margin parameter which gives us the best score. This face margin parameter value was found out to be 0.7 (crop margin of 0.15). At this point the score had now been improved to 0.218.
4) More experiments were now made by using other efficientnet models, and it was found that a b4 model with the same face margin parameter value gives an even more improved score of 0.214.

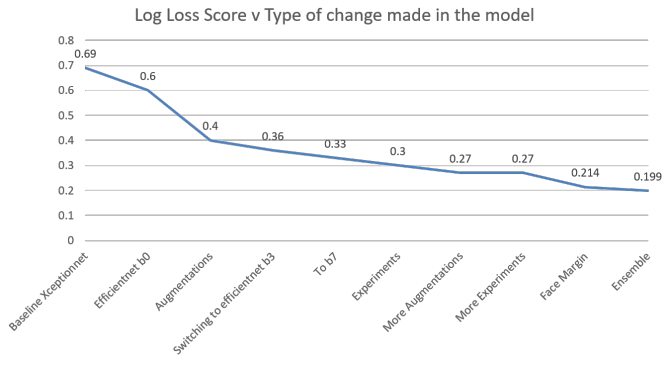

Fig. 8. Graph displaying the results achieved over different experiments made when making the deepfake detection model

J. Ensemble Model

As previously stated, an ensemble model can potentially improve the score of the model. Hence, experiments were made by combining different models to make an ensemble

model, after which a final ensemble model was made using 4 b4 efficientnet models, and 2 b5 efficientnet models.

The final model was made using a simple average, and the resultant score now improved to 0.199.

K. Block Schematic Diagram

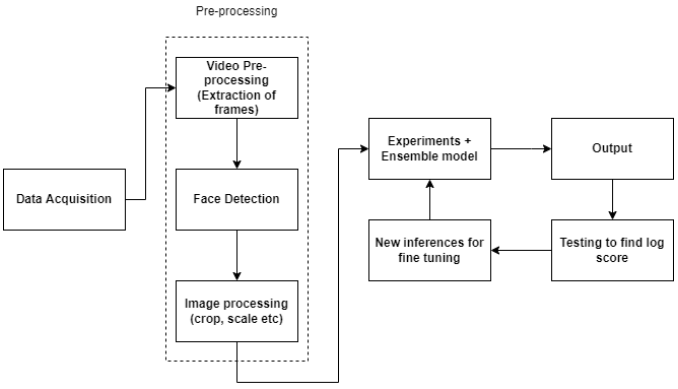

Fig. 9. Block Diagram

The section briefs the steps used for developing the deepfake detector. Initially, the input videos were split into individual frames, out of which only a select few frames were selected. Each frame was then further subjected to retinaface, so as to crop the face from the given images.

A base model was then selected to train the data, for this some initial runs were made using XceptionNet, which was then later changed to EfficientNet and its variants.

The model was then subjected to different augmentations (in the images extracted) and experiments in the model parameters using which a log loss score of 0.199 was achieved.

IV. CONCLUSION

The resultant model is able to detect deepfake videos, though still it is only able to make detections on the video itself and not the audio part of the video. There is a possibility that the lower accuracy of the model is being caused due to the model not able to predict the audio deepfakes.

REFERENCES

[1] S. Karnouskos, "Artificial intelligence in digital media: The era of deepfakes," *IEEE Transactions on Technology and Society*, vol. 1, no. 3, pp. 138–147, 2020.

[2] G. P. Zachary, "Digital manipulation and the future of electoral democracy in the u.s." *IEEE Transactions on Technology and Society*, vol. 1, no. 2, pp. 104–112, 2020.

[3] A. S. Uçan, F. M. Buçak, M. A. H. Tutuk, H. b. Aydin, E. Semiz, and e. Bahtiyar, "Deepfake and security of video conferences," in *2021 6th International Conference on Computer Science and Engineering (UBMK)*, 2021, pp. 36–41.

[4] H. A. Khalil and S. A. Maged, "Deepfakes creation and detection using deep learning," in *2021 International Mobile, Intelligent, and Ubiquitous Computing Conference (MIUCC)*, 2021, pp. 1–4.

[5] D. Yadav and S. Salmani, "Deepfake: A survey on facial forgery technique using generative adversarial network," in *2019 International Conference on Intelligent Computing and Control Systems (ICCS)*, 2019, pp. 852–857.

[6] Y. Li, C. Zhang, P. Sun, L. Ke, Y. Ju, H. Qi, and S. Lyu, "Deepfake-o-meter: An open platform for deepfake detection," in *2021 IEEE Security and Privacy Workshops (SPW)*. Los Alamitos, CA, USA: IEEE Computer Society, may 2021, pp. 277–281. [Online]. Available: https://doi.ieeecomputersociety.org/10.1109/SPW53761.2021.00047

[7] K. N. Ramadhani and R. Munir, "A comparative study of deepfake video detection method," in *2020 3rd International Conference on Information and Communications Technology (ICOIACT)*, 2020, pp. 394–399.

[8] D. Pan, L. Sun, R. Wang, X. Zhang, and R. O. Sinnott, "Deepfake detection through deep learning," in *2020 IEEE/ACM International Conference on Big Data Computing, Applications and Technologies (BDCAT)*, 2020, pp. 134–143.

[9] A. Rössler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Niessner, "Faceforensics++: Learning to detect manipulated facial images," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 1–11.

[10] H. Nguyen, J. Yamagishi, and I. Echizen, "Use of a capsule network to detect fake images and videos," 10 2019.

[11] A. Kumar and A. Bhavsar, "Detecting deepfakes with metric learning," 3 2020.

[12] S. I. Serengil and A. Ozpinar, "Hyperextended lightface: A facial attribute analysis framework," in *2021 International Conference on Engineering and Emerging Technologies (ICEET)*. IEEE, 2021, pp. 1–4. [Online]. Available: https://doi.org/10.1109/ICEET53442.2021.9659697

[13] L. Li, J. Bao, T. Zhang, H. Yang, D. Chen, F. Wen, and B. Guo, "Face x-ray for more general face forgery detection," 06 2020, pp. 5000–5009.