A Novel Machine Learning based Method for Deepfake Video Detection in Social Media

Alakananda Mitra
Dept. of Computer Science and Engineering
University of North Texas, USA.
Email: AlakanandaMitra@my.unt.edu

Saraju P. Mohanty
Dept. of Computer Science and Engineering
University of North Texas, USA.
Email: saraju.mohanty@unt.edu

Peter Corcoran
School of Engineering and Informatics
National University of Ireland, Galway, Ireland.
Email: peter.corcoran@nuigalway.ie

Elias Kougianos
Dept. of Electrical Engineering
University of North Texas, USA.
Email: elias.kougianos@unt.edu

Abstract—With the advent of deepfake videos, video forgery has become a serious threat. Videos in social media are the most common and serious targets. There are some existing works for detecting deepfake videos but very few attempts have been made for videos in social media. This paper presents a neural network based method to detect fake videos. A model, consisting of a convolutional neural network (CNN) and a classifier network is proposed. Three different structures, XceptionNet, InceptionV3 and Resnet50 have been considered as the CNN modules and a comparative study has been made. Xception Net has been chosen in the proposed model and paired with the proposed classifier for classification. We used the FaceForensics++ dataset to reach the best model. Our model integrated in the algorithm detects compressed videos in social media.

Index Terms—Deepfake, Deep Learning, Depthwise Separable Convolution, Convolutional Neural Network (CNN), Transfer Learning, Social Media, Compressed Video.

I. INTRODUCTION

Artificial intelligence, especially machine learning, manipulates images and videos in such a way that they are often visually indistinguishable from real ones. Among deep learning-based video falsification techniques, deepfake is a serious contender. The term ‘deepfake’ originates from the words ‘deep learning’ and ‘fake’. Use of deep learning networks (DNN) has made the process of creating convincing fake images and videos increasingly easier and faster. In social media, when images or videos are uploaded, they get compressed and resized. Compression causes losses. So, to detect deepfake social media videos we need techniques which will be applicable to highly compressed videos. In this paper we propose a novel method to detect deepfake videos in social media.

The rest of this paper is organized as follows: Section II presents the motivation for our work. Section III focuses on the novel contributions of this paper. Section IV is a review of related works in this field. Our detailed work for deepfake detection is described in Section VI. Section VII presents the theoretical perspective. Section VIII discusses experiments and results, while Section IX states the conclusion and directions for future works.

II. DEEPFAKE IS A SOCIAL AND ECONOMICAL ISSUE

In the last two decades face forgery in multimedia has increased enormously. Among the reported works, an image based approach [1] to generate a video in 1997, face replacement of an actor without changing the expression [2] and real time expression transfer [3] in 2015, are important. In 2017, a Reddit user named Deepfake, created some fake videos using deep learning networks. Use of convolution auto encoders [4] and generative adversarial networks [5] made this forgery so sophisticated that the synthesized videos are often visually indistinguishable from real ones. Smartphone applications to manipulate images, like FaceApp, are easily available to anybody.

This disruptive technological change distorts the truth. Many are intended to be funny, but others are not. They could be a threat to national security, democracy, and an individuals identity. People have started to lose faith in the news or images/videos brought to them by media.

III. NOVEL CONTRIBUTIONS OF THE CURRENT PAPER

In this paper we propose a DNN based framework and an algorithm to detect deepfake videos in social media. A system level overview of the network is shown in Fig.1.

A. The Problem and Challenges Addressed in the Current Paper

The problem addressed in this paper lies in the very origin of how a deepfake video is created. It is hard to distinguish between a real and a deepfake video in social media. Our goal is to model a framework which can detect those deepfake videos in social media generated mostly by an autoencoder. The challenge is threefold: (1) detecting deepfake videos, (2) creating a model applicable to compressed video, and (3) creating a lighter version of it. Our paper addresses the first two problems together.
future work.

each frame contains different features. These features are first extracted and then used as input to a deep learning classifier as CNN models can detect these artifacts. The classifiers are ResNet152 [10], VGG16, Inception V3, DenseNet etc. Certain works are associated with detection techniques based on eye blinking rate [11], noting the difference between head pose [12] of an original video and fake video, and detecting the artifacts of eyes, teeth and face [13]. A general capsule network based method has been proposed to detect manipulated images and videos [14]. A VGG-19 [15] network has been used for latent feature extraction along with a capsule network to detect different spoofs, replay attack etc. Two inception modules along with two classic convolution layers followed by maxpooling layers have been explored [16]. A combined network of CNN and LSTM architectures has been explored in [17]. A DenseNet structure combined with RNN has been used [18]. A blockchain based approach to detect forged videos is proposed in [19]. From Table II, it is evident that not much work has been done for compressed video which is predominantly used in social media. A triplet structure has been used to detect highly compressed videos [20].

V. WHY ARE DEEPFAKES HARD TO DETECT?

There are two main ways to create Deepfake videos - by autoencoders and by generative adversarial networks (GANs). Our method addresses deepfake videos generated by autoencoders, in which the creation of deep fake video consists of three steps - extraction, training and creation.

• In the extraction process all frames from video clips are extracted and faces are identified and aligned.

• The training stage is shown in Fig. 2(a). During training, common features for both image sets are created.

• The creation of a deepfake video frame is shown in Fig. 2(b).

VI. THE PROPOSED NOVEL METHOD FOR DEEPFAKE DETECTION

From Table I it is evident that GoogLeNet, ResNet50, InceptionV3 and Xception Net are smaller size than the other models. We preliminary chose three architectures except GoogLeNet as the feature extractor of our model as those models have better accuracy than GoogLeNet. We then compared the results and finally propose a final model. The end-to-end framework of our proposed method is shown in Fig. 3.

The framework consists of (1) CNN module (2) a classifier network. Three different CNN modules are used initially to get the best feature extractor for compressed video. Finally Xception Net has been used as our model feature extractor.

• Data processing: A dataset of videos each 4 sec long are clipped from the original and manipulated videos. Then frames are extracted from each compressed video with no decompression. We then detected the faces and cropped

### Table I

<table>
<thead>
<tr>
<th>Model</th>
<th>Size (MB)</th>
<th>Parameters (Millions)</th>
<th>Top-1 Accuracy (%)</th>
<th>Top-5 Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>GoogLeNet</td>
<td>40</td>
<td>7</td>
<td>71.8</td>
<td>90.7</td>
</tr>
<tr>
<td>AlexNet</td>
<td>217</td>
<td>60</td>
<td>57.1</td>
<td>84.7</td>
</tr>
<tr>
<td>VGG16</td>
<td>528</td>
<td>138.83</td>
<td>71.3</td>
<td>90.1</td>
</tr>
<tr>
<td>VGG19</td>
<td>549</td>
<td>143.67</td>
<td>71.3</td>
<td>90.0</td>
</tr>
<tr>
<td>InceptionV3</td>
<td>92</td>
<td>23.8</td>
<td>77.9</td>
<td>93.7</td>
</tr>
<tr>
<td>ResNet50</td>
<td>98</td>
<td>25.6</td>
<td>74.9</td>
<td>92.1</td>
</tr>
<tr>
<td>Xception</td>
<td>88</td>
<td>22.9</td>
<td>79.0</td>
<td>94.5</td>
</tr>
</tbody>
</table>

C. The Novelty of the Solution Proposed

In this paper, we propose a novel algorithm with a deep neural model to detect social media deepfake videos with high accuracy. We also consider a small size model as feature extractor so that we can extend our model to edge devices as future work.

IV. RELATED PRIOR WORKS

Most of the solutions proposed for video forensics are for easy manipulations such as copy-move manipulation [7], dropped or duplicated frames [8], or varying interpolation [9]. But the use of auto-encoders or generative adversarial networks made image/video forgery sophisticated. Existing works are presented in Table II.

![Fig. 1. System Level Overview of the Proposed Network.](image_url)
TABLE II
A COMPARATIVE PERSPECTIVE WITH EXISTING WORKS ON DEEPFAKE VIDEO DETECTION.

<table>
<thead>
<tr>
<th>Works</th>
<th>DataSet</th>
<th>Model Features</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Güera and Delp [17]</td>
<td>HOHA</td>
<td>Temporal inconsistencies of deepfake video is taken into account. Inception-V3 + LSTM.</td>
<td>Didn’t take into account of compressed videos.</td>
</tr>
<tr>
<td>Afchur et al. [16]</td>
<td>Downloaded from internet and processed. Mesonet structures - Meso-4 and MesoInception-4 used. 2 inception modules + 2 classic convolution layers + 2 FC layers.</td>
<td>Accuracy is less for highly compressed video.</td>
<td></td>
</tr>
<tr>
<td>Li et al. [21]</td>
<td>UADFV and DeepfakeTIMIT</td>
<td>Face warping artifacts. Used 4 CNN models. Measured resolution inconsistency between the warped face area and face.</td>
<td>Compression has not been considered.</td>
</tr>
<tr>
<td>Nguyen et al. [14]</td>
<td>Four major datasets. VGG-19 + Capsule Network.</td>
<td>Accuracy is low for highly compressed data.</td>
<td></td>
</tr>
</tbody>
</table>

![Fig. 2. Deepfake Video Creation by Autoencoder.](image)

![Fig. 3. A Detailed Representation of the Proposed Model.](image)

the faces from each frame. Finally all frames are normalized and resized as per the input of the various CNN modules. Imagesize is kept at (299, 299, 3) for InceptionV3 and Xception net and (224, 224, 3) for ResNet50. The data processing diagram is shown in Fig. 4.

- **Classification Network**: As the classification network (Fig. 5), we chose a combination of layers for better accuracy. The layers are a GlobalAveragePooling2D layer with 0.5 dropout followed by a fully connected layer with 1024 nodes, 0.5 dropout and ‘relu’ activation and finally a softmax layer which essentially classifies the detected video as real or manipulated. Fig 6 shows how the classifier works.
The Kernel or Filter or Feature Detector is a small matrix of numbers. When it is passed over the input image, new feature maps are generated from the convolution operation between the filter value and the pixel value of the input image at each point \((x, y)\). The complexity of the convolution operation is \(N \times D_G \times D_K^2 \times M\), where \(D_F \times D_F \times M\) is the size of the input image and the filter size is \(D_K \times D_K \times M\). \(M\) is the number of channels in the input image. The size of the feature matrix is denoted by \(D_G \times D_G \times M\). The complexity is decreased in Depthwise Separable Convolution. It divides the convolution operation in two parts (1) Depthwise Convolution - Filtering stage and (2) Pointwise Convolution - Combination stage. In depthwise convolution the complexity is \(M \times D_G^2 \times D_K^2\) while for pointwise convolution it is \(N \times D_G^2 \times D_K^2 \times M\). The total complexity is expressed by the following expression:

\[
\text{Total Complexity} = M \times D_G^2 \times D_K^2 + N \times D_G^2 \times D_K^2 \times M \quad (1)
\]

The relative complexity is the following expression:

\[
\frac{\text{Complexity Depthwise Separable Conv.}}{\text{Complexity Standard Conv.}} = \frac{1}{N} + \frac{1}{D_K^2} \quad (2)
\]

It is obvious from Eq. (2) that the complexity of standard convolution is much higher than the depthwise separable convolution, which implies that Xception Net is much faster and cheaper convolution than standard convolution.

**GlobalAveragePooling Layer:** It helps to reduce the number of parameters and eventually minimizing overfitting. It downsamples by computing mean or average of the width and height dimensions of the input.

**Dropout Layer:** It is very common for a deep network to overfit. The dropout layer prevents the overfitting of a neural network.

**Soft-Max Layer:** In order to predict the class of the video - pristine or manipulated, the softmax layer is used at the end of the network. It takes an \(M\)-dimensional vector and creates another vector of the same size but with values ranging from 0 to 1 making the sum of the values to 1.

**Training Loss:** During training, we minimize the Categorical Cross Entropy Loss to get optimal parameters of
the network to best predict the class. It is a measure of performance of a classification model whose output is the probability ranging from 0 to 1.

VIII. EXPERIMENTAL VALIDATION

A. Dataset

The FaceForensics++ dataset [6] by Google has videos at different compression levels. We used the deepfake videos of this dataset for training and evaluating our model, as it represents a realistic scenario for social media. The dataset details are given in Table III.

<table>
<thead>
<tr>
<th>Dataset Name</th>
<th>Compression = c23</th>
</tr>
</thead>
<tbody>
<tr>
<td>FaceForensics++</td>
<td>1000</td>
</tr>
</tbody>
</table>

B. Experimental Setup

Transfer Learning: We used transfer learning for better accuracy and save training time. All CNN modules are trained on the Imagenet dataset. Resnet50, InceptionV3 and Xception Net were used as feature extractors in our experiment. Lower level layers extract basic features like lines or edges whereas middle or higher layers extract more complex and abstract features and features defining classification. We detected faces and cropped them from each frame of the video to get a better features set. We finally used Xception Net as feature extractor and connected it to a classifier network. We trained the classifier with the dataset and then fine tuned the network end-to-end.

Parameter Settings: During the normalization of each frame the mean and standard deviation are both set to 0.5. The Adam optimizer [22] has been used for training the whole network. The whole work is shown in Fig. 7. The details of parameter settings are shown in Table IV.

<table>
<thead>
<tr>
<th>Training Parameters</th>
<th>Classifier</th>
<th>Fine Tuning</th>
</tr>
</thead>
<tbody>
<tr>
<td>learning rate</td>
<td>5e^{-4}</td>
<td>5e^{-5}</td>
</tr>
<tr>
<td>batch size</td>
<td>32</td>
<td>16</td>
</tr>
</tbody>
</table>

C. Experimental Verification

We verified our model with unseen data from FaceForensics++. Our model with Xception Net gave the best accuracy among all three CNN modules. The accuracy for compression level c=23 is better than c=40. Fig.8(a) shows the accuracy vs model with different CNN networks.

D. Analysis of Results

The results for the accuracy are shown in Table V.

All three CNN modules have been paired with our classifier network. Xception net combined with our classifier gave the best accuracy of 96.00% for compression level c = 23 and 93% for c = 40. Test accuracy for each model for different compression level is given in Table VI.

Test accuracy details for the proposed model are shown in Fig. 8(b). When we compared our result with the result published in the FaceForensics++ paper [6], we see accuracy
increase at both compression levels. For our experiment we did training with only $c = 23$ videos. Fig. 9 shows a comparative picture between our work and the FaceForensics++ paper.

**IX. CONCLUSION AND FUTURE WORK**

In this paper, we present a deep learning based approach to detect deepfake videos in social media with a high accuracy. We use a neural network based method to classify pristine and manipulated video. We compared three existing CNN modules and finally chose Xception net as the feature extractor paired with the proposed classifier for the most accurate model. We trained the network with intermediate compression and achieve high accuracy even at high loss scenario. Our proposed algorithm is the key factor in getting high accuracy even without training with highly compressed videos. The complexity of the algorithm is proportional to the number of frames extracted from the video. There is no restriction on video length. In future work we plan to deploy our model at edge devices with appropriate modification.

**REFERENCES**


