Deepfake detection using LSTM and CN

Dr.CHANDRA MOHAN
[1]*Assistant Professor (Adhoc),* [234] *under Graduate*
Dept. Computer Science & Engineering,
JNTUA College of Engineering(Autonomus) Pulivendula – 516390
[2] Boyaramya15@gmail.com [3] reddynaikbch@gmail.com [4]deepikanidadala@gmail.com

Abstract

Deepfake – a technology from the application of Artificial Intelligence, has just emerged since 2017 and has quickly become an Internet phenomenon. Deepfake creation are done on images containing video or on images. However, people have not yet foreseen the dark side of Deepfake. Especially, on social networks such as Facebook, Twitter and soon on .users still carelessly share Deepfake application or software without anticipating the harmful effects of this technology on their lives. Our work is aimed in addressing the social and Economical issues due to fake videos in social media. Accuracy can be increased by adding recent and flexible data set. The algorithms LSTM and CNN where they are abbreviated as Long Short-Term Memory and ResNext Convolution Neural Network for CNN, these algorithms are applied here. Long Short-Term Memory (LSTM) networks and RNN are connected to each other network. It not only checks a single data(such as images), but also entire sequences of data(such as video). A residual block called as ResNext is used as part of the ResNext CNN architecture. The working of this module is "split-transform-merge". This strategy is similar to an Inception module

Keywords: LSTM -Long Short-Term Memory,
CNN -Convolution Neural Network,
RNN -Recurrent Neural Network

INTRODUCTION

Deep fake is a synthesis process on a person's image based on different neurons related network tools like GAN(Generative Adversarial Network) or sophisticated Encoders etc. These tools super impose target images onto the source videos using a deep learning techniques and create a realistic featured deep fake video/Image. These deep-fake video are so realistic that it becomes highly difficult to find the variation between the real one and the faked one . In this process, we describe a new deepfake method that can effectively detect tool generated fake videos/Images from the actual real videos. We are using the limitation of the 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 might not be visible to the human being but the trained neuron related networks can spot the changes.

Our system uses a Res-Next Convolution Neural Networks to extract and analysis the frame-level features. These analysis are then used to feed a Long Short-Term Memory(LSTM) based to classify whether the video is subject to any kind of manipulation or not, i.e. whether video is/are faked or real piece. We thought to execute our model against a large set of deep fake videos pooled from multiple Sources. so to make the deep fake detection model to perform better on real time data. To achieve we trained our model on combination of available data-sets.so that our model can learn the features from different kind of images. We extracted a adequate amount of videos from various sources consisting of nearly 4000+ videos. We also evaluated our model against the large real time data like YouTube data-set to achieve best possible outcome in the real time scenario's.

Deepfake Creation:

Things where these transparent, super unnatural videos are used to create political promotion/mislead, black mail someone and other illegal activities, we have some many applications that can generate deepfake images like Face App and Fake App.

ISSN NO: 2249-3034

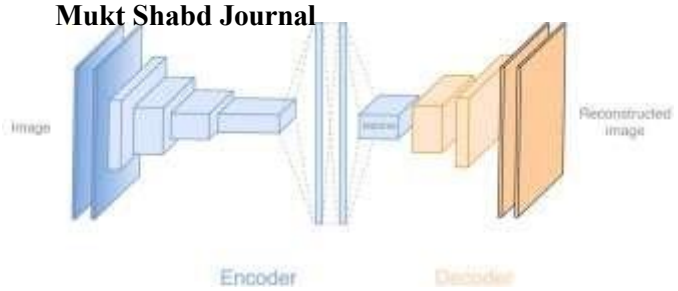

Fig: Deepfake Creation Overview

Deepfake can created by using different apps. some of them includes below

Fake App:

The attempt to deepfake creation was Fake App, the auto encoder extracts eminent features of face images and the decoder is used in building back the real face images. To swap faces between source images and final image pixels, there is a need of encoder decoder pairs. where each pair is taken to train on an image/video for deepfake creation. This approach enables the similar encoder to find and learn the similarity between two faces containing images, which are quite relatively unchallenging because the real and generated faces normally have quite same features as eye shape, nose, facial spots, and relative mouth position.

Face Swap Apps:

Face Swapping pictures are the new trend on social media. Face swaps apps are used for creation of harmless fun, deepfake videos. We have so many Face Swap Apps like Snapchat, B612, instagram, Reface, Cuoace etc.

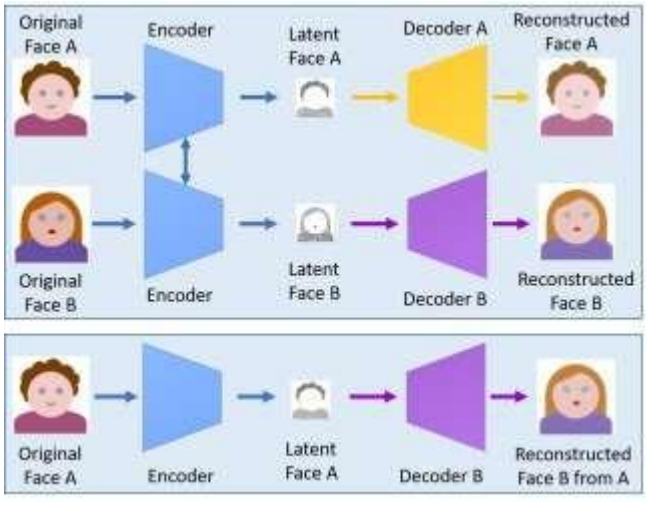

Fig: Deepfake Creation

Deepfake Detection:

Deepfake detection is normally used to find out the difference between naturally created videos and modified ones. This approach of methods requires a huge database of realistic and fake videos to train classification models. The programmed model learns from a given video sequence and generates the output by using LSTM and CNN. We will try to boost the accuracy of our model based on the outputs

Deepfakes are exponentially causing disturbance to privacy, and causing individual security threats and democracy of the common people. Methods for detecting deepfake shave been proposed over these activities to get relief.

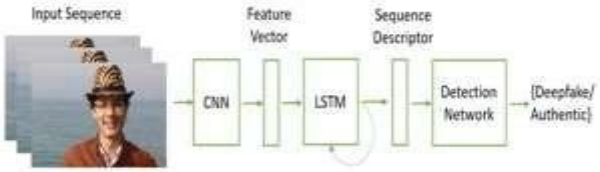

Fig. 4: A deepfake detection method using convolutional neural network (CNN) and long short term memory (LSTM) to extract temporal features of a given video sequence, which are represented via the sequence descriptor. The detection network consisting of fully-connected layers is employed to take the sequence descriptor as input and calculate probabilities of the frame sequence belonging to either authentic or deepfake class [77].

Fig: Deepfake Detection

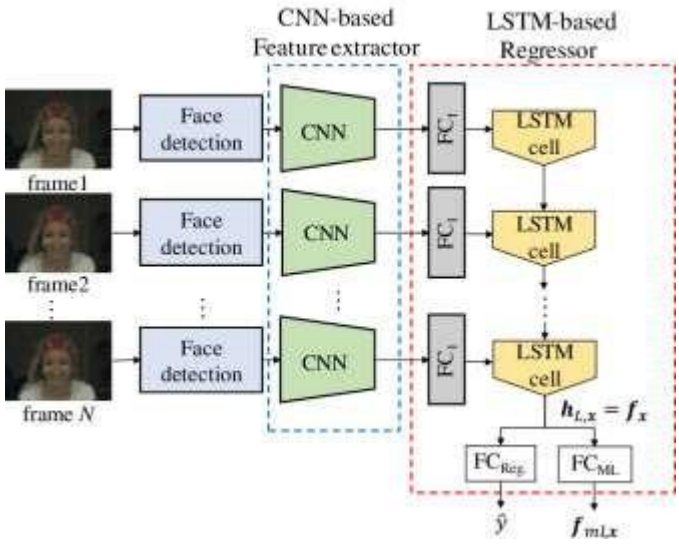

Fig: Block Diagram CNN and LSTM

Literature Survey

Face Warping Artifacts use different methods to find out artifacts by looking and analysing the generated face areas and their surrounding areas with a model know as Convolutional Neural Network model(CNN). Their process is depends upon observations that present deepfake model can only

generate images of limited resolutions, which are then needed to be further transformed to match the faces to be replaced in the source video. Their method has not considered the normal layer analysis of the frames. Detection by Blinking of eyes describes a new process for detecting the deepfakes by the eye blinking as a crucial parameter leading to classification of the videos as deepfake or pristine. Long-term repetitive folding network (LRCN) was used for time analysis of truncated flashing frames.

As it is today, the deepfake generation algorithm is so powerful that it is flawless. Winks aren't the only clues to detect deepfake. there Certain additional parameters need to be considered for deepfake detection Teeth enchantment, facial wrinkles, improper placement of eyebrows, etc. Capsule network for detecting fake images and videos uses the following methods: Use capsule networks to detect fake manipulated images and videos Various scenarios such as replay attack detection and computer-generated video recognition. Their method used random noise during the training phase, but it's not.

This is an opportunity. Still, the model was useful in the dataset, but it can fail in detecting a Real-time data due to noise during training. Our method should be trained A noise-free, real-time dataset. The recurrent neural network (RNN) for deepfake detection used this approach. Use RNNs for sequential processing of frames with ImageNet Pre-trained model. Their process used the HOHO dataset .From only 600 videos. Their dataset consists of a small number of videos and the same type of video Real-time data may not work. We will be training out model on large number of Realtime data.

Architectural Design

Module 1: Data-set Gathering

Module 2: Pre-processing

Module 3: Data-set split

Module 4: Model Architecture

Module 5: Hyper-parameter tuning

Module 1: Data-set Gathering

To be called an efficient model, it should need to detect the real world scenarios. We have gathered the data from different available data-sets like Face Forensic++(FF), Deepfake detection challenges, and Celeb-DF. Futher we have mixed the dataset the collected datasets and created our own new dataset, to accurate and real time detection on different kind of videos. To avoid the training bias of the model we have considered 50% Real and 50% fake videos.
Deep fake detection challenge (DFDC) dataset [3] consist of certain audio alerted video, as audio deepfake are out of scope for this paper. We pre-processed the DFDC dataset and removed the audio altered videos from the dataset by running
a python script.

Module 2: Pre-processing

In this step, the videos are pre-processed and all the unrequired and noise is removed from videos. Only the required portion of the video i.e. face is detected and cropped. The first steps in the pre-processing of the video is to split the video into frames. After splitting the video into frames the face is detected in the frame and that detected frame is cropped along the face. Later the cropped frame is again converted to a new video by combining each frame of the video. The process is followed for each video which leads to creation of processed dataset containing face only videos.

The frame that does not contain the face is ignored while pre-processing. To maintain the uniformity of number of frames, we have selected a threshold value based on the mean of total frames count of each video. Another reason for selecting a threshold value is limited computation power. As a video of 10 second at 30 fps will have total 300 frames and it is computationally very difficult to process the 300 frames at a single time in the experimental environment. So, based on our Graphic Processing Unit computational power in experimental environment we have selected 150 frames as the threshold value.

While saving the frames to the new dataset we have only saved the first 150 frames of the video to the

new video. To demonstrate the proper use of Long Short-Term Memory we have considered the frames in the sequential manner i.e. first 150 frames and not randomly. The newly created video is saved at frame rate of 30 fps and resolution of 112 x 112.

Module 3: Data-set split

The pooled dataset is split into train and test dataset with a ratio of 70% train data and 30% test data video. The dataset split is a balanced split i.e.50% of the real and 50% of fake videos in each split.

Module 4: Model Architecture

Our model is a combination of CNN along with RNN and LSTM. We have used the Pre- trained ResNext CNN model to extract the features at each frame level and based on the extracted features a LSTM network is trained to classify the video as deepfake or pristine. Using
the Data Loader on training split of videos the labels of the videos are loaded and fitted into the model for training.

ResNext:

Instead of typing the code from scratch, we used the pre-trained model of ResNext for feature extraction. ResNext is Residual CNN network optimized for high performance on deeper neural networks. For the experimental purpose we have used resnext50_32x4d model. We have used a ResNext of 50 layers and having 32 x 4 dimensions. The 2048-dimensionaity featured vectors after the last layer i.e. pooling layer of ResNext is used as the sequential input to the sequencing layer shortly know as LSTM.

LSTM for Sequence Processing:

2048-dimensional feature vectors is used as best fit as the input to the LSTM. We are using 1 LSTM layer with 1 2048 latent dimensions and . 2048 hidden layers along with $4*10^{-2}$ dropout chance, which is capable of achieving ,faked images detection objective. LSTM model has a hidden state that one of the major part used in Deepfake Video/image Detection process by checking each level frames in a sequential/ordered manner so that the temporary analysis of the video can be done, by comparing the frame at 't' ,'t-n' seconds w
difference. Where n can be any number of frames before t. The model also consists of Leaky Relu activation function. A linear layer of 2048 input features and 2 output features are used to make the model capable of learning the average rate of correlation between eh input and output. An adaptive average polling layer with the output parameter 1 is used in the model. Which gives the final output size of the image of the form H x W. For sequential processing of the frames a Sequential Layer is used. The batch size of 4 is used to perform the batch training. A SoftMax layer is used to get the confidence of the model during predication.

Module 5: .Hyper-parameter tuning

It is the practice of choosing the perfect hyper-parameters for achieving the maximum posible accuracy that a model can. After reiterating many times on the model. The best hyper-parameters for our dataset are chosen. To enable the adaptives learning outcome Adam optimizer with the model parameters is used. The learning rate is tuned to 1e-5 (0.00001) to achieve a better global minimum of gradient descent. The weight decay used is 1e-3. As this is a classification problem so to calculate the loss cross entropy approach is used. To use the available computation power properly the batch training is used.

The batch's size is taken of 4. Batch size of 4 is tested to be ideal size for training in our development environment. The input video is then passed to the model and prediction is made by the model. The model depicts the output by stating whether given video is real or fake along with the prediction confidence of the model on the face of the playing video.

References

[1] Güera, D. and Delp, E.J. (2018) Deepfake Video Detection Using Recurrent Neural Networks. 2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), Auckland, 27-30 November 2018, 1-6. https://doi.org/10.1109/AVSS.2018.8639163

[2] Mittal, T., Bhattacharya, U., Chandra, R., Bera, A. and Manocha, D. (2020) Emotions Don't Lie: An Audio-Visual Deepfake Detection Method Using Affective Cues. Proceedings of the 28th ACM International Conference on Multimedia, Seattle, 12-16 October 2020, 2823-2832. https://doi.org/10.1145/3394171.3413570

[3] VidTIMIT database. Available at http://conradsanderson.id.au/vidtimit/

[4] Korshunov, P., and Marcel, S. (2019). Vulnerability assessment and detection of deepfake videos. In The 12th IAPR International Conference on Biometrics (ICB), pp. 1-6.

[5] Amerini, I., and Caldelli, R. (2020, June). Exploiting prediction error inconsistencies through LSTM-based classifiers to detect deepfake videos.

[6] Parkhi, O. M., Vedaldi, A., and Zisserman, A. (2015, September). Deep face recognition. In Proceedings of the British Machine Vision Conference (BMVC) (pp. 41.1-41.12).

[7] Korshunova, I., Shi, W., Dambre, J., and Theis, L. (2017). Fast face-swaping using convolutional neural networks(CNN). In Proceedings of the IEEE International Conference on Computer Vision(CV) (pp. 3677-3685).

[8] Bai, S. (2017). Growing random forest (RM) on deep Convolutional Neural Network (CNN) for scene categorization. Expert Systems with Applications,71, 279-287.

[9] Wang, X., Thome, N., and Cord, M. (2017). Gaze latent support vector machine for image classification improved by weakly supervised region selection. Pattern Recognition, 72, 59-71.

[10] Atra Akandeh and Fathi M. Salem,"Slim LSTM NETWORKS: LSTM 6 and LSTM C6",2019 IEEE.

[11] Chao Dong, Chen Change Loy, Member, IEEE, Kaiming He, Member, IEEE, and Xia oou Tang, Fellow,"Image Super-Resolution Using Deep Convolutional Networks",IEEE Transaction on Pattern Analysis and Machine Intelligence(2016).