IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

DETECTING DEEPFAKE FACE WITH LSTM AND RESNEXT

Prof. Ravindra Patil
computer science and engineering
KLS VDIT Haliyal ,India

Ms.Shivani Chavan
computer science and engineering
KLS VDIT Haliyal ,India

Vinita naik
computer science and engineering
KLS VDIT Haliyal
Dandeli, India

Ankita Patil
computer science and engineering
KLS VDIT Haliyal
haliyal,India

Rashmi Tigadi
computer science and engineering
KLS VDIT Haliyal
Dharward, India

Abstract**—**With the increasing computational power, the creation of indistinguishable human synthesized videos, known as deepfakes, has become remarkably easy. These realistic face-swapped deepfakes have raised concerns as they can be utilized for malicious purposes such as causing political unrest, fabricating terrorism events, spreading revenge porn, and blackmailing individuals. In this research, we present a novel deep learning-based method that effectively distinguishes AI- generated fake videos from real ones. Our approach focuses specifically on detecting replacement and reenactment deepfakes. We harness the power of Artificial Intelligence (AI) to combat the challenges posed by AI itself. The core of our system lies in a ResNext Convolutional Neural Network, which extracts frame-level features. These features are then used to train a Long Short-Term Memory (LSTM)-based Recurrent Neural Network (RNN) that classifies videos as either manipulated (deepfake) or authentic (real).To ensure real-time applicability and enhance the model's performance on real- world data, we evaluate our method using a large and balanced dataset. This dataset is prepared by blending various available datasets, including FaceForensic++[1], Deepfake Detection Challenge[2], and Celeb-DF[3]. Additionally, we demonstrate how our system achieves competitive results through a simple and robust approach.In summary, our research aims to address the challenges posed by deepfakes by utilizing AI technologies. By leveraging a ResNext CNN and LSTM-based RNN, we successfully detect and classify manipulated videos. Through extensive evaluation on mixed and balanced datasets, we showcase the effectiveness and efficiency of our approach in real-time scenarios.

Keywords**—***Deepfake Video Detection, convolutional Neural network (CNN), recurrent neural network (RNN), Long short term memory(LSTM).*

INTRODUCTION

Deepfake videos are manipulated videos that use advanced artificial intelligence and machine learning techniques to create a fake video that looks very realistic. We are using the limitation ofthe deep fake creation tools as a powerful way to distinguish between the pristine and deep fake videos. During the creation of the deep fake the current deep fake creation tools leaves some distinguishable artifacts in the frames which may not be visible to the human being but the trained neural networks can spot the changes. They can be used to create fake news, manipulate public opinion, and harm individuals. Detecting deep fake videos is a complex task, and there are several techniques and approaches that can be used. Some of the most common methods include Facial analysis: This technique involves analyzing the facial expressions, movements, and inconsistencies in the video to determine if it is a deep fake audio analysis. Metadata analysis: Metadata can provide valuable information about the video, such as the location, date, and time it was recorded, which can help determine if the video is authentic or not, Source analysis: This technique involves tracing the origin of the video and analyzing its source to determine if it has been tampered with or manipulated. [Machine learning: Using machine learning algorithms can help detect deep fake videos by training the algorithms to recognize patterns and anomalies in the video. It is important to note that no single technique is foolproof, and a combination of these techniques may be necessary to detect deepfake videos accurately. Additionally, as technology advances, so do the techniques used to create deep fakes, making it a continuously evolving field that requires ongoing research and development. Several machine learning algorithms are used in fake video analysis. Here are some examples: Convolutional Neural Networks (CNNs): CNNs are commonly used in image and video analysis, and they have been shown to be effective in detectingdeep fake videos.

LITERATURE SURVEY

The first method mentioned, "Detecting Face Warping Artifacts in Exposing DF Videos,"[1] focuses on identifying artifacts present in deep fake videos. This technique utilizes a specialized Convolutional Neural Network to compare the generated face regions with their neighboring areas. By examining these areas, the method aims to detect specific visual irregularities that are indicative of deep fakes. The study categorizes these artifacts into different types and takes advantage of the fact that deep fake algorithms often generate low-resolution images. These low-resolution images are then transformed to match the faces being replaced in the original video. By analyzing and comparing these transformations, the method can effectively identify the presence of face warping artifacts, thus aiding in the detection of deep fake videos. The second method, "Detecting Eye Blinking in Exposing AI Created Fake Videos,"[2] tackles the issue of uncovering fake face videos produced using deep neural network models. The approach relies on the detection of eye blinking, which is a physiological signal that is typically not well represented in synthesized fake videos. By examining the presence or absence of eye blinking in the videos, this method can distinguish between genuine and fake content. The evaluation of this technique involves using eye-blinking detection benchmark datasets, which provide a standardized basis for assessing its performance. The results obtained so far show promise in effectively detecting videos generated by the Deep Neural Network-based software DF. However, it's worth noting that other factors, such as teeth enhancement and wrinkles on the faces, should also be taken into account for a more comprehensive detection approach. The third method, "Detecting Forged Images and Videos Using Capsule Networks,"[3] introduces the use of capsule networks for identifying manipulated images and videos in various scenarios. This technique employs a network architecture that is specifically designed to capture hierarchical relationships and spatial arrangements of visual features. During the training phase, random noise is introduced as a means to enhance the network's robustness against different forms of manipulation. However, it should be acknowledged that the inclusion of random noise may not be an ideal solution and could potentially affect the method's performance on real-time data. Although the model shows good results on the dataset used for evaluation, its effectiveness in real-world scenarios where noise is present throughout the training phase remains to be thoroughly examined. The fourth method, "Detecting Synthetic Portrait Videos using Biological Signals,"[4] focuses on extracting and analyzing biological signals from both authentic and fake portrait videos. This approach leverages the concept of spatial coherence and temporal consistency to capture the unique signal characteristics exhibited in genuine videos. By applying specific transformations and processing steps, such as feature set extraction and PPG (Photoplethysmogram) map generation, the method aims to differentiate between real and synthetic content. The system known as "Fake Catcher" demonstrates an accurate detection capability regardless of the video's resolution, quality, generator, or content. However, one of the main challenges associated with this method lies in formulating a differentiable loss function that effectively follows the proposed signal processing steps. Ensuring the preservation and discrimination of biological signals throughout the detection process is crucial to achieve optimal performance.

METHODOLOGY

Algorithm: Convolution Neural Network

1. Input Video/Image
2. Preprocess the Input
   a. Detect Faces
   b. Crop Image
   c. Enhance Image
3. Input to the Model and Predict
   a. Detect Features
   b. Downsize the Image
   c. Predict
4. Display The Result

Convolution Neural Network

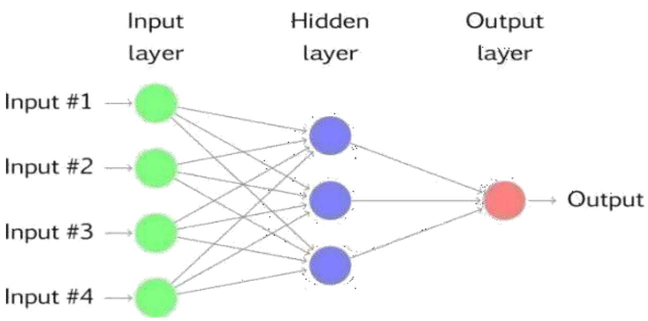


Fig 1.CNN model

To predict image class multiple layers operate on each other to get best match layer and this process continues till no more improvement left.

LSTM

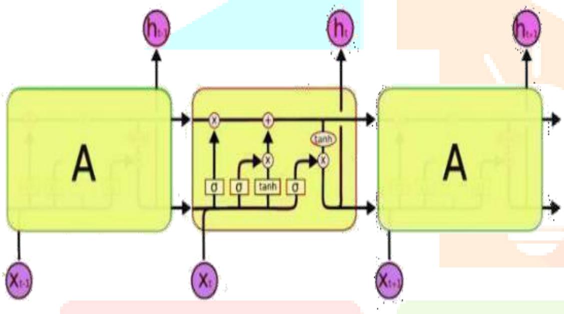


Fig2 .LSTM model

Long short-term memory (LSTM) is an artificial recurrent neural network (RNN) architecture[1] used in the field of deep learning. Unlike standard feedforward neural networks, LSTM has feedback connections. It can not only process single data points (such as images), but also entire sequences of data (such as speech or video). For example, LSTM is applicable to tasks such as unsegmented, connected handwriting recognition,[2] speech recognition[3][4] and anomaly detection in network traffic or IDSs (intrusion detection systems). A common LSTM unit is composed of a cell, an input gate, an output gate and a forget gate. The cell remembers values over arbitrary time intervals and the three gates regulate the flow of information into and out of the cell. LSTM networks are well-suited to classifying, processing and making predictions based on time series data, since there can be lags of unknown duration between important events in a time series. LSTMs were developed to deal with the vanishing gradient problem that can be encountered when training traditional RNNs. Relative insensitivity to gap length is an advantage of LSTM over RNNs, hidden Markov models and other sequence learning methods in numerous applications.

ResNext CNN :

For Feature Extraction Instead of writing the rewriting the classifier, we are proposing to use the ResNext CNN classifier for extracting the features and accurately detecting the frame level features. Following, we will be fine-tuning the network by adding extra required layers and selecting a proper learning rate to properly converge the gradient descent of the model. The 2048-dimensional feature vectors after the last pooling layers are then used as the sequential LSTM input.

SYSTEM ARCHITECTURE

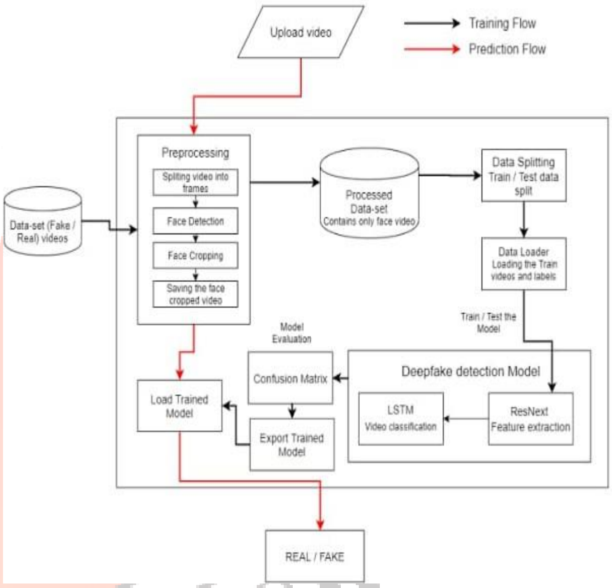

Upload video
Training Flow
Prediction Flow
Preprocessing
Spliting video into frames
Face Detection
Face Cropping
Saving the face cropped video
Processed Data-set Contains only face video
Data Splitting Train / Test data split
Data Loader Loading the Train videos and labels
Train / Test the Model
Data-set (Fake / Real) videos
Model Evaluation
Confusion Matrix
Deepfake detection Model
LSTM Video classification
ResNext Feature extraction
Load Trained Model
Export Trained Model
REAL / FAKE

Fig 3.System architecture

Explanation of System Architecture:

Module 1: Data-set Gathering: To create an efficient real-time prediction model, we gathered data from FaceForensic++ (FF), Deepfake Detection Challenge (DFDC), and Celeb-DF datasets. We combined these datasets with our own collected data, ensuring a balanced mix of 50% real and 50% fake videos. Preprocessing involved removing audio-altered videos from the DFDC dataset. Our final dataset comprised 81 real and 82 fake videos, totaling 163 video.

Module 2: Pre-Processing: In the pre-processing module, videos undergo various steps to remove noise and extract the required content, specifically the face. The initial step involves splitting the videos into frames. Each frame is then analyzed to detect and crop the face. The resulting cropped frames are recombined to form new videos, containing only the face regions.

Frames without detected faces are excluded during pre-processing.

To ensure uniformity in the number of frames, a threshold value is determined based on the mean frame count of each video. This threshold value is chosen considering computational limitations, such as the GPU's processing power.

Module 3: Data-set split: The dataset is divided into a train and test dataset, with a ratio of 70% train videos and 30% test videos. The split is balanced, ensuring an equal distribution of 50% real and 50% fake videos in both the train and test sets.

Module 4: Model Architecture: Our model architecture combines both CNN and RNN components. We utilize a pretrained ResNext CNN model for feature extraction at the frame level. These extracted features are then fed into an LSTM network to classify the videos as either deepfake or pristine. During the training process, the labels of the videos from the training split are loaded using a Data Loader and fitted into the model.

ResNext: To avoid starting from scratch, we leverage a pre-trained ResNext model for feature extraction. ResNext is a Residual CNN network specifically optimized for achieving high performance in deeper neural networks. For our experiments, we utilize the resnext50_32x4d model, which consists of 50 layers and dimensions of 32 x 4. Next, we fine-tune the network by adding additional necessary layers and selecting an appropriate learning rate to ensure proper convergence of the model's gradient descent. The 2048-dimensional feature vectors obtained from the last pooling layers of ResNext serve as the input for the sequential LSTM component.

LSTM for Sequence Processing: For sequence processing, the 2048-dimensional feature vectors are fed into a single LSTM layer. The LSTM layer has 2048 latent dimensions and 2048 hidden layers, with a dropout probability of 0.4, which aids in achieving our objective. The purpose of using LSTM is to process the frames in a sequential manner, allowing for temporal analysis by comparing the frame at time 't' with the frame 't-n' (where 'n' represents the number of frames before 't'). The model also incorporates the Leaky ReLU activation function. A linear layer with 2048 input features and 2 output features is employed to enable the model to learn the average correlation between the input and output. To obtain an output size in the form of H x W (height x width), an adaptive average pooling layer with an output parameter of 1 is included. Sequential layer is used for sequential processing of frames, and a batch size of 4 is employed for batch training. Finally, a SoftMax layer is utilized to obtain the model's confidence during prediction.

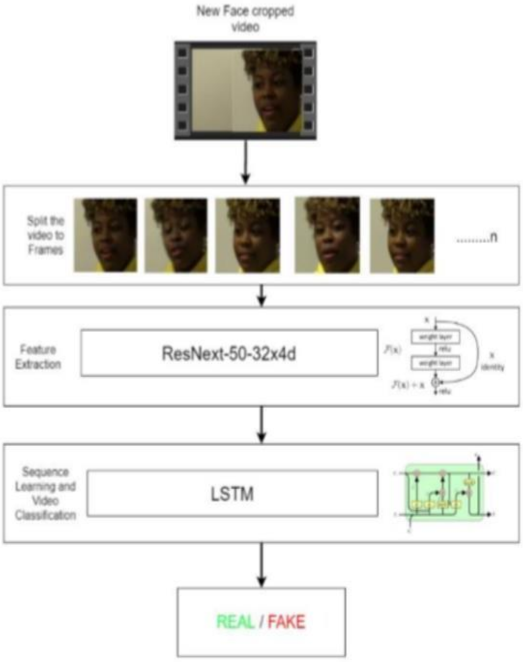
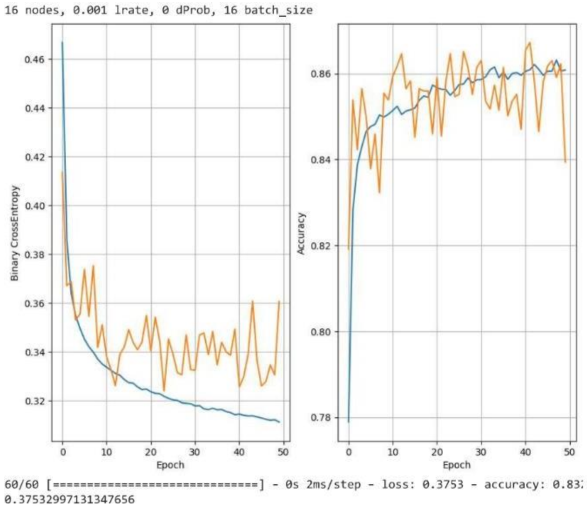


Fig 4 overview of model

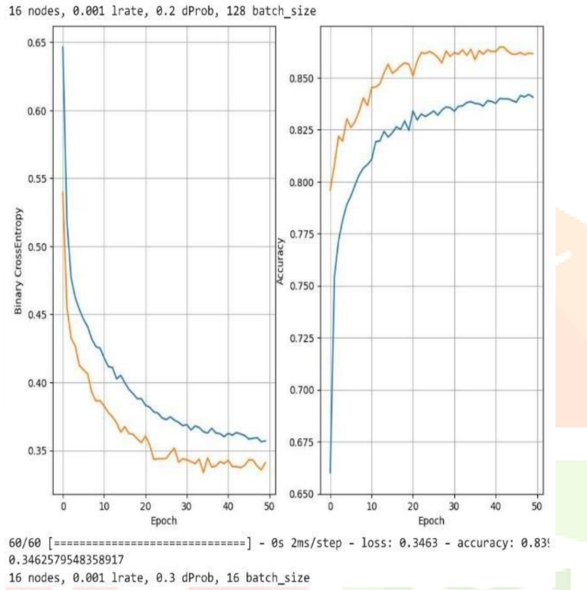
Module 5: Hyper-parameter Tuning: Hyper-parameter tuning involves selecting the optimal hyper-parameters to maximize accuracy. After multiple iterations on the model, the best hyper-parameters for our dataset are determined. To enable adaptive learning rate, we utilize the Adam optimizer [21] with a learning rate of 1e-5 (0.00001), which helps achieve a better global minimum during gradient descent. A weight decay of 1e-3 is employed. Since this is a classification problem, the cross-entropy loss approach is utilized to calculate the loss. To make efficient use of the available computational power, batch training is implemented with a batch size of 4. This batch size has been determined to be ideal for training in our specific development environment.

RESULTS



```
16 nodes, 0.001 lrate, 0 dProb, 16 batch_size
```

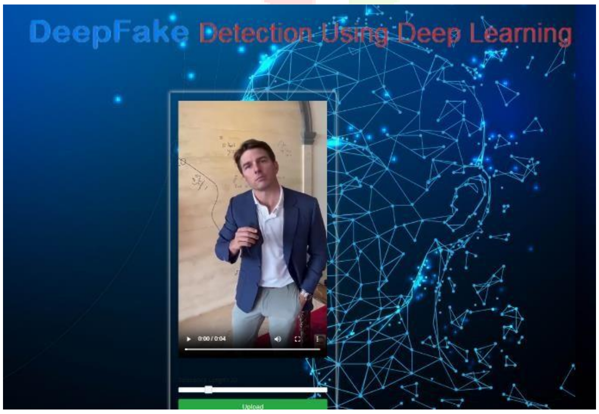
```
60/60 [==============================] - 0s 2ms/step - loss: 0.3753 - accuracy: 0.83:
0.37532997131347656
```



```
16 nodes, 0.001 lrate, 0.2 dProb, 128 batch_size
```

```
60/60 [==============================] - 0s 2ms/step - loss: 0.3463 - accuracy: 0.83!
0.3462579548358917
16 nodes, 0.001 lrate, 0.3 dProb, 16 batch_size
```

Fig.5 Training and plotting Loss and accuracy graphs .



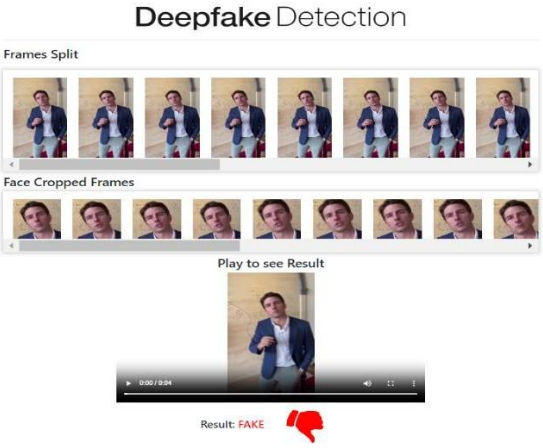


Fig.6 Display results

CONCLUSION

We present a novel solution that utilizes a neural network architecture for the classification of videos into deepfakes or real, providing a comprehensive measure of confidence in the model's predictions. Our approach stands out for its efficiency, as it achieves accurate results by analyzing only one second of video footage, corresponding to a frame rate of 10 frames per second. To construct the model, we leverage the power of a pre-trained ResNext CNN, which excels at extracting detailed features at the frame level. Furthermore, we incorporate an LSTM component that performs sequential analysis, enabling the identification of temporal changes between consecutive frames. Notably, our model is designed to handle video sequences of varying lengths, including options such as 10, 20, 40, 60, 80, and 100 frames. By considering a diverse range of frame sequences, our solution caters to different video contexts and offers enhanced flexibility in deepfake detection.

REFERENCES

[1] Tackhyun Jung, Sangwon Kim, Keecheon Kim - DeepVision:Deepfakes Detection Using Human Eye Blinking Pattern . IEEE 2020.

[2] Md Shohel Rana, Mohammad, Nur Nobi, Andrew H. Sung, Beddhu Murali, Deepfake Detection: A Systematic Literature Review. IEEE 2022.

[3] S Lyu , Deepfake Detection Current Challenges and Next Steps. IEEE 2020

[4] Deng Pan, Lixin Sun, Rui Wang, Richard O. Sinnott, Deepfake Detection through Deep Learning. IEEE 2020.

[5] TensorFlow: https://www.tensorflow.org/ (Accessed on 26 March, 2020)

[6] Yuezun Li, Siwei Lyu, “ExposingDF Videos By Detecting Face Warping Artifacts,” in arXiv:1811.00656v3.

[7] PyTorch : https://pytorch.org/ (Accessed on 26 March, 2020)

[8] G. Antipov, M. Baccouche, and J.-L. Dugelay. Face aging with conditional gen- erative adversarial networks. arXiv:1702.01983, Feb. 2017

[9] J. Thies et al. Face2Face: Real-time face capture and reenactment of rgb videos. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 2387–2395, June 2016. Las Vegas, NV.

[10] Face app: https://www.faceapp.com/ (Accessed on 26 March, 2020)

[11] Rossler, A., Cozzolino, D., Verdoliva, L., Riess, C., Thies, J. and Nießner, M. (2019) Faceforensics++: Learning to Detect Manipulated Facial Images. Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, 27-28 October 2019, 1-11. https://doi.org/10.1109/ICCV.2019.00009 .