Multimedia Tools and Applications (2023) 82:20101–20118
https://doi.org/10.1007/s11042-022-14307-8

Improvised detection of deepfakes from visual inputs using light weight deep ensemble model

Saroj Kumar Panda [1,2] · Tausif Diwan [1] · Omprakash G. Kakde [1] · Jitendra V. Tembhurne [1]

Received: 28 June 2022 / Revised: 3 October 2022 / Accepted: 10 December 2022 /
Published online: 23 December 2022

Abstract
Due to rapid growing image and video technology, synthesis or fabrication of the visual contents by replacing the original person of the visual contents with some other public figure has emerged as *Deepfake*. Mostly, the intention of such alterations is to spread propagandas, to create controversies, to defame a celebrity or public figure, or sometime just for fun. These deepfakes are then spread over the internet via social networking platforms such as Twitter, Facebook, etc. and the consequences of such spreads are very impactful in terms of embarrassment, legal actions, propagandas, and violence. The alterations are too realistic to detect the originality of the same. Lots of machine learning and deep learning models have been proposed for the detection of deepfakes but they report limited accuracy. Moreover, the model complexity is also high when we talk in terms of deep models. Herein, we propose light weight deep ensemble binary classification model utilizing pretrained convolutional neural networks for visual features' extraction and long short-term memory to extract the temporal features from the input frames after being preprocessed by OpenCV Haar cascade. The proposed model utilizes comparatively lesser frames and outperforms other state of the arts models on the well-known Celeb-DF-v2 dataset. We report 97% accuracy for an ensemble of VGG19 and BiLSTM

✉ Tausif Diwan
tdiwan@iiitn.ac.in

Saroj Kumar Panda
dtei20cse005@iiitn.ac.in

Omprakash G. Kakde
director@iiitn.ac.in

Jitendra V. Tembhurne
jtembhurne@iiitn.ac.in

[1] Department of Computer Science & Engineering, Indian Institute of Information Technology, Nagpur, India

[2] Tata Consultancy Services, Nagpur, India

and highly recommend the use of aforementioned ensemble for the deepfake detection and classification in case of balanced dataset.

Keywords Deepfakes · Deep learning · Convolution neural network · Long short-term memory · Ensemble

1 Introduction

Deepfake is a technique with which we can fabricate a fake or other person' image over the real image. Then, these deepfakes are spread over the internet. There are many advantages of advancement in technology, however, some of them can be easily utilized in a wrong way such as deepfakes creation and their spreading. There could be a large number of reasons behind creating and spreading deepfakes such as to spread propagandas, to create controversies, to defame a celebrity or public figure, or sometime just for fun. There could be various modalities in which deepfakes can be constructed such as visual, audio, textual, or multimodal. However, this research work only focuses on the visual modality of the deepfakes i.e., images or videos.

Deep learning models like auto-encoders and Generative Adversarial Networks (GANs) can be trained to synthesize these fake images. These networks are trained over a large set of images which specifically involve all kinds of facial expressions, thereby creating a model which can properly decode the face of the target onto the face of the person in the photo or video. This is very powerful because we can't identify the difference between a fake and a real image using traditional methods. These techniques are commonly used to target public figures such as politicians, movie stars, and other celebrities, etc. The first Deepfakes was made in 2017, where most of the targets belongs to popular celebrities. Their faces were superimposed over the bodies of porn stars. The problem with this technology is that, it can cause many threatening situations to the world peace. Deepfake models can be utilized to create fakes of politicians and make controversial fake statements which might lead to chaos in the world. Figure 1a shows the original images of famous personalities and Fig. 1b shows the Deepfake images generated by using synthesis and overlapping.

In Fig. 2, we see two images of different celebrities. We can design two encoder-decoder networks, one for each image and train the networks with the corresponding images. Once both the networks are trained, the encoder trained on image *A* is extracted and connected to the decoder trained on image *B*. When we input the image *A* to this mixed network, the resulting output is an overlapping of both image *A* and *B*. This is the standard way of synthesizing Deepfakes.

The identification of these deepfakes is very important as far as social harmony is concerned. The spread of such deepfakes creates humiliation, embarrassment, and guilt for the person in loop or for the public figures. Sometimes, it may lead to religious sentiment hurting that may lead to big chaos. Therefore, identification of such deepfakes is very important before its spreading. It becomes paramount to design an automated detection system which can correctly detect and classify the fake images or videos and refrain us from any conspiracies. Most of the regulatory bodies in various countries keep track of such creation and their spreading. It is important to assess the truthfulness of such online content also from an ethical point of view. Lots of website, firms, and companies have been established to identify such deepfakes. One such example is alt news that identify such types of incidents and report publicly.

Fig. 1 **a** Original Images of Celebrities **b** Synthesized Images of Celebrities considered as *Deepfakes*

There are various challenges in the detection of such deepfakes as these fake images or videos seems so realistic due to the advancement in the technology that it is becoming very difficult to differentiate between real and its fake counterpart. With the advent of machine learning and underlying algorithms, lots of models and techniques have been proposed in the recent time to detect and identify such deepfakes but these models also suffer from limited accuracy. In former models, identification of handcrafted features from artifacts and finding inconsistencies in the fake images were investigated. These methods include edge detection and recognizing vital parts of a face to identify fakes which didn't work well because of the complexity involved in the generation of Deepfakes itself. The features extracted from the artifacts were much simpler than the features of the generated fakes. So, the newer methods have been opted and the deep learning methods are applied to detect deepfakes.

Deep learning (DL) is a subdomain of Machine Learning (ML) that imitates the functioning of the human brain by recognizing patterns and features for the given data for further processing. Deep Learning can learn features and patterns without any human intervention

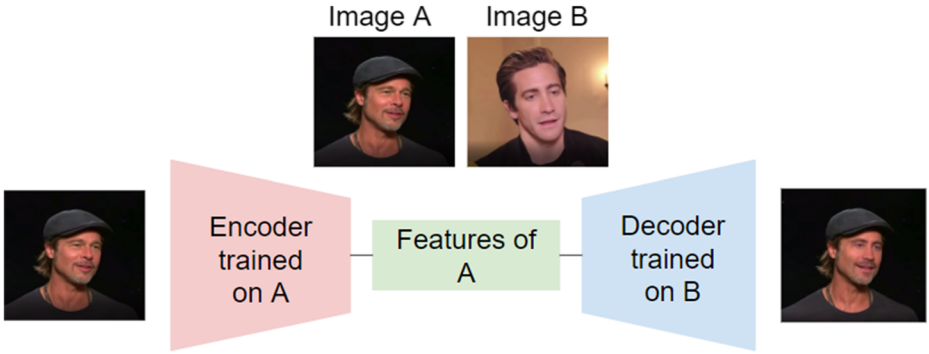

Fig. 2 A generic example of deepfake synthesis

in supervised or unsupervised manner for both structured and unstructured data. Deep learning hand-in-hand with the digital age has brought about an explosion of data around the world that can be utilized for making accurate decisions. Deep learning architectures such as deep neural networks (DNNs), graph neural networks (GNNs), recurrent neural networks (RNNs) and its architectural variants such as Long Short-Term Memory (LSTM), and convolutional neural networks (CNNs) can be applied in computer vision, image processing, natural language processing, bioinformatics, machine translation, and many more. Deep models require large amount of data to be processed to produce the considerable and significant improvement in the performance matrices. This will further increase the computational complexity of the model along with the processing time. The performance of such models completely relies on the efficient features' extraction.

Lots of techniques and models have been proposed to improve the performance but increases the model complexity too at the same time. Herein, we concentrate on the improvement in the model performance for the deepfakes detection with computational complexity keeping in mind. We propose light weight deep ensemble binary classification model utilizing pretrained CNN for visual features' extraction and LSTM to extract the temporal features from the input frames after being preprocessed by OpenCV Haar cascade. The proposed model utilizes comparatively lesser frames and outperforms other state of the arts models on the well-known Celeb-DF-v2 dataset. We report 97% accuracy for an ensemble of VGG19 and BiLSTM. We finally set the number of frames extracted from each video to five as we got the optimal results with this setting. Increasing number of frames per video beyond five is not producing the considerable gain in the model accuracy.

The major contributions of the proposed work can be summarized as follows:

1. We present a light weight deep ensemble model for the binary classification of the *deepfakes*.
2. With very fewer frames of an input video, the model is able to perform significantly better in comparison with the state-of-the-arts deep learning and machine learning models, producing a computationally inexpensive deep model with improved performance.
3. With the help of OpenCV' haar cascade classifier, the frames are preprocessed to extract the face objects from the visual frames and properly resized to provide standard sized input to the deep ensemble model. With intelligent usage of the processing techniques, deep models could outperform with handful number of input frames.

The organization of paper is as follows: Section 2 highlights the existing works and findings based on Deepfake detection using various machine learning and deep learning methods. The proposed system for Deepfakes detection with the underlying dataset is elaborated in Section 3. In Section 4, we demonstrated the results and discussion on experimentations performed using proposed ensemble models. Section 5 presents the conclusion and propose some future research directions.

2 Literature review

In this section, we have presented the literature review based on the various techniques implemented for Deepfakes identification over the recent period. In [20], the research work

is performed to differentiate the real and fake products. There are two types of products namely consumable and non-consumable. Even, if the sales depend on the quality of the items, sometimes, products with less quality are manufactured and marketed to create good business. Here, they deal by examining real or fake beady packages (i.e., image, hologram, logo, brand, etc.) to establish genuineness. Normally, both the packages i.e., real and fake are examined under Video Spectral Comparator (VSC), essentially, this method is not enough to draw any fruitful conclusions. Progressively, artificial intelligence (AI) is applied to produce Deepfakes namely Face Swap – digitally swapping the face of one object with another that barely leaves any hints of being fake resulting in political pain, extorting somebody, etc. The fake forensics using Digital Media Forensics with Recurrent Neural Network (RNN) is presented in [8], the framework utilized is Convolutional Long Short-Term Memory (C-LSTM) which is basically comprises of two parts – 1) CNN is adopted for outline highlight extraction and 2) LSTM is applied for transient grouping investigation. Authors presented a time-based aware system to automatically detect fake videos.

A series of recent incidents led to the inspection of online "*fib"* i.e., direct manipulation of data or presentation of unchanged content in a misleading context. Change in digital image - copy-move and slicing that constitute "*deliberate manipulation*" and hand image repurposing called "*misleading context*". Due to the spread of *fib* (image or video), generated synthetically but seems to be realistic is a significant problem. Recurrent Convolution models [19] is adopted for temporal features' extraction from image streams and best strategy that combine variations in the models along with domain specific face preprocessing techniques is utilized. Specifically, the model is used to detect (Deep-Fake, Face-to-Face and Face-Reciprocation) tampered faces in video streams of FaceForensics++ dataset and an improvement of around 4.55% in the accuracy is recorded over existing methods.

In social and professional networking forums, millions of images are uploaded, out of which 40% to 50% are manipulated for good-humored or mostly harmful reasons. Image manipulation (face manipulation) is a serious issue since it is widely used as a lead in biometric for identification and authentication services, moreover, due to the advancement in deep learning, generating and manipulating realistic faces become easier. In [2], an overview of recent technological supports for face manipulation generation, recognition, detection and underlying databases are presented. There are several challenges that remain unaddressed which includes – generalized manipulation detectors, adversary aware face recognition systems, wearable manipulation detection, and large-scale databases which definitely require attention in future research. Subsequently, refinement in mobile camera and effortless reach to social media, sharing videos has become very convenient. Out of all, an improvement in machine learning and computer vision techniques eliminated the need of manual editing, it takes a video with 'target' as input and the output is another video with replaced target's faces in the provided video. Backbone of Deepfakes detection is neural networks trained on images to automatically map expressions from source to target, a fake video can obtain high level of realism. This wrapping leaves some artifacts i.e., resolution inconsistency between surrounding context and wrapped face area. Eventually, these artifacts can be utilized to detect digital misinformation, a CNN based system is proposed to distinguish the difference between real and fake images/videos [13] without the need of large dataset.

Due to increase in popularity of electronic products such as mobile phones and digital cameras, large amount of data (images and videos) has been created and approximately 2

billion images are added to internet every day. In [26], facial fraud detection system is developed by using CNN and image segmentation. The system requires lesser training parameters and improved accuracy is reported with robustness. To do so, face area is extracted from the video frames followed by alignment and cropping, and dividing the blocks of pre-processed face area and training is performed using CNN. Finally, hard voting determines the label of image. In [3], a new forensic technique is introduced to differentiate between fake and real images within the videos. The adoption of optical flow fields is utilized for identifying inner frame dissimilarities to extract the features using CNN. Initially, motion vectors have been considered as 3-channel images and then input fed to CNN to obtained the results. FaceForensics++ dataset is utilized for the experimentation and promising performances were achieved.

Recently, heart rate of false videos is used to differentiate original and fake videos. In [7], the heart rate of original videos and trained state-of-the-art Neural Ordinary Differential Equation (ODE) model is applied to create fake videos using software. The system comprises: creating Deepfake dataset, extracting heart rate from facial videos, Neural-ODE training using heart rate from original videos, and applying trained Neural-ODE for predicting heart rates of Deepfake videos. The analysis shows that the average loss for first ten videos is 0.010926, average loss for ten donor videos is 0.010040 wherein trained Neural-ODE is able to predict the heart rate of Deepfake videos. In [1], a biometric-based forensic technique is designed for detecting face-swap Deepfakes. A static biometric – facial recognition and behavioral biometric – movement of head and facial expression are adopted. Here, 'Spatiotemporal behavior' is captured by Behavior-Net and VGG extracts facial identity and after ensuring that both the results are not tangled further processing is performed to detect deepfakes. An analysis of ways for visual media integrity verification is discussed in [23]. The analysis shows the limitations of the existing forensics tools and suggestions for further research in the aforementioned directions. At present, tools are being developed in large scale to break the norm of Deepfakes and to protect people from reaching fault information.

Lots of recent advancements have been proposed in classification and segmentation with the help of deep models [17, 24]. Deep learning is not complete black box now a days, each and every operation should be explainable and interpretable. Various recent approaches and advancements in the direction of explainable deep learning have been summarized in the form of a survey / review [4]. Various machine learning and deep learning-based models and approaches have been proposed by the research community for the *deepfakes* detection and classification [6]. Most of the deep learning models require a huge amount of data to be processed for getting the improved results. So, we obtain the computationally expensive deep models for the deepfakes detection, moreover, limited accuracy is obtained even by applying these latest deep models. In our proposed method, the main goal is to present a computationally inexpensive deep model that leverages the lesser frames and produces comparable or improved performance.

3 Methods and materials

Rapid mounting in artificial intelligence, deep learning techniques alongside refined mobile camera followed by easy reach to applications that support modification, and effortless access to internet, social media, sharing portals, etc. have made creating and spreading of fabricated

visual misinformation very easy. Fake videos/images are undoubtedly very interesting and can be utilized as social weapons to target public figures. There are various methods of doing so, however, we only target the most utilized and popular method known as Face-Swaps. It is digitally swapping one face object of a visual input by another face object with no hints of being faked is one of the hot choices for *deepfakes*. With enabling machine learning and deep learning technologies, we need an automated computationally inexpensive mechanism to detect, identify, and prevent such spreading of *deepfakes* through various social-media platforms.

We propose a deep learning-based light weight ensemble model for Deepfakes detection comprising two modules – Firstly, a CNN [25] is utilized to extract the visual features of the frames of an input video. Secondly, a sequence model is employed to extract and analyze the temporal and sequential features from the output of CNN.

We extracted only '*n*' frames which are distributed across the entire video in sequential order. Then, these frames are given as input to a CNN, which generates the feature sets for each of these frames. These features are then fed to the RNN (specifically, LSTM / BiLSTM) in sequential fashion and predict the final classification result using a dense network. Figure 3 shows the proposed schematic model for the detection of Deepfakes, the ensemble of CNN and RNN is utilized and random frames are selected over the time for processing in the binary classification model. This section covers the dataset preprocessing and preparation for model training and evaluation, architectural aspects of the pretrained CNNs for visual features' extraction using two variants of VGG i.e., VGG16 and VGG19, architectural details along with gating mechanisms of the utilized sequence models specially LSTM and BiLSTM for temporal features' extraction, and ensemble model training along with tuning of various hyperparameters.

3.1 Dataset: preprocessing and preparation

The dataset Celeb-DF-v2 [14] is used for training the proposed light weight ensemble deep model, is a vast collection of 6528 original and synthesized videos made by DeepFake Forensics using a refined synthesis algorithm that reduces the visual artifacts, which were considered as a bottleneck in the predecessor datasets. Currently, this dataset has the highest benchmark scores in comparison to all the available DeepFake datasets. It is more robust, quite stable, highly imbalanced, and designed for binary classifier systems. In this dataset, the real videos and corresponding number of frames are 590 and 225.4 k respectively. However, the DeepFake or synthesized video and corresponding number of frames are 5639 and 2116.8 k

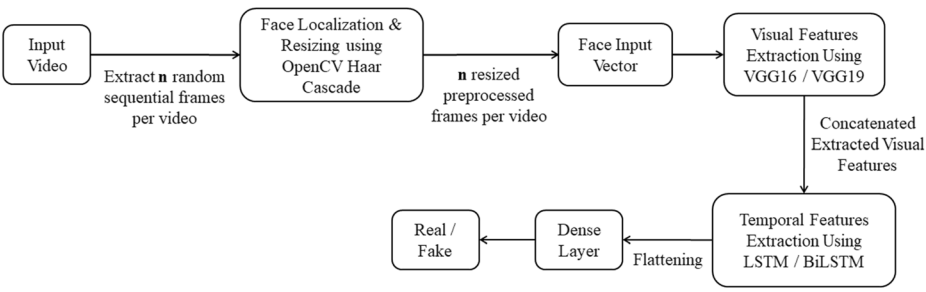

Fig. 3 Proposed light weight deep ensemble model for deepfake detection

respectively. As it seems to be a highly imbalanced dataset, we augment the real videos and corresponding frames to make the dataset almost balanced. We utilized the Keras ImageDataGenerator for augmenting the same and various underlying augmentation techniques are employed to achieve the goal.

The dataset is preprocessed to extract the faces from all the video frames. We utilized classifier in OpenCV i.e., Haar Cascade for these face cropping and extraction. This OpenCV classifier is highly optimized library that focuses on real-time application and includes both classic and state-of-the-art machine learning and computer vision algorithms for face cropping and extraction tasks. Haar Cascade classifier is an object detection algorithm – a cascade function is trained using lots of positive and negative images in this classifier, thus, helps to detect the faces in the video frames and cropping the face images accordingly. The schematic of the Haar Cascade classifier is demonstrated in Fig. 4. We resized the cropped images to 224 × 224 as per the standard input size, to be provided as input to our proposed lightweight ensemble deep model. The resulting face images are utilized as the dataset for training and evaluation of the proposed model.

Fig. 4 Localized face identification and cropping using OpenCV's Haar cascades

3.2 Visual features' extraction

After preprocessing the dataset using Haar Cascaded classifier, CNN based architecture is adopted for the visual features' extraction from the processed images. CNNs are heavily employed deep learning model for automatic features' extraction and representation of visual features in computer vision. Herein, we utilize the pretrained CNN, especially VGG and variants for the visual features' extraction. Convolution and pooling layer are the most important building block of any CNN model, transforming an input volume to an output volume. Convolutional layer is applied for learning various kernels; however, pooling is responsible for performing extraction of important or dominating features via down sampling. Generally, Rectified Linear Unit (ReLU) is preferred as a nonlinear activation function in the convolutional layer.

VGGs are pretrained CNN model trained on a large dataset, especially employed for an object detection and classification model. VGG was conceived out of the need to lessen the number of boundaries in the conv-layers and to reduce model training duration. This pretrained CNN is used quite too often due to its uniform architectural style. Due to its simplistic and consistent architectural characteristics, we prefer VGG over other pretrained CNNs such as AlexNet, ResNet, etc. The important point to note here is that all the conv kernels are of size 3 × 3 and max pool kernels are of size 2 × 2 with a stride of two, having overall 140 million parameters.

VGG comes in two flavors viz. VGG16 [5, 21] and VGG19 [5, 21] with 16 and 19 layers respectively, drawing a slight architectural difference, as presented in Table 1. Conv3-y indicates that we are applying 3 × 3 receptive fields and number of filters applied is y. However, FC – z indicates that it is a fully connected or dense layer with z number of neurons in the layer. By fully connected or dense layer, we mean that neuron in this layer is connected to all the neurons of the previous layer. We employ both in our experimentation for the comparative illustration of the proposed model's results. These extracted features are provided as input to the next sequence modeling layer in our proposed model i.e., LSTM/BiLSTM layer for sequential or contextual features' extraction.

3.3 Sequential and contextual features' extraction

LSTM is an architectural variant of RNN belongs to sequential deep learning architectures, designed to deal with vanishing gradient problem, generally occurred in the plain RNN. Unlike, standard feedforward neural networks, LSTM has feedback connections. LSTM applies to the tasks such as unsegmented, connected handwriting recognition, speech recognition and anomaly detection in network traffic or IDSs (i.e., intrusion detection systems).

A common LSTM unit is composed of a cell consisting of various gates: an input gate, an output gate and a forget gate. The forget gate processes the input of current time step and hidden output from the previous cell. Cell state gets manipulated due to the various gate operations in LSTM cell i.e., information is added, retained, or subsidized. Cell state gets modified by taking into account the input gate, forget gate and previous cell state. Output gate is responsible for the generation of the hidden state that shall be utilized in the next LSTM cell. The various equations pertaining to the aforementioned gates of the LSTM cell are presented in Eqs. 1–6, subscripts of the weight matrices and biases indicate the gate's initial. Moreover, LSTM cell is illustrated using Fig. 5. LSTM networks are well-suited to classifying,

Table 1 Architectural differences between VGG16 and VGG19

VGG16	VGG19
ConvNet Configuration	
16 layers	19 layers
Input layer (224×224 RGB image)	
Conv3–64	Conv3–64
Conv3–64	Conv3–64
Maxpool	Maxpool
Conv3–128	Conv3–128
Conv3–128	Conv3–128
Maxpool	Maxpool
Conv3–256	Conv3–256
Conv3–256	Conv3–256
Conv3–256	Conv3–256
	Conv3–256
Maxpool	Maxpool
Conv3–512	Conv3–512
Conv3–512	Conv3–512
Conv3–512	Conv3–512
	Conv3–512
Maxpool	Maxpool
Conv3–512	Conv3–512
Conv3–512	Conv3–512
Conv3–512	Conv3–512
	Conv3–512
Maxpool	Maxpool
FC - 4096	
FC - 4096	
FC - 1000	
Soft-max	

processing, and making predictions based on time series data since there can be lags of unknown duration between important events in a time series.

$$f_t = \sigma\left(W_f \times [x_t, h_{t-1}] + b_f\right) \tag{1}$$

$$i_t = \sigma\left(W_f \times [x_t, h_{t-1}] + b_i\right) \tag{2}$$

$$\widetilde{C}_t = \tanh(W_c \times [x_t, h_{t-1}] + b_c) \tag{3}$$

$$C_t = C_{t-1} \times f_t + \widetilde{C}_t \times i_t \tag{4}$$

$$o_t = \sigma(W_o \times [x_t, h_{t-1}] + b_o) \tag{5}$$

$$h_t = C_t \times \tanh(o_t) \tag{6}$$

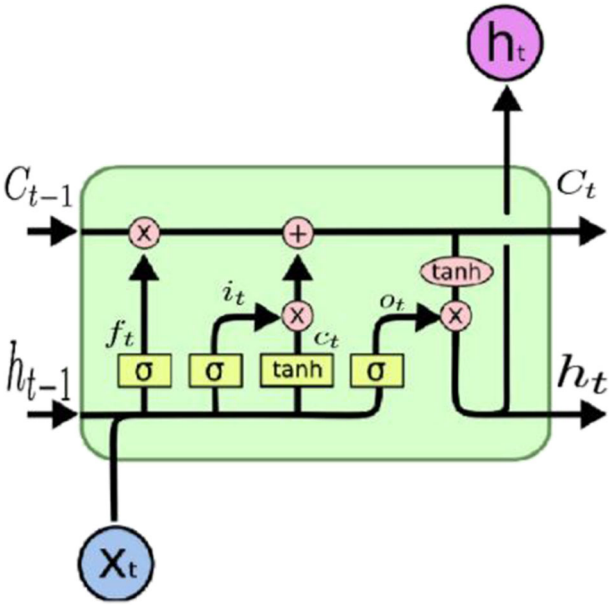

Fig. 5 Gating mechanism of an LSTM cell

Context size and direction play an important role in the sequential or temporal features' extraction. With the advancements in the architectural design, bidirectional sequence models are also proposed to better extract the context. Herein, we employ BiLSTM in addition to the plain LSTM model for the Deepfake classification. The schematic of traditional LSTM and BiLSTM is illustrated in the Fig. 6 wherein we can visualize the information flows in the both the directions i.e., in forward and in backward direction in case of BiLSTM. Hence, BiLSTM allows us to obtain both forward and backward temporal information over the traditional LSTM. We run the inputs in forward and backward directions so that the network is able to store features from past and future both. Generally, BiLSTM perform the task of understanding the context of the input sequence, much better than LSTMs.

3.4 Model training and hyperparameters tuning

We explore two variants of VGG i.e., VGG16 or VGG19 for the visual features' extraction. However, LSTM or BiLSTM are employed for the contextual features' extraction from the

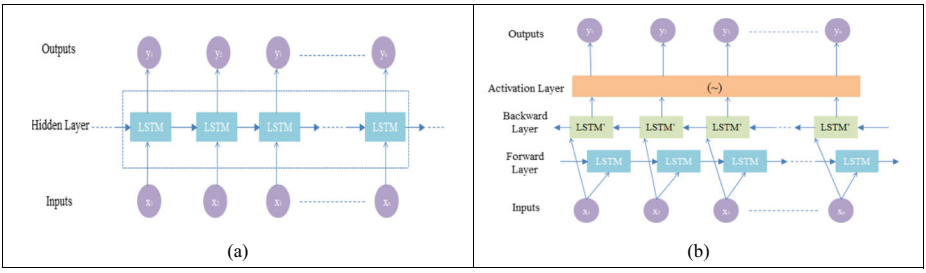

Fig. 6 LSTMs models: (**a**) Standard LSTM network, (**b**) Bi-directional LSTM network

extracted visual features through any one of the two VGG variants. Conclusively, VGG16 + LSTM, VGG16 + BiLSTM, VGG19 + LSTM, and VGG19 + BiLSTM are the four ensembles, experimented for the deepfake detection and classification. For the final classification, SoftMax layer is added to detect the probabilities of real or synthesized video. For avoiding model overfitting, we employ a dropout of 0.3 for each layer in the proposed deep ensemble model.

In the proposed system, dataset is pre-processed using OpenCV and Haar Cascade classifier. Thereafter, 'n' random frames are extracted that are distributed across the span of a video in a sequential order and given as an input to the pretrained CNN (i.e., VGG16 or VGG19). We partitioned the Celeb-DF-v2 dataset in an uneven split of 80%, 10% and 10% to create three disjoint sets i.e., train, validation, and test set respectively. We also ensure that proportionate contribution from real or synthesized videos in these set formations. Corresponding splits are then merged to form the entire train, validation, and test data. Here, we are not processing all the frames of a video as it increases the model convergence time as well as the memory requirement for model training. Hence, we randomly select the frames from the video and arrange in the sequence of appearance. We also experiment with varying number of frames in the deepfake classification and decided to used five random frames on the basis of obtained results.

Logistic loss or Binary cross entropy loss is used for computing the model loss and can be described using Eq. 7 for a minibatch. y_i and $p(y_i)$ indicate the ground truth label and the predicted output by the model respectively. N indicates the number of samples in a mini-batch in gradient descent algorithm. As suggested by the following equations, predicted outcome should be close to unity for minimizing the loss if the ground truth label is true. On the other side, predicted outcome by the model should be close to zero for minimization of the loss if the ground truth label of that instance is false. In case of a batch, we take the average of log loss of all the samples in a minibatch.

$$H_p(q) = -\frac{1}{N}\sum_{i=1}^{N} y_i \times \log(p(y_i)) + (1-y_i) \times \log(1-p(y_i)) \tag{7}$$

Backpropagation and gradient descent algorithm are employed for minimizing the logistic loss [18]. By adjusting the appropriate learning rate, parameters of deep ensemble networks are updated for minimizing loss or cost function. Instead of applying complete batch in an iteration, minibatch gradient descent is utilized by considering an appropriate minibatch size. The advantages of using smaller minibatch size are frequent updating the parameters space, efficient gradient direction, and faster model convergence. We employ a learning rate of 1e-3 and a decay of 1e-4 for training the ensemble model.

4 Results and discussion

This section reports and discusses the experimental results of proposed model in terms of overall predictive accuracy. All the experiments are accomplished using Deep Learning library on Intel® Core™ i7-8550U CPU @ 1.80GHz processor with 8GB RAM enabled with NVIDIA GeForce 940MX graphic card. Moreover, python 3.7 is utilized for the modelling and programming purpose. To detect Deepfakes, it's generally not the case that we find a video which is manipulated only in certain parts. A video is composed of frames, our

Table 2 Comparative illustration of all four lightweight ensemble deep models for the deepfakes detection

Model	Training Accuracy (%)	Validation Accuracy (%)	Test Accuracy (%)
VGG16+LSTM	96.17	96.14	96.12
VGG19+LSTM	95.89	95.88	95.86
VGG16+BiLSTM	96.27	96.25	96.24
VGG19+BiLSTM	96.6	96.4	96.4

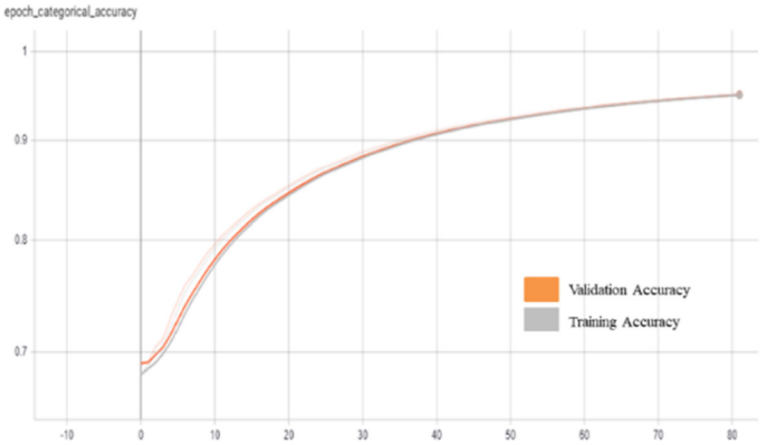

Fig. 7 Accuracy curves for VGG16 + LSTM ensemble

assumption for the deepfake detection lies on the fact that all the frames of the underlying video will be manipulated to make it deepfake if the video if considered as deepfake otherwise it would be a real video without any sort of manipulation. So, we extract the frames from different partitions of the video randomly and pass through our models for the possible tempering determination. This way of extraction is done for all the data splits sequentially. We presented the performance of all four ensemble models in terms of training, validation, and

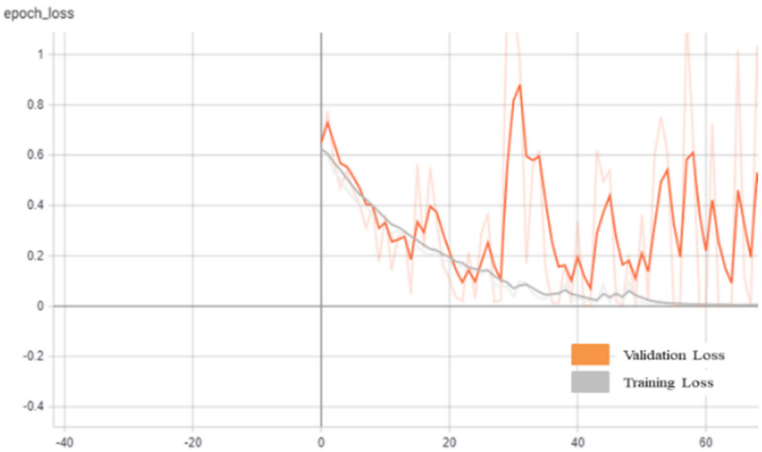

Fig. 8 Loss curves for VGG16 + LSTM ensemble

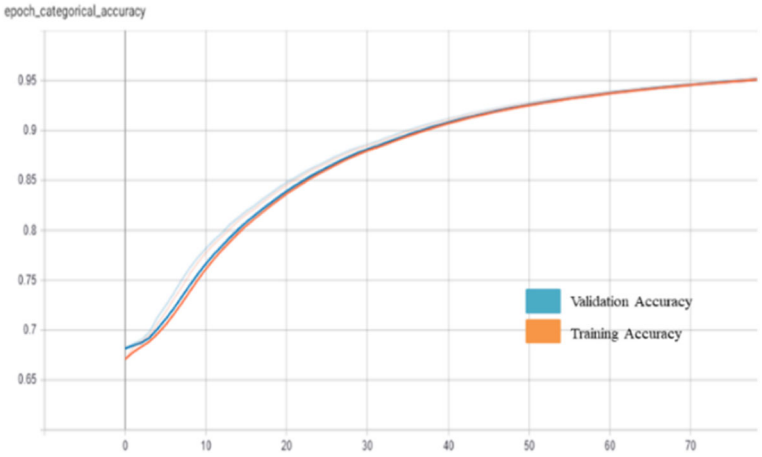

Fig. 9 Accuracy curves for VGG19+ LSTM ensemble

test accuracy in the Table 2, for number of considered frames as five. As we can see from the outcomes, our models are trying to predict whether the video has traces of manipulations or not. So, the comparison shows that the higher accuracy reported by different ensemble models wherein highest test accuracy of 96.4% is achieved by VGG19 + BiLSTM. Figures 7, 8, 9, 10, 11, 12, 13 and 14 demonstrate the training-validation accuracy curves and training validation loss curves for various ensembles for the deepfake detection and classification. We observe a significant improvement in the validation as well as test accuracy using VGG19 and BiLSTM. In all the experimentations, accuracy is the only metric used to demonstrate the model performance as all the experimentations are performed on the balanced version of the Celeb-DF-v2. It can be defined as the ratio of true predictions by the model to the total number of samples under experimentation.

We could achieve the considerable accuracies from all the four proposed lightweight deep ensembles. Moreover, the model is neither overfitting to the validation data nor to the test data. The probable reason behind this is the utilization of around 30% dropout in our model

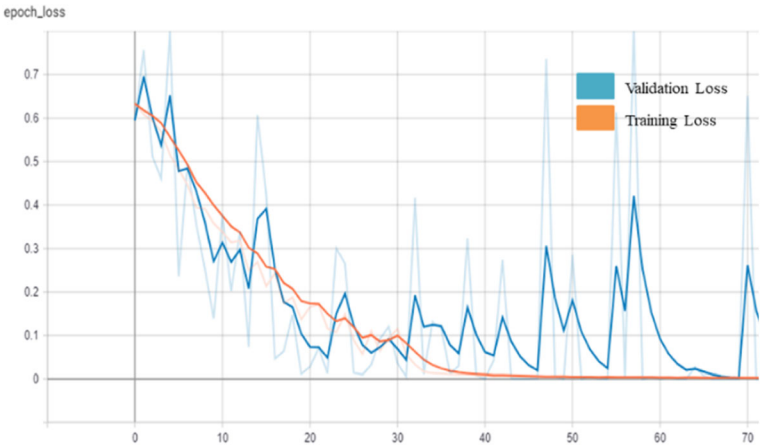

Fig. 10 Loss curves for VGG19 + LSTM ensemble

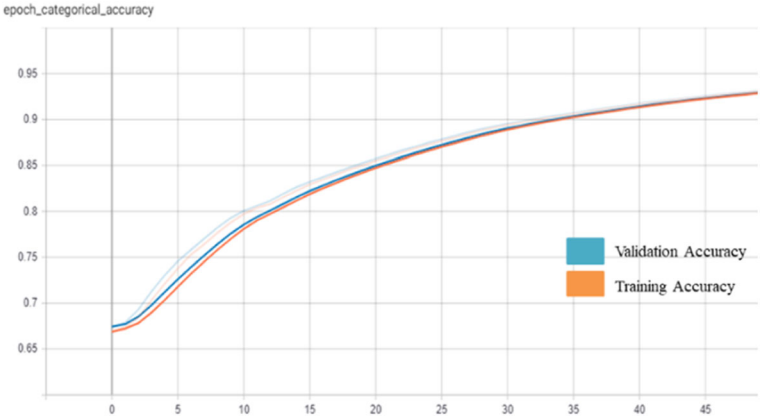

Fig. 11 Accuracy curves for VGG16 + BiLSTM ensemble

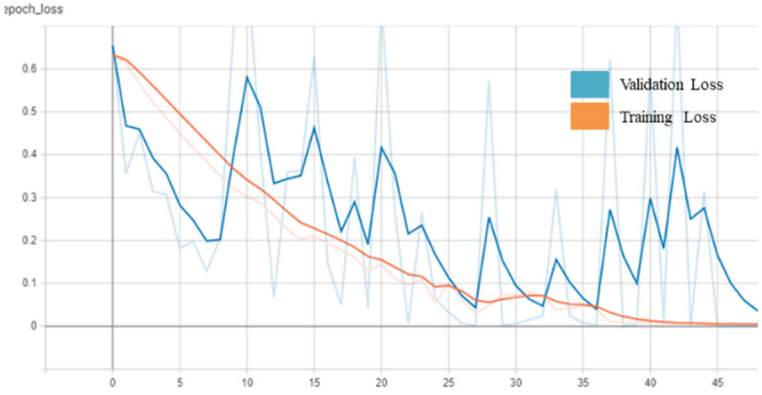

Fig. 12 Loss curves for VGG16 + BiLSTM ensemble

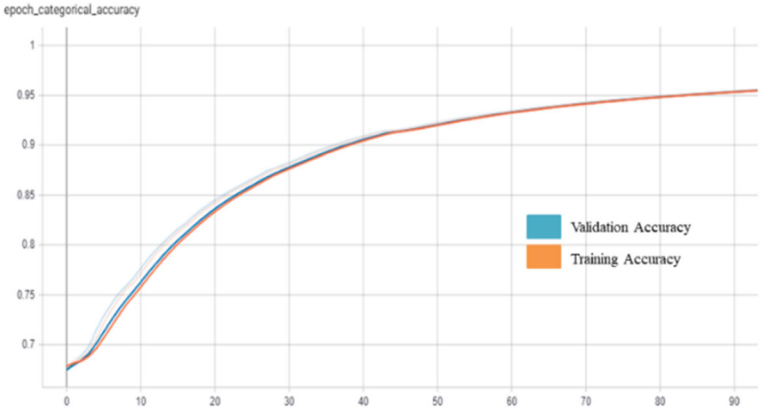

Fig. 13 Accuracy curves for VGG19 + BiLSTM ensemble

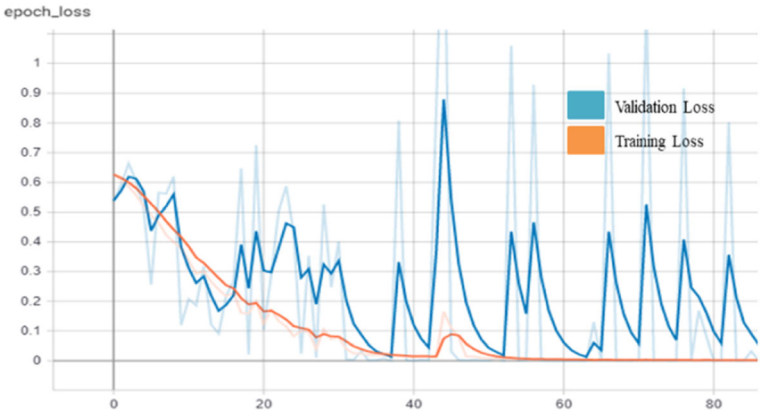

Fig. 14 Loss curves for VGG19 + BiLSTM ensemble

pipeline. Generalization ability of the model to the data which it has not seen before have been significantly improved. Moreover, we could claim that model generalization ability is not getting deteriorating by no more than 0.2% which is quite noticeable. Among all these four ensembles, VGG19 and BiLSTM combination is performing relatively better on both validation and test data. The probable reason behind this is the bidirectional contextual features' extraction is producing better in the course of identification of features pertaining to *deepfakes* along with the VGG19 variant of the pretrained CNN model. We have plotted learning curves for all four proposed ensembles by considering the training and validation set only just to demonstrate that both the sets are performing approximately to the similar line. Intentionally, we have not plotted the test accuracy as these curves were getting completely overlapped. However, we could observe the little bit fluctuations in loss curves corresponding to the validation set. In all these four ensembles, the number of epochs required to train the models is varying from 40 to 80.

We evaluated the performance of our proposed deep models with state-of-the-arts for Deepfake detection and classification tasks, as presented in Table 3. The outcomes of the comparisons shows that we achieved the benchmarking accuracy. The main achievement of this research is that we are able to predict the manipulation of video much faster than the

Table 3 Comparison with existing models and proposed model

Dataset	Method	Classifier	Test Accuracy
Face2Face	Optical flow	CNN / VGG16	81.61%
Custom	Conv + LSTM	CNN	97.1%
HOHA [12],	Conv-LSTM [9]	CNN	96.7%
Celeb-DF	Conv + LSTM [11]	CNN	96.46%
DFDC	Conv + LSTM [11]	CNN	96.99%
Celeb-DF	DenseNet169+Rayleigh blur [15]	NA	60.1%
DFDC	InceptionV3 [22]	NA	92.07
DFDC	XceptionNet [22]	NA	86.62
DFDC	EfficientNet-B0 [22]	CNN	96.24
DFDC	EfficientNet-B1 [22]	CNN	97.63
Own dataset (36,302 images)	CNN [16]	NA	95%
PGGAN	DeepFD [10]	GAN discriminator	94.7%
Celeb-DF-v2	**VGG19+BiLSTM**	**VGG19**	**96.4%**

existing models. The proposed model detects the fault just within 1 second of the entire part of video (i.e., videos frame rate is 30 fps) with a significant good accuracy of 96.4%.

5 Conclusions

We presented a deep learning-based light weight ensemble model for the binary classification of the deepfakes in visual inputs. CNN based pretrained model i.e., VGG16 and its architectural variant VGG19 are utilized for the visual features' extraction. LSTM and its architectural successor BiLSTM are experimented for the efficient sequence modelling of the extracted features from the CNN based pretrained classifier. The main contribution of the work is to efficiently predict the *deepfakes* with the help of very lesser frames of an input video. The visual frames of the input video are preprocessed with the help of OpenCV classifier i.e., Haar cascade is employed for the image preprocessing such as extracting the facial object from the entire image and resizing the same as per the standard input size of the proposed light weight ensemble deep model. The novelty in our approach is not only the usage of ensemble of deep models but also randomly picking few frames for providing as input to the pretrained CNN. With the help of five random frames only, the light weight ensemble of VGG19 and BiLSTM is outperforming several state-of-the-arts models for the *deepfakes* detection. Its generalization ability is around 96.4% on Celeb-DF-v2 dataset without any overfitting issues. Therefore, we strongly recommend the utilization of simpler pretrained deep models for visual features' extraction and bidirectional sequence model for processing the temporal dependencies from the extracted features by pretrained CNN in the course of *deepfakes* detection. With the help of handful number of frames and proper preprocessing techniques, ensemble of pretrained CNN and a sequence model achieve the significant performance. Further increasing the number of frames would unable to draw any considerable improvements in the model performance.

Herein, we experimented for a binary classifier of *deepfakes* detection with the help of balanced dataset. In future, we shall experiment the similar ensembles for imbalanced dataset and shall explore various other performance matrices. Moreover, cross domain transfer learning for the deepfakes detection across various similar datasets is also considered as the future scope of the proposed work.

Acknowledgments The authors, thanks to all the anonymous reviewers of Multimedia Tools and Applications Journal for their constructive remarks and fruitful suggestions to improve the manuscript.

Data availability The datasets generated during and/or analysed during the current study are available in the Celeb-DF (v2) repository, https://cse.buffalo.edu/~siweilyu/celeb-deepfakeforensics.html.

Declarations

Competing interests We do not have any conflict of interest related to the manuscript.

References

1. Agarwal S, Farid H, El-Gaaly T, Lim SN (2020) Detecting deep-fake videos from appearance and behavior. In 2020 IEEE international workshop on information forensics and security (WIFS), pp 1–6

2. Akhtar Z, Dasgupta D, Banerjee B (2019) Face authenticity: an overview of face manipulation generation detection and recognition. In Proceedings of International Conference of Communication and Information Processing (ICCIP)
3. Amerini I, Galteri L, Caldelli R, Del Bimbo A (2019) Deepfake video detection through optical flow based cnn. In Proceedings of the IEEE/CVF international conference on computer vision workshops
4. Bai X, Wang X, Liu X, Liu Q, Song J, Sebe N, Kim B (2021) Explainable deep learning for efficient and robust pattern recognition: a survey of recent developments. Pattern Recogn 120:108102
5. Chollet F (2021) Deep learning with Python, Simon and Schuster, 1st edition, pp 1–384
6. Deshmukh A, Wankhade SB (2021) Deepfake detection approaches using deep learning: a systematic review. Intelligent Computing and Networking 293–302. https://doi.org/10.1007/978-981-15-7421-4_27
7. Fernandes S, Raj S, Ortiz E, Vintila I, Salter M, Urosevic G, Jha S (2019) Predicting heart rate variations of deepfake videos using neural ode. In Proceedings of the IEEE/CVF international conference on computer vision workshops
8. Güera D, Delp EJ (2018) Deepfake video detection using recurrent neural networks. In 2018 15th IEEE international conference on advanced video and signal-based surveillance (AVSS), pp 1–6, IEEE
9. Güera D, Delp EJ (2018) Deepfake video detection using recurrent neural networks. In 2018 15th IEEE international conference on advanced video and signal based surveillance, pp 1–6
10. Hsu CC, Lee CY, Zhuang YX (2018) Learning to detect fake face images in the wild. In 2018 international symposium on computer consumer and control (IS3C), pp 388–391
11. Kaur S, Kumar P, Kumaraguru P (2020) Deepfakes: temporal sequential analysis to detect face-swapped video clips using convolutional long short-term memory. J Electron Imaging 29(3):033013
12. Laptev I, Marszalek M, Schmid C, Rozenfeld B (2008) Learning realistic human actions from movies. In 2008 IEEE Conference on Computer Vision and Pattern Recognition, pp 1–8
13. Li Y, Lyu S (2018) Exposing deepfake videos by detecting face warping artifacts. ArXiv preprint arXiv: 181100656
14. Li Y, Yang X, Sun P Qi H, Lyu S (2020) Celeb-df: a large-scale challenging dataset for deepfake forensics. In proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 3207–3216
15. Maksutov AA, Morozov VO, Lavrenov AA, Smirnov AS (2020) Methods of deepfake detection based on machine learning. In 2020 IEEE conference of russian young researchers in electrical and electronic engineering (EIConRus), pp 408–411
16. Marra F, Gragnaniello D, Cozzolino D, Verdoliva L (2018) Detection of Gan-generated fake images over social networks. In 2018 IEEE conference on multimedia information processing and retrieval (MIPR), pp 384–389
17. Ning X, Tian W, Yu Z, Li W, Bai X, Wang Y (2022) HCFNN: high-order coverage function neural network for image classification. Pattern Recogn 131:108873
18. Saad D (1998) Online algorithms and stochastic approximations. Online Learn 5(3):6
19. Sabir E, Cheng J, Jaiswal A, AbdAlmageed W, Masi I, Natarajan P (2019) Recurrent convolutional strategies for face manipulation detection in videos. Interfaces (GUI) 3(1):80–87
20. Saha B, Pal A, Pratihari HK (2017) Examination of genuine and fake images by histogram and edge detection method-a case report. J Forensic Investig 5(2):2
21. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. ArXiv preprint arXiv:14091556
22. Singh A, Saimbhi AS, Singh N, Mittal M (2020) DeepFake video detection: a time-distributed approach. SN Comput Sci 1(4):1–8
23. Verdoliva L (2020) Media forensics and deepfakes: an overview. IEEE J Sel Top Signal Process 14(5):910–932
24. Wang C, Ning X, Sun L, Zhang L, Li W, Bai X (2022) Learning discriminative features by covering local geometric space for point cloud analysis. IEEE Trans Geosci Remote Sens 60:1–15
25. Wu J (2020) Convolutional neural networks" LAMDA group
26. Yu CM, Chang CT, Ti YW (2019) Detecting deepfake-forged contents with separable convolutional neural network and image segmentation. ArXiv preprint arXiv:191212184