

Comparative Analysis of CNN and RNN Techniques for Deepfake Video Detection

Manasi Kasande¹, Rashmi Deshmukh², Disha Chaudhari³, Parul Sharma⁴, Tanishka Patel⁵ and Shilpa Pant⁶

¹⁻⁶ Cummins College of Engineering for Women, Pune/Computer Engineering, Pune, India

Email: manasi.kasande@cumminscollege.in, rashmi.deshmukh@cumminscollege.in, disha.chaudhari@cumminscollege.in, parul.sharma@cumminscollege.in, tanishka.patel@cumminscollege.in, shilpa.deogirkar@cumminscollege.in

Abstract—In the recent years photo shopping has been replaced by deepfakes. Deepfakes use deep learning to generate realistic media of fake events. The media can be of any form like audio, video or images. These deepfakes can be created without much efforts due to the presence of online tools and apps. If used with a bad intent, deepfakes are capable of causing harm to the society. Spread of mass misinformation, cyber bullying, spreading propaganda through fake news are few such examples. The rapid progression in deepfake media generation means that it is important to develop tools to identify such media. In this paper, we propose two methods to detect deepfake videos. The first method uses Convolutional Neural Networks (CNNs) and the second method uses Recurrent Neural Networks (RNNs). We train both the models on videos collected from three different open source datasets and also compare the performance of both methods.

Index Terms— Convolutional Neural Networks, Recurrent Neural Network and Deepfake detection.

I. INTRODUCTION

Computers have become very intelligent and better at simulating reality and deepfake technology is one such example. In recent years, a technology known as “deep fake” has been making headlines for its ability to create believable fake media such as audios, images and videos. The word “deepfake” comes from the technology Deep Learning. Deep learning algorithms are trained to create swaps in faces thereby generating “deep fakes”. Deepfake videos consist of a person doing or saying things which weren’t said or done by that person. Moreover, these videos leave very few traces of manipulation. Earlier, limited access to technologies such as Photoshop meant that the amount of fake media circulating on the internet was comparatively lesser than today. Limited access to such apps also meant that there were limited people who had the skills to generate fake media. The rise of the internet and the availability of free audios, videos and images has completely changed the landscape with regards to generation of fake media. The availability of free tools such as FaceApp, Wombo, Deepfakesweb, etc. [1] [2] [3] means that people can create fake media just by clicking a few buttons. FaceApp is a popular selfie editor which provides a plethora of AI filters, backgrounds and effects to create deepfakes [2]. One can create a deepfake video with ease on Deepfakesweb [1]. Wombo is an AI- powered lip sync app which creates ultra-realistic audio altered deepfakes [3].

Fig.1 contains an example of a deepfake image. Tools like these can open the door to more creative ways of filmmaking when it comes to visual arts and generating visual effects in movies. These tools can also be used for



Figure 1. Real Image (left) and its corresponding deepfake (right)

light hearted fun and enjoyment. However, there are many others who misuse deepfakes to cause harm. Spreading fake news, cyber bullying, defamation of public figures, revenge porn and internet scams are more commonplace now [4] [5] [6] [7]. Most recently, deepfakes were weaponized to spread fake news and propaganda during the Ukraine-Russia war. Videos of some pro-Russian Ukraine citizens were viral on social media which were later found out to be deepfake [9]. Deepfakes create a major social problem, especially for celebrities as their photos and videos are publicly available on the internet. It is evident that deepfakes circulated with a malicious intent on the internet can lead to dangerous consequences. Developing tools to detect such deepfakes is vital. Deepfake detection tools can help people like fact checkers and forensic experts in verifying the authenticity of such viral videos. Keeping this in mind, our work has three objectives:

- i. Develop a Convolutional Neural Network (CNN) model for deepfake detection.
- ii. Develop a Recurrent Neural Network model (RNN) for deepfake detection.
- iii. Analyse the performance of the developed CNN and RNN models.

The rest of the paper is structured into 5 parts. Section II contains a literature review of similar works. Section III explains our proposed architecture in detail. Section IV describes the experiments performed. Section V reports the results of all performed experiments. Finally, Section VI summarizes the paper and discusses the future scope.

II. LITERATURE SURVEY

Most deepfake detection methods are based on CNN and RNN. Multiple CNN and RNN techniques for deepfake detection have been developed in the last few years. Ref. [10] Bonettini et al. (2021), tried an ensemble of different CNN models and achieved promising results. The authors consider Efficientnet B4 as the base model and obtain two different ensemble models from it. The ensemble models are obtained by making use of two concepts – attention mechanism and Siamese training. Both models were trained and tested on two datasets, FaceForensics++ [11] and Kaggle’s Deepfake Detection Challenge dataset [12].

It has been observed that deepfakes often have inconsistencies in facial features. Ref. [13] Karandikar, Deshpande, Singh, Nagbhikar and Agarwal (2020) proposed a model that analyses the frames of the videos using CNNs to detect such inconsistencies in the videos. A pretrained VGG16 model [14] was used to train the model using transfer learning on the CelebDF dataset [15]. CNNs have been widely used to detect deepfake videos since they are capable of extracting frame level features. Along with CNNs, RNNs can be used to detect inconsistencies in frames of the videos. Ref. [16] Güera and Delp (2018) proposed a ConvLSTM model which uses both these properties of CNNs and RNNs. The proposed model adopted the Inception V3 model as a feature extractor which was followed by a LSTM network to analyze temporal inconsistencies in the sequences. The model was trained on 300 deepfake videos collected from multiple websites and 300 real videos collected from the HOHA dataset. Ref. [17] Jadhav, Patange, Patel, Patil and Mahajan (2020) also followed the approach of using CNN for feature extraction and LSTM for analyzing the extracted features. ResNext was used as a feature extractor followed by a 2 node neural network consisting of a LSTM layer and a fully connected layer to classify the videos. The authors used a mixed dataset consisting of videos from DFDC, CelebDF and FaceForensics++ [12] [11] [15].

Ref. [18] Chintla et. Al (2020) Proposed a solution capable of detecting both deepfake videos and audio-manipulated deepfakes. On the basis of cost functions, the authors have combined latent CNN architectures with bidirectional RNNs. From audios and videos, the latent representations are chosen in such a way that the information is meaningful for the model. These representations were passed to a Recurrent Neural Network. The RNN was able to identify faults within different frames as well as the whole sequence of frames. For video

detection, the authors used the FaceForensics++ and Celeb-DF video datasets and ASVspoof 2019 Logical Access audio datasets for deepfake audio detection. In order to speed up the development of deepfake detection techniques, Facebook hosted a Deep Fake Detection Challenge (DFDC). The models with the highest accuracy used EfficientNet models along with transfer learning. Based on this observation, [19] Pokroy and Egorov (2021) decided to perform a comparative study of different EfficientNet models for deepfake detection. For this, DFDC dataset was used [12]. Highest accuracy was obtained for the Efficientnet B5 model, followed by Efficientnet B4 and Efficientnet B6. Based on the literature survey, we choose EfficientNet as the CNN model to complete our first objective. We also observe that CNN models as a feature extractor followed by a 2 node neural network consisting of LSTM and a fully connected layer give good results for deepfake detection. Hence, to complete our second objective, we choose EfficientNet as a feature extractor followed by LSTM to detect the temporal inconsistencies within sequences of the videos.

III. PROPOSED ARCHITECTURE

We preprocess our dataset before training. The proposed model focuses on glitches caused around face boundaries while creating deepfakes. For preprocessing, from the video dataset we first extract the frames, then the cropped faces from these frames are extracted and finally we concatenate these cropped faces as videos which is now a new preprocessed dataset of cropped videos. This common dataset is used for both EfficientNet and LSTM models. Transfer learning is then implemented on this model.

Transfer Learning - While practically implementing the model, we can implement transfer learning for detecting Deepfakes. Transfer learning uses pre-trained weights of the neural network for training a fine tuned version of same model on different dataset for some specific application. The model is usually trained on a very large dataset and implementing this pre-trained model on our comparatively smaller dataset improves the performance and reduces the time required for training. Various models like XceptionNet, ResNext, MesoNet, EfficientNet are now available to reuse for specific applications. We have implemented EfficientNet & LSTM using transfer learning on our fine-tuned models for Deepfake Detection. In our solution, we freeze the weights of the convolutional layers of the EfficientNet Model and fine tune the fully connected layers according to our requirement.

EfficientNet - EfficientNet is a CNN architecture and a family of 8 CNN models. The model is designed in such a way that all dimensions of depth, width and resolution are scaled uniformly using a simple compound coefficient. [20] Efficientnet is trained on the ImageNet dataset consisting of 1000 classes. Results have shown that EfficientNet models consistently achieve better accuracy while using lesser parameters when compared with other state of the art CNN models like ResNet, VGGNet, InceptionNet and DenseNet. Moreover, EfficientNet models perform admirably on other transfer learning datasets – notably CIFAR-100 (91.7%), Flowers (98.8%) [20]. As EfficientNet is trained on the ImageNet dataset, its last fully connected layer outputs results for 1000 classes. We fine tune our EfficientNet model for 2 classes, as our goal is to classify a video into fake or real. Also, we develop 4 fine-tuned EfficientNet models based on EfficientNet B1, B2, B3 and B4. Fig. 2 shows the generalized architecture for EfficientNet.

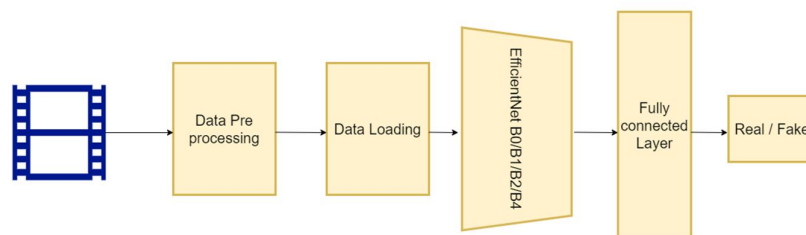


Figure 2. Proposed EfficientNet architecture

Long Short Term Memory (LSTM) – LSTM is a unique kind of Recurrent Neural Network as it is capable of handling long term dependencies and also solves the vanishing gradient problem of other RNNs. Because of this, LSTMs are mainly used in learning dependencies and patterns in large sequences. Since a video is a sequence of frames, LSTMs can perform well on the task of deepfake video detection. We propose a convolutional-LSTM model to detect deepfake videos. EfficientNet B4 is used for feature extraction followed by a 1792-wide LSTM unit for sequence processing. The width of the LSTM unit was chosen as 1792 because EfficientNet B4 outputs a

feature vector of the same number of features. The LSTM unit is followed by a fully connected layer to classify the videos as real or fake. Fig. 3 depicts proposed LSTM architecture.

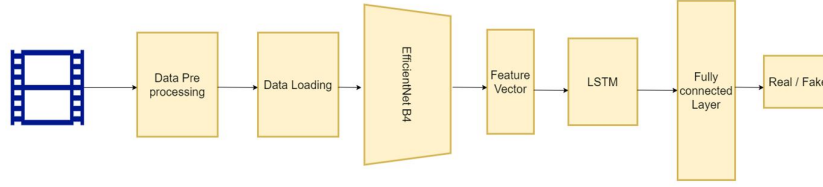


Figure 3. Proposed LSTM Architecture

IV. EXPERIMENTS

A. Dataset

In order to train the model on a variety of deepfake videos, our dataset consists of videos from three different state of the art deepfake detection datasets – CelebDF [15], DFDC [12] and FaceForensics ++ [11]. CelebDF consists of deepfake videos of celebrities and their corresponding real videos. The real videos in these dataset are obtained from YouTube interviews of 59 different celebrities belonging to a varied range of age, gender and ethnicity. The dataset has two versions – CelebDF v1 and CelebDF v2. We have chosen the v1 dataset and its structure is as follows:

- 158 Real videos
- 795 Fake videos

The Deepfake Detection Challenge Dataset (DFDC) is currently the largest dataset that is available for deepfake detection. It has over 100,000 total videos created from 3,426 paid actors. The deepfake videos are created using different methods. This ensures that the models are trained on a wide range of methods. The size of the whole dataset is 470 GB. However, the dataset is also available as 50 smaller chunks of approximately 10 GB each. Keeping in mind the computational resources and storage available to us, we have included the 25th chunk in our dataset and randomly chosen 921 videos from the said chunk. FaceForensics++ is also created using four deepfake creation methods - Face2Face, FaceSwap, DeepFakes, and NeuralTextures. The structure of the dataset is as follows:

- 363 Real Videos
- 3076 Fake Videos

In our dataset, we include all 363 real videos and randomly choose 507 fake videos. Our final dataset consists of 2731 videos out of which 737 are real videos and the remaining are fake.

B. Pre processing

Pre-processing includes extracting frames from the videos, detecting faces from those frames, cropping the faces from the frames and finally combining the cropped images into a video. In order to reduce the computational complexity, we are only considering the first 150 frames of each video. The frames are extracted at a rate of 60 frames per second. The dimensions of the cropped video are 500x500.

C. Training and Testing

First, we check if the cropped video dataset has corrupted videos and remove them. Then, the cropped video dataset is split into three sets – training set, validation set and testing set in the 80:10:10 ratio. Three corresponding instances of data loader are created for training, testing and validation. The data loader extracts frames from the cropped videos and stacks them together to create the corresponding tensor of the video. The stacked tensor and label of the video is loaded in the model. Before loading the data in the model, the frames are cropped again to suit the input requirements of our models:

- LSTM – 112x112
- EfficientNet B1 – 240x240
- EfficientNet B2 – 260x260
- EfficientNet B3 – 300x300
- EfficientNet B4 – 380x380

For both EfficientNet and LSTM, we train our model on 20 epochs with a learning rate of 0.00001. We use the Adam optimizer for reducing the loss and modifying the weights during the learning process. To avoid

overfitting, a weight decay of 0.00001 is used. Cross Entropy Loss is used to measure the loss after every epoch. For training, we use a mini-batch size of 4. After training and validation is concluded, we save the model and test it on our testing dataset. The training and testing flow can be visualised using Fig. 4.

D. Hyper parameter tuning for EfficientNet

For all four EfficientNet models, we perform hyper parameter tuning at the fully connected layers. Each EfficientNet model has two variations – one with three fully connected layers and the second with two fully connected layers. The number of input and output neurons for the first variation where the left hand term denotes input neurons and right hand term denotes output neurons:

- Layer 1: (in_features, 512)
- Layer 2: (512, 128)
- Layer 3: (128,2)

Where in_features represents the number of features output by the CNN layers of the EfficientNet model used. The number of input and output neurons for the second variation are:

- Layer 1: (in_features, 512)
- Layer 2: (512,2)

E. Hyper parameter tuning for LSTM

For LSTM, we perform hyper parameter tuning on the sequence length. The sequence length refers to the number of frames passed to the model. Our hypothesis is that more the number of frames passed to the model, greater the accuracy of the model. Hence, we train our model on two different sequence lengths – 40 and 50.

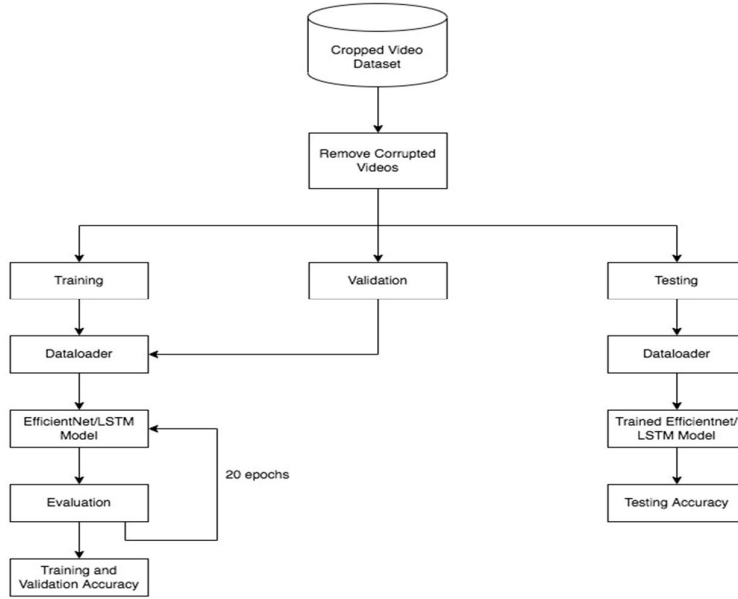


Figure 4. Training and Testing Flow

V. RESULTS

Table I shows the results obtained for EfficientNet models. Highest accuracy was obtained for EfficientNet B2 with two fully connected layers. We also observe that each EfficientNet model with two fully connected layers gives a higher accuracy than the corresponding model with three fully connected layers. Table II shows the results obtained for LSTM. Highest accuracy was obtained for LSTM model with a sequence length of 50. This confirms our hypothesis that the accuracy increases when the sequence length is higher. Both EfficientNet and LSTM show fairly good accuracies for deepfake detection. As LSTM has a higher accuracy than EfficientNet, it can be preferred for the task of deepfake detection. One possible explanation for this is that RNNs are better suited for analyzing sequences, which is why LSTM models were able to predict more accurately than EfficientNet.

TABLE I. EFFICIENT NET RESULTS

EfficientNet Model	Number of fully connected layers	Training loss	Validation loss	Testing Accuracy
B1	3	0.000288	0.002155	67.77%
	2	0.000272	0.002164	70.00%
B2	3	0.000297	0.002052	74.44%
	2	0.000274	0.002056	75.55%
B3	3	0.000295	0.002371	67.03%
	2	0.000268	0.002003	71.48%
B4	3	0.000288	0.002064	70.37%
	2	0.000271	0.002282	72.96%

TABLE II. LSTM RESULTS

Sequence Length	Training Accuracy	Validation Accuracy	Testing Accuracy
40	93.46%	74.34%	76.66%
50	97.12%	76.20%	80.37%

VI. CONCLUSION AND FUTURE SCOPE

With the growing advances in technology, creation and manipulation of digital data is easier than earlier times. Developing robust and accurate tools to detect such manipulated media is crucial to prevent them from negatively affecting the society. In this paper, we demonstrate the effectiveness of CNN and RNN models for deepfake detection. We also conclude that RNNs like LSTM are better suited for detecting deepfake videos accurately. There is scope for further improvement in the accuracy of our LSTM model. Currently, we have trained our model on a subset of three different datasets. The model can become more accurate and robust if trained on all the videos of these three datasets. Another promising line of research is to perform hyper parameter tuning for LSTM by increasing the sequence length to values greater than 100.

REFERENCES

- [1] "Deepfakes Web," [Online]. Available: <https://deepfakesweb.com/>. [Accessed 15 May 2022].
- [2] "Face App," [Online]. Available: <https://www.faceapp.com/>. [Accessed 15 May 2022].
- [3] "Wombo," [Online]. Available: <https://www.wombo.ai/>. [Accessed 15 May 2022].
- [4] Y. Pashaeva, "Scammers Are Using Deepfake Videos Now," Slate, 13 September 2021. [Online]. Available: <https://slate.com/technology/2021/09/deepfake-video-scams.html>. [Accessed 15 May 2022].
- [5] K. Fagan, "A video that appeared to show Obama calling Trump a 'dipsh-t' is a warning about a disturbing new trend called 'deepfakes'," Business Insider, 18 April 2018. [Online]. Available: <https://www.businessinsider.in/tech/a-video-that-appeared-to-show-obama-calling-trump-a-dipsh-t-is-a-warning-about-a-disturbing-new-trend-called-deepfakes/articleshow/63807263.cms>. [Accessed 15 May 2022].
- [6] M. Grothaus, "Deepfake videos are the latest cruel form of school bullying – parents and teachers must watch out," iNews UK, 9 November 2021. [Online]. Available: <https://inews.co.uk/news/technology/deepfake-videos-school-bullying-cyberbullying-ai-apps-parents-teachers-children-1290664>. [Accessed 15 May 2022].
- [7] B. S. Samuel, "A guy made a deepfake app to turn photos of women into nudes. It didn't go well.," Vox, 27 Jun 2019. [Online]. Available: <https://www.vox.com/2019/6/27/18761639/ai-deepfake-deepnude-app-nude-women-porn>. [Accessed 10 May 2022].
- [8] Crescendo, "YouTube," 16 April 2020. [Online]. Available: <https://www.youtube.com/watch?v=xkqfIKC64IM>. [Accessed 02 April 2022].
- [9] "Digital war: How Russia is using deep fakes in Ukraine for propaganda," Business Today, 02 March 2022. [Online]. Available: <https://www.businesstoday.in/latest/world/story/digital-war-how-russia-is-using-deep-fakes-in-ukraine-for-propaganda-324531-2022-03-02>. [Accessed 15 May 2022].
- [10] N. Bonettini, E. D. Cannas, S. Mandelli, L. Bondi, P. Bestagini and S. Tubaro, "Video Face Manipulation Detection Through Ensemble of CNNs," in 2020 25th International Conference on Pattern Recognition (ICPR), 2021.
- [11] A. Rossler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies and M. Nießner, "FaceForensics++: Learning to Detect Manipulated Facial Images," in IEEE/CVF International Conference on Computer Vision, 2019.
- [12] B. Dolhansky, J. Bitton, B. Pflaum, J. Lu, R. Howes, M. Wang and C. C. Ferrer, "The DeepFake Detection Challenge (DFDC) Dataset," in arXiv 10.48550/ARXIV.2006.07397, 2020.
- [13] Karandikar, V. Deshpande, S. Singh, S. Nagbhidkar and S. Agarwal, "Deepfake Video Detection Using Convolutional Neural Network," International Journal of Advanced Trends in Computer Science and Engineering, vol. 9, 2020.
- [14] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," in arXiv 10.48550/ARXIV.1409.1556, 2014.

- [15] Y. Li, X. Yang, P. Sun, H. Qi and S. Lyu, "Celeb-DF: A Large-scale Challenging Dataset for DeepFake Forensics," in IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020.
- [16] D. Güera and E. J. Delp, "Deepfake Video Detection Using Recurrent Neural Networks," in 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), 2018.
- [17] A. Jadhav, A. Patange, J. Patel, H. Patil and M. Mahajan, "Deepfake Video Detection Using Neural Networks," International Journal for Scientific Research and Development, vol. 8, no. 1, 2020.
- [18] Chintha, B. Thai, S. J. Sohrawardi, K. Bhatt, A. Hickerson, M. Wright and R. Ptucha, "Recurrent Convolutional Structures for audio spoof and video spoof detection," IEEE Journal of Selected Topics in Signal Processing, 2020.
- [19] A. Pokroy and A. D. Egorov, "EfficientNets for DeepFake Detection: Comparison of Pretrained Models," in IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering, 2021 .
- [20] M. Tan and Q. V. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," in International Conference on Machine Learning, 2019.