

## DETECTION OF DEEP FAKES USING DEEP LEARNING

By

ANJANI SUPUTRI DEVI D. \*

SAI KISHORE T. \*\*

VENKATA SRI SAI TEJASWI V. \*\*\*

VENKATA SUBRAHMANYA SIVARAM G. \*\*\*\*

SUBHAN SAHEB S. K. \*\*\*\*\*

VIKAS KUMAR K. \*\*\*\*\*

\*\_\*\*\*\*\* Computer Science and Engineering, Sri Vasavi Engineering College, Andhra Pradesh, India.

[\\* Assis](#)

Date Received: 06/05/2024

Date Revised: 10/05/2024

Date Accepted: 14/05/2024

### ABSTRACT

Deep learning algorithms have simplified the process of creating indistinguishable synthetic videos, or deep fakes, because of the unparalleled increase in processing power. It is concerning because these face-swapped manipulations are often used in a variety of contexts, such as blackmail and political manipulation. This paper presents a revolutionary deep learning-based approach to accurately discriminating between real and artificial intelligence (AI)-generated false films. Using a ResNext Convolutional Neural Network (CNN) for frame-level feature extraction, this method makes use of an automated mechanism intended to identify replacement and re-enactment deep fakes. A Recurrent Neural Network (RNN) equipped with Long Short-Term Memory (LSTM) training is utilized to classify videos and distinguish between real and modified ones. The system demonstrates the effectiveness of a straightforward and reliable methodology, in addition to utilizing complex neural network topologies. Through testing, this paper showcases how well the system can accurately identify videos playing a crucial role in ongoing initiatives to combat the increasing dangers posed by the proliferation of deep fake content in society.

Keywords: Deep Learning Algorithm, Deep Fakes, Artificial Intelligence, ResNext CNN, RNN, LSTM, Artificial Intelligence.

### INTRODUCTION

The deep fakes are created using deep learning GANs, a subtype of artificial intelligence. These can be exploited for a variety of purposes, including political disinformation, blackmail, and the spread of viral content. These occurrences highlight the urgent need for strict laws and public awareness campaigns to stop the spread of deepfake content. Social media companies are also being urged to strengthen their detection systems. In the meantime, persistent efforts to counter this growing danger to online authenticity and confidence are reflected in continuous developments, such as suggested tighter IT regulations and improvements in

detection technologies.

An online deepfake video purports to show a cricket player in India supporting dubious gaming software, sparking concerns about misinformation and technology exploitation. The cricket player quickly refuted the video and highlighted the dangers of deep fakes. The episode attracted media attention, underscoring the growing danger posed by deep fakes and the possible damage. These instances highlight the impending time when it will get harder to determine the real identity hiding behind digital interactions. This kind of uncertainty can reduce the credibility and engagement of virtual interactions by posing serious security threats and resource inefficiencies. Making sure online interactions are genuine is crucial to avoiding potential repercussions, as technology makes it harder to distinguish between fact and fiction.

In June 2019, a deepfake video emerged on Instagram



This paper has objectives related to SDG

depicting the CEO of a major tech company delivering a speech about data control using "Spectre," a technology he never mentioned. The video sparked concerns about misinformation and prompted discussions on social media platform policies regarding deep fakes. The video remains online with a false label, highlighting the ongoing challenges associated with deepfake technology. This incident underscores the need for continued efforts to address the misuse of AI-generated content (Li et al., 2018).

## 1. Related Works

Sun et al. (2021) proposed an approach that uses frequency domain features to extract edge geometric features of the image and spatial domain features integrated to extract the high-level features of the image under a neural network in a deep fake detection method based on cross-domain fusion. Low-level and high-level features are combined to create the image's feature vectors in order to obtain greater detail in facial features. Their detection method is effective and has higher cross-database performance when the target datasets are created using deepfake technology. This is because their approach includes carefully considered edge features. Their approach, however, is less successful with the neural textures dataset since it gives less thought to minor features like the nose, mouth, and eyes (Song et al., 2014). Gandhi and Jain (2020) produced adversarial perturbations to enhance deepfake images and fool common deepfake detectors in both black-box and white-box scenarios by utilizing the Carlini and Wagner L2 norm attack and the Fast Gradient Sign Method. When deep fakes were not disrupted, detector accuracy was over 95%, but it was less than 27% when deep fakes were disrupted. The paper investigated Lipschitz regularization and Deep Image Prior (DIP), two enhancements to deepfake detectors. Lipschitz regularization boosts the detector's resilience to input disturbances by constraining its gradient with respect to the input. The DIP defense employs generative convolutional neural networks to eliminate perturbations in an uncontrolled fashion. On average, regularization increased the identification of disturbed deep fakes, with a 10% accuracy increase in

the blackbox scenario. When disturbed deep fakes tricked the original detector, the DIP defense obtained 95% accuracy, in other circumstances, on a subsample of 100 images, it retained 98% accuracy.

Li et al. (2020) worked by analyzing an input face image and producing a grayscale image that shows whether the image can be divided into two separate images that were blended together. They observed that the majority of face-altering techniques currently in use all involve integrating the modified face into an already-existing background image. For this reason, the majority of face manipulation algorithms in use produce forgeries, and face X-rays offer an efficient means of identifying them. Research has shown that face X-rays are useful for detecting forgeries created by invisible face manipulation techniques, despite the majority of face forgery detection or deepfake detection algorithms in use (Ciftci et al., 2020).

Yang et al. (2019) founded a recently developed technique for exposing artificial intelligence (AI)-generated fake face images or videos, or 'Deep Fakes,' using 'Head Poses.' This technique is based on the observation that deep fakes are produced by splicing a synthetic face region into the original image, introducing errors that become visible when 3D head poses are estimated from the face images. They subsequently created a categorization technique based on this cue and conducted experiments to illustrate this occurrence. An SVM classifier is assessed with a set of authentic face photos and deep fakes using characteristics derived from this cue.

A free software program based on machine learning has made it possible to produce "deepfake" videos (King, 2009). Güera and Delp (2018) proposed a temporal-aware pipeline for automatically recognizing deepfake videos. According to Bayar and Stamm (2016), convolutional Neural Networks (CNNs) are used in their system to retrieve frame-level characteristics. A Recurrent Neural Network (RNN) is then trained with these features to determine whether or not a video has been altered. They tested their approach on a sizable collection of deepfake films that they had gathered from several different video

websites.

## 2. Proposed System

### 2.1 Deepfake Videos

An encoder and decoder network combination is used to create deepfake movies, frequently within the context of a Generative Adversarial Network (GAN). The important features and representations are extracted by the encoder network after it has analyzed the original content, which includes the original face. After that, the decoder network receives these features and creates new material, such as a modified face. This procedure is kept up until the AI produces the intended outcome. To maintain consistency in the case of video deep fakes, this procedure is done for every frame of the video. According to Do et al. (2018) and Xuan et al. (2019), the way GANs operate is by utilizing the network of discriminators to try and distinguish between actual and fraudulent content and a generator network to produce fake content.

### 2.2 Deep Fake Detection

Li and Lyu (2018) proposed an approach that leverages advanced deep learning techniques to detect deepfake videos with high accuracy. By employing a combination of LSTM-based neural networks and a pre-trained ResNext CNN, this paper effectively analyzes the sequential temporal patterns within video frames while extracting essential features specific to each frame. Through extensive training on diverse datasets like FaceForensic++, Deepfake Detection Challenge, and Celeb-DF, this model is optimized to handle real-time scenarios and ensure robust performance. Moreover, this paper developed a user-friendly front-end application for seamless interaction, where users can upload videos for the authenticity of the content, whether it is genuine representation or a deepfake, assessed alongside the confidence level of the model's assessment.

#### 2.2.1 Datasets

- *Face Forensics++*: This is one of the largest and most popular deepfake video datasets. It includes more than 1,000 authentic videos and more than 1,000 deepfake videos produced with a variety of

techniques, such as Face2Face, Deep Fake, and NeuralTextures.

- *Celeb-DF*: Another well-known dataset for identifying deepfake videos, it contains more than 5,639 deepfake videos created with the Deep Fake technique, along with over 890 authentic videos.
- *Dataset for the Deep Fake Detection Challenge (DFDC)*: This dataset was produced by Facebook as part of a competition to improve deepfake detection techniques (Schroepfer, 2019).
- *DeeperForensics-1.0*: This is a relatively new dataset with more than 5,000 videos, both real and deepfake, created using multiple techniques.
- *Own-Made Dataset*: This is the own dataset, produced to enhance the system's performance, increase training and prediction accuracy, and identify fraudulent videos in real-time scenarios.

Usually, these datasets include tagged videos, where each one has a label designating it as authentic or fraudulent. They are frequently employed in the evaluation and training of deepfake video detection models. Nevertheless, there are drawbacks to using these datasets, such as the possibility of bias in the categorization process and the limited range of deepfake videos that are included (Güera & Delp, 2018).

## 3. System Architecture

### 3.1 Making of Deepfake Videos

Knowing how deepfake videos are made is necessary in order to comprehend the deepfake detection procedure. Deep learning algorithms are employed in a complex procedure to create deepfake videos (El-Gayar et al., 2024). A generative adversarial network (GAN), a type of deep neural network, is first trained using enormous datasets of real-world facial emotions and motions. This trained model is then used to produce artificial face expressions and traits that resemble the target person. The intended video material is then expertly combined with these generated features to provide a convincing and lifelike image of the target individual performing acts they have never done. As a result, we get a misleading video that is difficult to tell apart from real

footage. Figure 1 describes how deepfake videos can be generated using real images and source video.

### 3.2 Description of Model's Operation

In the suggested framework, the utilization of Res-Next Convolutional Neural Network is employed for the extraction of frame-level features from videos. Subsequently, these features are inputted into an RNN (Recurrent Neural Network) based on LSTM (Long Short-Term Memory) architecture. This RNN is designed to discern the temporal relationships among individual frames, facilitating the classification of videos into the categories of either authentic or manipulated. Figure 2 shows the Model's system architecture and Figure 3 shows the flow of deep fake detection process.

#### 3.2.1 Gathering Data

We have combined data from several datasets, including

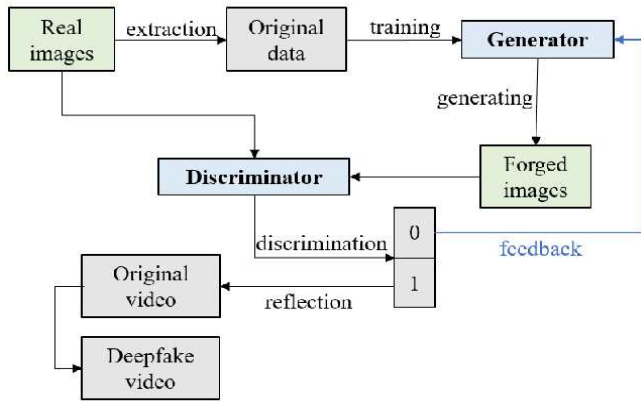


Figure.1 Deepfake Videos Generated Using Real Images and Source Video

the FaceForensics++, Celeb-DF, Deepfake Detection Challenge to create a new dataset (Yang et al., 2017).

#### 3.2.2 Splitting of Data

To ensure that our model is not biased, we trained it on an equal number of real and fraudulent videos. We took an existing dataset, preprocessed it, and produced a new processes dataset in the development phase that only contained the face-cropped videos.

#### 3.2.3 Pre-processing

Dataset preprocessing comprises the breaking the video into frames and identifying faces within each frame and precisely cropping the frame to focus solely on the detected face. To maintain uniformity in the dataset, the mean number of frames per video is calculated. Subsequently, a new dataset is created, containing face-cropped frames with this mean frame count. Frames lacking detected faces are excluded during the preprocessing stage.

#### 3.2.4 Model

The architecture of the model consists of a single LSTM layer after resnext50\_32x4d. The videos are divided into train and test sets by the Data Loader after they have been preprocessed and face cropped. During training and testing, the frames from the processed videos are divided into smaller batches. These batches facilitate efficient model training and evaluation.

#### 3.2.5 Prediction

To make predictions, a fresh video is fed into the learned

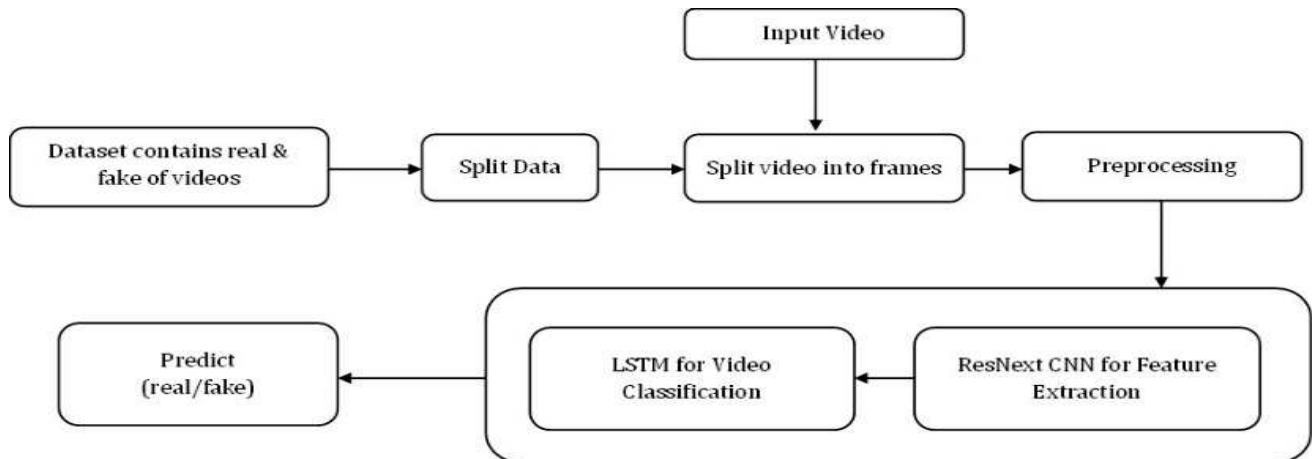


Figure 2. Model's System Architecture

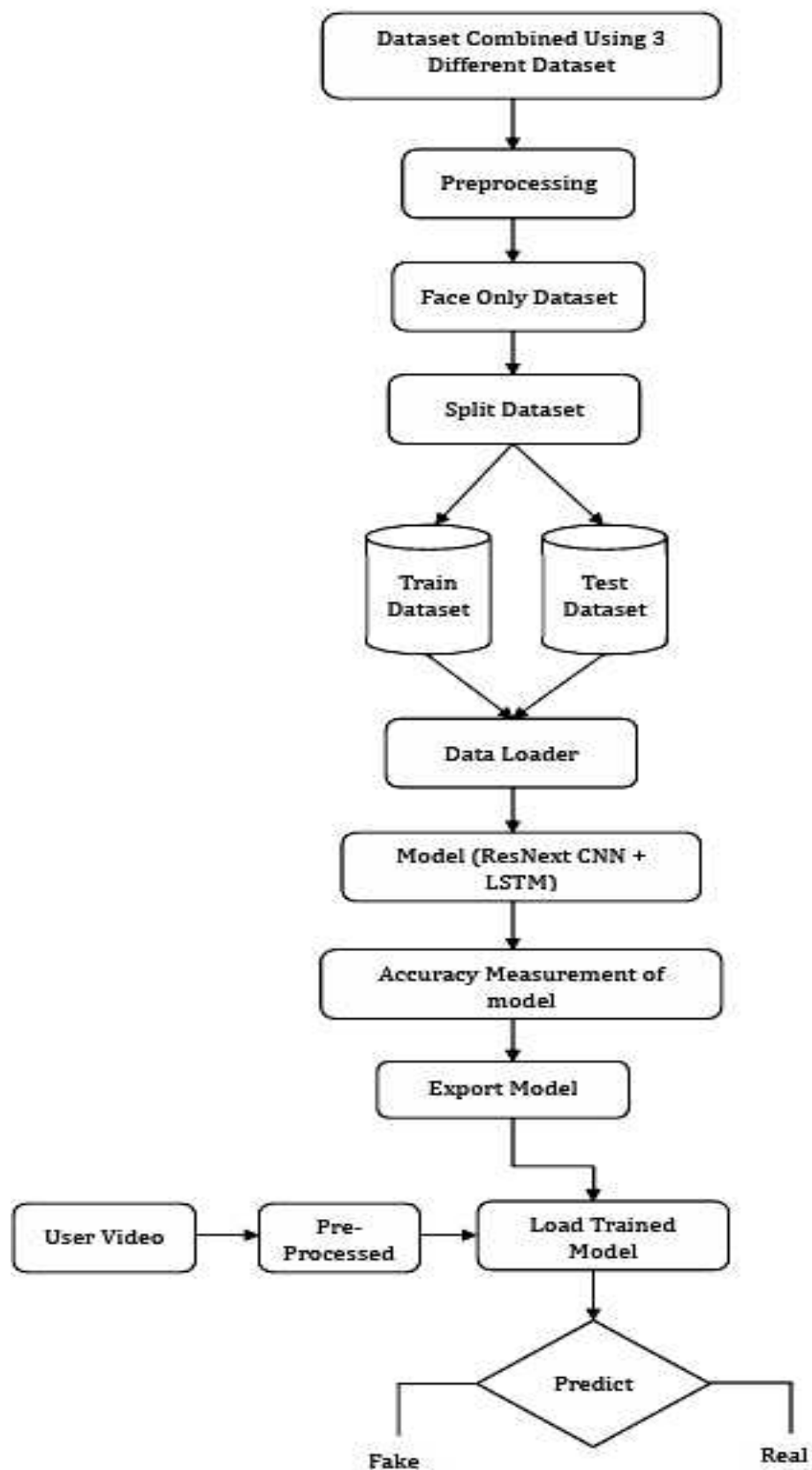


Figure 3. Flow of Deep Fake Detection Process

model. To incorporate the trained model's format, a fresh video is additionally preprocessed. Upon dividing the video into frames and cropping them to focus on faces, the face-cropped frames are directly fed to the trained model for identification. There is no local storage involved in this streamlined approach.

### 3.2.6 Output

The results, from the model will show if the video is real or a deepfake along with the level of certainty, in the model.

A hybrid dataset comprising real and altered videos from multiple sources was used to train and assess the suggested approach for identifying phony videos in real-time situations. At first, the videos have been taken from the dataset of 3 sources Face Forensics++, Celeb-DF, Deepfake Detection Challenge. Lastly, we also have a self-created dataset that we use to enhance training accuracy and generate real-time video results. Additionally, we combined the gathered datasets to create a brand-new dataset. We have taken into consideration 50% Real and 50% Fake videos in order to prevent the model's training bias. Using a Python script, we pre-processed the DFDC dataset and eliminated the audio-altered videos. After that, the videos undergo preliminary processing in which the faces are removed and then saved as a different face-cropped video dataset.

The model consists of two parts: a ResNext CNN and an LSTM. The ResNext CNN is used to extract features from each video frame. The LSTM is used to classify each video as real or fake based on the features extracted by the ResNext CNN.

A ResNext CNN is used to extract features from each video frame. The ResNext CNN is a type of convolutional neural network (CNN) that has been shown to be effective for image classification. The features extracted by the ResNext CNN are used to represent the visual content of each video frame. To determine whether a video is authentic or fraudulent, an LSTM is employed. An LSTM is a type of network (RNN) that is specialized in recognizing long term connections, within datasets. Using the features that were recovered by the ResNext CNN as input, the

LSTM makes predictions on the authenticity of the video. On the validation set, the model's accuracy is evaluated. By doing this, you can make sure that the model isn't overfitting the training set. Exporting the trained model is the last step. This makes it possible to classify fresh videos as true or fraudulent using the model.

The experiment's outcomes demonstrated the effectiveness of the suggested technique in identifying video modifications. The system demonstrated a high degree of accuracy in differentiating between actual and modified videos during the training of the Hybrid dataset, with an accuracy of 95.83% and a loss value of 0.177. The user can upload a video label it as genuine or fake and stop it from being circulated on the internet through a version of our platform. This project may be easily integrated with popular applications like WhatsApp, Facebook, and Instagram to enable the pre-detection of manipulated or fake films before they are sent to another user.

## 4. Methodology

### 4.1 ResNext CNN

ResNext, a powerful CNN architecture, which improves feature extraction compared to standard CNNs like ResNet. Instead of a single path in each block, ResNext uses multiple "lanes" (groups) of mini-convolutions, each capturing diverse features. These groups then combine their learnings for a richer representation. This "split-transform-merge" approach boosts accuracy and efficiency, making ResNext perform in tasks like image classification, object detection, and even video analysis. To use it for feature extraction, we can start with pre-trained models and fine-tune them on the data, adjusting the "lanes" for optimal performance. The best choice depends on factors like data size and hardware. In deep fake detection, ResNext CNN is used for feature extraction.

### 4.2 LSTM

A specific kind of neural network called an LSTM, or long short-term memory, is made to process sequential input by holding onto crucial information for lengthy periods of time. In contrast to traditional networks, LSTMs have a



special "cell" that can hold important information for extended periods of time; it's like a conveyor belt that holds only the information that's relevant at that time. LSTMs control information flow by removing unnecessary material, letting pertinent information pass via gates that function as security guards, and updating the cell's memory as necessary. Text, voice, and time series are just a few of the sequential data types that LSTMs excel at handling thanks to their unique architecture. They excel in tasks like speech recognition, sentiment analysis, machine translation, and forecasting future patterns in time series data. Long-term dependencies within sequences are essentially captured and understood by LSTMs, which makes them invaluable for applications requiring context-aware processing of historical data.

## 5. Results and Discussion

After user has given input to the system with a video, video undergoes pre-processing, and is fed to the preloaded trained model. Figure 1 explains how input is given to the web application and also mentions the sequence length. i.e. How many frames need to be splitted to get the prediction. Therefore, by giving these two inputs user can get to know the prediction of that video whether the video is a deep fake or an authentic recording. The outcome is known, together with the prediction's degree of confidence and the label including real/fake. Figure 2 explains how frame splitting is done, and showcase the face only cropped frames, i.e. how much sequence length given that many frames are given to the user. And finally, we extended our tests to real-time, and tested against different cases to complete 6 set real world testcases.

### 5.1 Accuracy

Accuracy in deep learning reflects how well a model performs across all classes, assuming equal importance. It's calculated by dividing the number of correct predictions by the total predictions made. After successful training on 80% (960) of videos, we tested the model against 20% (240) of videos.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} = 95.40$$

Where,

TP = True Positives (correctly classified positive instances)

TN = True Negatives (correctly classified negative instances)

FP = False Positives (incorrectly classified positive instances)

FN = False Negatives (incorrectly classified negative instances)

### 5.2 Testing the (Detectify) System with Two Cases Which Happened Recent Times:

#### 5.2.1 Case 1

An AI-generated deepfake video depicting the Ram Lalla idol at the Ayodhya Temple blinking has gone viral on social media, eliciting awe and excitement from netizens. The video showcases intricate details of the idol's movements, including blinking and smiling, creating a lifelike impression. Shared by various users with hashtags like #RamMandir and #AyodhaRamMandir, the video has garnered over 1.4 million views and sparked widespread discussion and admiration. The video was actually created using artificial intelligence (AI) software and not a real wink by the idol. This case talks about the viral video that used AI to make the Ram Lalla idol at the Ayodhya temple site wink, though it was not actually a real wink.

Figure 4 shows our application's home page. Figure 5 shows the page that appears after selecting Get Started. The user can upload a video to this page to see if it has been altered or not. The user has uploaded a video, as seen in Figure 6. Figure 7 illustrates how the video was divided into frames based on the length of the sequence. After cropping the faces from the split frames as indicated, the only face-contained frames are used as



Figure 4. Application's Home Page

# RESEARCH PAPERS



Figure 5. Page that Appears after Selecting Get Started

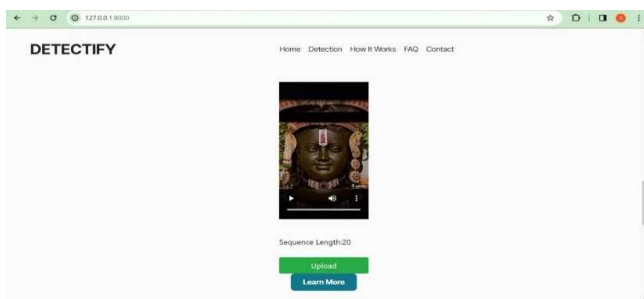


Figure 6. User Uploaded Video

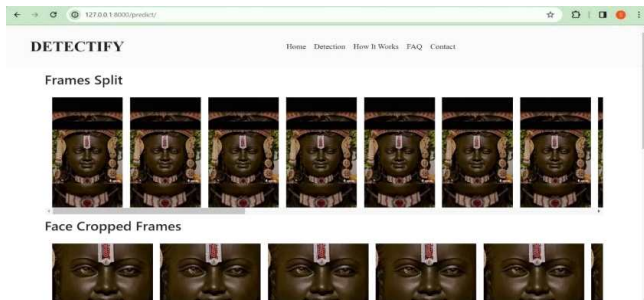


Figure 7. Video Divided into Frames Based on the Length of the Sequence

input. Figure 8 displays the final result along with a percentage to indicate whether the video is real or not.

## 5.2.2 Case 2

In January 2022, a deepfake video of Sachin Tendulkar started circulating on social media platforms. In the video, Tendulkar's face and voice were manipulated to promote an online gambling app. Tendulkar warned people about this fake video on social media and asked them to report such videos. He also tagged government authorities to take swift action. The Mumbai Police registered an FIR and investigated the deepfake video

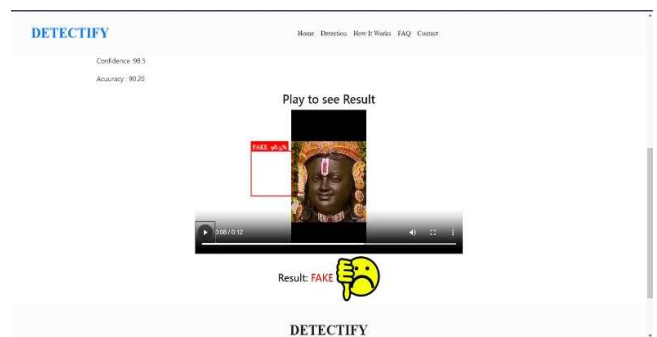


Figure 8. Final Result along with a Percentage to Indicate Whether the Video is Real or Not

case and traced the IP address used to upload it to the Philippines. They are still in the process of getting details from the email provider and checking if a VPN was involved. The input video (Sachin Tendulkar's video) was intended to test the system to check whether the video is real or fake and displayed in Figure 9.

The video's frames are split as seen in Figure 10, and after that, it only gave regard to the frames that featured Sachin Tendulkar's face. Figure 11 displays the result, indicating if the video is genuine or not.

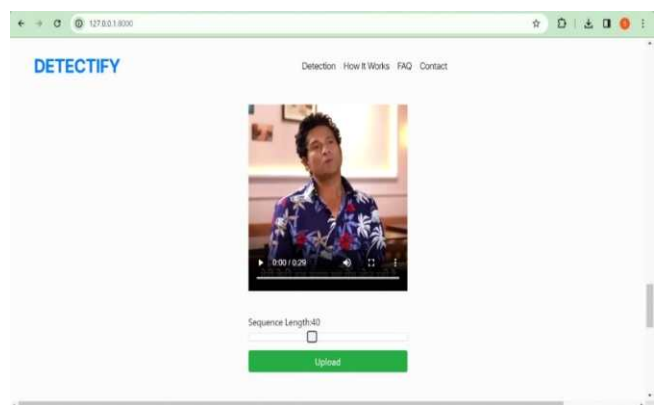


Figure 9. User Uploaded Video

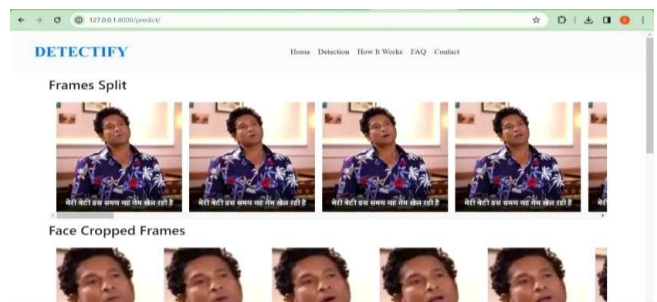


Figure 10. Video split into Frames and Face only Cropped Frames



# RESEARCH PAPERS



Figure 11. Final Result Indicating Whether the Video is Genuine or Not

## 5.2.3 Case 3

Giving real video to the system and checking whether it gives real as output or not.

## 5.3 Key Features

API support is available in Linux, Windows, Python and JS and many more platforms.

Figure 15 and Figure 16 show how any application can access the services of this work through API.

Figure 16 shows the output of the downloaded video in windows when it accessed the API services of this work.

## 6. Limitations

This algorithm does not take audio into account. For this

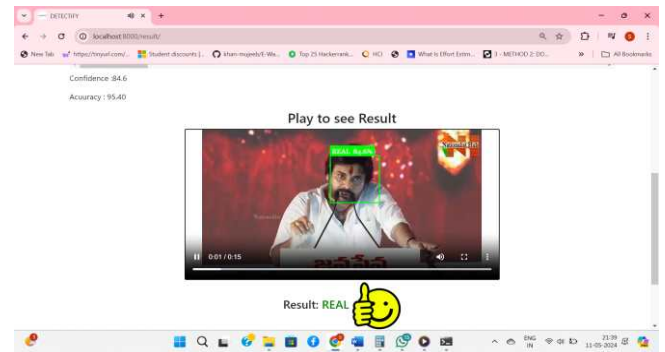


Figure 14. Final Result Indicating that Video is Real

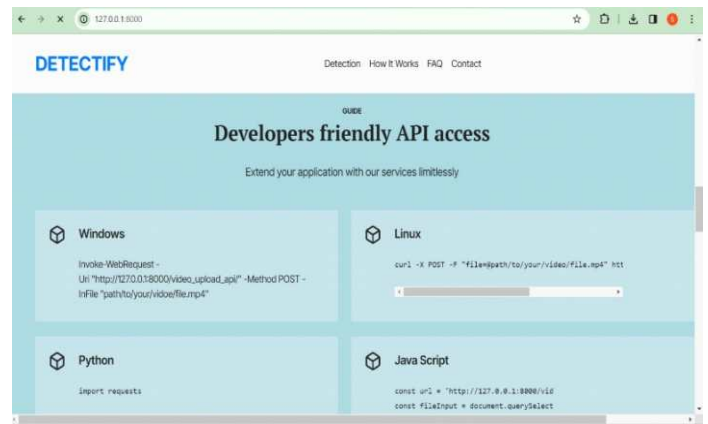


Figure 15. Access to Services through API

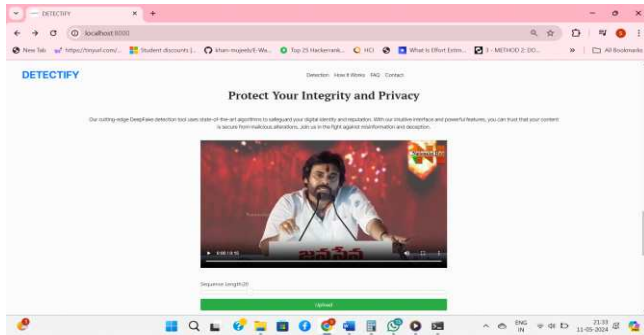


Figure 12. User Uploaded Video (Real Video)

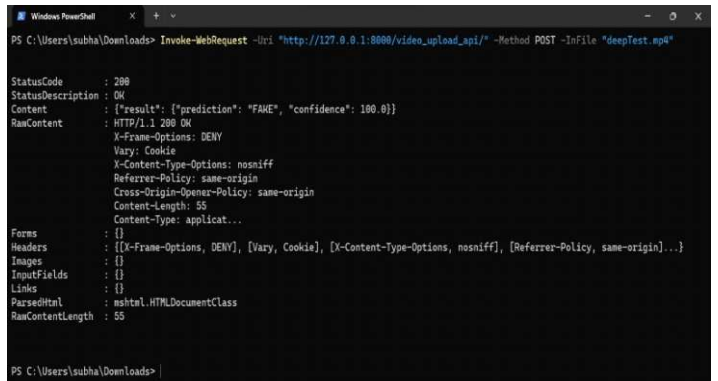


Figure 16. Output of the Video in Windows when Accessed Through API

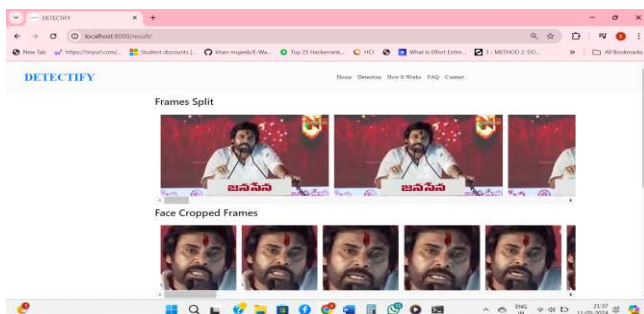


Figure 13. Frames Splitting & Face only of the Video

reason, the audio deep fake will go undetected by this method. However, in the future, we want to be able to detect audio deep fakes. Also, if multiple faces are detected, prediction is inaccurate and sometimes not predicted.

## Conclusion

While there are several tools available for producing deep

fakes, there are very few accessible for detecting them. Our method for identifying deepfakes will significantly aid in preventing the spread of these false impressions over the internet. We want to offer a web-based platform that allows users to post videos and categorize them as authentic or fraudulent. The goal of this work may be expanded from creating a web-based platform to creating a browser plugin that detects deep fakes automatically. This research may be integrated with popular apps like Facebook and WhatsApp to enable simple pre-detection of deepfakes before they are sent to another user.

## References

- [1]. Sun, F., Zhang, N., Xu, P., & Song, Z. (2021). Deepfake detection method based on cross-domain fusion. *Security and Communication Networks*, 2021, 1-11.  
<https://doi.org/10.1155/2021/2482942>
- [2]. Li, L., Bao, J., Zhang, T., Yang, H., Chen, D., Wen, F., & Guo, B. (2020). Face x-ray for more general face forgery detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 5001-5010).
- [3]. Schroepfer, M. (2019). *Creating a Dataset and a Challenge for Deepfakes*. Retrieved from  
<https://ai.meta.com/blog/deepfake-detection-challenge/>
- [4]. El-Gayar, M. M., Abouhawwash, M., Askar, S. S., & Sweidan, S. (2024). A novel approach for detecting deep fake videos using graph neural network. *Journal of Big Data*, 11(1), 22.  
<https://doi.org/10.1186/s40537-024-00884-y>
- [5]. Song, F., Tan, X., Liu, X., & Chen, S. (2014). Eyes closeness detection from still images with multi-scale histograms of principal oriented gradients. *Pattern Recognition*, 47(9), 2825-2838.  
<https://doi.org/10.1016/j.patcog.2014.03.024>
- [6]. King, D. E. (2009). Dlib-ml: A machine learning toolkit. *The Journal of Machine Learning Research*, 10, 1755-1758.
- [7]. Do, N. T., Na, I. S., & Kim, S. H. (2018). Forensics face detection from GANs using convolutional neural network. *The International Securities Association for Institutional Trade Communication (ISITC)*, 2018, 376-379.
- [8]. Yang, P., Ni, R., & Zhao, Y. (2017). Recapture image forensics based on Laplacian convolutional neural networks. In *Digital Forensics and Watermarking: 15th International Workshop, IWDW 2016, Beijing, China, September 17-19, 2016, Revised Selected Papers 15* (pp. 119-128). Springer International Publishing.  
[https://doi.org/10.1007/978-3-319-53465-7\\_9](https://doi.org/10.1007/978-3-319-53465-7_9)
- [9]. Li, Y., & Lyu, S. (2018). Exposing deepfake videos by detecting face warping artifacts. *arXiv preprint arXiv:1811.00656*.
- [10]. Ciftci, U. A., Demir, I., & Yin, L. (2020). Fakecatcher: Detection of synthetic portrait videos using biological signals. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (pp 1-2).  
<https://doi.org/10.1109/TPAMI.2020.3009287>
- [11]. Bayar, B., & Stamm, M. C. (2016, June). A deep learning approach to universal image manipulation detection using a new convolutional layer. In *Proceedings of the 4<sup>th</sup> ACM Workshop on Information Hiding and Multimedia Security* (pp. 5-10).  
<https://doi.org/10.1145/2909827.2930786>
- [12]. Güera, D., & Delp, E. J. (2018, November). Deepfake video detection using recurrent neural networks. In *2018 15<sup>th</sup> IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)* (pp. 1-6). IEEE.  
<https://doi.org/10.1109/AVSS.2018.8639163>
- [13]. Li, Y., Chang, M. C., & Lyu, S. (2018, December). In ictu oculi: Exposing AI created fake videos by detecting eye blinking. In *2018 IEEE International Workshop on Information Forensics and Security (WIFS)* (pp. 1-7). IEEE.
- [14]. Yang, X., Li, Y., & Lyu, S. (2019, May). Exposing deep fakes using inconsistent head poses. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 8261-8265). IEEE.  
<https://doi.org/10.1109/ICASSP.2019.8683164>
- [15]. Xuan, X., Peng, B., Wang, W., & Dong, J. (2019,

# RESEARCH PAPERS

October). On the generalization of GAN image forensics. In *Chinese Conference on Biometric Recognition* (pp. 134-141). Springer International Publishing.

[https://doi.org/10.1007/978-3-030-31456-9\\_15](https://doi.org/10.1007/978-3-030-31456-9_15)

[16]. Gandhi, A., & Jain, S. (2020, July). Adversarial perturbations fool deepfake detectors. In *2020 International Joint Conference on Neural Networks (IJCNN)* (pp. 1-8). IEEE.

---

## ABOUT THE AUTHORS

Anjani Suputri Devi D. is working as a Senior Assistant Professor at Sri Vasavi Engineering College, Tadepalligudem, Andhra Pradesh, India. She is pursuing her Ph.D. in Image Processing at Jawaharlal Nehru Technological University, Kakinada, Andhra Pradesh, India. She received her B.Tech degree in Computer Science and Engineering from Andhra University, Vishakhapatnam, India, in 2003, and her M.Tech degree in Computer Science and Engineering from JNTUK, Kakinada, in 2010. Her current research interests include Image Processing, Computer Vision, Data Mining, and Machine Learning.



Sai Kishore Tadi completed his B.Tech degree in Computer Science and Engineering from Sri Vasavi Engineering College, Pedatadepalli, Andhra Pradesh, India. He has a strong passion for software development and AI, having completed over 10 NPTEL courses and earning 4 stars for his dedication. Sai Kishore has received multiple placement offers and has contributed to research on disease prediction using the Random Forest Algorithm. His interests include machine learning, web development with Django, and data analytics.



Vangala Venkata Sri Sai Tejaswi, a recent Computer Science and Engineering graduate from Sri Vasavi Engineering College, Pedatadepalli, Andhra Pradesh, is interested in Java Full Stack, Artificial Intelligence, and React.js. She has completed internships in these areas and holds an AWS Cloud Practitioner certification as well as a Data Science certification from NPTEL. She has been recognized for academic excellence and has worked on numerous projects leveraging Django, React.js, AI, and other web technologies.



Venkata Subrahmanya Sivaram Gandhi recently completed his B.Tech degree in Computer Science and Engineering from Sri Vasavi Engineering College, Pedatadepalli, Andhra Pradesh, India. His fervor for Data Science and AI is evident through his successful completion of over five NPTEL courses. He achieved impressive scores in the GATE 2024 examination. In Computer Science, he secured a remarkable score of 328, while in Data Science and Artificial Intelligence, he excelled with a score of 303.



Subhan Saheb Shaik recently completed his B.Tech degree in Computer Science and Engineering from Sri Vasavi Engineering College, Pedatadepalli, Andhra Pradesh, India. His passion for software development and artificial intelligence is evident through his completion of over ten NPTEL courses and multiple recognitions. His interests include Machine Learning, Web Development with Django, and Data Analytics. He is proficient in Python, Java, C++, and SQL, and holds a Gold Badge in Python from HackerRank.



Vikas Kumar Koppoju recently completed his B.Tech degree in Computer Science and Engineering from Sri Vasavi Engineering College, Pedatadepalli, Andhra Pradesh, India. He has interned as a Software Development Engineer at ObSoft, where he led multiple successful projects, including an automated education web app and a Python Full Stack Development internship program. Vikas has expertise in machine learning, web development using Django, and AI. He also holds the position of Technical Lead for GDSC at his college.

