An Exploratory Analysis on Visual Counterfeits Using Conv-LSTM Hybrid Architecture

MOHAMMAD FARUKH HASHMI1, (Member, IEEE), B. KIRAN KUMAR ASHISH2, AVINASH G. KESKAR3, (Senior Member, IEEE), NEERAJ Dhanraji Bokde4, JIN HEE YOON5, AND ZONG WOO GEEM6, (Member, IEEE)

1Department of Electronics and Communication Engineering, National Institute of Technology, Warangal 506004, India
2Tericsoft Technology, Hyderabad 500036, India
3Department of Electronics and Communication Engineering, Visvesvaraya National Institute of Technology, Nagpur 440010, India
4Department of Engineering, Renewable Energy and Thermodynamics, Aarhus University, 8000 Aarhus, Denmark
5School of Mathematics and Statistics, Sejong University, Seoul 05006, South Korea
6Department of Energy IT, Gachon University, Seongnam 13120, South Korea

Corresponding author: Zong Woo Geem (geem@gachon.ac.kr)

This work was supported in part by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education under Grant NRF-2017R1D1A1B03034813, and in part by the Energy Cloud Research and Development Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT, under Grant 2019m3f2a1073164.

ABSTRACT In recent years, with the advancements in the Deep Learning realm, it has been easy to create and generate synthetically the face swaps from GANs and other tools, which are very realistic, leaving few traces which are unclassifiable by human eyes. These are known as ‘DeepFakes’ and most of them are anchored in video formats. Such realistic fake videos and images are used to create a ruckus and affect the quality of public discourse on sensitive issues; defaming one’s profile, political distress, blackmailing and many more fake cyber terrorisms are envisioned. This work proposes a microscopic-typo comparison of video frames. This temporal-detection pipeline compares very minute visual traces on the faces of real and fake frames using Convolutional Neural Network (CNN) and stores the abnormal features for training. A total of 512 facial landmarks were extracted and compared. Parameters such as eye-blinking lip-synch; eyebrows movement, and position, are few main deciding factors that classify into real or counterfeit visual data. The Recurrent Neural Network (RNN) pipeline learns based on these features-fed inputs and then evaluates the visual data. The model was trained with the network of videos consisting of their real and fake, collected from multiple websites. The proposed algorithm and designed network set a new benchmark for detecting the visual counterfeits and show how this system can achieve competitive results on any fake generated video or image.

INDEX TERMS DeepFakes, generative adversarial network (GANs), facial landmarks, convolutional neural networks (CNN), recurrent neural network (RNN), visual counterfeits.

I. INTRODUCTION

Image or Video manipulation [1], [2] is been carried since photography and videography were born. The artifact platforms for images and videos, powerful editing tools have been playing a crucial role in image/video manipulation and animations [3], [4]. However, these visual artifacts are easily detectable and distinguishable with human-centric vision, and hence do not have much scope in producing them. Started with animations, commercial elements for promoting and entertainment purposes, with rapid advancement in technology, using image processing techniques, production of face warping [5], [6] on target from source for a realistic appearance emerged on large scale in recent months [7], [8]. Now, fake multimedia has become a central problem, especially after the advent of the so-called DeepFakes started revolving in the social media platforms by defaming famous personalities [7], [9].

DeepFakes have become a new form of altering reality that is spreading very fast than expected [10]. DeepFakes is a machine learning technology that fabricates or manipulates video and audio recordings to show people doing and saying things that they never did or said. DeepFakes appear authentic and realistic, but they are not [5], [8], [10]–[12]. It could be
superimposing a face on to a body, so it looks like they are doing something that they have never done [10]. There are two methods of altering a person’s identity with DeepFakes: one is counterfeiting images and videos and another one is counterfeiting audio. Visual DeepFakes are much more complicated to identify than Audio DeepFakes. Most of the existing DeepFakes are in form of vision-based and are easily manipulated for a realistic look [5], [7]. Generally, DeepFakes was limited to imitate actors and politicians to say funny things. As the improvement in the accuracy begins, it has spread to circulate fake news and create ruckus in society by disturbing peace and harmony. The recent troubling floating in the media is the swaying away from the opinions during an election or implicit a person in a crime [13]. These manipulations of facial features of any historical figure, politician, sportsmen, or any CEO by synthesizing a re-enacted face swapped by the movements and actions of another person are realistic in such a way where any human vision cannot distinguish a real or fake video [8].

Today, with recent advancements in deep learning, it has become very easy to build a model using the existing photos and videos of a person [12]. Here, people perform face detection on each of these images using various real-time face detection algorithms. The backbone of a DeepFake is the deep neural networks trained on faces that outputs the facial expressions of the source to the target after proper post-processing, resulting in a high level of realism. To do this, one existing method is to use an autoencoder [2], [8] which is a convolutional network that tries to reconstruct the input image and lets it learn a lower-dimensional representation of the input image which will be later used to swap the faces. Specifically, DeepFakes uses one encoder and two decoders. During training it trains two networks; both the networks share a common encoder and different decoders. The encoder transforms the input image into a base vector which is a set of numbers that identify the face. The decoder transforms that vector back into the image. There is an error function that measures how good that transformation works. The model then learns to lower that training error. During training, both images are trained with the same encoder but two different decoders. After the network training is done, the network is fed with a video which is a collection of image frames one frame at a time. Finally, the concatenation of all the images together and the video results are displayed.

If this kind of passive information is used, just photos and videos that are out there, that’s the key to scaling to anyone as shown in Figure 1a. A sample dataset of DeepFakes creation on actors taken from YouTube is released by Google AI as shown in Figure 1b. These examples shown below describes how deep the images are in a realistic view with a perfect warping of target faces onto the source faces using GANs.

II. BACKGROUND AND RELATED WORK
A. DeepFakes
At the end of 2017 and the starting of 2018 DeepFakes came out. This is a deep learning approach to swap faces into videos. Comparing the progress of GANs [12], in four years the world has upgraded from blurry to photorealistic faces. To create a DeepFake you just need a few thousand frames of training data both for the source face and the target face [8], [14]. After the training face, you will have a trained swap face network. So, you start with the trained network and then the video you wish to modify and it might take minutes to hours depending on the length of the video you wish to alter [5], [10], [13], [15], [16].

One of the popular DeepFakes approaches uses a machine learning approach known as auto-encoders which has three parts [2]. The first is the encoder and the second is the encoding or the bottleneck and the third is the decoder. It’s the job of the encoder to take a high dimensional input image and compress it down into a semantically meaningful representation in the encoding and it is the job of the decoder to reconstruct that encoding to make an approximation for the input image. It gives us an unsupervised way to learn a semantically meaningful representation of the data and by comparing the differences between the input and the output data that provides us an error signal which can be used to update the weights in the network during training. Here the encoder is common to both the two faces. So, it shares weights and learns representations and two decoders are trained individually on two people. At the production time, person A is taken and encoded, then the decoder is used for person B to reconstruct the input image that has the effect of swapping the faces across the two videos [2], [8], [17], [18].

B. DIGITAL MEDIA FORENSICS
It is the process of identifying, preserving, analyzing and presenting digital evidence [9], [10]. Since the 1970’s the field of digital forensics has evolved to keep up with the widespread adoption of technology. The use of computers for financial crimes in the 1980s helped shape digital forensics methods with what they are today. Due to the rise of large datasets such as ImageNet [19] which were created for further research in computer vision community such as object detection and image classification. It contains millions of manually labeled examples across thousands of classes and then in 2012 a breakthrough in performance in image classification AlexNet [20]. It used a deep learning technique for large scale visual recognition challenge that uses ImageNet. It provided such significant gain in performance that is essentially reshaped the computer vision community. Two years after AlexNet [20] was proposed, Ian Goodfellow [12] published a paper on generative adversarial networks and GANs are broadly applicable because they give new ways to train deep neural networks, they give us ways to generate new synthetic data for training. What’s more important is the ability of GANs to generate synthetic faces [5], [9], [21]. In 2016 an approach called face to face was published and this uses real-time tracking to monitor both a source and a target face and to transfer expressions across them [18]. The techniques used here are as simple as simple Image processing techniques. The accuracy levels are very low and
an impossible task to identify/classify it into ‘real’ or ‘fake’. The recent work includes tracing the IP address and history of the source from which the video is playing [22], however, this would become a mere challenge and time consuming to track and trace each suspected IP address.

### C. VISUAL SYNTHESIS

The availability of a huge number of datasets on DeepFakes can now be used to train a simple classification model or even objection detection models of RPN or segmentation. Transfer Learning such as InceptionV3 and ResNet models...
can be used for advanced training [10], [23]. The CNN-based system with a global pooling layer which computes statistical equations such as mean, variance, maximum and minimum pool using a pre-trained weight from ImageNet [19] database for skipping common features and reducing computation power. This powerful method with different architectures using Transfer Learning such as ResNet, Inception V3, XceptionNet, ShuffleNet, BubbleNet.

Most of the DeepFake datasets are video type, existing state-of-art methods are using LSTM to classify videos into ‘real’ or ‘fake’ [6], [10], [18], [23], [24]. Video classification methods have evolved from hand craft features until 2D and recent 3D Conv nets [25]–[27]. Recurrent neural networks have also been utilized to model video sequences. Siamese networks in a supervised way are been used in a few cases for few-shot learning and one-shot learning. The training video dataset is taken as one episode and trained. Vinyals et al. [28] proposed metric learning for a few-shot classification by measuring the distance between the sequential vectors from the attention kernel. Zhu and Yang [29] proposed compound memory networks for few-shot learning. This network learns a fixed size matrix for a variable-length video-sequence. This architecture is a variant of a key-value memory network. Visual features are stored in the key part and its corresponding labels are stored in the value part. This model, however, isn’t enough to train in the micro-level since it uses distance to plot the differences to differentiate the class branch and it is obvious the differentiate cannot be done with few shots of learning and requires an in-depth micro-level learning framework. The differences are very minute and thanks to GANs for that, a micro difference leads to differentiate the video into ‘real’ or ‘fake’.

Some methods include training the neural networks for video analysis where 3D ConvNets [26] are included to extract temporal information but are much time-consuming. Xu et al. [30] acclimate frame-wise features in ConvNets using VLAD pooling for video classification. This has shown great advantages than conventional average pooling. Sutskever et al. [31] adopt a more sophisticated sequence to sequence using up-convolution and down-convolution which uses multi-layered RNN to encode input frame into a hidden state. Another layer of RNN takes this encoded state as input and decodes it into a sequence of outputs. In the recent advancements, encoder-decoder, LSTM autoencoders are being used to pact having a very large-scale dataset due to its efficiency in training and decent performance than the rest [32], [33]. With the recent advancements, the above methods are being used in the Image Captioning applications where LSTM is being used to understand the scenario behind the frames. LSTM is mainly used as an RNN language model to generate a sequence of words framewise based on visual features extracted framewise. The model deploys an encoder-decoder network for image captioning. Also called as vision language, a combination of visual features and language encoded framewise using natural language unit (NLU). However, to the best of our knowledge, the above-mentioned models are heavyweights and memory consuming models that require high-end GPU systems to train on large scale datasets. One-shot learning and few-shot learning is generally used for small scale dataset. Deep fakes require much in-depth training in order to persist the features at a micro-level. Therefore, CNN and LSTM are preferred with a large amount of training rather than a quick training methodology. RNN would extract the features from the video frames as a whole where the features get diluted since the whole image frame is given for extraction. This would consume the overloading of memory space and computational resources. The overall accuracy is somewhat satisfactory, i.e., mAP 85% accuracy. Datasets may be available in large numbers, but the visual forgery is in microscopic scale and specific parameters work precisely rather than a global feature scaling.

**D. ARTIFACTS FROM GANs**

A reverse engineering technique is a useful read the patterns in GANs generated DeepFakes [5]. While the GAN network [12] is being trained to generate DeepFakes [21], the features that are impeded to generate realistic fake images/videos need to get stored under the ‘Fake’ class. This feature vector can differentiate between ‘real’ and ‘fake’ videos. The CNN feature extractor [34] used here to extract the features stores in the form of an array containing all the essential image warping features to generate a DeepFake. These features of original real videos also follow the same procedure and both features under their respective labels are trained using CNN architecture. This can distinguish the ‘real’ and ‘fake’ video base on the features extracted. One such best GAN network [12] for extracting facial attributes during face warping is the StarGAN [35] network which is used by Face App for swapping users face to famous personality’s faces [16], [36].

The computation power that takes is much larger when compared with all other state-of-art methods. Often the features at the end while merging of fake into real are mostly similar. Hence, based on the density of the videos, we must choose a trade-off for ending the storage of features from a particular hidden layer. The overall accuracy decreases and is not reliable because the Black Box is indistinguishable on what kind of features is extracted in each hidden layer. The patterns can be visualized on training, but the exact depth is unknown.

Zhang et al. [37] find that classifiers generalize poorly between GAN models. The purpose AutoGAN model for generating images which consists of generator and discriminator architecture. The other work includes detecting GAN images using hand-crafted co-occurrence features and anomaly detection models on faces by isolating faces using face detector models. Most of the recent works include GANs generated images and videos and detecting counterfeits using GANs itself from both GANs generated deepfakes and GANs detection of deepfakes. Bau et al. [38] concluded that GANs have limited role and capacity in the generation and analyzed that the pre-trained GANs model is
unable to grab the image structures from the given datasets. Having limitations in a generation, it is obvious that the GANs model is not so reliable in detecting deepfakes alone. Recent works include state-of-the-art architectures namely, ProGAN [14], StyleGAN [39], BigGAN [40] trained on ImageNet [20] dataset. As mentioned earlier, the training needs to be computed at micro-level features for classifying deepfakes. To perform micro-level feature extraction, the network needs to extract high-frequency details framewise [41]. StarGAN injects large, per-pixel noise in each frame to introduce high frequency. BigGAN uses self-attention layers [42], [43] on very large scale datasets. Recent works include image-to-image translation approaches such as GauGAN [44], StarGAN [39], and CycleGAN [36]. Wang et al. [45] adapt frequency analysis on each dataset with different GAN approaches. They performed CNN image synthesis with a simple form of high-pass filtering for subtracting median blur version frame-wise and Fourier transform for extracting informative visualization. The real image spectra and their distinct patterns are visible by CNN generated models. Agarwal et al. [46] proposed a novel approach by matching the lip-synch movement and mouth shape. The main focus is on the visemes which are occasionally inconsistent with a spoken phoneme. The sounds of M (mama), B (baba), or P (papa) which the mouth must completely close to pronouncing these phonemes are the primary focus.

However, this is research concludes a similar approach but with grasping almost all facial movements. Generally, deep fakes are the face swaps and the mouth movements are micro variations. Mouth shape alone can not help in accurately detecting the deep fakes. Other than these visemes, the model would not give an accurate result. Hence, mouth shape alone is not sufficient to conclude the best possible way for classifying deepfakes.

E. CNN-BASED VISUAL MANIPULATIONS

Recent works have addressed CNN-based deepfakes detection using basic Conv nets training of real and fake images. These models are not reliable since the deepfakes variations are at the microlevel and need to be compared framewise to detect the anomaly. Rossler et al. [47] evaluated methods on detecting facial manipulations, while they have shown simple CNN classifiers to classify deep fakes by using the same model to generate deep fakes, but these models fail to generalize the models are deep fakes datasets. Marra et al. [48] showed a similar approach that image classifiers can create and detect deep fakes, but the same problem and cross-model transfer are not considered. Cozzolino et al. [49] concluded that forensic classifiers are very poor between models and accomplish near-chance performance in classifying deepfakes. They propose a new representation learning model based on autoencoders in zero and low shot training regimes. This work is similar to Wang et al. [45] the first one took an orthogonal approach. Both try to improve the transfer learning performance but the latter one tries to study the performance of simple baseline classifiers under training and testing conditions for CNN-based image generations. Researchers have shown enough work that common CNN models contain artifacts that reduce their representational power. The work mainly focused on the up-sampling and down-sampling approach. Durall et al. [50] proposed spectral regularization which is based on spectral effects on CNN generated adversarial networks. The power spectrum uses the input data to generate similar realistic data with a slight tweak in the power spectrum. Even the low-resolution data can be used in this approach to generate and detect the need. This approach is similar to the proposed method while the difference is in the plotting spectrum in the first one and plotting the motion framewise in the latter one. Both works promise high accuracy levels. Mittal et al. [51] proposed the Siamese network on the audio-visual dataset and visual datasets. This further includes audio and video of moving lip-synch in videos. It includes modality embeddings and emotion embeddings given to triplet loss.

This research is purely based on facial analysis at the micro-level. Each parameter extracted is dug in-depth and trained at a high-frequency notion [41]. Alongside the facial landmark patterns motion is trained on a decent bandwidth scale frame-wise, which makes this work different from other SOTA.

F. ATTENTION NETWORKS

The best architecture used for image captioning to the best of our architecture is the attention network. Xu et al. [52] adopt the extraction of convolution features in the encoder section. Frame-wise feature extraction is done from every single raw image and for each frame, the model generates a caption encoded as a sequence of total encoded words. At the decoder end, the LSTM network produces the caption by generating each word on a context vector. The proposed architecture by the authors is inspired by the attention networks since the main reason for choosing this network is the ‘forget’ mechanism which saves the memory and would not become a burden for training and resource usage. However, the convolutional features are implemented on a selected part such as face but not the whole frame using a face detector. This extracts the micro-level features which cannot be detected by the landmarks ‘alone’. These conv-features are combined with the landmark movements which makes this work unique.

III. TECHNICAL APPROACH

This paper gives an essence of combining two-novel architecture esteems, i.e., facial landmarks movement and convolutional long short-term memory (Conv-LSTM). The complete layout of the architecture is shown in Figure 2. This pipeline gives the essence of the stepwise framework and tracks to train the model from scratch till the prediction layer.

Using convolutional drift neural network architecture, the features are extracted from the frames, without saving the frames for memory constraints that require minimal training to achieve competitive performance on Spatio-temporal tasks. LSTM alone as a feature extractor and handling
temporal data tend to be an expensive task to train. Hence, combining CNN for feature extraction and LSTM for storing the feature vector for frames [53] would be efficient for training such a humongous amount of data without any memory expense. LSTMs and GRUs are the most common recurrent neural networks used to compute temporal sequence problems. The two are having different data flows with a common component which is called a memory state. Mangal et al. [54] concluded that GRUs takes less time than LSTMs and Bidirectional RNN in training limited datasets. However, LSTM outperforms GRUs in training and descent into loss with just having few epochs [54]. The main reason for using LSTM is memory constraints. LSTM has three gates (input, output, forget gates) whereas GRU has two gates (reset and update gates). Since accuracy is critical in this work, LSTM is being in place. The Forget gate in LSTM determines which part of the previous cell state to retain, the Input gate determines the amount of new memory to be added. These two gates are independent of each other, meaning that the amount of new information added through the Input gate is completely independent of the information retained through the Forget gate [54], [55].

A. FACIAL LANDMARKS

Given the input dataset, i.e., video frames, the network runs the face detector algorithm to locate the face region in the frame \( k \). The face parametric model denoted by \( S(0) \) represents the mean face shape in the frame. The landmark vertices \( S(0) \) are projected onto the frame to determine the initial 2D landmark locations as well as their visibility in the frame. The alignment of 2D landmarks on the frame is done via 128 landmark vertices. The initial 2D locations of these landmarks are denoted by \( X(0) \in R^{136} \) [10].

Besides, a dense set of landmark vertices is used to evaluate dense face feature descriptors, to capture more global and deep insight information of the face and produce more robust results in the facial movements. A dense sample of the face mesh is used to obtain 512 landmarks vertices, which includes the previous 128 alignments. The landmark locations are denoted by \( U(0) \in R^{320} \) and their visibility in the frame is indicated using a binary vector \( V(0) \in R^{160} \). During the training, updating landmarks in each frame \( k^{th} \) iteration evaluates the dense face feature descriptors of the dense landmarks using their current locations \( U(k) \) and visibility on the frame \( V(k) \). The dense face features descriptors are concatenated into a feature vector \( F(k) \in R^{5120} \). For invisible landmarks, such as extreme side-angles, dark pixels, their corresponding alignments/components \( F(k) \) are set to zero. The target location defined by \( \hat{X}(k+1) \) which improves the accuracy of forgery detection are customized since those are the key parts in detecting the swapping from real ones. The camera parameters or the angle at which the target face is located is given \( w_{k+1} \). The proposed solution adapts the state-of-art [10], [23] approach to determine \( w_{k+1} \) and \( X(k+1) \), by computing their displacements proposed as a linear function \( w_{k+1} - w_k \) and \( \hat{X}(k+1) - X(k) \) of dense face feature
The feature vectors are shot from the CNN feature extractor vector \([4], [34]\) are fed as input to the LSTM cell inputs. 

In LSTM, there is a mean pooling layer to extract frame-level visual features without saving it by holding it temporarily in its memory unit and the output video-level feature vector \([4]\) is compared with movement detected of each part in the face. This is compared with the fake video, where landmarks are aligned in 2D space in its memory unit and the output video-level feature vector. 

The proposed approach has a 3-phase architectural design, which is designed to reduce the memory constraints on physical GPU servers. AI-generated DeepFake videos are given to training and the reflection is given to testing.

### IV. RESULTS AND DISCUSSION

The first step would be the facial landmarks extraction from each frame \(v\). 128D and 512D landmarks are extracted...
in 2D space and training both of them separately to checks the accuracies of both the facial alignments. The initial step would be the face detection of the frame \([v_1, v_2, v_3, \ldots, v_n]\). Then the landmarks extractor is applied to extract landmarks and is saved under target label \(T\). For detecting face swapping, the authors mainly considered the movements of ideal parts such as eyes, nose, eyebrows, and mouth. The third experiment is training with these specific parts of both 128D and 512D landmarks. The person’s movement in ‘Real’ and ‘Fake’ class videos are noted in an Nd-array and are sent for training for each frame \(v\).
2) CNN FEATURE EXTRACTOR
The further advance step is feature extraction from the facial landmarks and their locations. The in-depth insights are extracted using CNN feature extractors alongside with the facial landmark movements for better and precise accuracies, to avoid false positives and false negatives. CNN layers [11], [56] are used to extract features and give the feature vector to the LSTM cell as input. The image size is set as (224, 224). The network is loaded with the face detection model through Multi-Cascaded Convolutional Network (MTCNN) architecture. The bottleneck feature extraction method is initialized in the CNN layers with GlobalAverage Pooling of the features extracted from the video frame \( v \). The features are stored as Nd-array and are combined with a frame-level feature vector for each frame \( v \). The CNN network starts with the Sequential model, first and second layers consist of Conv2D filters of random numbers with size \( 3 \times 3 \) followed by ReLu activation function, with the first layer having Padding of ‘1’. Based on the number of frames \( v \), the filters are taken in the hidden layer from here, i.e., Conv2D filters based on frames with size \( 3 \times 3 \) and padding ‘1’ followed by ReLu activation function. MaxPooling2D is then applied with a pool size of \( 2 \times 2 \). Dropout function is applied to remove unnecessary features for eliminating the memory constraints. The final output layer consists of Flatten the features extracted with is fed to Dense node of size 256 and ReLu activation function. A final Dropout is given to clean the features for a final time. This output is fed to the LSTM cell unit.

3) LSTM UNIT
Due to heavy and larger datasets, extracting frames from the videos and give them to the network consumes loads of memory and requires very high computational power. LSTM network works best for video classification and it allows storing the patterns or features without saving the frame in the physical server. It extracts the frames from the videos, extracts the facial landmarks and features but never saves the frame physically. The pattern recorded by the LST memory cell stores the pattern and learns the patterns until the next pattern arrives and then the current gets erased from the cell [4], [7], [18]. These patterns are given weights and are stored in the LSTM memory units. The loop continues throughout the video frames \( v \). The facial extractions of faces in frames is shown in Figure 4, which tracks the movement of essential parts such as eyes, eyebrows, nose, mouth and head using 512 facial embedding.

The LSTM network is built using recurrent neural network functions. Under the Sequential model, LSTM nodes are initiated of random choice based on the input frames. The first LSTM node with return sequences and Recurrent Dropout of ‘0.2’ is added, followed by Dropout function. This loop is continued throughout the total number of layers in the network. Following the loop of initial layers is the LSTM node with Recurrent Dropout of ‘0.2’, followed by Dropout function. This is given as input to Dense layers of total 256 with activation function ReLu. The final layers consist of Dense layers with several classes, i.e., ‘2’ and with activation function Sigmoid since it is binary class. Applying more complex convolutions for CNN and LSTM pipeline for prediction, the authors now concatenate the features coming down from CNN feature extractor and LSTM for training. 1D-Conv layers with ReLu activation function is initialized for embedded sequence followed by MaxPooling x 5 for 1D-Conv. These are appended and are concatenated with the 1D-Conv of filter size ‘5’ and ReLu activation function.
A Dropout + MaxPooling for size ‘5’ is applied and appended with the master vector. Now, a 1D-Conv layer with filter size ‘5’ and ReLu activation function followed with a Dropout and MaxPoolingx 30 which is flattened into 1024 vector.
Now, the Dense layer with ReLu activation function is used followed by Dropout which gives 512 vectors. The above layers are appended and concatenated with each other in a loop for each frame $v$. Finally, Dense layer with ReLu activation function and Dropout with Sigmoid of $[0, 1]$ which gives the prediction.

**B. TRAINING ON DFDC DATASET**

The dataset [57] is collected from Kaggle DeepFake Detection Challenge. The total dataset consists of 470 GB of videos which are AI-generated datasets of multiple persons. The whole dataset is given for training to the Conv-LSTM network into 2 classes: ‘Real’ and ‘Fake’.

**C. RESULTS AND ANALYSIS**

Some of the major dataset on deep fakes has been tested such as Google AI DeepFake raw dataset [58], Kaggle DFDC testing dataset [57], Youtube raw videos from Google AI deepfake dataset [58], Celeb-DF dataset, shown in Figure 5 and 6.

The training almost lasted for 7 days on the humongous data (Kaggle DFDC training set). The trained model is tested on different datasets for testing the accuracy. Some standard deepfake datasets are mentioned above. Figure 7 describes the prediction results on Google AI deepfake dataset [58]. Google AI deepfake dataset is humongous as compared with other testing datasets selected by the authors. As visualized in the Figure 7, landmarks play a pivotal role in identifying the facial parts movements which can be visualized in a graph as shown in Figure 8. The authors have plotted the variations in each frame with respective to each parameter such as eyebrows, nose movement, mouth movement consistency as shown in Figure 6. It clearly shows the spike in variations in each frame. Since, the original face is being warped with
the another, the facial parameters does not be consistent with respect to the original target facial movements. A normal original video would have quite consistent waves of consistency with few decent spikes unlike heavy spikes as shown in Figure 8 on a visual deepfake video. The variations in each frame shall be the deciding factor for classifying it into ‘real’ or ‘fake’.

As we can see the subjective results in Table 2, side facial movements will fail to classify since the model needs to analyze the whole facial part for comparing proper movements frame-wise. There are some deepfake videos where just one side of the face is shamed and the other half such as eye, eyebrow, the cheek is kept normal. In these extreme cases of forgery, the model gets stuck at low accuracy levels. Hence, a proper full-face video is necessary to get proper results.

D. PERFORMANCE METRICS

Figure 9 resumes the performance of the network trained on the Kaggle DFDC dataset [58]. The dataset is very huge weighing around 470 GB in size. Overall 50 training videos, each with ~ 10 GB in size is being trained. The training patch accuracy is noted at 95.12% and the validation patch at 89.01%. The network having Adam optimizer, the loss curve has shown a decreasing behavior. The network took $1024 \times 1024$ image patches to train the model. The batch size was defined as 1 to prevent memory allocation errors. Overall the model was trained till 50K epochs which took

![Similarity Scores](image-url)
over a week. The learning rate which started at 0.100 at training initiating was reduced to half of it in each further 10K step. Further, the loss did not decrease post 46K epoch.

The widely used model evaluation metrics are average precision and F1 score. The face detector compares the predicting boundary boxes and landmarks inside it according to

FIGURE 9. Loss graph during Training period on Kaggle DFDC dataset [57].

FIGURE 10. Landmarks motion framewise taken with reference with real and fake videos. The red line represents real video graph and blue line fake video graph.
the intersection over union (IOU), where its parameters get updated at each epoch. F1 score is used for calculating the success rate of precision and recall. Precision is the ratio of actual matches and recall is defined as the ratio of correct predictions to that of total ground truths. However, neither of them is sufficient to measure the performance of the network. F1 score is calculated with the precision and recall as dependent parameters to compute the evaluation of the network on data. F1 score is termed by the parameters true positives (TP) as correct predictions, false negatives (FN) as false non-detections, false positives (FP) as false correct predictions. The mathematical computations for the above-mentioned parameters are:

\[
\text{Precision} = \frac{TP}{TP + FP} \quad (11)
\]

\[
\text{Recall} = \frac{TP}{TP + FN} \quad (12)
\]

\[
F1 \text{ score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (13)
\]

### TABLE 3. F1 score results on each approach validated on the testing dataset.

| Model                      | Precision | Recall | F1 Score |
|----------------------------|-----------|--------|----------|
| Eye Blinks                 | 0.3       | 0.21   | 0.24     |
| Eye Blinks + Eyebrows Movement | 0.43     | 0.35   | 0.38     |
| Eyes + Eyebrows + Head Movement | 0.61     | 0.58   | 0.59     |
| Eyes + Eyebrows + Head + Mouth Movement | 0.97   | 0.92   | 0.95     |
| Side Face Movements        | 0.87      | 0.65   | 0.74     |

The model trained and saved is a heavyweight model and computationally complex. Training is performed on Nvidia 1080x TI GPU for 7 days without disruption. Inference speed on GPU systems is around ~ 0.5 ms and on CPU systems the speed is ~ 10 ms.

As shown in Figure 10, the labeled dataset from DFDC which is used for training, the landmarks motion framewise is plotted and visualized. The unusual massive spikes can found in fake videos. These frame-wise results of landmarks motion are the key but not alone sufficient to classify, hence, along with the landmarks motion results, CNN feature results which are results of the landmarks patterns given to the network are combined to classify the video into ‘real’ and ‘fake’.

### V. CONCLUSION

In this paper, the authors have designed a benchmark-setting architecture Conv-LSTM which uses facial landmarks and convolutional features to automatically detect the visual forgery in videos and images. Visual forgery on videos and images can be automatically detected with precise accuracies from the proposed approach. As this architecture was built and trained on a humongous amount of data, any visual forgery can be easily detected giving the best accuracy when compared to other art-of-states. The proposed approach is simple and best in terms of usage although the black-box training inside neural nets is much complex, memory constraints are addressed as the frames are not saved from the videos during the training process.

This research has addressed on detecting the visual forgery through convolutional methods, the further scope of this research includes the reverse engineering of AI-generated visual forgery by saving the patterns and features while generating visual forgery and directly training them under forgery class. This would be much complex than expected but opens the gate for light-weight model approaches.

However, the primary limitations of this work are having real videos of the subject. Although the model works well without having any real videos of the subject, as comparing the variations from frame-to-frame as shown in Figure 8. But there would be a challenge in predicting the ‘real’ or ‘fake’ classification without real video for having precise results. The real video of the subject would act as a backbone for the predicted results. The next major limitation is the position of the subject in the video. Zoomed angles are preferred rather than side angles since the accuracy levels are low in the side angles. The model is heavyweight in terms of computational complexity. The video must be at least 10 seconds for getting decent results, 15-30 seconds would be much better for higher accuracy.

### VI. FUTURE SCOPE AND FUTURE RESEARCH

The authors have performed various computational training with several architectures on just videos of ‘real’ and ‘fake’ classes resulting in high training accuracy but very low prediction accuracy. This is due to the mismatch of the process for identifying the DeepFakes patterns in videos. Having got the proper pattern recognition obtained by the authors, i.e., the facial landmark patterns approach and CNN feature extraction on specific main-course parts, these features can be further applied in the Reinforcement Learning architecture where the rules (here the features of ‘real’ and ‘fake’) are given to the system and humongous amount of data is given for training and testing. This can have a future scope in successfully nullifying any kind of DeepFake videos generated in the future course. The authors will be contributing by extending this approach using Reinforcement Learning in the upfront.

### REFERENCES

[1] G. Lample, N. Zeghidour, N. Usunier, A. Bordes, L. Denoyer, and M. Ranzato, “Fader networks: Manipulating images by sliding attributes,” in Proc. Adv. Neural Inf. Process. Syst., 2017, pp. 5967–5976.

[2] Y. Liao, Y. Wang, and Y. Liu, “Graph regularized auto-encoders for image representation,” IEEE Trans. Image Process., vol. 26, no. 6, pp. 2839–2852, Jun. 2016.

[3] H. Averbuch-Elor, D. Cohen-Or, J. Kopf, and M. F. Cohen, “Bringing portraits to life,” ACM Trans. Graph., vol. 36, no. 6, pp. 1–13, Nov. 2017.

[4] P. Bestagini, S. Milani, M. Tagliasacchi, and S. Tubaro, “Local tampering detection in video sequences,” in Proc. IEEE 15th Int. Workshop Multimedia Signal Process. (MMSP), Sep. 2013, pp. 488–493.
V. Conotter, E. Bodnari, G. Boato, and H. Farid, “Physiologically-based detection of computer generated faces in video,” in Proc. IEEE Int. Conf. Image Process. (ICIP), Oct. 2014, pp. 248–252.

K. Dale, K. Sunkavalli, M. K. Johnson, D. Vlasic, W. Matusik, and H. Pfister, “Video face replacement,” ACM Trans. Graph., vol. 30, no. 6, pp. 1–10, Dec. 2011.

M. Brundage et al., “The malicious use of artificial intelligence: Forecasting, prevention, and mitigation,” 2018, arXiv:1802.07228. [Online]. Available: http://arxiv.org/abs/1802.07228

I. Laptev, M. Marszałek, C. Schmid, and B. Rozenfeld, “Learning realistic human actions from movies,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2008, pp. 1–8.

H. Farid, Photo Forensics. Cambridge, MA, USA: MIT Press, 2016.

I. Sutskever, O. Vinyals, and A. G. Hauptmann, “A discriminative CNN video classification guided by batch normalized LSTM translator.” 2017, arXiv:1707.04045. [Online]. Available: http://arxiv.org/abs/1707.04045

Keras. Accessed: Dec. 31, 2019. [Online]. Available: https://www.keras.io/

I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” in Proc. Adv. Neural Inf. Process. Syst., 2014, pp. 2672–2680.

The Outline: Experts Fear Face Swapping Tech Could Start an Out?zd=1&zi=hbmf4svs

T. Karras, T. Aila, S. Laine, and J. Lehtinen, “Progressive growing of GANs for improved quality, stability, and variation,” 2017, arXiv:1710.10196. [Online]. Available: http://arxiv.org/abs/1710.10196

Faceapp. Accessed: Dec. 31, 2019. [Online]. Available: https://www. faceapp.com/

H. Farid, “CNN-generated images are surprisingly easy to spot...For now,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 38–45.

A. Brock, D. Donahue, and K. Simonyan, “Large scaleGAN training for high fidelity natural image synthesis,” 2018, arXiv:1809.11096. [Online]. Available: http://arxiv.org/abs/1809.11096

N. S. Ivanov, A. V. Arzhakhov, and V. G. Ivanenko, “Combining deep learning and super-resolution algorithms for deep fake detection,” in Proc. IEEE Conf. Russian Young Researchers Electr. Electron. Eng. (EICONRus), Jan. 2020, pp. 326–328.

X. Wang, D. Girshick, A. Gupta, and K. He, “Non-local neural networks,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 7794–7803.

H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena, “Self-attention generative adversarial networks,” 2018, arXiv:1805.08318. [Online]. Available: http://arxiv.org/abs/1805.08318

T. Park, M.-Y. Liu, T.-C. Wang, and Y.-J. Zhu, “Semantic image synthesis with spatially-adaptive normalization,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 2337–2346.

S.-Y. Wang, O. Wang, R. Zhang, A. Owens, and A. A. Efros, “CNN-generated images are surprisingly easy to spot...For now,” 2019, arXiv:1912.11035. [Online]. Available: http://arxiv.org/abs/1912.11035

S. Agarwal, H. Farid, Y. Gu, M. He, K. Nagano, and H. Li, “Protecting world leaders against deep fakes,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops, Jun. 2019, pp. 38–45.

F. Rossler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Niessner, “FaceForensics++: Learning to detect manipulated facial images,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 1–10.

G. Toderici, D. Gragnaniello, D. Cozzolino, and L. Verdoliva, “Detection of GAN-generated fake images over social networks,” in Proc. IEEE Conf. Multimedia Inf. Process. Retr. (MIPR), Apr. 2018, pp. 218–226.
MOHAMMAD FARUKH HASHMI (Member, IEEE) received the B.E. degree in electronics and communication engineering from R. G. P. V Bhopal University, the M.E. degree in digital techniques and instrumentation from SGSITS Indore/R. G. P. V Bhopal University, in 2010, and the Ph.D. degree from the Visvesvaraya National Institute of Technology (VNIT), Nagpur, under the supervision of Dr. Avinash G. Keskar. He is currently an Assistant Professor with the Department of Electronics and Communication Engineering, National Institute of Technology, Warangal. He has published up to 60 articles in international/national journals/conferences. He has a teaching and research experience of ten years. His current research interests include computer vision, machine vision, machine learning, embedded systems, digital signal processing, image processing, and digital IC design. He is a member of LMIETE, ISTE, and LMIAENG.

AVINASH G. KESKAR (Senior Member, IEEE) received the B.E. degree from the Visvesvaraya National Institute of Technology (VNIT), Nagpur, in 1979, and the M.E. degree from IISc, Bengaluru, in 1983. He has 26 years of teaching experience and seven years of industrial experience. He is currently a Professor with the Department of Electronics and Communication Engineering, VNIT Nagpur. His current research interests include computer vision, soft computing, and fuzzy logic. He is a member of IAENG. He is a Senior Member of FIETE, LMISTE, and FIE. He received the Gold Medal during his B.E. and M.E. degrees.

MOHAMMAD FARUKH HASHMI (Member, IEEE) received the B.E. degree in electronics and communication engineering from R. G. P. V Bhopal University, the M.E. degree in digital techniques and instrumentation from SGSITS Indore/R. G. P. V Bhopal University, in 2010, and the Ph.D. degree from the Visvesvaraya National Institute of Technology (VNIT), Nagpur, under the supervision of Dr. Avinash G. Keskar. He is currently an Assistant Professor with the Department of Electronics and Communication Engineering, National Institute of Technology, Warangal. He has published up to 60 articles in international/national journals/conferences. He has a teaching and research experience of ten years. His current research interests include computer vision, machine vision, machine learning, embedded systems, digital signal processing, image processing, and digital IC design. He is a member of LMIETE, ISTE, and LMIAENG.

AVINASH G. KESKAR (Senior Member, IEEE) received the B.E. degree from the Visvesvaraya National Institute of Technology (VNIT), Nagpur, in 1979, and the M.E. degree from IISc, Bengaluru, in 1983. He has 26 years of teaching experience and seven years of industrial experience. He is currently a Professor with the Department of Electronics and Communication Engineering, VNIT Nagpur. His current research interests include computer vision, soft computing, and fuzzy logic. He is a member of IAENG. He is a Senior Member of FIETE, LMISTE, and FIE. He received the Gold Medal during his B.E. and M.E. degrees.

B. KIRAN KUMAR ASHISH received the B.Tech. degree in electronics and communication engineering from the Anurag Group of Institutions, Hyderabad, in 2019. He is currently the Tech Lead at Hexvel Disruptor, an AI-IoT based startup company in India, which deals with leveraging IoT and AI in the Sports realm. Prior to this, he has a year experience as an AI Engineer. He has completed and awarded a certificate in Artificial Intelligence and Machine Learning as an Excellent Performer grade from IIIT-Hyderabad and Talentsprint Combined AI/ML Certification Programme. He has particular expertise in computer vision and been an Investigator in facial recognition, facial anti-spoofing, facial emotion and micro-expressions, GAIT analysis, and deep fakes. He recently published a conference paper at the International Conference on Machine Vision on GAIT analysis. In addition to this work, he currently collaborates with his mentor Dr. Farukh Hashmi and other researchers to build momentum in computer vision space. His research interests include computer vision and pattern recognition.

ZONG WOO GEEUM (Member, IEEE) received the B.Eng. degree from Chung-Ang University, the M.Sc. degree from Johns Hopkins University, and the Ph.D. degree from Korea University. He researched at Virginia Tech; the University of Maryland, College Park; and Johns Hopkins University. He is currently an Associate Professor with the Department of Energy IT, Gachon University, South Korea. He invented a music-inspired optimization algorithm, harmony search, which has been applied to various scientific and engineering problems. His research interest includes phenomenon-mimicking algorithms and their applications to energy, environment, and water fields. He has served for various journals as an Editor (an Associate Editor for Engineering Optimization and a Guest Editor for Swarm and Evolutionary Computation, the International Journal of Bio-Inspired Computation, the Journal of Applied Mathematics, Applied Sciences, and Sustainability).

JIN HEE YOON received the B.S., M.S., and Ph.D. degrees in mathematics from Yonsei University, South Korea. She is currently a Faculty Member with the School of Mathematics and Statistics, Sejong University, Seoul, South Korea. Her research interests include fuzzy regression analysis, fuzzy time series, optimizations, intelligent systems, and deep learning. She is a Board Member of the Korean Institute of Intelligent Systems (KIIS) and has been working as an Associate Editor, the Guest Editor, and an Editorial Board Member of several journals, including SCI and SCIE journals. In addition, she has been regularly working as an Organizer and a Committee Member of several international conferences.

NEERAJ DHANRAJ BOKDE received the Ph.D. degree from the Department of Electronics and Communication, Visvesvaraya National Institute of Technology, Nagpur, India, and the M.E. degree in embedded systems from the EEE Department, Birla Institute of Technology and Science (BITS) at Pilani, Pilani, India. He is currently a Post-doctoral Researcher with the Department of Engineering, Renewable Energy and Thermodynamics, Aarhus University, Denmark. His research interests include time series analysis, data mining, and prediction methodologies. He is serving as a Guest Editor for Energies journal.

IEEE Access

M. F. Hashmi et al.: Exploratory Analysis on Visual Counterfeits Using Conv-LSTM Hybrid Architecture

[52] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio, “Show, attend and tell: Neural image caption generation with visual attention,” in Proc. Int. Conf. Mach. Learn., Jun. 2015, pp. 2048–2057.

[53] A. Verma, V. Mittal, and S. Dawn, “FIND: Fake information and news detections using deep learning,” in Proc. 12th Int. Conf. Contemp. Comput. (IC3). Aug. 2019, pp. 1–7.

[54] S. Mangal, P. Joshi, and R. Modak, “LSTM vs. GRU vs. Bidirectional RNN for script generation,” 2019, arXiv:1908.04332. [Online]. Available: http://arxiv.org/abs/1908.04332

[55] A. A. Mohamed and C. Claudel, “MCRM: Mother compact recurrent memory,” 2018, arXiv:1808.02016. [Online]. Available: http://arxiv.org/abs/1808.02016

[56] M. Abadi et al., “Tensorflow: A system for large-scale machine learning,” in Proc. 12th USENIX Symp. Operating Syst. Design Implement. (OSDI), 2016, pp. 265–283.

[57] Kaggle—Deepfake-Detection-Challenge. Accessed: Dec. 31, 2019. [Online]. Available: https://www.kaggle.com/e/deepfake-detection-challenge/data

[58] Github—Faceforensic. Accessed: Dec. 31, 2019. [Online]. Available: https://github.com/ondyari/FaceForensics/