International Journal of Information Management Data Insights 2 (2022) 100054

Contents lists available at ScienceDirect

International Journal of Information Management Data Insights

journal homepage: www.elsevier.com/locate/jjimei

Exposing deepfakes using a deep multilayer perceptron – convolutional neural network model

Santosh Kolagati, Thenuga Priyadharshini, V. Mary Anita Rajam*

Department of Computer Science and Engineering, College of Engineering Guindy, Anna University, Chennai, India

ARTICLE INFO

Keywords:
CNN
Multilayer perceptron
Deepfake detection
Mixed data
classification

ABSTRACT

Creating deepfakes has rapidly become easier and more accessible due to advancements in hardware and computing. The harmful nature of deepfakes urges immediate action to improve detection of such doctored videos. In this work, we build a deep hybrid neural network model to detect deepfake videos. Using facial landmarks detection, we extract data pertaining to various facial attributes from the videos. This data is passed to a multilayer perceptron to learn differences in real and deepfake videos. Simultaneously, we use a convolutional neural network to extract features and train on the videos. We combine these two models to build a multi-input deepfake detector. A subset of the Deepfake Detection Challenge Dataset along with the Dessa Dataset is used to train the model. The proposed model provides good classification results with an accuracy of 84% and an AuC score of 0.87.

1. Introduction

Deepfakes are synthesized images and videos that employ a myriad of powerful artificial intelligence and deep learning techniques. Existing images and videos are enough to create convincing, highly deceptive[1] deepfakes by combining or superimposing them onto source images and videos, which may then go on to be used to spread fake news, malicious hoaxes, and financial fraud among other use. For instance, deepfakes have been used to alter the appearances and speech patterns of well-known politicians to depict them in a negative light.[2]

With the increasing importance of social media as a means of disseminating news, online disinformation campaigns have received significant attention in recent years. While social media makes the dissemination of fake news easier, computer vision tools have contributed to this trend by making it easier to generate fake imagery. While an image manipulator in prior years would need significant experience with rendering and/or image manipulation software, modern data-driven approaches have made it much easier to generate artificial imagery from scratch. Thus, deepfake videos or images can cause an unprecedented amount of damage in a political environment as well as the personal lives of many people.

For instance, deepfakes have been used to misrepresent well-known politicians on video portals or chat rooms. Former US President Donald Trump routinely spread nefariously created deep fake videos of his opponent, current US President Joe Biden (Frum, 2020). The videos were edited and spoofed in an attempt to mock Biden, but as more of Trump's supporters viewed and shared the video even during the crucial days leading up to the 2020 US Election, the videos were moderately effective in disparaging Biden's image, at least for some people (Burns, 2020). If left unchecked, deepfakes could sow disarray.

Deepfakes can also have positive implications, however, as they can be used to create voices and AI-based personalities to aid individuals who are blind. It is also possible to recreate and update scenes in movies without reshooting them, which is especially useful in preserving legacy

Abbreviations: CNN, Convolutional Neural Network.

* Corresponding author at: Department of Computer Science and Engineering, CEG, Anna University, Chennai, India.
E-mail addresses: ksantosh1399@gmail.com (S. Kolagati), thenugapriya@gmail.com (T. Priyadharshini), anitav@annauniv.edu (V. Mary Anita Rajam).

[1] DFDC dataset. Retrieved from https://www.kaggle.com/c/deepfake-detection-challenge/data.

[2]

MLP-CNN	0.84	0.83	0.87	0.877
CNN-only	0.84	0.72	0.74	0.669

DFDC dataset. Retrieved from https://www.kaggle.com/c/deepfake-detection-challenge/data.

https://doi.org/10.1016/j.jjimei.2021.100054
Received 7 June 2021; Received in revised form 12 December 2021; Accepted 12 December 2021

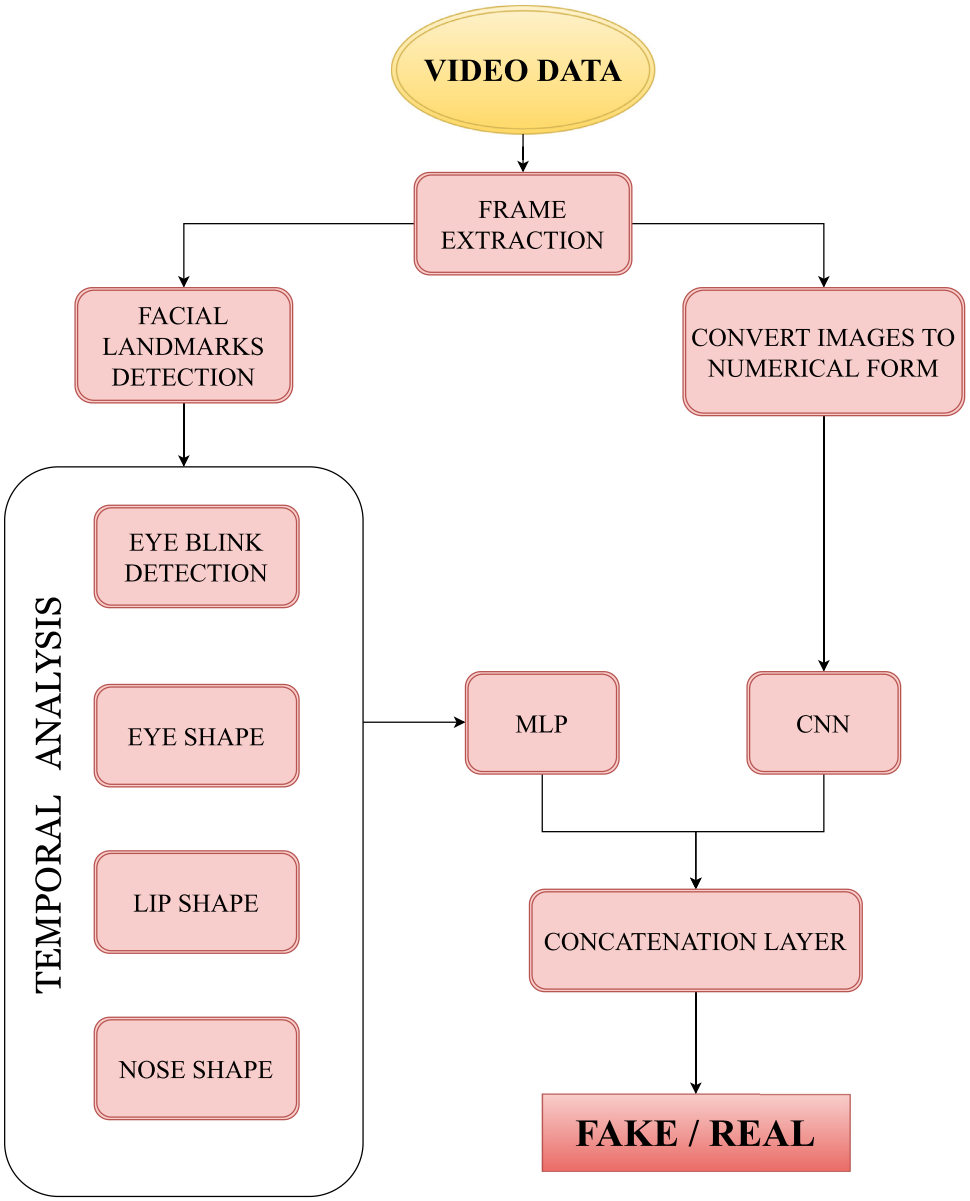

Fig. 1. Steps involved in the proposed system to detect deepfakes.

films where actors in those movies may not be alive (Westerlund, 2019). But the positive applications of deepfakes are not enough to outweigh the negative ones.

The most high-profile technology available to detect deepfakes was unveiled by Microsoft in September 2020, just in time for the U.S Elections. It is called the Video Authenticator (Burt & Horvitz, 2020) tool which uses the FaceForensics++ dataset and the DeepFake Detection Challenge Dataset, the same dataset we have used to train and test our deepfake detector. It uses the boundaries of the deep fake where the blending occurs, and detects faded or greyscale elements undetectable by the naked eye. However, deepfake detection still remains a largely obscure interest as most research being conducted is still evolving in order to keep up with the ephemeral nature of the chase between deepfake detection and generation.

This work proposes a deep learning approach using Multilayer Perceptron (MLP) and Convolutional Neural Networks (CNN) to detect fake videos. Fig. 1 shows the steps involved in the proposed system. Initially, the video is provided as input, from which image frames are extracted. Using a facial landmarks detector, the coordinates of the eyes, nose and lips are extracted. Using this data, the number of eye blinks, the shape of the eyes, nose, and lips, are also extracted. This information is fed to the MLP model. Meanwhile, the pre-processing stage converts the images into their numerical form. This is fed to the CNN model which performs feature extraction and trains on the extracted features. The classification stage, combining the results of the MLP and the CNN models can predict whether the given video is a deepfake or not.

The ease and accessibility of deepfakes have opened a new realm of social engineering attacks for which current cybersecurity systems may not be prepared. Since everything is happening on regular information channels like social media and emails, one does not need to have special hacking skills to deploy cybersecurity attacks based on and around deepfakes. Attackers can create extremely damaging video and audio clips and extort money, data, or both. Deepfake ransomware is among the most feared cyberattack vectors as of recent. While combating deepfake technology is challenging, it is possible to keep data secured. Coupled with cybersecurity measures and detection methods, the solution to stopping the spread of deepfake is well within the power of traditional deep learning and computer vision.

The rest of this paper is organized as follows. The related work is detailed in Section 2. Section 3 details the various stages in the proposed

system. The dataset, parameter tuning, metrics used for evaluation, experiments conducted, and the results obtained are provided in Section 4. Finally, the conclusion is given in Section 5.

2. Related work

While the foundations for face-swapping techniques have been around for a considerable amount of time, it is the advent of AI and its ease of accessibility that ballooned the seriousness of deepfakes. There have been notions to detect deepfakes only fairly recently, and we discuss a few in detail. This section details the origin of deepfake techniques and the existing work in the area of creation and detection of deepfake videos.

2.1. Generating deepfakes

It is surprisingly easy to create deepfakes with good end results as it is possible to create quality deepfakes without much skill. Many applications exist that can be used both by the novice user and a professional. Applications to generate deepfakes are based on deep learning techniques because they are capable of representing complex and high dimensional data required for dimensionality reduction and image compression. Autoencoders and generative adversarial networks (GAN) are two such techniques used widely to create deepfakes. Several works such as FakeApp (FakeApp 2.2.0, 2021), DeepFaceLab (DeepFaceLab, 2021), DFaker (DFaker, 2021) and DeepFake-tf (tensorflow-based deepfakes) (DeepFake-tf, 2021) have used autoencoders.

Though Generative adversarial networks (GAN) (Goodfellow et al., 2014) and variational autoencoders (VAE) (Kingma & Welling, 2013) were used widely for a variety of applications such as generating images (Aggarwal, Mittal & Battineni, 2021), the images produced were blurred and were easily identifiable. Karras et al. (Karras, Aila, Laine & Lehtinen, 2021) proposed ProGAN which generated images of up to 1024×1024 pixels. Flow-based generative models (Dinh, Krueger & Bengio, 2014, 2017; Kingma & Dhariwal, 2018) were also used to generate such images. A general-purpose solution proposed by Isola et al. (Isola, Zhu, Zhou & Efros, 2017) generated images having a relatively low resolution. Wang et al. (T.-C. Wang et al., 2018) improved upon this method by using multi-scale generators and discriminators to generate images with a resolution of up to 2048×1024 pixels. This method was further extended to video-to-video translation problems as well (T.-C. Wang et al., 2018).

2.2. Existing detection methods

The quality of deepfakes has been increasing, putting more pressure on developing detection methods that improve along with it. There are two types of classifiers when it comes to deepfake detection: shallow classifiers and deep classifiers. Shallow classifiers use inconsistency of features to differentiate between fake and real images or videos. For example, the eyes may have missing reflections and other details. The teeth areas may also have similar inconsistencies which are exploited as well. The texture and color around a face, along with other features extracted from the facial region (Matern, Riess & Stamminger, 2021) are also used for deepfake detection.

On the other hand, deep classifiers depend on the fact that affine face warping techniques like scaling, rotating, or shearing are used to create deepfake videos, usually with limited resolutions. Finally, artifacts can also be detected by CNN models such as VGG16 (Simonyan & Zisserman, 2021), ResNet50, ResNet101 and ResNet152 (He, Zhang, Ren & Sun, 2016). Y. Li et al. developed a deep learning method to detect deepfakes based on these models (Li & Lyu, 2019). Darius et al. (Afchar, Nozick, Yamagishi & Echizen, 2021) also use a deep learning approach to detect face tampering in videos.

Yang et al. (Yang, Li & Lyu, 2019) propose a way of finding deepfakes using inconsistent head poses. They use 3D head pose estimation and SVM classifiers for their system. Nguyen et al. (Nguyen, Nguyen, Nguyen, Nguyen & Nahavandi, 2019) discuss how deep learning can be used for the creation and detection of deepfakes. Eye blinking was also used to detect deepfakes (Li, Chang & Lyu, 2021). Deepfake algorithms often use images of faces available online for training, which normally have people with their eyes open. Since it's not really feasible to have algorithms train on images that show people blinking, they usually learn to mimic eye blinking rather poorly, as evidenced by the fact that the deepfakes generally have lower frequency eye blinks.

The following solutions address feature extraction and constraints relating to it, since it is an important step in most methods proposed to detect deepfakes. Lewis et al. describe a deep learning approach that uses a multi-modal network for combining spatial, spectral and temporal inconsistencies (Lewis et al., 2020). They also use discrete cosine transforms to improve deepfake detection. Wodajo and Atnafu (Wodajo & Atnafu, 2021) propose a method that uses a convolutional neural network to extract learnable features which is then provided as input to a Vision Transformer, which sequences them into pixels for detection. In a solution by Burroughs et al. (Burroughs, Gokaraju, Roy & Khoa, 2020), a discrete wavelet transform (DWT) for frame extractions, in conjunction with a convolutional neural network, is used to detect deepfakes. A solution using *MoviePy* to exclude extraneous images from input videos in order to focus only on images with a particular feature, in this case the mouth area, was proposed in Jafar, Ababneh, Al-Zoube and Elhassan, (2020).

Several solutions also employ methods that stray from a deep-learning based approach. Haya and Khaled (Hasan & Salah, 2019) propose a way to combat deepfake videos using blockchain and smart contracts. In Asnani, Yin, Hassner and Liu, (2021), Facebook AI revealed a method to detect deepfakes by reverse engineering generative models from deepfake images. It works by picking up image fingerprints left behind by the generative model, which are unique and are able to identify the generative model. Fernandes at al. explicate an attribute-based confidence matrix that uses attribution over features to decide whether an input video is fake (Fernandes et al., 2020). In Sethi, Dave, Bhagwani and Biwalkar, (2020), a more interdisciplinary method is explored where a semi-fragile, binary watermark encrypted using the AES encryption algorithm is placed around the general region of a face, which is detected using a Haar feature extraction filter. This watermark is then extracted and decrypted when checking for authenticity.

3. Materials and methods

In our solution, we make use of two neural networks, the multilayer perceptron (MLP) and a convolutional neural network (CNN). As shown in Fig. 1, the structured data extracted from the image frames of the videos using facial landmarks detection is given as input to the MLP. The image frames of the videos are directly input to the CNN for automatic feature extraction. The output of the MLP and the CNN are combined together and linked to a fully connected neural layer and an activation layer, which gives the final output. This section details the proposed hybrid system for classifying a video as FAKE or REAL.

3.1. Frame extraction and facial landmarks detection

The input video is first split into image frames. For each image, the face region is identified. From the face region, the locations (x, y coordinates) of 68 facial landmarks are extracted. We use a pre-trained facial landmark detector included in the dlib[3] library. This landmark detector estimates the face's landmark positions from the pixel intensities of the images using an ensemble of regression trees (Kazemi & Sullivan, 2014).

[3] Dessa dataset. Retrieved from https://github.com/dessa-oss/DeepFake-Detection.

3.2. Eye blink detection

Deepfakes require an increased sophistication to not compromise on eye blinking as most amateur deepfakes either have no blinking or rapid unnatural blinking. This step detects the blinking sequence of the person in the frame.

From the facial landmarks, the eye coordinates (points 37 – 46 in the list of extracted facial landmarks) are extracted and given as input to the eye blink detector. The eye blink detector is based on computing the eye aspect ratio (EAR) introduced by Soukupová and Čech (Soukupová & Čech, 2016). The eye is represented by 6 (x, y) coordinates, starting at the left corner (p_1) of the eyes and then plotting points (p_2, p_3, p_4, p_5, p_6) clockwise from p_1. The *EAR* is calculated using these points as given in Equation 1.

$$EAR = \frac{||p_2 - p_6|| + ||p_3 - p_5||}{2||p_1 - p_4||} \quad (1)$$

The value of *EAR* remains constant when the eye is open, but falls to zero when the blink is taking place. If the average of the value of *EAR* of both the eyes is less than a threshold (*EYE_AR_THRESH*) in a fixed number of consecutive frames (*EYE_AR_CONSEC_FRAMES*), the number of blinks is incremented by one. In this work, we have used the values 0.3 and 3 for *EYE_AR_THRESH* and *EYE_AR_CONSEC_FRAMES* respectively.

3.3. Extraction of shape features

Features such as the eye, lip, and nose coordinates are extracted from the facial landmarks detector. The motivation behind the extraction of features from the eyes is to record the widely varying eye shapes introduced by the face-swapping technique that is found in most deepfake videos. We found that the eye shape of a subject in a real video remains mostly consistent. That is not the case when the video was tampered in any of the numerous ways to make it fake. Similarly, most facial inconsistencies happen around the mouth region including facial warping. Different lip shapes are also caused by deepfake manipulation. Thus, we aim to exploit the inconsistencies of the shape of the facial features across frames to train our classifier.

The eye coordinates (points 37–46) extracted from the facial landmarks are given as input to the eye shape detector. The Euclidean distance (d1) between the endpoints of the left eye and the Euclidean distance (d2) between the end points of the right eye are calculated by the eye shape detector. The lip coordinates (points 49–68) extracted from the facial landmarks are input to the lip shape detector. The Euclidean distance between the inner lip coordinates is found to get the length of the inner lips (d3). Similarly, the Euclidean distance between the outer lip coordinates is found to get the length of the outer lips (d4).

From the facial landmarks, the nose data (points 28–36) is extracted and input to the Nose Shape detector. The Euclidean distance (d5) between the edges of the base of the nose is found to get the base width of the nose. Similarly, the Euclidean distance (d6) between the edges of the top of the nose is found to get the top width of the nose.

Thus, the shape features extracted are the width of both the eyes (d1, d2), distance between the outer and inner lip coordinates (d3, d4), and the top and base width of the nose (d5, d6).

3.4. Multilayer perceptron (MLP)

The number of eye blinks and the shape features extracted in the previous steps are first normalized and then given as input to the MLP. Thus, a total of seven features are fed to the MLP. The MLP is a simple, layered neural network with activation functions. In this work, the MLP consists of two layers (Fig. 2):

1 A fully connected (Dense) input layer with ReLu (activation).
2 A fully connected (Hidden) layer, also with ReLu (activation).

ReLu is linear for all positive values and zero for all negative values. That makes it cheap to compute and takes less time to train. It converges quickly as well. We convert the categorical representation of 'FAKE' and 'REAL' into a real-value representation using one hot encoding. The output layer of the MLP is concatenated with the output layer of the CNN module using a concatenation layer of the functional Keras API. The MLP architecture is shown in Fig. 2.

3.5. Convolutional neural network (CNN)

The image frames that are extracted from the video in the first step are transformed into numerical data (NumPy arrays). Each image is further reshaped into a size of 224 × 224 × 3 for simpler processing as well as standardization purposes. This data pertaining to the image frames is fed to the convolutional neural network. A tuple of progressively larger filters (16, 32, and 64) is input as a parameter so that the network can learn more discriminate features in each step. The CNN consists of multiple iterations of CONV => RELU => BN => POOL layers in addition to a densely connected layer. We flatten the next layer and add a fully-connected layer along with appropriate Batch Normalization and Dropout functions. This is done to increase the efficiency of the training process and to enable the model to generalize better. Finally, a fully-connected layer is added to match the nodes coming out of the multilayer perceptron.

3.6. Concatenation layer

The output from the CNN module is concatenated along with the output from the MLP module. Like the MLP, the CNN is also fully connected. Together, the output of the MLP and CNN are merged to a final set of two layers. The first is a fully connected layer with a ReLu activation function, which is followed by another dense layer with a sigmoid activation function. The different layers of the proposed work are shown in Fig. 2.

4. Experimental results

4.1. Dataset description

We use 249 videos from the dataset obtained from the Deepfake Detection Challenge, out of which 199 videos are fake videos and 53 videos are real videos. Each of these videos is 10 s long. To balance the number of real and fake videos, we have also used 66 videos from a YouTube dataset obtained from Dessa containing real videos. Thus, in total we have used 318 videos, out of which 199 are fake and 119 are real.

4.2. Parameter settings

The cv2[4] library and the dlib library are collectively used for frame extraction from videos and facial landmarks detection. From the videos, the frames are extracted at a frame rate of 1 frame per second. There are around 3114 frames including 2189 fake video frames and 925 real video frames. We have split the frames into a 60/20/20 split for training, validation, and testing respectively. The optimizer is set to Adam with a learning rate of *1e-3* and a decay of *1e-3/50*.

4.3. Performance evaluation metrics

The classification performance of the proposed architecture is evaluated over the test set by computing the accuracy and analyzing the ROC curve. The positive class is assigned to be 'FAKE' while the negative class is assigned to be 'REAL'.

True positive (TP) refers to the total number of fake video frames that are correctly labeled as fake by the classifier. True negative (TN)

[4] Dlib C++ Library. Retrieved from http://dlib.net/

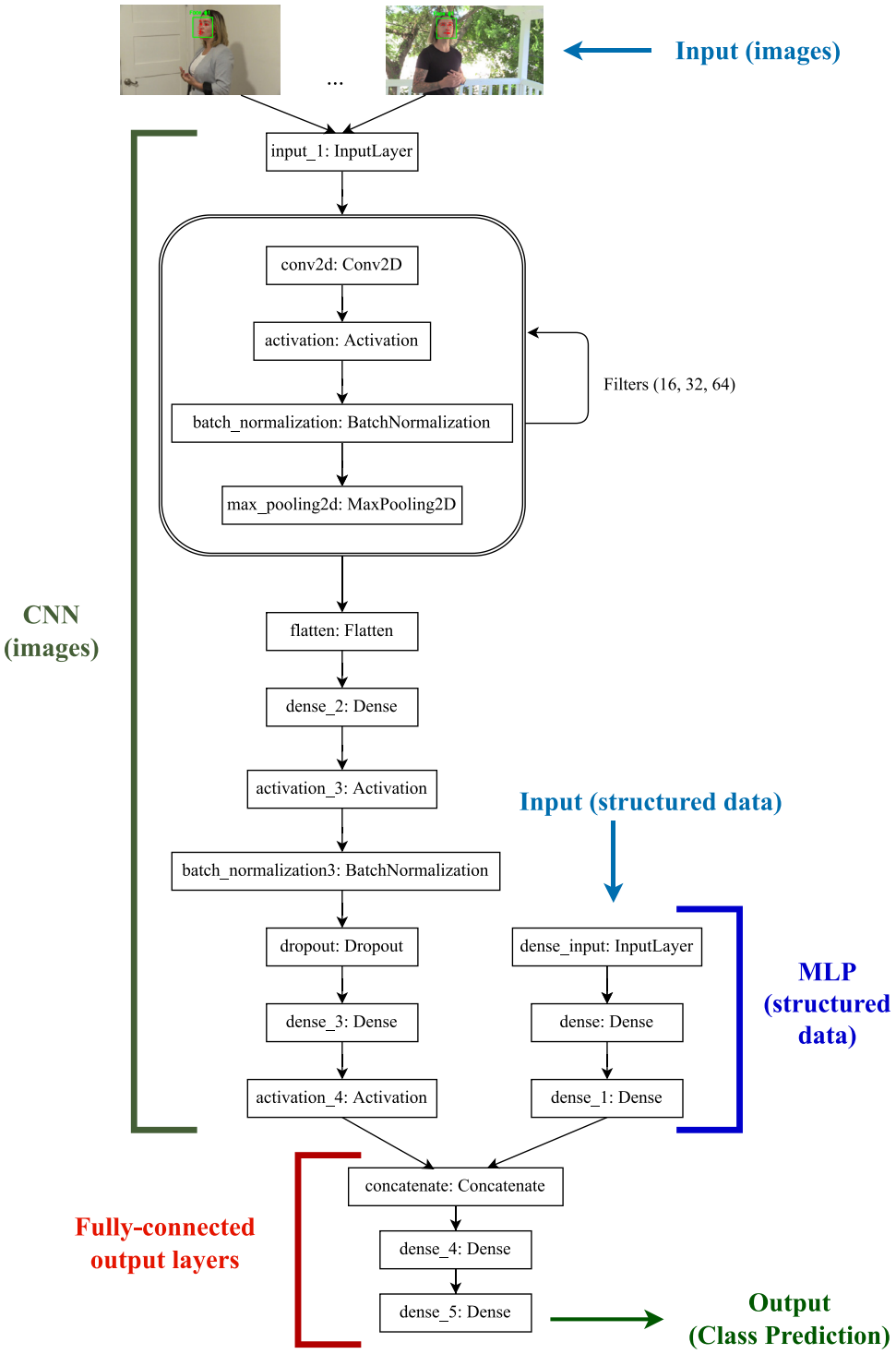

Fig. 2. Proposed architecture diagram showing the concatenation of the CNN and the MLP models.

refers to the total number of real video frames that are correctly labeled as real by the classifier. False positive (FP) refers to the total number of real video frames that are incorrectly labeled as fake by the classifier. False negative (FN) refers to the total number of fake video frames that are incorrectly labeled as real by the classifier.

Using these values, the accuracy (the fraction of correct predictions) is calculated as:

$$Accuracy = \frac{TruePositive + TrueNegative}{TruePositive + TrueNegative + FalsePositive + FalseNegative}$$

The receiver operating characteristic (ROC) curve is a plot of the false positive rate (x-axis) versus the true positive rate (y-axis). It is useful in our case because it is important to consider at what rates the model is recognizing fake videos as real and real videos as fake since both contribute unfavorably to the model's credibility.

4.4. Performance evaluation

The model was trained for 40 epochs with an Adam learning rate scheduler. It was observed that the model began to overfit if it was run for more epochs. The classifier is tested with randomly downloaded real

Table 1
Performance Evaluation of the two models.

Model	Training accuracy (%)	Validation accuracy (%)	Testing accuracy (%)	AUC Score
MLP-CNN	0.84	0.83	0.87	0.877
CNN-only	0.84	0.72	0.74	0.669

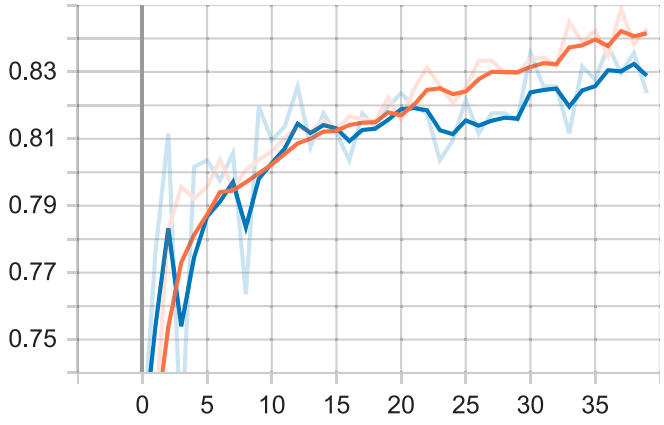

Fig. 3. Model Accuracy (MLP-CNN).

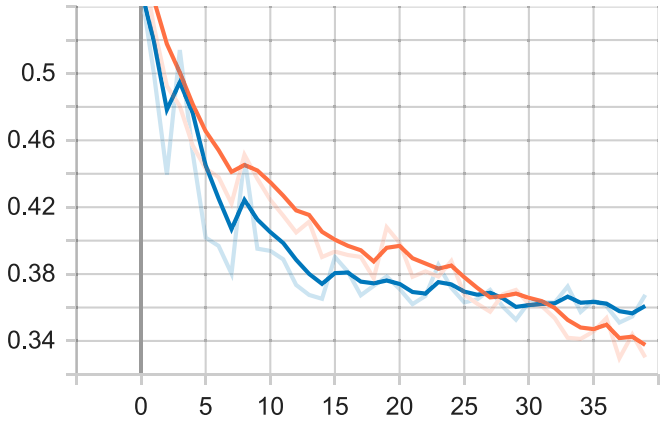

Fig. 4. Model Loss (MLP-CNN).

and deepfake videos from YouTube. The classifier predicts correctly in most cases although in certain cases it fails. This is discussed further in the Limitations section. Fig. 3 shows the model accuracy and Fig. 4 shows the model loss, where the orange line indicates training accuracy and loss, and the blue line indicates validation accuracy and loss, respectively.

The ROC curve is shown in Fig. 5. From the ROC, another important metric known as the Area Under the Curve (AUC) is determined. Since the dataset used is imbalanced, AUC is a good metric to use. This model provides an AUC score of 0.87 which means that the model has an 87% chance of flagging a fake video correctly.

For comparison purposes, we built another deepfake model that only has the CNN architecture and does not consider the facial features data that was input into the MLP. It was found that the absence of this information considerably worsened the performance of the model. Fig. 6 shows the model accuracy and Fig. 7 shows the model loss for the CNN-only model with the orange and blue lines indicating training and validation respectively.

The ROC Curve for this CNN-only model is shown below, in Fig. 8. With an AUC score of 0.669, we can quantify that the model exhibits poor performance in the absence of the extracted facial features that we input to the MLP network.

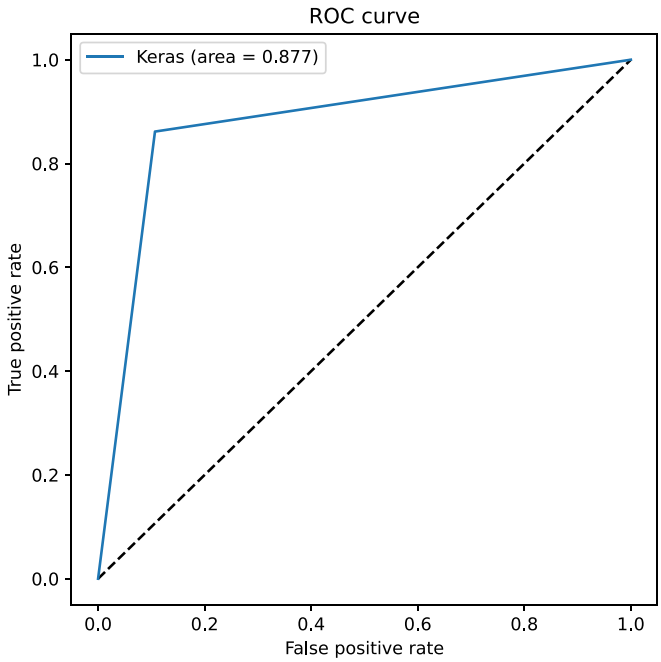

Fig. 5. Receiver Operating Characteristic (ROC) Curve (MLP-CNN).

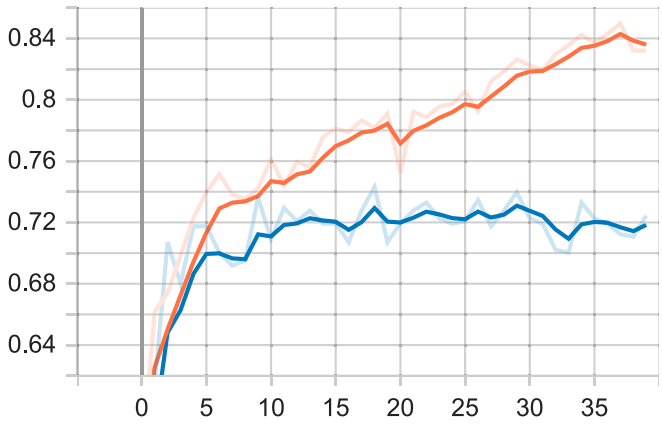

Fig. 6. Model Accuracy (CNN-only).

In Table 1, we present the accuracies of both the models along with their AUC scores.

An important point to note is that in the case of the CNN-only architecture, the model overfits much earlier and provides poor performance. This is the reason why the model exhibits a training accuracy of 84% but the accuracy drops to 74% during testing. The AUC score is thus a much better indicator of the model's performance.

We would also like to mention the time taken to train our model. For a training sample size of 1992 image frames (which is equal to 199 videos of approximately 10 second duration each), our hybrid model takes 280 s (4.6 min) to train on a Tesla K80 GPU provided by Google Colab. Thus, our model provides high performance at a relatively faster training rate despite its limited training sample.

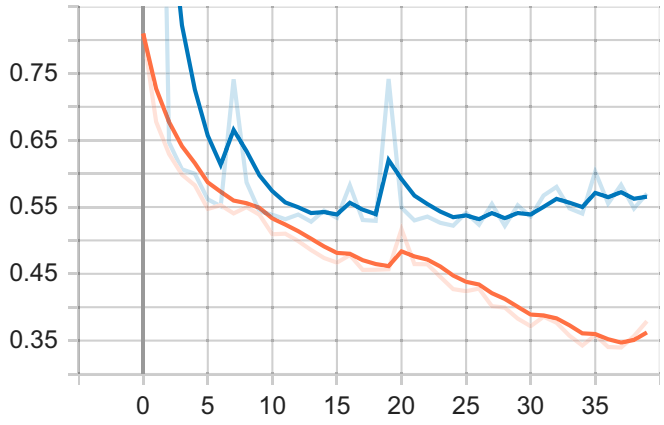

Fig. 7. Model Loss (CNN-only).

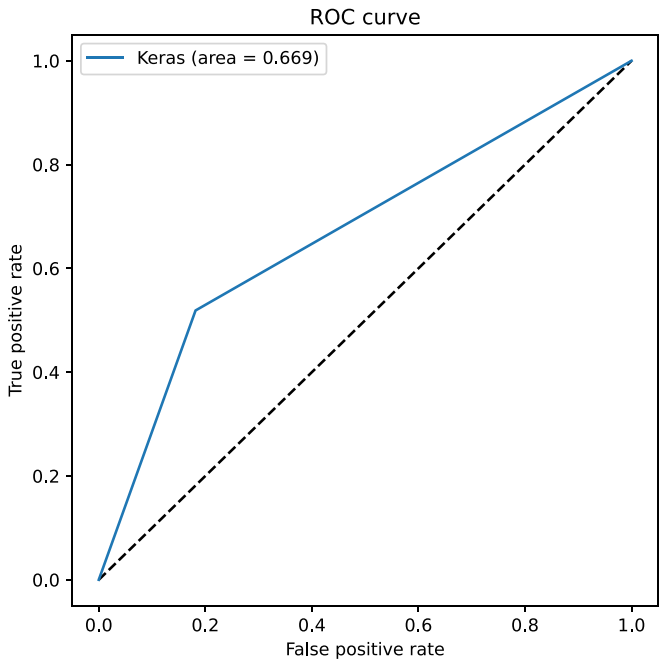

Fig. 8. Receiver Operating Characteristic (ROC) Curve (CNN-only).

5. Discussion

The generation and proliferation of deepfake material will only get more competitive and by extension, more harmful in the future. In a world that is fast to adapt to technological advancements, but is slow to change socially and structurally, the effects of deepfake have the potential to be catastrophic. Even while we develop solutions to combat the rise and spread of deepfake, equally interested and competent parties are developing ways to generate more and more realistic, and sometimes even entirely convincing images of "fake humans" as generated by the StyleGAN (Karras et al., 2020; Karras, Laine & Aila, 2019). In this paper, we propose one such method to expose AI-generated deepfake videos, in hopes that it will aid other security verification processes to bar deepfake material from causing irreparable damage.

5.1. Theoretical implications

This paper introduces a unique method that utilizes facial data directly instead of feature extraction. This could be considered in future models as we have achieved good accuracy for a really small sample in a short time. There are also various other directions that this paper can be taken forward in order to improve the accuracy of our model. First, we found that our model flounders when presented with faces that are dark. We hope this will encourage exploration of other computer vision techniques on improving detection of dark faces and faces lost in dark environments, as well as contribute to the overall public discourse about the biases and ethical issues involved in employing intelligent machines (Akter et al., 2021; Coombs et al., 2021). Second, there are more obscure ways to detect deepfakes such as color segmentation, to improve face warping and noise-pattern disturbances in the video itself. It will be interesting and helpful to combine those techniques with the current model to build a more comprehensive deepfake detector.

5.2. Implications for practice

Deepfakes will only exacerbate current social and political issues and can even disrupt the very foundations of society. Therefore, there is a pressing need to combat the rise of deepfakes, especially ones that will be used in a malicious manner.

As governments and businesses move toward adopting facial recognition technology as a means of identification, deepfake detection models like ours can be used to as an additional layer of verification in order to prevent fraud and identity theft. This research will also be invaluable to law enforcement bodies as it can be used to verify the authenticity of visual evidences. It should be noted that the problem of deepfakes is unique in the sense that both false positives and false negatives are detrimental. Flagging a fake video as real is what we have considered so far, but in instances of informational videos or videos of journalistic quality, flagging a real video as fake is also equally damaging. So, due to the myriad of ways in which deepfakes can be used to defame, trick or scam people, a deepfake detection tool will be immensely helpful in combating these different instances. An example of such an instance could be workplace oppression that may rise from the use of deepfakes as detailed in Young, Majchrzak and Kane, (2021). The presence of a deepfake detection tool in a workplace setting may help hasten the process of verifying evidence, thereby contributing to quicker resolution times for employee disputes.

Therefore, our research is crucial as it provides stakeholders a means to take proactive measures against deepfakes.

5.3. Limitations

The proposed system suffers from a few limitations. As this project was undertaken with limited access to computing resources, we could not take advantage of the entirety of the DFDC dataset. Although the goal of this paper was to provide a new method and not to beat existing performance standards for deepfake detection systems, future work could scale the proposed system for a larger dataset for completeness. The proposed system also performs poorly at detecting faces in low-light conditions or dark environments. Further, the system did not take into account videos that had multiple people in frame although this could be easily implemented in the future.

6. Conclusions and future work

In this work, we propose a novel method to expose AI-generated deepfake videos, combining structured and unstructured data. Our method is based on observations that such deepfakes are created by splicing a synthesized face region into the original image, and in doing so, introducing errors like improper eye, lip, and nose locations that are not normally found in real videos. While existing methods focus on using deep neural networks to extract facial features from video frames directly, they do not investigate the discrepancies across frames directly. In this paper, we propose a way to combine the knowledge of these inconsistencies (input to an MLP) along with the powerful feature extraction of a CNN. The proposed method exhibits an accuracy of 84% and an AUC score of 0.87 despite training on a small subset of data. Although it

isn't a comprehensive detection tool, we believe that the proposed hybrid system offers a great baseline for screening deepfake videos with limited computational resources and at a relatively faster speed.

The future scope of this work can be to find ways to expand the range of people that the model can detect accurately, such as people of color, in order to ensure fairness and reduced bias. It is also worth considering inculcating more facial data that are spatial and temporal in an acceptable blend. Further, it is necessary to test improved models on wider, more balanced datasets.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Afchar, D., Nozick, V., Yamagishi, J., & Echizen, I. (2021). MesoNet: a compact facial video forgery detection network. *2018 IEEE International Workshop on Information Forensics and Security* doi: 10.1109/WIFS.2018.8630761.

Aggarwal, A., Mittal, M., & Battineni, G. (2021). Generative adversarial network: An overview of theory and applications. *International Journal of Information Management Data Insights, 1*(1), Article 100004. 10.1016/j.jjimei.2020.100004.

Akter, S., McCarthy, G., Sajib, S., Michael, K., Dwivedi, Y. K., D'Ambra, J., et al. (2021). Algorithmic bias in data-driven innovation in the age of AI. *International Journal of Information Management, 60*, Article 102387. 10.1016/j.ij3infomgt.2021.102387.

Asnani, V., Yin, X., Hassner, T., & Liu, X. (2021).Reverse engineering of generative models: inferring model hyperparameters from generated images, ArXiv:2106.07873 [Cs],http://arxiv.org/abs/2106.07873 (Accessed August 23, 2021).

Burns, K. (2020).A deceptively edited video of Joe Biden illustrates a big problem in 2020, *Vox,* 2 January, Available at: https://www.vox.com/policy-and-politics/2020/1/2/21046605/joe-biden-viral-video-deceptive-edit (Accessed: August 16, 2021).

Burroughs, S. J., Gokaraju, B., Roy, K., & Khoa, L. (2020). DeepFakes detection in videos using feature engineering techniques in deep learning convolution neural network frameworks. In *2020 IEEE Applied Imagery Pattern Recognition Workshop (AIPR)* (pp. 1–4). IEEE. 10.1109/AIPR50011.2020.9425347.

Burt, T., & Horvitz, E. (2020). New steps to combat disinformation. *Microsoft On The Issues*. 1 SeptemberAvailable at https://blogs.microsoft.com/on-the-issues/2020/09/01/disinformation-deepfakes-newsguard-video-authenticator . Accessed on: August 16, 2021.

Coombs, C., Stacey, P., Kawalek, P., Simeonova, B., Becker, J., Bergener, K., et al. (2021). What is it about humanity that we can't give away to intelligent machines? A European perspective. *International Journal of Information Management, 58*, Article 102311. 10.1016/j.ijinfomgt.2021.102311.

DeepFaceLab. (2021). Retrieved from https://github.com/iperov/DeepFaceLab.

DeepFake-tf: Deepfake based on tensorflow. (2021). Retrieved from https://github.com/StromWine/DeepFaketf.

DFaker. (2021). Retrieved from https://github.com/dfaker/df.

Dinh, L., Krueger, D., & Bengio, Y. (2014). NICE: Nonlinear independent components estimation. *2015 International Conference on Learning Representations* arXiv preprint arXiv:1410.8516v6.

Dinh, L., Sohl-Dickstein, J., & Bengio, S. (2017). Density estimation using Real NVP. *2017 International Conference on Learning Representations* arXiv preprint arXiv:1605.08803.

FakeApp 2.2.0. (2021). Retrieved from https://www.malavida.com/en/soft/fakeapp/.

Fernandes, S., Raj, S., Ewetz, R., Pannu, J. S., Kumar Jha, S., Ortiz, E., & Salter, M. (2020). Detecting deepfake videos using attribution-based confidence metric. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)* (pp. 1250–1259). IEEE. 10.1109/CVPRW50498.2020.00162.

Frum, D. (2020).The Very Real Threat of Trump's Deep fake, *The Atlantic*, 27 April,Available at: https://www.theatlantic.com/ideas/archive/2020/04/trumps-first-deepfake/610750/ Accessed: August 16, 2021).

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., & Bengio, Y. (2014). Generative adversarial nets. In *Proceedings of the 24th International Conference on Neural Information Processing Systems: 2* (pp. 2672–2680).

Hasan, H. R., & Salah, K. (2019). Combating Deepfake Videos Using Blockchain and Smart Contracts. *IEEE access : practical innovations, open solutions, 7*, 41596–41606. 10.1109/ACCESS.2019.2905689.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition* (pp. 770–778).

Isola, P., Zhu, J.-Y., Zhou, T., & Efros, A. A. (2017). Image-to-Image Translation with Conditional Adversarial Networks. In *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition* (pp. 5967–5976). 10.1109/CVPR.2017.632.

Jafar, M. T., Ababneh, M., Al-Zoube, M., & Elhassan, A. (2020). Forensics and Analysis of Deepfake Videos. In *11th International Conference on Information and Communication Systems (ICICS)* (pp. 053–058). IEEE. 10.1109/ICICS49469.2020.239493.

Karras, T., Aila, T., Laine, S., & Lehtinen, J. (2021). Progressive Growing of GANs for Improved Quality, Stability, and Variation. *2018 International Conference on Learning Representations*.

Karras, T., Laine, S., & Aila, T. (2019). A Style-Based Generator Architecture for Generative Adversarial Networks. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 4396–4405). 10.1109/CVPR.2019.00453.

Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., & Aila, T. (2020). Analyzing and Improving the Image Quality of StyleGAN. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 8107–8116). 10.1109/CVPR42600.2020.00813.

Kazemi, V., & Sullivan, J. (2014). One millisecond face alignment with an ensemble of regression trees. In *2014 IEEE Conference on Computer Vision and Pattern Recognition* (pp. 1867–1874). 10.1109/CVPR.2014.241.

Kingma, D.P., & Dhariwal, P. (2018).Glow: Generative flow with invertible 1x1 convolutions. arXiv preprint arXiv:1807.03039v2.

Kingma, D. P., & Welling, M. (2013). Auto-encoding variational bayes. *2014 International Conference on Learning Representations* arXiv preprint arXiv:1312.6114 .

Lewis, J. K., Toubal, I. E., Chen, H., Sandesera, V., Lomnitz, M., Hampel-Arias, Z., & Palaniappan, K. (2020). Deepfake Video Detection Based on Spatial, Spectral and Temporal Inconsistencies Using Multimodal Deep Learning. In *2020 IEEE Applied Imagery Pattern Recognition Workshop (AIPR)* (pp. 1–9). IEEE. 10.1109/AIPR50011.2020.9425167.

Li, Y., Chang, M., & Lyu, S. (2021). In Ictu Oculi: Exposing AI Created Fake Videos by Detecting Eye Blinking. *2018 IEEE International Workshop on Information Forensics and Security (WIFS)*. 10.1109/WIFS.2018.8630787.

Li, Y., & Lyu, S. (2019). Exposing deepfake videos by detecting face warping artifacts. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops* (pp. 46–52).

Matern, F., Riess, C., & Stamminger, M. (2021). Exploiting Visual Artifacts to Expose Deepfakes and Face Manipulations. *2019 IEEE Winter Applications of Vision Workshops (WACVW)*. 10.1109/WACVW.2019.00020.

Nguye, T.n., .T.., Nguyen, C.M., Nguyen, D.T., Nguyen, D.T., & Nahavandi, S. (2019).Deep learning for deepfakes creation and detection: A survey, arXiv:1909.11573.

Sethi, L., Dave, A., Bhagwani, R., & Biwalkar, A. (2020). Video security against deepfakes and other forgeries. *Journal of Discrete Mathematical Sciences and Cryptography, 23*, 349–363. 10.1080/09720529.2020.1721866.

Simonyan, K., & Zisserman, A. (2021).Very deep convolutional networks for large-scale image recognition.arXiv preprint arXiv:1409.1556.

Soukupová, T., & Čech, J. (2016).Real-Time Eye Blink Detection using Facial Landmarks. 21st Computer Vision Winter Workshop.

Wang, T.-. C., Liu, M.-Y., Zhu, J.-Y., Liu, G., Tao, A., Kautz, J., et al. (2018a). Video–to-Video Synthesis. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems* (pp. 1152–1164).

Wang, T.-. C., Liu, M.-Y., Zhu, J.-Y., Tao, A., Kautz, J., & Catanzaro, B. (2018b). High Resolution Image Synthesis and Semantic Manipulation with Conditional GANs. In *Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 8798–8807). 10.1109/CVPR.2018.00917.

Westerlund, M. (2019). The emergence of deepfake technology: A review. *Technology Innovation Management Review, 9*, 39–52. 10.22215/timreview/1282.

Wodajo, D., & Atnafu, S. (2021).Deepfake video detection using convolutional vision transformer. arXiv:2102.11126 [Cs],http://arxiv.org/abs/2102.11126 (accessed August 23, 2021).

Yang, X., Li, Y., & Lyu, S. (2019). Exposing deep fakes using inconsistent head poses. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 8261–8265). 10.1109/ICASSP.2019.8683164.

Young, A. G., Majchrzak, A., & Kane, G. C. (2021). Organizing workers and machine learning tools for a less oppressive workplace. *International Journal of Information Management, 59*, Article 102353. 10.1016/j.ijinfomgt.2021.102353.