”DEEPFAKE DEFENDER – REAL-TIME VIDEO CALL INTEGRITY CHECKER USING DEEP LEARNING AND EXPLAINABLE AI ”

1st Joe William A
Department Of Advanced Computing and Analytics
Vels Institute of Science, Technology & Advanced Studies
Chennai, India
joewilliam147@gmail.com

2nd Dr.S.Prathiba
Department Of Advanced Computing and Analytics
Vels Institute of Science, Technology & Advanced Studies
Chennai, India
prathiba.scs@vistas.ac.in

***Abstract*—Abstract— The Deepfake technology has developed at an incredible speed in the recent years and has become a threat to our digital security. There are many instances where we communicate with people in real time and thus have to worry about their identities being verified. Despite the numerous techniques and methods that have been proposed to resolve this deepfake detection problem, none of the current solutions have proven to be robust enough to work accurately in a real time scenario, due to inconsistencies in their performance.**

In this study, a system for the real-time detection of deepfake is proposed focusing on image analysis method and relying on the analysis of audio signals as well. For the analysis of visual signals, an EfficientNet is used after processing the data and providing the appropriate training. Audio signals are converted into Mel- spectrograms and are also analyzed using a CNN to highlight the characteristics of the voices.

Real time Deepfake Video Detection using CNN with Mel Spectrogram and Image Pre processing Index Terms—Deepfake Detection, Efficient Net, CNN, Mel Spectrogram The proposed system performs video detection of deepfakes in real time. It captures frames from the webcam of the device during runtime and classifies whether the input is real or fake. The image model achieves a classification accuracy in between 76–85% and the model performs stably during real time classification. Although it is not a perfect model, it shows that deep learning with both image and audio inputs can be used to strengthen trust in digital communication.

Spectrogram, Real-Time Systems, Digital Security.

***Index Terms*—Deepfake Detection, Efficient Net, CNN, Mel Spectrogram, Real-Time Systems, Digital Security.**

I. INTRODUCTION

Deep learning and AI have evolved over the years at a very rapid pace. In this era of rapid evolving technology, fake images, audio and videos are also on the rise. In particular, the evolution of deepfake technology has attracted worldwide attention. Deepfakes are a type of artificial video that use advanced AI to transform and edit a real video or audio into a fake video that is almost realistic and is hard to distinguish from real and fake [3], [10]. The major concern of deepfakes is that they pose threats to online security, privacy and can cause a tremendous amount of fake information to spread over the internet [13]. Many researchers have made various datasets for this purpose. They use real and fake samples in their dataset such as FaceForensics++ [2] and Deepfake Detection Challenge (DFDC) dataset [5]. The majority of the deepfake detection research is focused on images and videos. Researchers used many different approaches to identify the deepfakes. Some researchers have used the CNN-based model like MesoNet [11] that detects mesoscale changes in facial images to detect deepfakes. Others have used the RNNs [1], [8] to compare the frame changes in the videos. Some researchers also have a focus on the simple features like eye blinking in videos. If a video does not have a proper blinking feature in the eyes then that video is considered as a deepfake video. Audio deepfakes are also getting popular these days. New researches and technologies can make a voice sound very natural and human, making it hard to believe that it is a fake audio [9]. Hence, relying on only visual or audio features is not sufficient for the detection of deepfakes. Some researchers also used both audio and video features together. The FakeAVCeleb dataset [7] is an example of such a dataset. The deepfake detection using both audio and video is producing better results but they are not yet real time. This project aims to develop a simple and yet practical system for the deepfake detection, known as Deepfake Defender: Real-Time Video Call Integrity Checker Using Deep Learning and Explainable Artificial Intelligence. The proposed model is based on EfficientNet for image feature extraction and a CNN model for audio feature extraction after Mel spectrograms conversion. The output from the EfficientNet and the CNN model are then combined for the final detection. The proposed system also utilizes the webcam and microphone for real time video call deepfake detection. It also includes some basic explainable features that helps the users to know how the model produces the output. In this project, the focus is to develop a practical system that helps to improve trust in online communication [7], [3], [10].

II. LITERATURE SURVEY

Deepfake detection has become an increasingly important topic in recent years because of the rapid rise in DeepFake images, videos and audio as a result of the evolution of deep learning. As a matter of fact, works such as Verdoliva [15] and Nguyen et al. [10] illustrated the DeepFake generation process, and discussed the difficulties of the deepfake detection problem that however heavily change depending on the specific context at hand.

Datasets are very crucial in building up a detection system. FaceForensics++ [3] is one of the most used datasets, which has a large collection of real and fake face videos. DFDC dataset [5] is also used which has variety of samples with different types of fake contents. This is very helpful for the training and testing of the models.

Recently, several works exploited deep learning for both face detection and face forgery identification. Later, the research started focusing on multimodal approaches. Indeed, Khalid et al. introduced the FakeAVCeleb dataset and demonstrated with an experimental evaluation how the detection performances are degraded if only one modality (fake video or fake audio) is available for the test. In the literature, several approaches have been recently proposed to deal with deep learning-based Fake Video Detection. In [1] the authors proposed an approach relying on Recurrent Neural Networks (RNNs) to investigate temporal dynamics in video sequences for a strong fake video detection framework. In a more recent work, instead, Sabir et al. have implemented a novel fake video detection framework leveraging both Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs).

Most existing works for forgery detection utilize audio or other than visual features. In this paper, we propose a new approach called EyeCount which detects forgery by examining abnormal eye blinking behaviors. Other works based on visual cues involve detecting facial microscopic anomalies [11], or exploiting both spatial and motion features [12].

Jia et al. [9] demonstrated that with the recent advancements in audio deepfakes, it is possible to generate very natural and human sounding audio which further emphasizes the importance of multi-modal detection. Other works have also proposed the use of Capsule Networks [6] as well as lip-sync analysis [4].

A large number of researches have already been carried out in order to solve the problems mentioned in the preceding subchapters. However, these researches do not completely solve the problems of real time image and audio processing and integration. This project is intended to solve those problems.

III. PROPOSED METHODOLOGY

There are many detection techniques designed for deepfakes already. However, most of the approaches fail to capture the whole spectrum. As of now, most models only rely on images or only check a few video frames. Few of them explore the details of faces and audio samples. We are not the first one to notice these little aspects like blinking. There are several papers on this subject such as this paper "Eye Blinking Detection Based on Face Tracking" from 2010. Some papers explore deep learning based approaches like "FaceForensics" which trains CNN models (like MesoNet [11]) to detect slight modifications to input faces. Another similar approach is [1], [8] which focuses on analyzing temporal patterns within individual video frames. Recently, a paper called "FakeAVCeleb" [7] comes out and they have a nice dataset of fake videos and audio that could be highly beneficial for the proposed research as it has both video and audio. However, these current approaches have the following drawbacks. • One source: Most of these models rely heavily on one single aspect such as face or audio samples. It limits the performance of these models. They may not classify as "fake" when they should because some face or audio aspects have not been caught. • Low accuracy and speed: Most of these approaches may work for a few particular cases but not for all scenarios, due to heavy dependence on quality of datasets used, complexity of their algorithms and many other parameters which restricts their overall performance. Hence they tend not to be used at an application level. Also many deep learning models like the RNN or Resnets have a long duration of training. Hence they tend to be slow to classify and therefore they do not fit into real-time applications. In addition, their intermediate outputs are mostly not explained and this creates mystery to the users on how the models actually classify and aspect which will be discussed later. This research addresses the following: - Improves the detection accuracy of Deepfakes - Uses both visual and audio signals for classification - It is a real-time application which supports the use case of video calling. The model is based on using the webcam and microphone. For face analysis, the EfficientNet is used for classifying the input faces into real or fake. For the audio part, the audio is preprocessed using Mel spectrogram technique. Then the audio is fed to a CNN. The final result is the weighted sum of the classifications given by both the models. However, the weighted sum gives higher preference to the classification given by the EfficientNet as the accuracy is greater for the images. The overall model can detect Deepfakes in real time and it can be also be used for video calling purposes. A number of explainable features have also been included to provide further clarity to users about the actual reason of classifying the video or audio as fake or real. Hence the proposed research improves the detection of the quality of Deepfakes by using multiple sources, provides an almost real-time application and increases user satisfaction to the proposed system as the users will have an idea how the model detects the Deepfakes.

IV. ARCHITECTURE DIAGRAM

Figure 1 provides a simplified overview of how this system operates. The two inputs to the system are provided by taking video frames from a video call using a webcam and audio recorded from the microphone.

The video component is analyzed using the face region to extract and analyze facial features through the image model for detection of the face being real or fake (i.e. based on minute differences in the face being examined by EfficientNet).

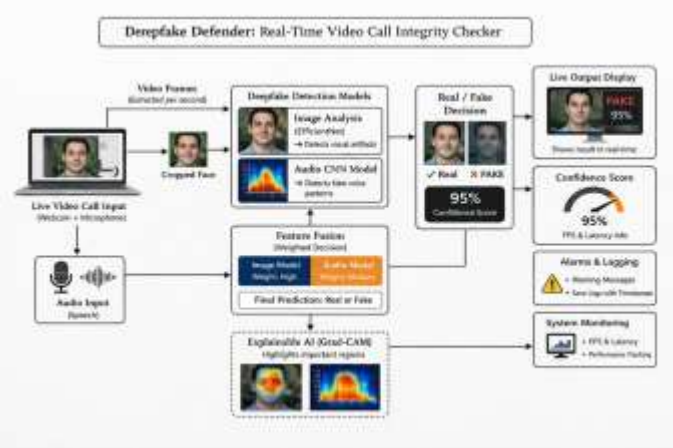


Fig. 1. DEEPFAKE ARCHITECTURE DIAGRAM

Along with processing video input, the audio input is processed by converting the recorded voice into a Mel spectrogram and then analyzed with the use of a CNN model to determine if the audio is original or created.

Once both inputs have been processed and the two results have been combined (with greater weight afforded to the image analysis and less weight to audio which is used to provide additional evidence to confirm the detection), the model will provide the final output indicating if the input is real or fake along with a confidence value for the detection of real or fake (the results is displayed live through the screen).

Also included in the output is Grad-CAM which identifies critical areas of the face that supported the model's determination of the presence or absence of the similarity in facial features. Additional output includes FPS (frames per second), delays, warnings and logs.

V. METHODOLGIES

In this project, deepfake is detected by using both image and audio. Earlier methods mostly use only one input, but here both are taken together. Because of that, the result becomes better in many cases.

A. DATA COLLECTION

First, some data is needed. For image, both real face images and fake images are used. For audio also, real voice and generated voice samples are taken.

While running, the system is not only using stored data. It also takes live input. Webcam is used for video and microphone is used for audio. These inputs are directly given to the model. So it can work in real time also.

1) Data Set:

B. PREPROCESSING

Before giving input, some changes are done.

Video is split into frames using OpenCV. From that, face part alone is taken using Haar Cascade. Background is removed in this step.

Fig. 2. A). 1. IMAGE DATASET

For audio, the signal is changed into Mel spectrogram.his helps to understand sound pattern.

It is written as:

$$S = \log \quad 1 + |\mathrm{STFT}(x)|^2 \tag{1}$$

where,

- S is the Mel spectrogram representation of the audio signal
- x is the input audio signal
- STFT(x) is the Short Time Fourier Transform of x
- $|\cdot|$ represents the magnitude
- log is the logarithmic scaling used to compress values

After that, normalization is done:

$$X' = \frac{X - \mu}{\sigma + \epsilon} \tag{2}$$

where,

- X' is the normalized output
- X is the original input data
- μ is the mean of the data
- σ is the standard deviation
- ϵ is a small constant to avoid division by zero

This step is mainly to keep values in same range.

C. FEATURE EXTRACTION

Here, model tries to take useful information.

For image, EfficientNet_B0 is used. It checks face details like small changes, edges, etc. Sometimes fake images have small errors, so this helps.

For audio, spectrogram is given to CNN. It learns voice pattern like pitch and frequency.

Basic formula:

$$Y = X * W + b \tag{3}$$

where,

- Y is the output feature map
- X is the input data (image or feature map)
- W represents the filter or kernel
- $*$ denotes the convolution operation
- b is the bias term

Activation:

$$\text{ReLU}(x) = \max(0, x) \quad (4)$$

where,

- x is the input value
- $\max(0, x)$ returns 0 if $x < 0$, otherwise returns x
- ReLU is used to introduce non linearity in the model

So overall, raw input is converted into features.

D. MODEL TRAINING AND PREDICTION

Now model is trained using real and fake data.
Loss is calculated:

$$\text{Loss} = -\sum_{i} y_i \log(p_i) \quad (5)$$

where,

- y_i is the true label (actual value)
- p_i is the predicted probability of class i
- log is the logarithmic function
- Σ represents summation over all classes

Weights are updated using Adam optimizer.
During testing, Softmax is used:

$$P(y_i) = \frac{e^{x_i}}{\Sigma e^{x_j}} \quad (6)$$

where,

- $P(y_i)$ is the probability of class i
- x_i is the input score (logit) for class i
- e is the exponential function
- Σe^{x_j} is the sum of exponentials over all classes

After this, output is given as real or fake.

E. FUSION

Both outputs are combined.
Final Score = (0.7 × Image) + (0.3 × Audio)
Image is given more weight because it gives stronger results. Audio is just support.

F. OUTPUT

Finally, result is shown.
It shows real or fake. Also confidence value is displayed. Output keeps updating continuously.
So it can be used during video calls.

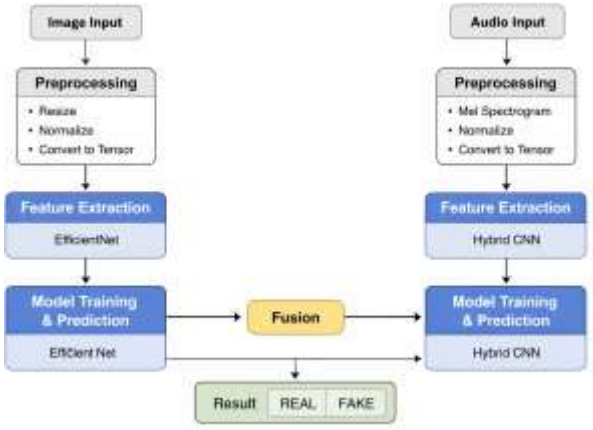


Fig. 3. IMAGE AND AUDIO INPUT FLOWCHART

VI. INPUT

We will be using the face images. These could be from the Real and Fake Face images or simply from the webcam if our app is running. These are prepared for the model. This involves resizing the images so that they are all of the same size. The pixel values are then normalised and changed into tensors. The images are then passed into our EfficientNet B0 model. These images are analysed by the model to see if the image is of a real face. It also checks the contours, skin patterns and texture of the real face. If the model is unsure that the images is a real face then it must be a fake face. The output of the model is then the answer. For audio, voice is used. It can be taken from a dataset or recorded using a microphone. Both real and fake voice samples are used for training. Audio cannot be used directly, so it is first changed into a Mel-spectrogram. This helps to see how the sound changes over time and makes it easier for the model to understand. After that, the values are adjusted. Then it is converted into tensor format. This is given to the CNN model. The model checks voice features like pitch and tone. If something is not normal, it may be fake. So in this system, both image and audio are used. Because of this, the checking becomes better. If only one input is used, sometimes mistakes can happen. Using both helps to reduce that.

VII. PSEUDO CODE AND IMPLEMENTATION

1) PSEUDO CODE: ALGORITHM 1: Deepfake Detection (Multimodal)

Input: Image (I), audio (A) and respective output of result and confidence.

Step 1: Start
Step 2: Load image model (EfficientNet)
Step 3: Load audio model (CNN)
Step 4: Take image input I from webcam or dataset
Step 5: Take audio input A from mic or dataset
Step 6: Process image Resize to 224 × 224 Adjust values Convert into tensor
Step 7: Process audio Convert to Mel spectrogram. Adjust values

Step 8: Give the image to the model. Get Image_Prob

Step 9: Give audio to model Get Audio_Prob

Step 10: Combine both: Final_Prob = (0.7 × Image_Prob) + (0.3 × Audio_Prob)

Step 11: the overall result (decision) will be determined based on the overall probability. Any value equal to or greater than 0.5 will be a determination of fake and anything lower will be determined to be a real.

Step 12: Show result with confidence

Step 13: Repeat for next input

Step 14: Stop

2) IMPLEMENTATION: The system is built using Python programming language along with Streamlit for user interface, PyTorch for neural network, OpenCV, NumPy and Librosa.

The system has two main parts: image analysis and audio analysis, which work separately at the initial stage. For image processing, the EfficientNet B0 model is used. It is a pre-trained model, and its final layer is modified to classify inputs as real or fake. The trained model is saved and loaded during execution.

Audio classification is done using a CNN model. Audio is first preprocessed into a Mel spectrogram using Librosa. This model then classifies the pitch and tone of audio based on the given training data to classify whether audio is real or fake.

OpenCV is being used here to capture a video from the webcam and a classifier of type Haar Cascade is being used to find the face features in that video. The audio from the microphone is being captured and processed too.

Both models produce probability scores, which are combined using a weighted formula:

Final Probability = (0.7 × Image) + (0.3 × Audio).

This application uses a buffering technique to maintain stability of the output. The application is implemented with Streamlit library and uses live video, face detection and confidence values.

VIII. OUTPUT

The output of the system is the decision that the output panel displays. In other words, the output of the system is the decision whether the input image, audio or video is real or fake. The output is provided after the analysis of both image and audio segments. The output panel provides a probability value as a measure of how artificial the input is and a decision based on a predefined threshold value, which classifies the input as real or fake. For our implementation, if the probability value is above the predefined threshold value, the system decides that the input is fake, otherwise the input is classified as real. In addition to the decision, a confidence measure that is calculated as the square of the probability value, is provided to the user. Therefore, for large values of the probability measure, the system has more confidence on its decision. For the image and video input, the system provides the decision right on the output panel by drawing a bounding box around the face of the input person and displaying the corresponding decision at the top right corner of the output panel. Also, the system works in real time and hence when it is used in a video call,

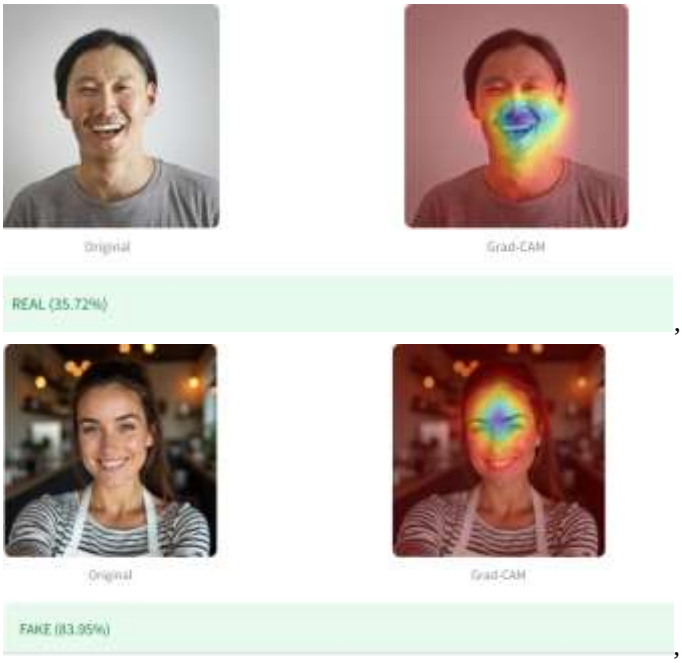


Fig. 4. 1. REAL IMAGE

it is always active and tries to classify the new video frames and audio segments that arrive in time. Hence the average probability value for the new video frame and audio segment is calculated to prevent sudden switching of decisions which is undesirable for smooth operation of the system. In this respect, the output panel that displays the confidence measures of the classification provided by the system is quite simple and clear. It displays confidence measures corresponding to the real or fake decisions of the input image, audio or video segments provided to the system separately.

IX. RESULT AND DISCUSION

1) IMAGE:

2) ACCURACY GRAPH:

3) LOSS GRAPH:

4) VIDEO DETECTION:

A. DISCUSSION

In summary, the findings suggest that using both image and audio together enhances the effectiveness of deepfake detection. To analyse data, researchers used two different models: EfficientNet for images and a conventional CNN for audio; each trained on both real and fake data, then combined their results.

When detecting differences between real and fake images, the 'image' model was accurate for many types of changes, such as the presence of a texture difference at the face or the presence of an unusual facial expression. However, there is also a possibility that the image will be very convincing and the model will fail to recognise it as a deepfake. For audio models, variations between voice features (e.g., pitch and tone) can generally be recognised; however, external noise or poor

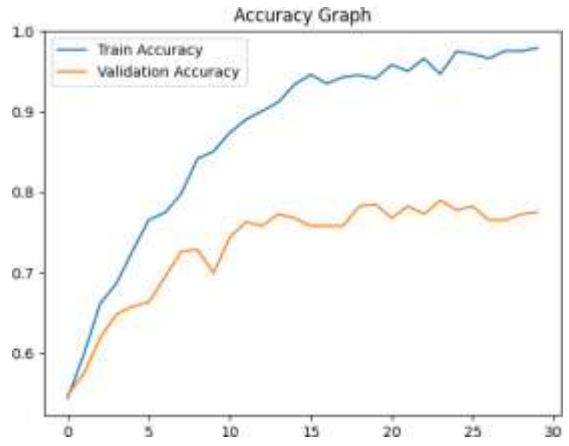


Fig. 5. 2. TRAIN IMAGE

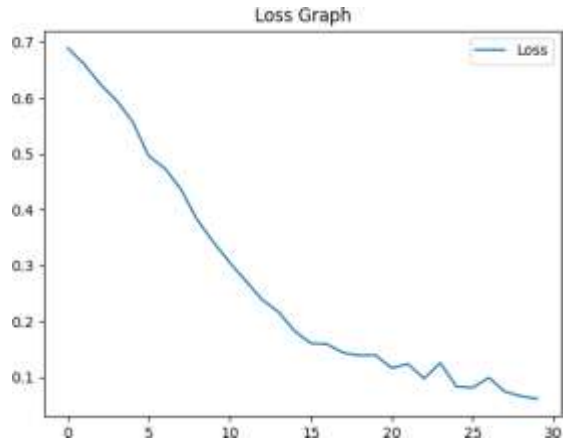


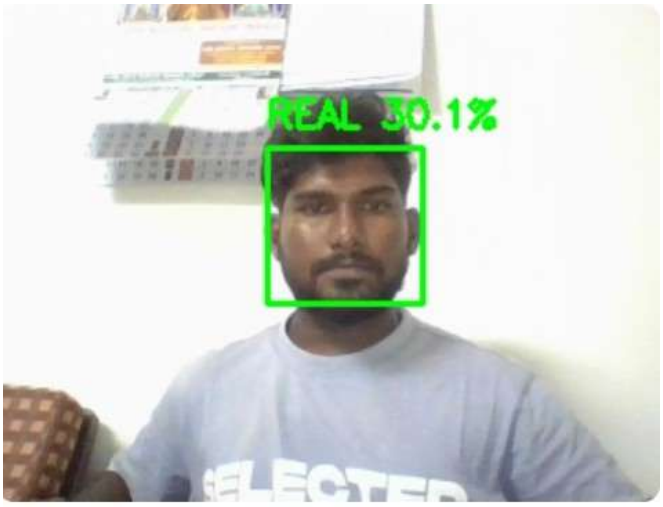
Fig. 6. 3. TRAIN IMAGE

sound quality could also limit the audio model's ability to detect these variations.

Based on the results of these two models and their limitations, the use of one type of input alone (i.e., either audio or image) is ineffective, therefore both audio and video (with an emphasis placed on the video component) need to be combined for this task.

The work also makes use of Grad-CAM techniques to produce visualisations for specific regions of the face relevant to the decision-making process, thus allowing for an explanation of how decisions have been made in regard to deepfake detection. A webcam and an associated microphone can be used for real-time testing of this system; averaging methods are used to produce reliable results.

The performance of this system ultimately depends upon the quality of captured data as well as surrounding environmental conditions; nonetheless, it is an excellent and viable option for use in various applications (e.g., video conferencing, security).



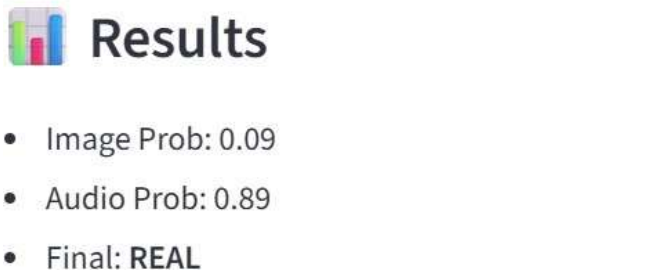


Fig. 7. 4. WEBCAM IMAGE

X. CONCLUSION

My conviction is that this Deepfake Detection project will provide additional facial and accompanying tailored data, which is needed as input in the detection of deepfakes. As a result, the overall quality of deepfake detection will significantly exceed that of any previous single source methods currently in use. Many factors cause the negative consequences of deepfakes, including limited visibility, lack of background noise or at least noise that is not relevant to the discussion, etc. More information available to train the model means better separation of real vs. fake (in this example, it will be fairly easy to separate but is still a good starting point to help address security concerns related to both live video chat, social networking sites and broadcasts).

REFERENCES

[1] 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.

[2] Shruti Agarwal, Hany Farid, Ohad Fried, and Maneesh Agrawala. Detecting deep-fake videos from phoneme-viseme mismatches. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 660–661, 2020.

[3] Shruti Agarwal, Hany Farid, Yuming Gu, Mingming He, Koki Nagano, and Hao Li. Protecting world leaders against deep fakes. In *CVPR workshops*, volume 1, 2019.

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

[5] David Gu¨era and Edward J Delp. Deepfake video detection using recurrent neural networks. In *2018 15th IEEE international conference on advanced video and signal based surveillance (AVSS)*, pages 1–6. IEEE, 2018.

[6] Xintong Han, Vlad Morariu, Peng IS Larry Davis, et al. Two-stream neural networks for tampered face detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 19–27, 2017.

[7] Ye Jia, Yu Zhang, Ron Weiss, Quan Wang, Jonathan Shen, Fei Ren, Patrick Nguyen, Ruoming Pang, Ignacio Lopez Moreno, Yonghui Wu, et al. Transfer learning from speaker verification to multispeaker text-to-speech synthesis. *Advances in neural information processing systems*, 31, 2018.

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

[9] Yuezun Li, Ming-Ching Chang, and Siwei Lyu. In ictu oculi: Exposing ai created fake videos by detecting eye blinking. In *2018 IEEE International workshop on information forensics and security (WIFS)*, pages 1–7. Ieee, 2018.

[10] Huy H Nguyen, Junichi Yamagishi, and Isao Echizen. Capsule-forensics: Using capsule networks to detect forged images and videos. In *ICASSP 2019-2019 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 2307–2311. IEEE, 2019.

[11] Andreas Rossler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, and Matthias Nießner. Faceforensics++: Learning to detect manipulated facial images. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1–11, 2019.

[12] Ekraam Sabir, Jiaxin Cheng, Ayush Jaiswal, Wael AbdAlmageed, Iacopo Masi, and Prem Natarajan. Recurrent convolutional strategies for face manipulation detection in videos. *Interfaces (GUI)*, 3(1):80–87, 2019.

[13] Luisa Verdoliva. Media forensics and deepfakes: an overview. *IEEE journal of selected topics in signal processing*, 14(5):910–932, 2020.