Deepfakes: temporal sequential analysis to detect face-swapped video clips using convolutional long short-term memory

Sawinder Kaur,[a,*] Parteek Kumar,[b] and Ponnurangam Kumaraguru[c]
[a]Thapar Institute of Engineering and Technology, Doctoral Research Lab-II, Department of Computer Science and Engineering, Patiala, Punjab, India
[b]Thapar Institute of Engineering and Technology, Department of Computer Science and Engineering, Patiala, Punjab, India
[c]Indraprastha Institute of Information Technology, Department of Computer Science and Engineering, New Delhi, Delhi, India

Abstract. Deepfake (a bag of "deep learning" and "fake") is a technique for human image synthesis based on artificial intelligence, i.e., to superimpose the existing (source) images or videos onto destination images or videos using neural networks (NNs). Deepfake enthusiasts have been using NNs to produce convincing face swaps. Deepfakes are a type of video or image forgery developed to spread misinformation, invade privacy, and mask the truth using advanced technologies such as trained algorithms, deep learning applications, and artificial intelligence. They have become a nuisance to social media users by publishing fake videos created by fusing a celebrity's face over an explicit video. The impact of deepfakes is alarming, with politicians, senior corporate officers, and world leaders being targeted by nefarious actors. An approach to detect deepfake videos of politicians using temporal sequential frames is proposed. The proposed approach uses the forged video to extract the frames at the first level followed by a deep depth-based convolutional long short-term memory model to identify the fake frames at the second level. Also the proposed model is evaluated on our newly collected ground truth dataset of forged videos using source and destination video frames of famous politicians. Experimental results demonstrate the effectiveness of our method. [DOI: 10.1117/1.JEI.29.3.033013]

Keywords: deepfake; news; politicians; face swap; forged; frames; videos.

Paper 200155 received Feb. 24, 2020; accepted for publication May 21, 2020; published online Jun. 8, 2020.

1 Introduction

Social media is inundated with deepfake (using face-swapping tools) videos of users. Deepfake videos are artificial intelligence (AI) generated clips that use open source libraries such as Google image search, Tensorflow,[1] social media,[2,3] and websites (YouTube[4] videos, stock photos, Instagram,[5] etc.) to insert famous people faces onto similar preexisting background videos by creating a machine-learning-based algorithm.[6]

In 2017, an anonymous user of Reddit posted defamatory videos of multiple celebrities in compromising positions.[7] Such videos were not real and damaged the identity of various world leaders, famous celebrities, and politicians, thus showing the alarming impact of deep-fakes. Deepfakes exist in three different forms.[8]

- *Lip sync*. In this form, a source video is modified to make an arbitrary audio recording showing a consistent movement along the mouth region. For instance, the director and actor Jordan Peele used this technique to create a viral video of Obama saying inflammatory things about President Trump.

*Address all correspondence to Sawinder kaur, E-mail: skaur_phd17@thapar.edu

1017-9909/2020/$28.00

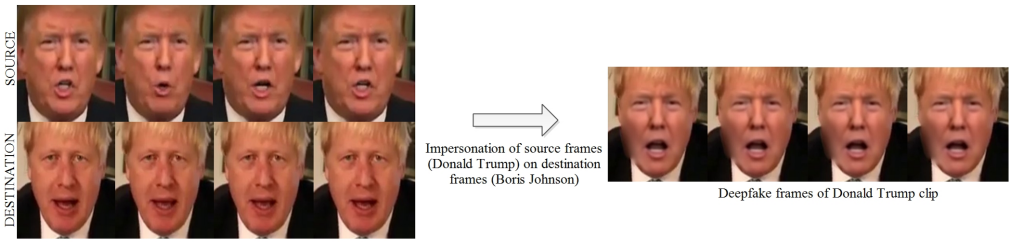

Fig. 1 Impersonation of source frames retrieved from the source clip onto the destination clip to extract the frames for generating the deepfake video clip.

- *Puppet master*. This is a technique in which a target person (puppet) is animated using their facial expressions, eye movements, and head movements by an actor. The actor performs by acting out in front of the camera to show the puppet's actions.
- *Face swap*. Two videos (source and destination) are considered in this technique. A face in the destination video is replaced by the source face. For example, one could create a deepfake of Donald Trump (US president) by superimposing his face onto a video of Boris Johnson (UK president) doing an impersonation of Trump as shown in Fig. 1.

Among these (lip sync, puppet master, and face swap), the most common is the face swap form of deepfake (also known as AI-synthesized media).[9,10] In this paper, we will focus on face-swap deepfake news clips related to famous politicians. Also the deepfake technology acts as a double-edged sword by playing a creative role for advancements in virtual reality, production, and film editing.

Software tools are designed to allow people without a technical background or programming experience to create deepfakes. With enough photos and video content of the target, usually sourced through publicly available social media profiles, hiring an expert to develop fake videos of unsuspecting victims is not difficult. Multiple frames pulled from one or more videos can provide a few hundred images to develop fake videos.[11]

Deepfakes can have a devastating impact on their victims, leading to traumatic stress disorders, depression, anxiety, and in extreme cases, driving them to the point of suicide. It can smear the victim's public reputation, resulting in unemployment, layoff, etc. The technology is relatively easy to use, especially with readily available machine learning algorithms and open-source codes. Also, the process requires only a basic consumer-grade graphics card to get the job done within hours.

In this paper, we present the first publicly available database of deepfake videos (100 source and 100 destination) of famous politicians using generative adversarial networks (GAN) based approach,[12] which is an open-source method developed from the original autoencoder-based Deepfake algorithm[13] for face swaps. We manually selected 200 similar looking pairs of people from publicly available videos using the YouTube platform[4] with high definition (HD) quality. Based on the collected dataset, we generated 100 deepfake videos of reputed politicians. There was no modification done to the audio tracks of the original (destination) videos.

To the best of our knowledge, no work has been done to detect deepfake face-swapped video[7] clips of politicians using temporal sequential frames retrieved from the clips to protect the political leaders against deepfakes within a timespan of initial 2 s of an uploaded video clip. The proposed approach uses the forged video[13] to extract the frames at the first level followed by a deep depth-based convolutional long short-term memory (C-LSTM) model to identify the fake frames at the second level.

The key contributions of this research paper are as follows.

- A ground truth dataset (deepfake and real frames) is prepared by swapping famous politicians face with the pre-existing video clips.
- An approach is proposed that works under a two-level structure. At the first level, the forged frames from the deepfake video are extracted using "OpenCL" and in the next phase preprocessing is performed on the extracted frames to feed it to the next level. At the

second level, a deep temporal-based C-LSTM model is used to identify the fake frames to detect the fake face-swap video clips.

- The deepfake video clips are predicted on the basis of temporal sequences and inconsistencies between the retrieved frames from the LSTM layer of the C-LSTM model to build a highly efficient model.
- The proposed deep temporal-based C-LSTM model is compared with the state-of-the-art models on the basis of performance metrics and training time.

The structure of this paper is organized as follows. Section 2 gives a brief overview of the related work done in the field of forged video classification. The problem statement is defined in Sec. 3. Training and generation of deepfake videos using the "DeepFaceLab" application are explored in Sec. 4. The dataset generated to perform the experiment is introduced in Sec. 5. The methodology followed to detect deepfake video clips and the architecture of our proposed system are discussed in Sec. 6. The evaluation phase of our proposed model is presented in Sec. 7. Section 8 concludes this paper along with presenting future work.

2 Related Work

Many approaches have been proposed in the last decade for detecting target face manipulations in both images and videos, which are known as deepfakes. CNN and feature-based methods have been studied in the literature to detect deepfake videos and images.

A universal forensic approach was proposed by Bayar and Stamm[11] to detect image editing operations using deep learning. A new form of the convolutional layer was designed to learn the manipulated features and to suppress an image's content. The proposed model was tested on a collected dataset (from 12 different models) creating a set of greyscale images. The model gave an accuracy of 99.10% for multiclass classification (detected four different types of forgery) and gave an accuracy of 99.31% for binary classification.

A two-stream (face classification and patch triplet) network for face tampering detection was proposed by Zhou et al. A CNN model was trained during a face classification stream to classify the face images as authentic or tampered. Steganalysis features were used to train a second stream for effective image splicing detection and to capture the hidden information from an image.[14] The proposed model was evaluated on a newly collected dataset using FaceSwap and SwapMe tools.

Korshunova et al. studied the problem of transforming the face identity of a source image into the target image using face swapping. The transformation was performed by preserving the lighting, facial expressions, and position of the identity in images, with a goal to render the image style of the source into the destination image. The authors used CNN to capture the appearance of the target identity using the CelebA dataset and developed new loss functions to produce high photorealistic results.[10]

Realistic fake face videos have been developed by deep generative networks with high efficiency and quality. Li et al. proposed a method using neural networks (NNs) to expose fake face videos on the basis of the eye-blinking feature. The first step involved the detection of faces in each frame of the video. In the next step, the detected faces were aligned into the same coordinate system to discount the changes in orientations of facial landmark points and head movements.[6] Then the eye-blinking was detected in each frame of video using the long-term recurrent convolutional neural network (LRCN). It is becoming easier to create face swaps in videos with machine learning-based freely available software. Such scenarios cause fake terrorism events, political distress, blackmailing of unknowns, etc. A temporal-aware pipeline-based system was proposed by Guera and Delp to automatically detect deepfake videos. Frame-based features were extracted by the proposed system using CNN.[15] The retrieved features by CNN model were then fed to RNN to classify the manipulated video within a timespan of 2 s.

A capsule network was proposed by Nguyen et al. to detect forged videos and images in a wide range of GANs generated video or image detections. The proposed network consisted of five capsules, three as primary and two as secondary or output capsules, in which the secondary capsules were used to predict the final output (real or fake). The method used deep convolutional networks and gaves promising results for all four datasets used in Ref. 13.

Table 1 Comparative analysis of research studies.

Authors	Proposed approach	Model	Dataset	Analysis
Bayar et al. (2016)	A universal forensic approach is proposed to detect multiple image manipulations using deep learning	CNN + RNN	Created own database of edited and unaltered images using 12 different camera models	The approach gives an accuracy of 99.10% in automatically detecting the multiple image manipulations
Zhou et al. (2017)	A two-stream network is proposed for tampered face detection	CNN (face classification stream) + SVM (patch triplet stream)	GoogleNet dataset using steganalysis feature extraction technique for triplet stream	The proposed approach learns both hidden noise residual features and tampering artifacts
Guera et al. (2018)	An automatic detection system is proposed for deepfake videos based on the temporal-aware pipeline	CNN + RNN	Deepfake video collection from multiple video websites	A video is predicted as a subject to manipulation or not within 2 s of temporal frames
Li et al. (2018)	A method is proposed to expose fake face videos based on eye-blinking generated by NN	LRCN	Created own database eye blinking video	LRCN gives a 0.99 ROC curve
Li et al. (2019)	A method is proposed to detect the difference between deep neural generated images and real scene images by analyzing the disparities in color components	LDA by extracting various features from the colour components	CelebA, HQ-CelebA, and LFW datasets that contain various face images with different resolutions	Average accuracy achieved is >99%
Yang et al. (2019)	A method is proposed to detect AI-generated videos or images	SVM using the difference between the estimated 3-D head poses as features	Fake video data (UADFV) fake image data (DARPAMedi for GAN image or video challenge)	SVM gives 0.89 AUC on UADFV and 0.843 AUC on DARPA corpora
Nguyen et al. (2019)	A novel capsule network with random noise is proposed to detect image and video forgeries	CNN	Faceforensic, deepfake, REPLAY-ATTACK, computer generated images, and photographic images	Capsule random noise network gives an accuracy of 95.93% and 99.23% for image and video deepfake dataset and an accuracy of 99.37% and 99.33% for image and video faceforensic dataset with no compression, respectively. Also the model gives an accuracy of 97% for CGIs and PIs dataset and an HTER of 0% for REPLAY-ATTACK dataset
Proposed approach	A deep temporal-based C-LSTM model is proposed to detect the deepfake video clip using temporal sequential frames	CNN + LSTM	Trained and generated own deepfake dataset of famous politicians	C-LSTM model gives an accuracy of 98.21% on collected ground truth dataset

A method was proposed by Yang et al. to detect AI-based generated fake videos and images. The authors compared the head poses of the identity using only the central regions and all landmarks of the face in images and videos. They used the differences of the established head poses as a feature vector to train the binary classifier support vector machine (SVM) to detect deepfakes and original images or videos.[16] The results revealed that the SVM model achieves an area under receiver operating characteristic (ROC) (AUC) of 0.843 and 0.89 on defense advanced research projects agency (DARPA) and UADFV corpora, respectively.

It is assessed that the research in the field of deepfake video or image detection is mainly restricted to CNN and SVM classifiers without extracting any unique feature vectors from deepfake content as shown in Table 1. No dataset is available for face-swapped deepfake videos of famous politicians. So we have created a ground truth dataset for both real and deepfake video clips by swapping famous politicians face with the preexisting video clips in this paper. Also an approach is proposed that works under a two-level structure. At the first level, the forged frames from the deepfake video are extracted using OpenCL and further preprocessing is performed on the extracted frames to feed it to the next level. At the second level, a deep temporal-based C-LSTM model is used to identify the fake frames to detect the fake face-swap video clips. To the best of our knowledge, no work has been done to detect deepfake face-swapped video clips of politicians using temporal sequences of the frames. The problem statement to identify the deepfake videos is discussed in the next section.

3 Problem Statement

The work proposed in this paper addresses the issue of identifying fake news videos of famous politicians. To solve the listed problem, we propose an approach to analyze the deepfake features in tampered videos.

The videos are classified using a binary {deepfake_video, real_video} classifier, and the classification process is performed at two levels. At the first level, real and deepfake video clips are selected. Further, frames are extracted from the referenced video (real and deepfake) clips. Following this, face alignment is performed on the extracted frames to perform preprocessing. At the second level, a deep temporal-based C-LSTM model is applied to identify the inconsistencies between the extracted frames to predict deepfake video clips.

4 Deepfake Videos Exposed

As the DeepFaceLab generates the deepfake videos, temporal and the intraframe inconsistencies between frames are created.[17] To detect the deepfake manipulation, the temporal and intraframe anomalies can be exploited. Let us briefly explain the way we can exploit such video anomalies and the way a deepfake video is generated to detect the deepfake video clips.

4.1 *Training and Generation of Video Using DeepFaceLab*

NNs (autoencoders) are used to compress or decompress images.[18] Figure 2 shows that the chosen image (retrieved face) in the case of face swap is fed to an encoder that gives a low-dimensional representation of the face, which is also known as a latent face. The latent face is then passed to the decoder for its reconstruction. To make face-swapping possible, the DeepFaceLab uses two autoencoders with the same encoder and different decoders.[10]

For training an NN, two sets (source and destination) of training images are required. Among these two sets, the first set consists of original frames that are replaced by the destination video frames, resulting in the creation of a manipulated destination clip. The second set consists of the faces that are swapped with the source video frames.[18]

The two sets (source and destination clips) are treated separately. The decoder of X (Donald Trump) is only trained with the retrieved frames of its clip (source); the decoder Y (Boris Johnson) is only trained with the frames retrieved from its clip (destination) as shown in Fig. 2. All of the common features are identified by the encoder from the latent faces.[19] The encoder can automatically identify common features in both faces as these faces share a similar structure.[20]

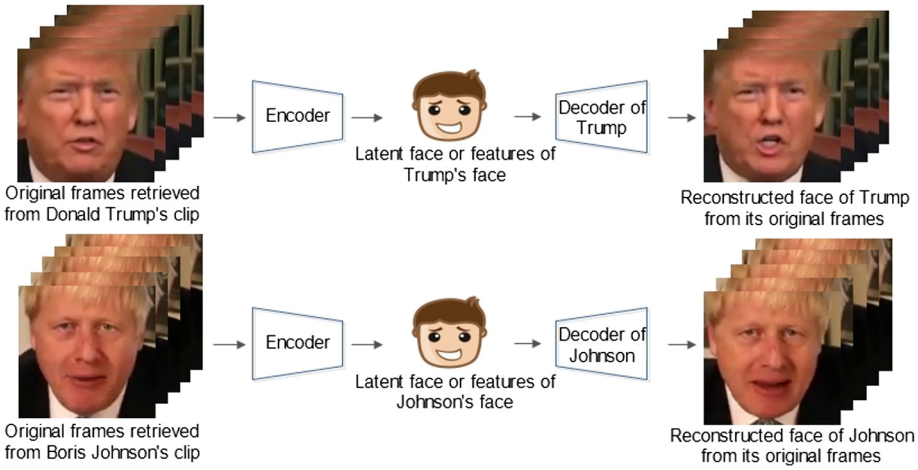

Fig. 2 Representation of two networks (encoder and decoder) for the source and destination clips during the training phase.

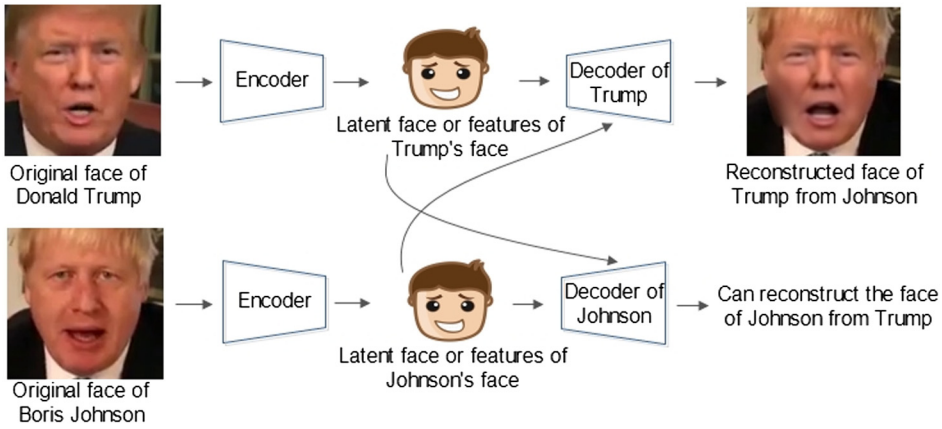

Fig. 3 Representation of two networks (encoder and decoder) for the source and destination clips during the generation phase.

After completion of the training process, a latent representation of a face generated from the source clip X (Donald Trump) present in the source video is passed to the decoder of Y (Boris Johnson) to reconstruct subject X from Y as shown in Fig. 3. Also the intermediate iterations extracted during the merging phase to generate the deepfake video clips are shown in Fig. 4.

Further, a swapped video[15,21] is retrieved and is used as a first frame-level feature to be taken as an input for our proposed deep temporal-based C-LSTM model. To perform the frame-level feature extraction, a face detector is used to extract the face regions from the whole frame as shown in Fig. 5, which is further passed to the trained autoencoder to generate the deepfake video clips.[12]

5 Data Collection

Many sources such as Twitter,[22] LinkedIn,[3] Facebook,[23] and YouTube[4] are common trading platforms used to publish inappropriate information. Such types of information are disseminated in the form of images, videos, and posts on social media websites.[3]

5.1 *Experimental Setup*

To perform the experiment, the deepfake dataset is collected by replacing a politician's face from a source video and imposing it onto the destination video using an NN.[21,24] The minimum requirements to generate deepfake video clips are discussed in Table 2.

Fig. 4 Merging of (a) Arvind Kejriwal frames of source clip onto Ashok Chavan frames of destination clip at iteration 60, 113,570 and 234,050; (b) Donald Trump frames of source clip onto Boris Johnson frames of destination clip at iteration 60,980, 197,301, and 138,254; and (c) Jaya Bachchan frames of source clip onto Smriti Irani frames of destination clip at iteration 37,594, 97,927, and 156,121 using DeepFaceLab.

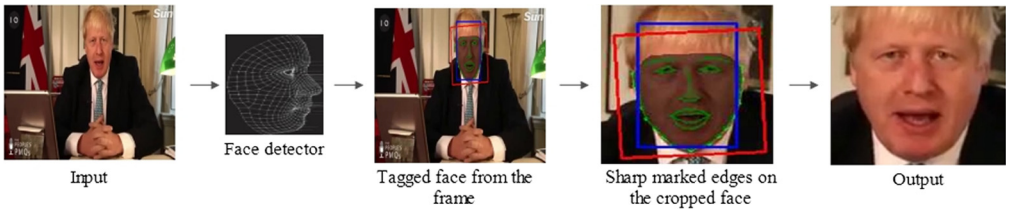

Fig. 5 Extraction of the face region from the retrieved frames of the video clip using face detector.

To generate a deepfake video, a DeepFaceLab application is required to be downloaded from the Github repository.[17] Further, cloning of the repository by building each Python library is required. The software required to run the application is CUDA toolkit 9.0 with all four patches (CuBLAS update with RNN and convolutional, CuBLAS update of GEMV operation, CuBLAS update with new GEMM kernels, and CuBLAS update with GEMM heuristics) along with the base installer.

The other software required to run the application is the NVIDIA CuDNN 7.0 for CUDA 9.0 toolkit. The step-wise instructions to execute the repository using the above discussed software are given in Ref. 25. With the help of the DeepFaceLab application, we generate deepfake videos using source and destination video clips (a sample of 20 videos from our collected dataset).

To perform the experiment, a large dataset (in the form of video frames) that consists of 200 videos (100 real and 100 deepfake) is created to evaluate our proposed model. The whole dataset

Table 2 Summary of configurations required for the collection of dataset.

Developer	Url	Configuration used
iperov	https://github.com/iperov/DeepFaceLab	16 GB RAM
		NVIDIA video card with 8 GB video
		Windows 7 and higher
		OpenCL-compliant graphics card (NVIDIA)
		Processor supporting streaming SIMD extensions instructions
System	**Item**	**Details**
Graphics processing unit (GPU)	Titan Xp	CUDA cores: 3840
		Graphic clock: 1404 MHz
		Memory data rate: 11,410 MHz
		Memory interface: 384-bit
		Memory bandwidth: 547.68 GB/s
		Bus: PCI express × 16 Gen3
		Dedicated video memory: 12,288 MB GDDR 5×

was curated from YouTube channel related to famous politician's news or speeches.[26] Two types of video clips (source and destination) are considered for our dataset, in which the face of the politician used in the dataset in a source video is swapped with the face of the politician in the destination video. The focus was to concentrate on the videos of a person of interest (POI) talking in formal scenarios such as public speech, news interview, and weekly addresses.

All videos were manually downloaded with the primary focus being on a POI facing toward the camera. The proposed approach is tested on the famous politicians as shown in Table 3. The SD_i represents the face swap between source and destination clips, where $i \in \{1,2,\ldots,100\}$.

Table 3 A sample of 20 POI chosen for source and destination video clips from our collected dataset with high-quality downloads.

	POI		Video duration (minutes) + quality (HD)	
Name	Source interest	Destination interest	Source clip	Destination clip
SD_1	Donald Trump	Boris Johnson	01:15 + 360	01:05 + 360
SD_2	Arvind Kejriwal	Ashok Chavan	00:34 + 360	00:38 + 360
SD_3	Jaya Bachchan	Smriti Irani	00:35 + 720	00:30 + 360
SD_4	Rahul Gandhi	Narendra Modi	01:15 + 360	00:23 + 360
SD_5	Arvind Kejriwal	L. K. Advani	00:31 + 360	00:30 + 720
SD_6	Akhilesh Yadav	Arvind Kejriwal	00:49 + 360	00:47 + 360
SD_7	Mahua Moitra	Hema Malini	00:32 + 360	00:37 + 360
SD_8	Smriti Irani	Sasikala Pushpa	00:58 + 480	00:46 + 480
SD_9	Hina Rabbani Khar	Mahua Moitra	00:41 + 360	00:38 + 480
SD_{10}	Narendra Modi	Hukmdev Narayan Yadav	00:30 + 480	00:32 + 480

Table 4 A sample of 20 POI video clips from our collected dataset with total time taken to train the model, total loss after training the model, and total number of frames retrieved from these clips.

Name	Loss value	Training duration (h)	Number of frames (at 2 s)	Total frames
SD_1	0.01	10:08:34	62	1962
SD_2	0.11	15:32:08	60	1143
SD_3	0.07	19:00:58	62	924
SD_4	0.02	16:23:41	64	717
SD_5	0.07	12:30:48	26	373
SD_6	0.01	21:07:38	54	1261
SD_7	0.04	14:55:12	36	668
SD_8	0.06	13:28:05	36	792
SD_9	0.07	12:47:53	50	923
SD_{10}	0.02	15:14:11	34	541

In total, 200 video clips (source and destination) are collected to perform the experiment. The source video clips are swapped with the destination by overlapping the facial features from source to destination using the DeepFaceLab application in which the background of the destination video remains unaltered. It is observed from Table 4 that the number of frames retrieved for 2-s video clip is highest for SD_4. Also the time-based representation (in millisecond) per frame retrieved from the deepfake clip of Donald Trump is shown in Fig. 6. The three-frame sequence of a fake clip from the original clip using DeepFaceLab from our collected dataset is shown in Fig. 7, where X (Arvind Kejriwal), Y (Donald Trump), and Z (Jaya Bachchan) (from the collected source clips), P (Ashok Chavan), Q (Boris Johnson), and R (Smriti Irani) (from the collected destination clips) are swapped and deepfake frames of source clips are returned by impersonating it onto the destination frames. The approach followed in this paper is discussed in the next section.

Fig. 6 Time-based representation of video frames retrieved from the source, destination, and deepfake video clips.

Fig. 7 Three deepfake example frame sets generated from the source video clips using DeepFaceLab.

6 Methodology

An overview of the proposed automatic detection system for deepfake clips is seen in Fig. 8. A set of unseen test sequences is passed to the CNN model to retrieve the features for each sequential frame generated during video training and generation as discussed in the previous section. The extracted features are then passed as an input in the form of sequential frames to the LSTM model. At last, an estimation of the likelihood of the sequential frames being real or deepfake is detected.

The detailed architecture of the proposed C-LSTM model for automatic detection of deepfake clips is shown in Fig. 9. The system is composed of two levels, i.e., image preprocessing at the first level, CNN[27] and LSTM subnetworks[28] at the second level for processing sequential frames of both video clips (source and destination).

6.1 *Image Preprocessing*

The input frames retrieved from the video clip use the "ImageDataGenerator" class to perform image preprocessing. Various operations performed for the image preprocessing phase are discussed below.

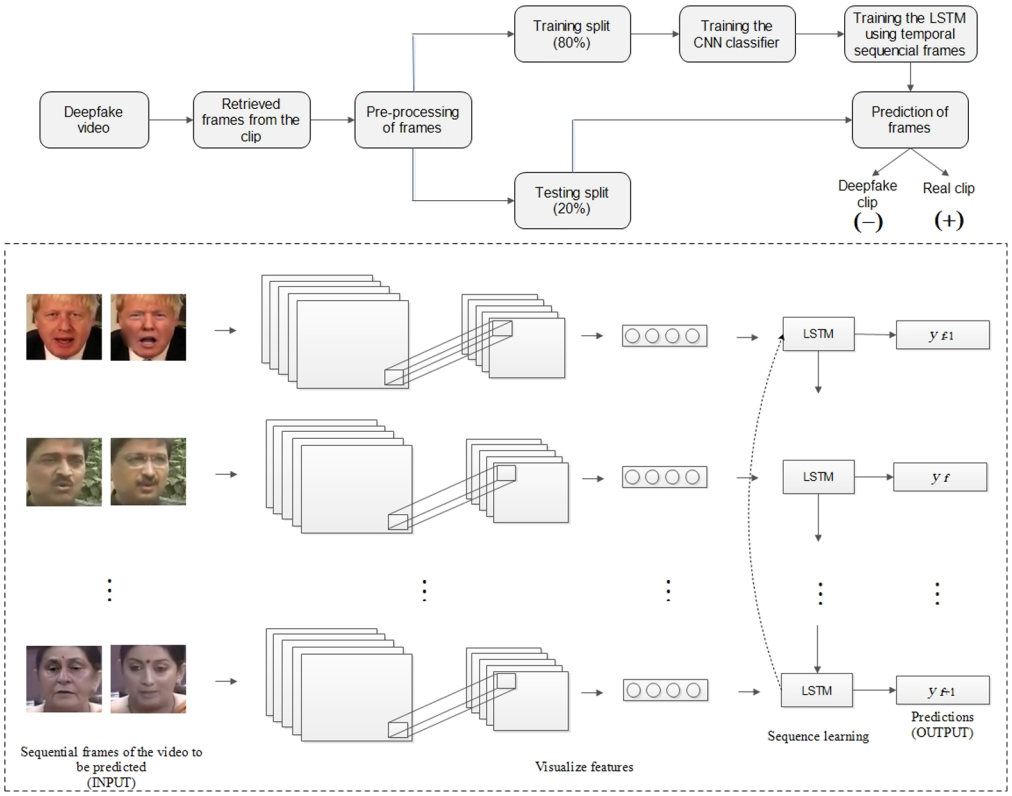

Fig. 8 Overview of our proposed deep temporal-based C-LSTM model for automatic detection of deepfake video clips of politicians. It consists of C-LSTM for processing the input temporal frames.

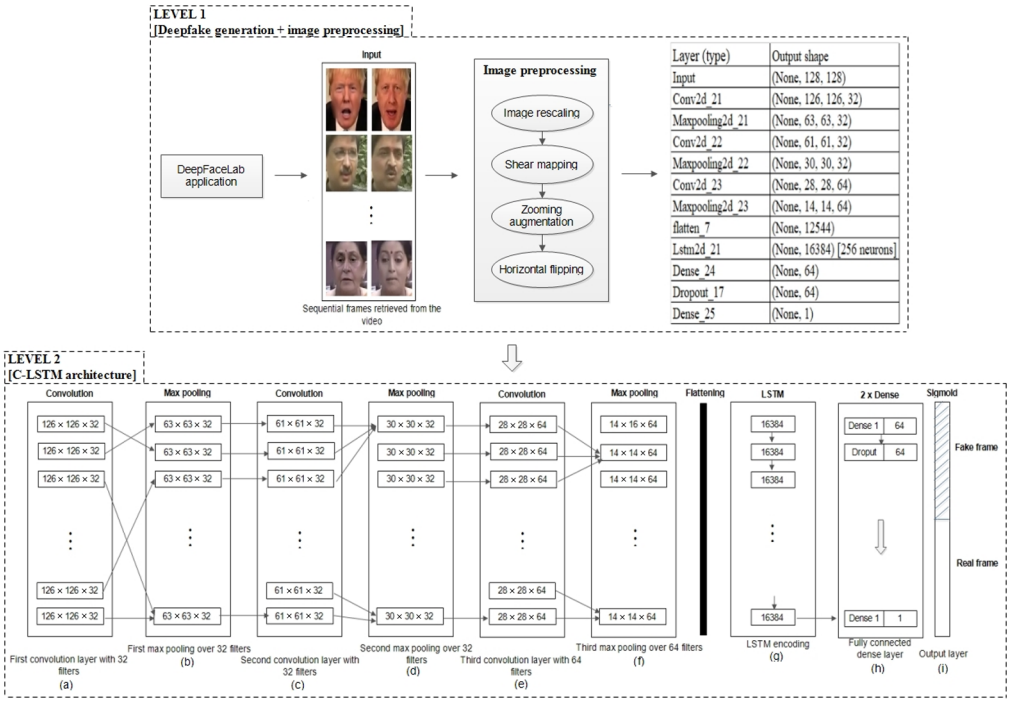

Fig. 9 Two-level structure of the proposed deep temporal-based C-LSTM model for automatic detection of deepfake face-swapped video clips of politicians.

- *Image rescaling*. The original frames in the form of images consist of RGB coefficients in the range (0 to 255). These values are too high to feed directly into the proposed model, so the values are rescaled between 0 and 1 using the 1/255 scaling factor.
- *Shear mapping*. For the set of frames, each image is displaced from its edge to the vertical direction. The "shear_range" parameter controls the displacement rate and the deviation angle between the horizontal line of the original frame and the frame of a line in the transformed frame (shear_range = 0.2).
- *Zooming augmentation*. To make the appearance of the face in the frame larger, zooming augmentation is configured by the zoom_range (0.2) parameter. The range of this parameter varies from [1 − value, 1 + value], i.e., 0.2 zoom value will have the range from [0.8, 1.2].
- *Horizontal flipping*. The zoomed images are then flipped horizontally by setting the boolean value of the horizontal_flip parameter to "true."

The two other essential components of the proposed system are CNN and LSTM as discussed below.

6.2 *Convolutional-Long Short Term Memory*

An end-to-end learning of fully connected layers is used to detect the deepfake clip of politicians as shown at level 2 in Fig. 9. Our proposed model is divided into CNN and LSTM components.[29] CNN is used to extract the high-level features from the sequential frames of the source and destination video clips. LSTM is used to capture the inconsistencies and temporal-based sequences and reduces the training time of the model. With the help of LSTM, it becomes easy to analyze the temporal sequences of the video frames to improve the efficiency of the model as an automatic detection of a deepfake video clip is done within a timespan of 2 s of inputting the video clip.

The input sequence of frames retrieved after preprocessing is then fed to a C-LSTM.[30] The extracted features of frames are detected at different regions using a sliding filter vector evolved in the convolution layer. Consider $u_i \in \mathbb{R}^m$ the m-dimensional vector of a particular frame from the input video clip for the i'th position. Let $u \in \mathbb{R}^{l \times m}$ be the input frame sequence, where l denotes the length of the sequential frame. Consider p the length of the filter and vector $v \in \mathbb{R}^{p \times n}$ the filter for performing the convolution operation. A window vector w_j with p consecutive frame vectors, where j is the position of a pixel in a frame, is given by

$$w_j = [u_j, u_{j+1}, \ldots, u_{j+p-1}], \tag{1}$$

where commas represent the row vector concatenation. The vector v revolves at each position with the window vectors to generate a feature map $fm \in \mathbb{R}^{l-p+1}$, where each element fm_i of the feature map for the window vector w_j is represented by

$$fm_j = n_t(w_j \odot v + b), \tag{2}$$

where $\odot$ is the element-wise multiplication, n_t is the nonlinear transformation function, and b is the biased term that belongs to $\mathbb{R}$.

The number of filters used in the CNN model for our experiment is 32 of 3×3 dimension, i.e., the convolution window is set to 3. ReLu is the nonlinear activation function chosen for the hidden layers, and 2×2 is the pool size for the max-pooling layer, which remains the same for all three max-pooling layers in the CNN model.[27]

During the learning process, our model is trained for 25 epochs. The input shape of the frame considered in the architecture is 128×128. The frames are split into an 80:20 (training:testing) ratio. The input dimension (128×128) is fed to the first convolutional layer (a). The output dimension of the first convolutional layer (a) is $126 \times 126 \times 32$, which is fed as an input to the first max pooling layer (b), giving an output of $63 \times 63 \times 32$. The second convolution layer (c) takes $63 \times 63 \times 32$ dimensions as the input and trains the vector. In the next step, the second max-pooling layer (d) takes the $61 \times 61 \times 32$ input shape from the previous convolutional layer

(c) and gives $30 \times 30 \times 32$ as the output shape. Further, the third convolutional layer (e) uses 64 feature maps and gives $28 \times 28 \times 64$ as the output shape, which is fed to the third max-pooling layer (f).

After the third max-pooling layer (f), flattening is performed. The 2048-dimensional feature vectors retrieved as an output from the flattening layer are used as the sequential input to the LSTM (g) with 256 neurons through input gate (i_g) as shown in the following equation:

$$i_g = \sigma[W_j \cdot (h_{t-1}, x_t) + b_i], \tag{3}$$

where h_{t-1} is the previous output of the hidden state, x_t is the input at current time step t, and σ is the logistic sigmoid function that gives output between [0,1]. The sequence of the CNN feature vectors is recursively processed in LSTM (g) to address the temporal sequence of video frames with 16,384 (values) × 256 (neurons) + 256 (bias values for neurons) as output. LSTM is followed by a dense layer (h) with a 0.5 chance of dropout, which is capable of remembering the temporal sequence of input frames using the ReLu function. In the dropout layer (h), randomly selected neurons are ignored during training, which helps to make the network less sensitive to the specific weights of neurons. Hence, the network becomes less likely to overfit the training data. A second dense layer (with Sigmoid function) (i) is used to give the final prediction of the proposed network classifying a sequence as a deepfake or real video.

7 Evaluating C-LSTM Model

The proposed model was implemented using Theano.[31] Theano is a Python library that supports the efficient use of a GPU and symbolic differentiation. The model was trained on a GPU (Titan Xp) system to get better efficiency.

To evaluate the proposed deep temporal-based C-LSTM model, a comparison of performance analysis using MesoNet,[32] capsule,[33] and CNN state-of-the-art models is discussed in this section. Classification of deepfake video clip[34] is performed on the collected dataset using our proposed model. Various results are discussed in this section to test our proposed model.

The analysis of training and validation data on the basis of accuracy metric and loss information for the collected deepfake video dataset using state-of-the-art models and our proposed model at different epochs is shown in Fig. 10. It is observed from Figs. 10(b), 10(d), and 10(f) that the MesonNet,[32] capsule,[33] and CNN models give a validation loss of 1.81, 46.93, and 6.31, respectively, which is greater than the proposed model with a validation loss of 1.75 as shown in Fig. 10(h). Similarly, the CNN model gives a validation accuracy of 97.06% as observed from Fig. 10(e), which is less than the proposed model with a validation accuracy of 98.21% as shown in Fig. 10(g), which thus affects the model efficiency.

The comparative analysis between the proposed and other state-of-the-art models on the basis of validation accuracy and loss history at every epoch level is shown in Fig. 11. The comparative analysis of the precision metric is shown in Table 5.

Consider the number of epochs $(\text{epoch})_i$, where $i \in \{1,2,\ldots,25\}$. It is observed from Fig. 11 that the C-LSTM model accuracy metric < the CNN model accuracy metric for $(\text{epoch})_i$, where $i \in \{1,2,3,4\}$. A similar behavior of validation loss history at each epoch level is analyzed. It is observed that the C-LSTM model has a validation loss > the CNN validation loss at $(\text{epoch})_i$ where $i \in \{1,2,3\}$, and gradually the loss rate decreases until 1.75 for C-LSTM at $(\text{epoch})_{25}$. Hence, the proposed C-LSTM model is better than the simple CNN based on the accuracy and value-loss information metrics.

The performance of the C-LSTM is evaluated in terms of accuracy, precision, recall, and $F1$-score as shown in Table 5. The results of the performance metrics for all state-of-the-art and the proposed temporal-based C-LSTM models are shown in the same table as discussed above. It is assessed that the proposed model outperforms the MesoNet,[32] capsule,[33] and CNN state-of-the-art models when LSTM is used for detecting deepfake clips on the basis of temporal sequences and inconsistencies seen between retrieved frames from our own collected dataset. The total number of real and deepfake frames retrieved from the collected dataset is 181,608. Our proposed model achieves an accuracy of 98.21% for 0.9962 precision and 0.9391 recall values. The loss value of validation data is 1.75, which is quite lower than the CNN (6.31) and capsule

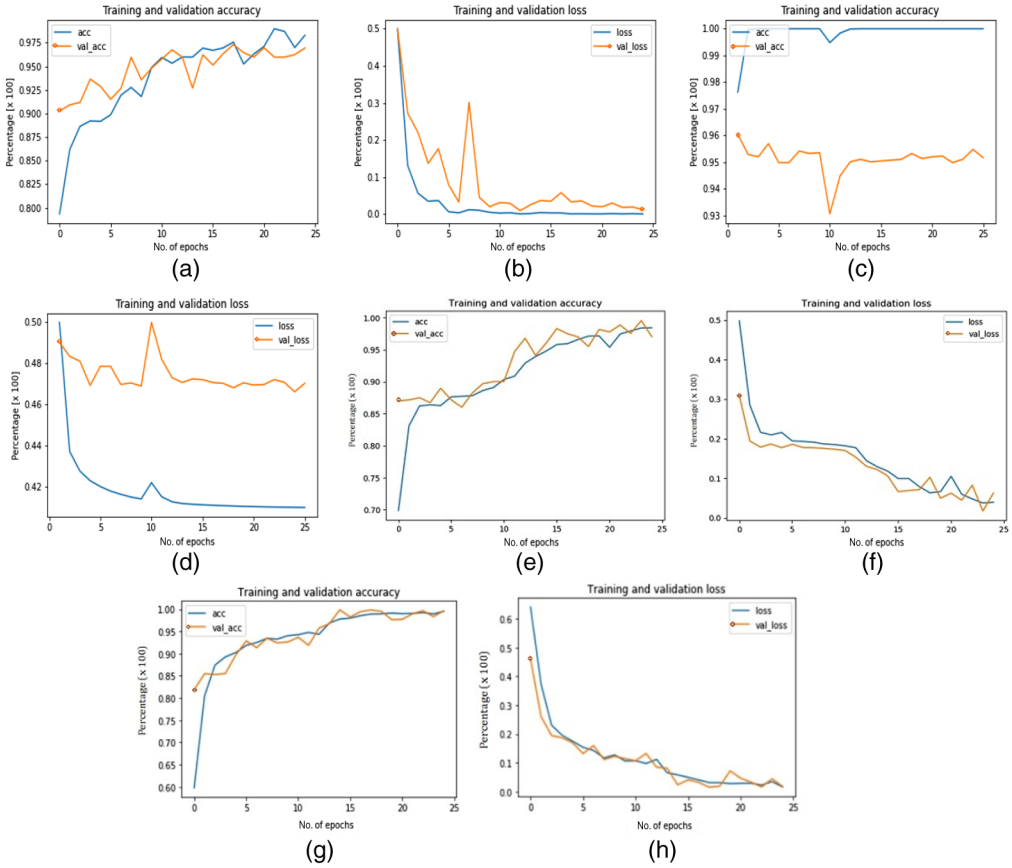

Fig. 10 Analysis of (a) training versus validation accuracy and (b) training versus validation loss for the collected deepfake video dataset using the MesoNet model, (c) training versus validation accuracy and (d) training versus validation loss for the collected deepfake video dataset using the capsule model, (e) training versus validation accuracy and (f) training versus validation loss for the collected deepfake video dataset using the CNN model, (g) training versus validation accuracy and (h) training versus validation loss for the collected deepfake video dataset using the proposed deep temporal-based C-LSTM model at different epochs.

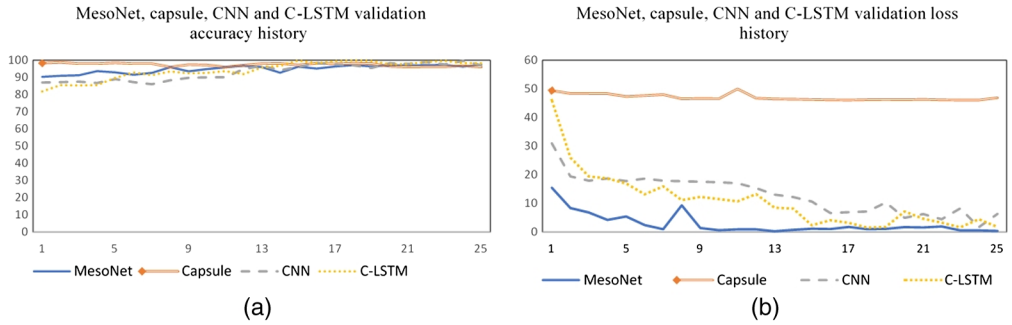

Fig. 11 Comparison of state-of-the-art versus proposed deep temporal-based C-LSTM: (a) validation accuracy history and (b) validation loss history at different epochs.

(46.93) validation loss. Also the total training time taken by the MesoNet (00:58:43 h), capsule (03:49:23 h), CNN model (01:22:23 h) > the C-LSTM model (00:39:42 h), which is collectively greater than the existing state-of-the-art models. Through these observations, it is analyzed that the proposed model improves the overall performance by 1.15% in comparison with the simple CNN model. The proposed model reduces the total training time to train the features of the deepfake video frames.

Table 5 Comparative analysis of performance measures using MesoNet, capsule, CNN, and the proposed deep temporal-based C-LSTM architecture on our collected dataset.

	Frames							
Architecture	Fake frames	Real frames	Accuracy	Precision	Recall	*F*1-score	Loss	Training time (h)
MesoNet	90,804	90,804	97.30	98.91	93.97	96.37	1.81	00:58:43
Capsule			96.03	97.99	93.29	95.58	46.93	03:49:23
CNN			97.06	97.91	94.24	96.05	6.31	01:22:23
Temporal-based C-LSTM			**98.21**	99.62	93.91	96.68	**1.75**	**00:39:42**

Table 6 Comparative analysis of performance measures using Celeb-DF, DFDC, and ground dataset using proposed deep temporal-based C-LSTM architecture.

Proposed temporal-based C-LSTM model							
Datasets	Fake frames	Real frames	Accuracy	Precision	Recall	*F*1-score	Training time (h)
Celeb-DF	123,900	123,900	96.46	98.99	98.99	**98.99**	00:56:91
DFDC	187,900	187,900	96.99	99.07	97.15	**98.10**	01:35:26
Ground truth	90,804	90,804	98.21	99.62	93.91	**96.68**	00:39:42

No standard dataset is available for detecting deepfake video clips of politicians. So we used a mixed domain dataset (Celeb-DF[9] and DFDC[35]) that is available online, and we observe that our proposed deep temporal-based C-LSTM model gives effective results in terms of *F*1-score and accuracy metric as shown in Table 6. Also from Table 5, it is assessed that our proposed model outperforms the state-of-the-art techniques that were used to detect the deepfakes on our collected ground truth dataset. A primary prototype of the developed system is available at IsItFake.[36]

8 Conclusion and Future Scope

A two-level deep temporal-based C-LSTM model has been proposed in this paper. In the proposed approach, the forged frames from deepfake videos are extracted using OpenCL and a frame-level preprocessing is performed at the first level. The preprocessed frames are further fed to a C-LSTM algorithm to analyze the deepfake (face-swapped) videos. The deepfake clips are predicted on the basis of temporal sequences and inconsistencies between the frames retrieved from the LSTM layer of the C-LSTM model, which achieves better performance than the state-of-the-art models. Also it is observed that if a POI in the clip consistently looks away from the camera, the deepfake video detection is compromised.

In the future, the detection of deepfakes can be done using the edited audio by changing the audio speech of a person along with the lip-sing. The source clip frames can be further extended to include the images from publicly available sources to analyze more realistic results. Also a tool named FaceSwap video Inspector will be made publicly accessible (comprising front-end and back-end) for detecting a video clip as tampered or original in real time. The back end will consist of a pretrained model (deep temporal-based C-LSTM) and the front end will consist of a browser plug-in (Google Chrome) that will communicate over HTTP RESTful APIs. Such an application can help to control the various types of fraudulent videos (related to political speeches) from growing over social media.

Acknowledgments

This publication is an outcome of the R&D work undertaken in the project under the Visvesvaraya PhD Scheme of Ministry of Electronics and Information Technology, Government of India, being implemented by Digital India Corporation (formerly Media Lab Asia). We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPU used for this research.

References

1. M. Abadi et al., "Tensorflow: large-scale machine learning on heterogeneous distributed systems," Software available from tensorflow.org (2015).
2. S. Kaur, P. Kumar, and P. Kumaraguru, "Automating fake news detection system using multi-level voting model," *Soft Comput.* **24**, 9049–9069 (2019).
3. F. Benevenuto et al., "Detecting spammers and content promoters in online video social networks," in *Proc. 32nd Int. ACM SIGIR Conf. Res. and Dev. in Inf. Retrieval*, ACM, pp. 620–627 (2009).
4. K. R. Canini, B. Suh, and P. L. Pirolli, "Finding credible information sources in social networks based on content and social structure," in *IEEE Third Int. Conf. Privacy, Secur., Risk and Trust and 2011 IEEE Third Int. Conf. Social Comput.*, IEEE, pp. 1–8 (2011).
5. H. Hosseinmardi et al., "Detection of cyberbullying incidents on the Instagram social network," (2015).
6. Y. Li, M. Chang, and S. Lyu, "In ictu oculi: exposing AI created fake videos by detecting eye blinking," in *IEEE Int. Workshop on Information Forensics and Security (WIFS)*, pp. 1–7 (2018).
7. S. Tariq et al., "Detecting both machine and human created fake face images in the wild," in *Proc. 2nd Int. Workshop Multimedia Privacy and Security*, ACM, pp. 81–87 (2018).
8. S. Agarwal et al., "Protecting world leaders against deep fakes," in *Proc. IEEE Conf. Comput. Vision and Pattern Recognit. Workshops*, pp. 38–45 (2019).
9. Y. Li et al., "Celeb-df: a new dataset for deepfake forensics," (2019).
10. I. Korshunova et al., "Fast face-swap using convolutional neural networks," in *Proc. IEEE Int. Conf. Comput. Vision*, pp. 3677–3685 (2017).
11. B. Bayar and M. C. Stamm, "A deep learning approach to universal image manipulation detection using a new convolutional layer," in *Proc. 4th ACM Workshop Inf. Hiding and Multimedia Secur.*, ACM, pp. 5–10 (2016).
12. C. Vondrick, H. Pirsiavash, and A. Torralba, "Generating videos with scene dynamics," in *Adv. Neural Inf. Process. Syst.*, pp. 613–621 (2016).
13. H. H. Nguyen, J. Yamagishi, and I. Echizen, "Capsule-forensics: using capsule networks to detect forged images and videos," in *ICASSP 2019-2019 IEEE Int. Conf. Acoust., Speech and Signal Process. (ICASSP)*, IEEE, pp. 2307–2311 (2019).
14. P. Zhou et al., "Two-stream neural networks for tampered face detection," in *IEEE Conf. Comput. Vision and Pattern Recognit. Workshops (CVPRW)*, IEEE, pp. 1831–1839 (2017).
15. D. Güera and E. J. Delp, "Deepfake video detection using recurrent neural networks," in *15th IEEE Int. Conf. Adv. Video and Signal Based Surveill. (AVSS)*, IEEE, pp. 1–6 (2018).
16. X. Yang, Y. Li, and S. Lyu, "Exposing deep fakes using inconsistent head poses," in *ICASSP 2019-2019 IEEE Int. Conf. Acoust., Speech and Signal Process. (ICASSP)*, IEEE, pp. 8261–8265 (2019).
17. B. Lyon, "Faceswap," https://github.com/deepfakes/faceswap (accessed 28 September 2019).
18. A. Rössler et al., "Faceforensics: a large-scale video dataset for forgery detection in human faces," (2018).
19. G. Goswami, M. Vatsa, and R. Singh, "RGB-D face recognition with texture and attribute features," *IEEE Trans. Inf. Forensics Secur.* **9**(10), 1629–1640 (2014).
20. N. Abudarham and G. Yovel, "Reverse engineering the face space: discovering the critical features for face identification," *J. Vision* **16**(3), 40–40 (2016).

21. L. K. Hansen and P. Salamon, "Neural network ensembles," *IEEE Trans. Pattern Anal. Mach. Intell.* **12**(10), 993–1001 (1990).
22. A. H. Wang, "Don't follow me: spam detection in Twitter," in *Int. Conf. Secur. Cryptogr. (SECRYPT)*, IEEE, pp. 1–10 (2010).
23. P. Dewan and P. Kumaraguru, "Facebook inspector (FBI): towards automatic real-time detection of malicious content on facebook," *Social Network Anal. Mining* **7**(1), 15 (2017).
24. F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: a unified embedding for face recognition and clustering," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, pp. 815–823 (2015).
25. I. Perov et al., "DeepFaceLab," https://github.com/iperov/DeepFaceLab (accessed 28 September 2019).
26. J. Caetano et al., "Analyzing and characterizing political discussions in Whatsapp public groups," (2018).
27. F. Chollet, "Xception: deep learning with depthwise separable convolutions," in *Proc. IEEE Conf. Comput. Vision and Pattern Recognition*, pp. 1251–1258 (2017).
28. A. Ullah et al., "Action recognition in video sequences using deep bi-directional LSTM with CNN features," *IEEE Access* **6**, 1155–1166 (2017).
29. Z. Xu, S. Li, and W. Deng, "Learning temporal features using LSTM-CNN architecture for face anti-spoofing," in *3rd IAPR Asian Conf. Pattern Recognit. (ACPR)*, IEEE, pp. 141–145 (2015).
30. M. Sivaram et al., "Detection of accurate facial detection using hybrid deep convolutional recurrent neural network," *ICTACT J. Soft Comput.* **9**(2), 1844–1850 (2019).
31. J. Bergstra et al., "Theano: a CPU and GPU math expression compiler," in *Proc. Python Sci. Comput. Conf. (SciPy)*, Austin, Texas, vol. 4, No. 3 (2010).
32. D. Afchar et al, "MesoNet: a compact facial video forgery detection network," in *IEEE Int. Workshop Inf. Forensics and Secur. (WIFS)*, IEEE, pp. 1–7 (2018).
33. H. Nguyen, J. Yamagishi, and I. Echizen, "Use of a capsule network to detect fake images and videos," (2019).
34. P. Korshunov and S. Marcel, "Vulnerability assessment and detection of deepfake videos," in *12th IAPR Int. Conf. Biometrics (ICB)*, pp. 1–6 (2019).
35. B. Dolhansky et al., "The deepfake detection challenge (DFDC) preview dataset," (2019).
36. S. Kaur and P. Kumar, "IsItFake," https://faceswap.isitfake.co.in/ (accessed 20 February 2020).

Sawinder Kaur received her ME degree from Thapar Institute of Engineering and Technology, Patiala, India, in 2016. She has been working as a research scholar in the Computer Science Department, Thapar Institute of Engineering and Technology, Punjab, India, since July 2017. Her research interests include online fake news detection, machine learning, deep learning, and data retrieval from social media platforms.

Parteek Kumar received his PhD in computer science and engineering from Thapar University, Patiala, India, in 2012. He is currently working as an associate professor in the Department of Computer Science and Engineering, Thapar Institute of Engineering and Technology, Patiala, India. He has authored more than 110 research papers published in various conferences and journals. His research interest includes machine learning, natural language processing, big data analytics, and online social media. He has authored six books and acted as PI and co-PI in three government-funded projects.

Ponnurangam Kumaraguru received his PhD in computer science from Carnegie Mellon University, USA, in 2009. He is currently working as a professor in the Department of Computer Science and Engineering, Indraprastha Institute of Engineering and Technology, Delhi, India. He has 3 registered patents and more than 120 research papers in reputed conferences and journals. His research interests include online social media, e-crime, privacy, and usable security. He has acted as a PI and a co-PI in various government-funded projects.