Deepfake Manipulation Detection

NOVOJIT SAHA, University of Twente, The Netherlands

The advent of generative AI have led to the rise of synthetic media, also referred to as deepfakes. Just like any new technology, deepfakes can be a boon and a bane. Hence it is imperative to identify deepfakes that may be used for malicious purposes. This paper presents a lightweight video deepfake manipulation detection method based on the temporal differences of facial mesoscopic properties between frames. Mesoscopic properties refer to the characteristics of images that lie between the microscopic and macroscopic scales, such as textures and edges. Most contemporary deepfake detection methods perform excellently at detecting deepfakes, but they often require high computational power. This makes them unsuitable for real time applications or deployment on resource constraint devices. Taking these observations into account, I propose a lightweight deepfake manipulation detection framework that utilizes the combination of a lightweight CNN network and an LSTM network to take both spatial and temporal dimensions into account. Through experiments on open source datasets, I show that this framework is effective in identifying deepfakes to a certain extent at a low computational cost.

Additional Key Words and Phrases: Video deepfake detection, Facial manipulation detection, Machine learning for video verification, Synthetic media, Computer vision, Video forensics, MesoInception-4 network, LSTM network, Time-aware neural networks

1 INTRODUCTION

Deepfakes are synthetic media: images, audio, video, and videos with audio, produced by Generative Adversarial Networks (GANs). GANs are a combination of two different neural networks, a generator network and a discriminator network, that contest against each other over multiple iterations. The generator tries to create data that mimics real data, while the discriminator tries to classify between real and synthetic data. The two networks share feedback with each other, and thus improve their capabilities over multiple iterations. Eventually, the generator becomes good enough to produce data that closely resembles real data. These GAN models are improving dramatically every year. The latest state-of-the-art models are so sophisticated that they can produce real time deepfakes that can easily fool human eyes. Even the people who are aware of deepfakes are biased towards thinking they can detect deepfakes, but that is not true [1].

While GANs can be a useful tool in many use cases, malicious actors can also use this technology to create deepfakes that harm individuals and societies in many different ways. Malicious actors often use deepfakes to impersonate a specific person and make them say or do things they never did. Moreover, deepfakes can also be used to create new persona that never existed. A multitude of cyber crimes have already been committed in the past few years using deepfakes. In December 2023, the Hong Kong office of a multinational company lost $25.6 million to a deepfake video conference call impersonating its chief financial officer [2]. At the time of writing this paper, as the Russo-Ukrainian War progresses, deepfakes of presidents of both the nations have been used over the internet multiple times to distribute propaganda. A more sinister use of deepfakes is creating sexually explicit content of specific individuals or even of children. Fraud, deception, identity theft, illicit content generation are some of the many crimes that can be committed with deepfakes.

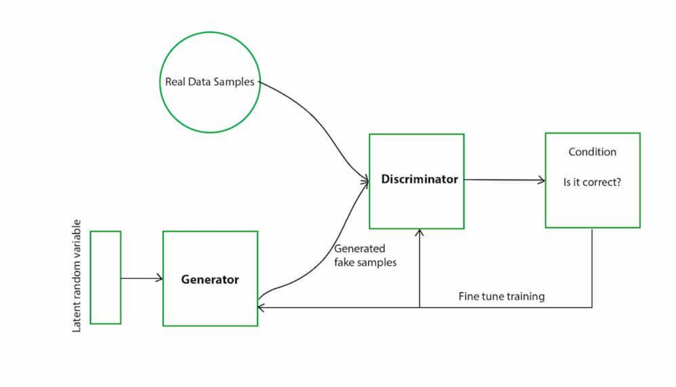

Fig. 1. Generative Adversarial Network (GAN) [8]

Deepfakes can be classified into two main categories:

- Face Swap
- Face Reenactment

Face Swap refers to the technique that replaces the original face with a donor face. Face Reenactment refers to the technique that alters the original face to follow the expressions and movements of another person's face. Most deepfakes use a combination of both of these techniques. Therefore it is important for the proposed method to generalize well on videos manipulated by either one or both of the techniques combined.

TScIT 41, July 5, 2024, Enschede, The Netherlands

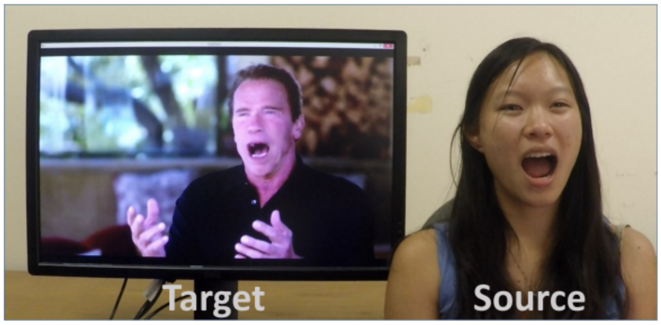

Fig. 3. Example of face reenactment (Thies et al.) [19]

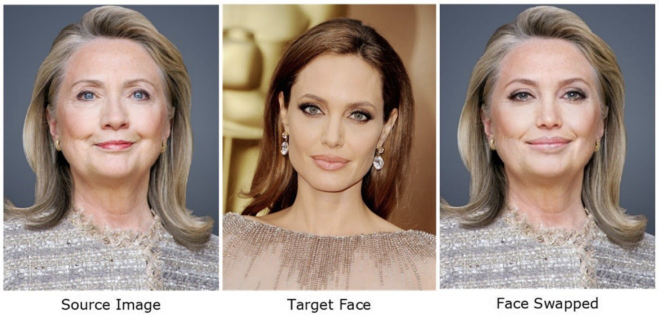

Fig. 2. Example of face swap (Masood et al.). [20]

It has been shown that temporal dependent features perform superior in terms of both detection accuracy and generalizability in contrast to temporal independent features [9, 10]. There are many different temporal dependent features that can be analyzed for deepfake detection. These features can vary from biological signals, such as eye blinking rate analysis [11], to CNN-RNN based spatio-temporal features [10]. The biological signal based deepfake detection methods are not reliable and can be fooled by modern deepfakes. On the other hand, most contemporary deepfake detectors use large CNN models such as ResNet or XceptionNet neural networks for binary classification.

While these modern frameworks have state-of-the-art performance results, they are not suitable for real time detection or deployment on resource constrained computers, given their large computational demands. Taking these weaknesses into account, I propose a lightweight deepfake detection framework that uses a combination of the compact MesoInception-4 network and a 2 layer LSTM network [7]. This framework is significantly more compact than most contemporary deepfake detection frameworks. Compared to the other models, where the number of trainable parameters are in the tens of millions, the proposed model contains only about 215,000 parameters.

The findings of this research will add value to a multitude of domains, namely, real time video applications, IoT industry, law enforcement, social media, and media and journalism.

2 RELATED WORKS

Related works on deepfake video detection methods can be categorized into three distinct categories:

- Spatial features based methods (inconsistencies in individual frames)
- Spatio-temporal features based methods (inconsistencies between adjacent frames)
- Multi-modal features based methods (inconsistencies between different modalities)

Some studies have used biological clues such as eye blink rate analysis [10], photoplethysmography (PPG) signal analysis [15], and multimodal lip sync analysis [16]. But these studies come with limitations. Eye blink rate analysis does not output state-of-the-art performance and is also not futureproof. PPG signal analysis performs well on most deepfake algorithms but fail to generalize over deepfakes generated by the NeuralTextures network, which is good in resembling real life PPG signals [15]. Lip sync based methods perform satisfactorily, but it requires both the modalities to be present and different neural networks in the architecture to process the different modalities, which makes them computationally expensive detection frameworks.

Another class of deepfake detection studies have taken the data driven approach. They use pixel level features with traditional classifiers such as support vector machines [17] or linear regressions. Feature representations include but not limited to local binary patterns, histogram of oriented gradient (HOG), Gaussian scale space analysis (SURF), etc. These types of analysis have modest outcomes and are also likely to suffer from generalizability over different datasets.

Finally, all the state-of-the-art systems have used deep learning based approaches. Their architectures usually consist of Convolutional Neural Networks (CNNs) or Hybrid CNN and Recurrent Neural Networks (RNNs) or Hybrid CNN and Long Short Term Memory Networks (LSTMs). A very recent paper has used a Graph Neural Network (GNN) [17] as well. While these architectures offer superior performance and very high generalization ability over cross-dataset testing, they often demand a lot of computational power, making them unsuitable for real time applications and/or deployment in resource constrained devices.

3 RESEARCH QUESTION

How effective is a lightweight neural network architecture in detecting deepfake manipulations across various deepfake generation techniques and datasets, and how does it compare to larger models in terms of accuracy and efficiency?

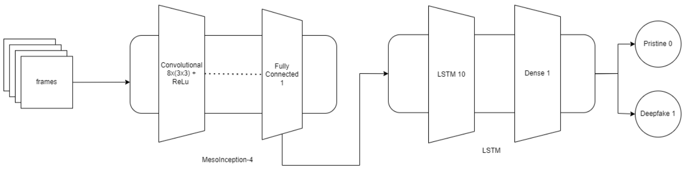

Fig. 4. Proposed Model Architecture

4 PROPOSED METHOD

The primary objective is to create a lightweight neural network architecture that can effectively detect deepfakes without demanding much computational power, during both the training and inference phase. In contrast to the contemporary deep learning based architectures, the most striking feature in my proposed method is the very low number of trainable parameters, around 215,000. The proposed hybrid architecture contains two independent neural networks: the MesoInception-4 network and a two layer LSTM network.

The MesoInception-4 network is used for spatial feature extraction. It captures the mesoscopic details of the faces present in the video. The MesoInception-4 Network is shown to have a very high average detection rate given its lightweight architecture: 98% for *Deepfake* manipulated frames and 95% for *Face2Face* manipulated frames. This suggests that the features extracted by this neural architecture are very effective in deepfake manipulation classification in videos as well. The MesoInception-4 network is created from the Meso-4 network by replacing its Convolutional Layers with the Inception Modules. Features are extracted from the last Dropout layer before the Dense classifier layer. These penultimate layer features contain the most amount of useful information for deepfake detection. After removing the final dense classification layer, this model has a total 28,708 parameters.

The LSTM (long short term memory) network is specialized for capturing temporal patterns. The spatial features collected by the MesoInception-4 network are packaged in a sequence and fed into the final LSTM network to capture the temporal inconsistencies between frames of a video. This LSTM network is also lightweight with 50 hidden layers and 215,051 total parameters.

Layer (type)	Output Shape	Param #
lstm_1 (LSTM)	(None, 50)	215,000
dense_1 (Dense)	(None, 1)	51

Table 1. Architecture of the LSTM Network

5 EXPERIMENTS

5.1 Datasets

Two different publicly available datasets have been used for training and testing:

- FaceForensics++
- Celeb-DF (V2)

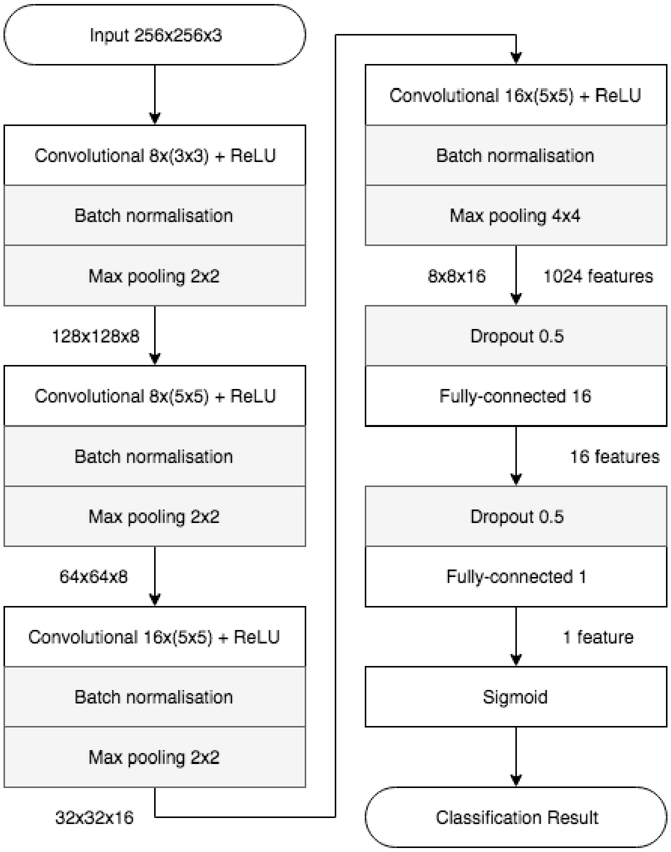

Fig. 5. Architecture of Meso-4 Network ("MesoNet") [7]

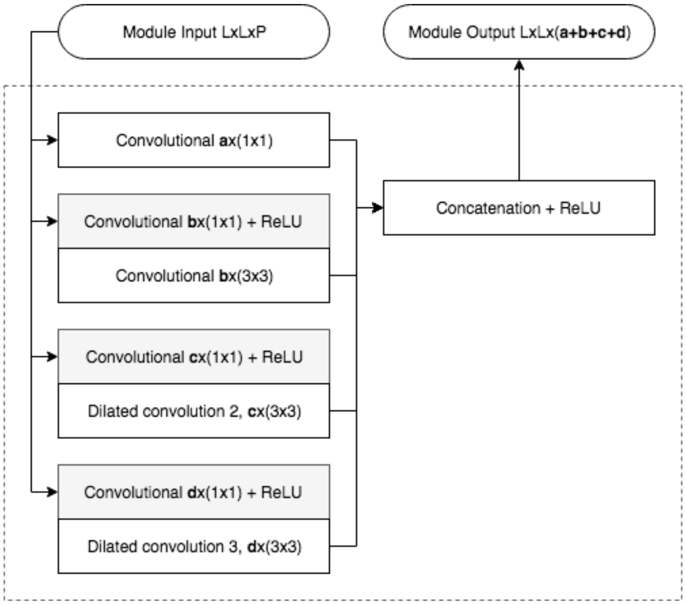
Module Input LxLxP
Module Output LxLx(a+b+c+d)
Convolutional ax(1x1)
Convolutional bx(1x1) + ReLU
Convolutional bx(3x3)
Convolutional cx(1x1) + ReLU
Dilated convolution 2, cx(3x3)
Convolutional dx(1x1) + ReLU
Dilated convolution 3, dx(3x3)
Concatenation + ReLU

Fig. 6. Architecture of the Inception Modules inside the MesoInception-4 Network ("MesoNet") [7]

Type	Training set	Validation set	Test set
Real	192 vids 15 × 192 imgs	48 vids 15 × 48 imgs	60 vids 15 × 60 imgs
Deepfakes	192 vids 15 × 192 imgs	48 vids 15 × 48 imgs	60 vids 15 × 60 imgs
FaceShifter	192 vids 15 × 192 imgs	48 vids 15 × 48 imgs	60 vids 15 × 60 imgs
Face2Face	192 vids 15 × 192 imgs	48 vids 15 × 48 imgs	60 vids 15 × 60 imgs
FaceSwap	192 vids 15 × 192 imgs	48 vids 15 × 48 imgs	60 vids 15 × 60 imgs
Celeb-DF (V2)	192 vids 15 × 192 imgs	48 vids 15 × 48 imgs	60 vids 15 × 60 imgs

Table 2. Dataset Summary

The FaceForensics++ dataset comes with synthetic media generated from four different types of deepfake generation systems:

- Deepfakes
- Face2Face
- FaceShifter
- FaceSwap

All the data in this dataset is collected from YouTube. The dataset consists of videos of newscasters at 854 X 480 pixels (480p). Most videos are between 10 to 30 seconds long. Deepfakes method uses two autoencoders to swap the original face with a target face. Face2Face uses face reenactment techniques to alter the emotions of the target face to that of a source face. FaceShifter is another face swap technique that uses an encoder and a generator [23]. FaceSwap uses a graphics based approach to transfer the face region from a source video to a target video.

The Celeb-DF dataset is comprised of celebrity interview videos of 59 different celebrities collected from Youtube at 2560 X 1440 pixels (1440p). Then deepfakes are generated by swapping faces for each pair of the 59 subjects. As a result, this dataset contains 590 real videos and 5,639 deepfake videos. Each videos are approximately 13 seconds.

Due to memory constraints on the computer used for experimentation, only 300 randomly selected real videos and 300 randomly selected synthetic videos from each dataset are used for each experiment. All the videos are compressed using H.264 encoding. Python 3.12 along with Keras and MediaPipe libraries have been used for the programming phase. All the computation is done on a 12th Gen Mobile Intel i-5 processor.

5.2 Metrics

For metrics, the following formulae have been used to evaluate the performance of the proposed classifier.

Accuracy measures the overall correctness of the model.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{1}$$

Precision measures the accuracy of the positive predictions.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{2}$$

Recall measures the proportion of true positive predictions out of all positive predictions.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{3}$$

5.3 Preprocessing

300 synthetic videos and 300 original videos are taken during the experiments from each dataset. Each video is converted to frames at 15 frames per second. Then the first 15 frames are loaded in the data loader.

BlazeFace face detection model from Google's MediaPipe AI library is used to detect faces in the frames. Then the frames are cropped according to the facial bounding box dimensions provided by the BlazeFace model. This is done to not take backgrounds in the training and testing process as they are noise, since the facial deepfake algorithms only affect textures around the facial region.

Each of the frames are downsampled to (256 x 256 x 3) to match the input dimensions of the MesoInception-4 Network. The input dimensions of the XceptionNet and MobileNet are also set to this shape.

5.4 Feature extraction

For the MesoInception-4 network, trained weights were already provided by the authors of MesoNet. They trained the network on both the FaceForensics++ dataset and the Celeb-DF dataset separately and provided two different weight files. But only the weights acquired from training on FF++ dataset were used, since they performed better than the alternative.

Therefore, the weights of this network are frozen and the input data is fed through the network to collect features. The output of the penultimate layer is collected as features to be fed into the LSTM network.

The MesoNet considers each frame as one unit of data, while the LSTM considers multiple frames from the same video to be one unit of data, also referred to as the sequence length. Therefore after the features are extracted from the MesoInception-4 network, the data is reshaped to match the input dimensions of the LSTM network.

Finally, a train test split is created for the LSTM model training. 80% of the data is used for training and validating while the rest of the 20% is used for testing. For the XceptionNet and MobileNet architectures, the data is split into train-test split right after it's loaded.

5.5 Hyperparameter details

To keep the comparisons between the different architectures fair, all common hyperparameters were kept the same.
MesoInception-4 Network: Since pre-trained weights are provided by

the original authors of the MesoInception-4 Network, no training is done on this network in this paper.

LSTM Network: The LSTM Network is set to have 50 hidden layers and a Dense sigmoid activation layer. This network was tested with variable hidden layers (25 to 100) to see if there's any improvement in performance. However, there was no improvement in performance by increasing the number of hidden layers but degradation in performance by using less than 50 hidden layers. Therefore 50 hidden layers is considered in this research.

In the training phase, multiple experiments were carried out by varying the number of epochs and batch sizes. There were no changes in the training accuracy, so a standard of 20 epochs with 32 batch size, and a default learning rate of 0.001 was decided for all three of the networks: LSTM, MobileNet, and XceptionNet to keep the comparisons fair.

6 RESULTS

For the experiments, the metrics mentioned before have been used to evaluate performance of the proposed model and the comparison models. Table 3 summarizes all the metrics of the models on each datasets.

Fig. 8 and Fig. 9 shows a positive correlation between accuracy vs inference time and accuracy vs training duration correspondingly. Higher accuracy comes with a cost in both training and inference times.

From Table 3, we can calculate the mean inference time of the XceptionNet Model to be 154.2 ms. Fig. 7 suggests that this model can classify somewhere around 10 frames per second on the computer used for experimentation. Needless to say, computational power on mobile devices are expected to be lower. This suggests that XceptionNet is not a suitable choice for real time inferences on mobile devices and most laptops.

On the other hand, MobileNet can classify videos of around 25 FPS on the computer used for experimentation. This is better than XceptionNet, however it is also likely to be computationally intensive for most present-day mobile devices.

Finally, the proposed model has dramatically lower both training and inference times, with not much of a significant drop in accuracy. Once again, referring to Fig. 7, this model can classify more than 1000 frames per second real time on the experimentation computer. This suggests that resource constrained devices can comfortably handle real time classification using this model.

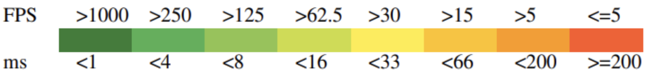

Fig. 7. inference time/ms to FPS visualization scale [23]

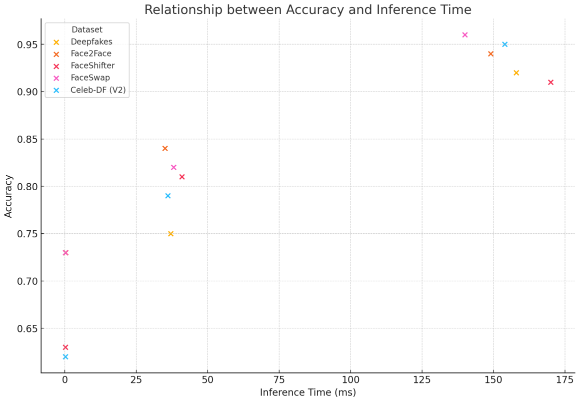

Fig. 8. Accuracy vs inference times/ms

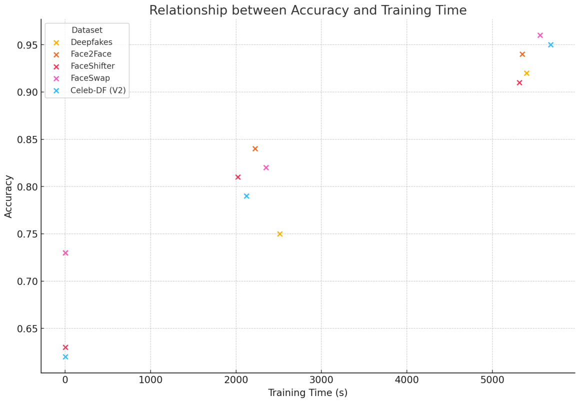

Fig. 9. Accuracy vs training duration/s

7 LIMITATIONS AND CONCLUSION

Note that these models can also be pruned after training to reduce their size and improve inference times. However, pruning was not carried out in this research. Moreover, due to resource constraints on the computer used for these experiments, a small subset of data from each of the datasets were used for training and testing. But if trained and tested with larger quantity of data, the metrics of all the models are expected to improve.

With increase in edge devices, focusing on effective lightweight machine learning models is becoming crucial. In this paper, a very lightweight model has been shown to be able to identify deepfake manipulated videos to a certain extent.

REFERENCES

(1) Fooled Twice: People Cannot Detect Deepfakes but Think They Can - PubMed. https://pubmed.ncbi.nlm.nih.gov/34820608/. Accessed 10 May 2024.

Dataset	Network	Accuracy	Precision	Recall	Training Time/s	Inference Time/ms	Number of parameters
Deepfakes	MesoInception4 + LSTM	0.73	0.70	0.86	2.89	0.17	215,051
	MobileNet	0.75	0.71	0.91	2,511	37	4,279,489
	XceptionNet	0.92	0.92	0.92	5,400	158	22,960,681
Face2Face	MesoInception4 + LSTM	0.73	0.72	0.77	3.48	0.24	215,051
	MobileNet	0.84	0.67	0.82	2,223	35	4,279,489
	XceptionNet	0.94	0.90	0.89	5,350	149	22,960,681
FaceShifter	MesoInception4 + LSTM	0.63	0.62	0.81	3.05	0.18	215,051
	MobileNet	0.81	0.78	0.66	2,020	41	4,279,489
	XceptionNet	0.91	0.94	0.89	5,315	170	22,960,681
FaceSwap	MesoInception4 + LSTM	0.73	0.75	0.74	2.76	0.19	215,051
	MobileNet	0.82	0.72	0.74	2,351	38	4,279,489
	XceptionNet	0.96	0.90	0.88	5,556	140	22,960,681
Celeb-DF (V2)	MesoInception4 + LSTM	0.62	0.76	0.51	3.92	0.22	215,051
	MobileNet	0.79	0.70	0.75	2,122	36	4,279,489
	XceptionNet	0.95	0.89	0.87	5,681	154	22,960,681

Table 3. Performance Comparison of Different Networks on Various Datasets

(2) "Criminal Exploitation of Deepfakes in South East Asia." Global Initiative, https://globalinitiative.net/analysis/deepfakes-ai-cyber-scam-south-east-asia-organized-crime/. Accessed 10 May 2024.

(3) Deepfake Presidents Used in Russia-Ukraine War. 18 Mar. 2022, https://www.bbc.com/news/technology-60780142.

(4) Salvi, Davide, et al. "A Robust Approach to Multimodal Deepfake Detection." Journal of Imaging, vol. 9, no. 6, June 2023, p. 122, https://doi.org/10.3390/jimaging9060122.

(5) Yu, Peipeng, et al. "A Survey on Deepfake Video Detection." IET Biometrics, vol. 10, no. 6, 2021, pp. 607–24,https://doi.org/10.1049/bme2.12031.

(6) Gupta, Gourav, et al. "A Comprehensive Review of Deep-Fake Detection Using Advanced Machine Learning and Fusion Methods." Electronics, vol. 13, no. 1, Jan. 2024, p. 95, https://doi.org/10.3390/electronics13010095.

(7) "MesoNet: A Compact Facial Video Forgery Detection Network." Ar5iv, https://ar5iv.labs.arxiv.org/html/1809.00888. Accessed 29 June 2024.

(8) "Generative Adversarial Network (GAN)." GeeksforGeeks, 15 Jan. 2019, https://www.geeksforgeeks.org/generative-adversarial-network-gan/.

(9) "The Effectiveness of Temporal Dependency in Deepfake Video Detection." Ar5iv, https://ar5iv.labs.arxiv.org/html/2205.06684. Accessed 30 June 2024.

(10) Ganiyusufoglu, Ipek, et al. Spatio-Temporal Features for Generalized Detection of Deepfake Videos. 2020, https://doi.org/10.48550/ARXIV.2010.11844.

(11) Li, Yuezun, et al. In Ictu Oculi: Exposing AI Generated Fake Face Videos by Detecting Eye Blinking. 2018, https://doi.org/10.48550/ARXIV.1806.02877.

(12) Saikia, Pallabi, et al. A Hybrid CNN-LSTM Model for Video Deepfake Detection by Leveraging Optical Flow Features. 2022, https://doi.org/10.48550/ARXIV.2208.00788.

(13) Nadimpalli, Aakash Varma, and Ajita Rattani. Facial Forgery-Based Deepfake Detection Using Fine-Grained Features. 2023, https://doi.org/10.48550/ARXIV.2310.07028.

(14) Li, Meng, et al. "Deepfake Detection Using Robust Spatial and Temporal Features from Facial Landmarks." 2021 IEEE International Workshop on Biometrics and Forensics (IWBF), IEEE, 2021, pp. 1–6, https://doi.org/10.1109/IWBF50991.2021.9465076.

(15) Mao, Maoyu, and Jun Yang. Exposing Deepfake with Pixel-Wise AR and PPG Correlation from Faint Signals. 2021, https://doi.org/10.48550/ARXIV.2110.15561.

(16) Shahzad, Sahibzada Adil, et al. "Lip Sync Matters: A Novel Multimodal Forgery Detector." 2022 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), 2022, pp. 1885–92, https://doi.org/10.23919/APSIPAASC55919.2022.9980296.

(17) Kharbat, Faten F., et al. "Image Feature Detectors for Deepfake Video Detection." 2019 IEEE/ACS 16th International Conference on Computer Systems and Applications (AICCSA), 2019, pp. 1–4, https://doi.org/10.1109/AICCSA47632.2019.9035360.

(18) El-Gayar, M. M., et al. "A Novel Approach for Detecting Deep Fake Videos Using Graph Neural Network." Journal of Big Data, vol. 11, no. 1, Feb. 2024, p. 22, https://doi.org/10.1186/s40537-024-00884-y.

(19) Thies, Justus, et al. Face2Face: Real-Time Face Capture and Reenactment of RGB Videos. 2020, https://doi.org/10.48550/ARXIV.2007.14808.

(20) Masood, Momina, et al. "Deepfakes Generation and Detection: State-of-the-Art, Open Challenges, Countermeasures, and Way Forward." Applied Intelligence, vol. 53, no. 4, 2023, pp. 3974–4026, https://doi.org/10.1007/s10489-022-03766-z.

(21) Li, Yuezun, et al. Celeb-DF: A Large-Scale Challenging Dataset for DeepFake Forensics. 2019, https://doi.org/10.48550/ARXIV.1909.12962.

(22) Rössler, Andreas, et al. FaceForensics++: Learning to Detect Manipulated Facial Images. 2019, https://doi.org/10.48550/ARXIV.1901.08971.

(23) Li, Lingzhi, et al. FaceShifter: Towards High Fidelity And Occlusion Aware Face Swapping. 2019, https://doi.org/10.48550/ARXIV.1912.13457.

(24) Bianco, Simone, et al. Benchmark Analysis of Representative Deep Neural Network Architectures. 2018, https://doi.org/10.48550/ARXIV.1810.00736.