A Novel Method of Deepfake Detection

Rutuja Hande¹,*, Sneha Goon¹,**, Aaditi Gondhali¹,***, and Navin Singhaniya****

¹Department of Electronic Engineering, Ramrao Adik Institute of Technology, Nerul, Navi Mumbai

Abstract. Deep-Fake is a novel artificial media technology that uses the likeness of someone else to replace people in existing photographs and films. Deep Learning, as the name implies, is a type of Artificial Intelligence that is used to create it. It is critical to develop counter attacking approaches for detecting fraudulent data. This research examines the Deep-Fake technology in depth. The Deep-Fake Detection discussed here is based on current datasets, such as the Deep-Fake Detection Challenge (DFDC) and Google’s Deep-Fake Detection dataset (DFD). The creation of a bespoke dataset from high-quality Deep-Fakes was utilised to test models. The results of both with and without Transfer Learning were analysed. Finally, the trained models were used to spot well-known deep-fakes of former US President Barack Obama and well-known actor Tom Cruise. A comparison study was performed on all three models. The findings show that the detection are generally domain-specific tasks, however that using Transfer Learning considerably improves the model performance parameters, whereas convolutional RNN gives sequence detection advantage.

1 Introduction

The internet and social media platforms have brought people from all corners of the globe closer together. In a world where recording and sharing are an inevitable part of everyday life, it’s critical to be aware of the far-reaching consequences that Deep-Fakes can have [1]. However, This is no longer the case, thanks to the appearance of Deep Fake Video. Technology has the ability to convince others that something is real when it isn’t." Deep-Fakes is a combination of the words "deep" and "fake." "Deep learning" and "fake" are two words that come to mind.

In this technology, autoencoders and GANs are utilised to create visual and auditory data. The resulting information is rife with deception [2]. Autoencoders are neural networks that retrieve and change the original image dimension by extracting crucial facial traits. Traditionally in the latent space The latent space is used to depict data analysis in a more realistic way. In a transparent way The de-noising process will overlay the original image’s features [3]. This image was coded by a coder who had been specifically trained for it. Incorporation of a Generative The addition of an adversarial network to the decoder side helped to increase the technique’s generation.

If unmanaged, this cutting-edge technology can be disastrous, as the era of believing what one sees is over. However, malevolent Deep-Fakes dominate the productive applications of Deep-Fakes, such as film dubbing, special effects, and instructional objectives [5]. Information

Figure 1. Frames from Deep-Fake video of former US President Donald Trump

© The Authors, published by EDP Sciences. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (http://creativecommons.org/licenses/by/4.0/).
dealt with by each of them on all three model architectures as discussed, and is also represented using tabular and graphical representation for better visual understanding [13].

2 IMPLEMENTATION

APPROACHES BASED ON CNN+RNN NETWORKS

The extraction of features from video frames is done using a Convolutional Neural Network. RNN is trained on these features in order to classify videos into one of two categories: fake or authentic. Another model based on the ConvLSTM hybrid architecture [2] examines the minute visual markers on the faces using CNN and then evaluates the visual data using an RNN network based on the features. The widespread availability of resources such as GPUs has resulted in the widespread distribution of these videos. RNN is well-known for its ability to analyse sequential data. We already know that the temporal information extracted from video frames is sequential. As a result, RNN can be used to process it. The work presented by Sabir et al. [3] used the best model for recognising facial alterations, which uses the RNN network. The FaceForensics++ dataset was used for testing, and efficiency was enhanced by 4.55 percent above the prior state-of-the-art. Some approaches focus on the faces in the films to reliably detect modifications, extracting visual and temporal information from the faces. These methods employ a recurrent neural network-based architecture to investigate temporal trends. D. M. Montserrat et al. [4] explored one such strategy. They compared the results with previous methodologies using the DFDC dataset. The proposed method produces more accurate findings than existing methods. MTCNN is the CNN model employed, and the model is used for temporal feature extraction. The proposed model has a validation accuracy of 92.61 percent and a test accuracy of 91.88 percent.

APPROACHES BASED ONLY CNN NETWORKS

Face warping is used in Deep-Fake generation, as are numerous morphing techniques like as scaling, translation, and affine deformations. Deep-Fake generation’s warping method causes resolution discrepancy [7] between the warped face area and the surrounding background area. Visual artefacts [8] are created in the video frames as a result of this. This type of technique takes advantage of the errors that occur throughout the generation process. Convolutional Neural Network models are used in these methods to detect the artefacts. Convolutional Neural Networks could be quite effective for computer-assisted detection difficulties. H.-C. Shin et al. present a paper [9] that uses deep CNN factors to solve Computer-Aided Detection (CADE) problems. Another CNN-based effort on image modification detection [10] use a high pass filter in conjunction with a CNN network to detect the image’s hidden features.

2.1 ALGORITHM

For the detection of Deep-Fake films, a variety of existing models have been applied. A comparison of CNN-based video deep fake detection with CNN followed by RNN-based Video DeepFake Detection is discussed in this paper. The performance of two RNN variants, LSTM and GRU, on temporal feature exploitation is compared. The algorithm of the method presented in this paper is discussed in this part. The major goal of the system under discussion is to determine whether the video input is authentic or Deep-Fake. This goal can be accomplished in three different ways. Three different models are compared based on their performance for Video Deep-Fake Detection. The following is the algorithm for these processes:

STEP 1: Frames from the input video are extracted. The input video is preprocessed and turned into frames in this step. The CNN architecture is fed a predetermined amount of frames as an input.

STEP 2: Each frame’s features are extracted. For an unseen test sequence input, CNN extracts a collection of facial traits for each frame. The feature set is concatenated when this operation is performed on a preset number of frames at the same time.

STEP 3: Analyze Temporal Sequences[1] This stage is different in each of the three models covered in the paper. To acquire a single probability value for the CNN model, the concatenated features are input to dense, followed by a global average pooling layer. Concatenated features are given to LSTM and GRU layers for temporal
feature analysis in CNN followed by LSTM and GRU models, and a single probability value is obtained.

STEP 4: Classification of video as real video or Deep-Fake video From the probability values obtained in step 3, the unseen test input video is classified as either manipulated i.e. DeepFake video, or non-manipulated video i.e real video on the basis of decided threshold

2.2 Mathematical Modelling

Each layer of CNN can be as follows:

- Convolutional Layer
- Pooling Layer
- Fully connected layer
- Convolutional Layer

Convolutional product is applied to this layer using many filters followed by an activation function.

- Pooling Layer The pooling layer’s goal is to down sample the input’s features without reducing the the following notations are taken into account:
- Fully Connected Layer It is a finite number of neurons. It takes input as a vector as well as returns a vector.

Maths Behind LSTM-

\[
C_t = C_{t-1} + (H(C_t) \cdot t)
\]
Calculating output
\[
H_t = \text{tanh}(C_t)
\]

Maths Behind GRU-

Consider time step \( t \) and input is a minibatch \( X_{t \in \mathbb{R}} \), where \( n \) is number of examples and \( d \) is number of inputs. Hidden state of earlier time step is \( H_t \in \mathbb{R}^n \). From this we can calculate reset gate \( R_t \in \mathbb{R}^n \) and update gate \( Z_t \in \mathbb{R}^n \) as follows:

\[
R_t = (X_t W_r + H_t W_rh + b_r)
\]
\[
Z_t = (X_t W_z + H_t W_zh + b_z)
\]

Here \( W \) are the weights and \( b \) are the biases. Candidate Hidden State \( H'_{t \in \mathbb{R}} \) state is given as:

\[
H'_{t} = \text{tanh}(X_t W_x + (R_t H_t h) + b_h) J
\]

Table 1. Quantitative comparison of fakeAVCeleb to existing publicly available deepfake dataset

<table>
<thead>
<tr>
<th>Dataset</th>
<th>realvideos</th>
<th>fakevideos</th>
<th>totalvideos</th>
<th>rightcleared</th>
<th>agreeingsubject</th>
<th>total subjects</th>
<th>synthesismethods</th>
</tr>
</thead>
<tbody>
<tr>
<td>UADFV</td>
<td>49</td>
<td>49</td>
<td>98</td>
<td>NO</td>
<td>0</td>
<td>49</td>
<td>1</td>
</tr>
<tr>
<td>DeepFakeTIMIT</td>
<td>640</td>
<td>320</td>
<td>960</td>
<td>NO</td>
<td>0</td>
<td>32</td>
<td>2</td>
</tr>
<tr>
<td>FF++</td>
<td>1000</td>
<td>5000</td>
<td>5000</td>
<td>NO</td>
<td>0</td>
<td>N/A</td>
<td>4</td>
</tr>
<tr>
<td>CelebDF</td>
<td>590</td>
<td>5639</td>
<td>6229</td>
<td>NO</td>
<td>0</td>
<td>59</td>
<td>1</td>
</tr>
<tr>
<td>GoogleDFD</td>
<td>363</td>
<td>3000</td>
<td>3363</td>
<td>YES</td>
<td>28</td>
<td>28</td>
<td>5</td>
</tr>
<tr>
<td>DeepForensics</td>
<td>10000</td>
<td>60000</td>
<td>YES</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>1</td>
</tr>
<tr>
<td>DFDC</td>
<td>23654</td>
<td>104500</td>
<td>128154</td>
<td>YES</td>
<td>960</td>
<td>960</td>
<td>8</td>
</tr>
<tr>
<td>KoDF</td>
<td>62166</td>
<td>175776</td>
<td>237942</td>
<td>YES</td>
<td>403</td>
<td>403</td>
<td>6</td>
</tr>
<tr>
<td>FakeAVCeleb</td>
<td>500</td>
<td>19500</td>
<td>20000</td>
<td>NO</td>
<td>0</td>
<td>500</td>
<td>4</td>
</tr>
</tbody>
</table>

Hadamard Product Operator.

Proposed Model Architecture

More precisely, at the 1st layer, we denote:

- INPUT: \( a^{l-1} \), \( d^{o} \) being the image in the input
- PADDING: \( p[l], S, t \)
- NUMBER OF FILTER: \( n_l^{i}[l] \) where each \( K^{(o)} \) has the dimension \( \left( f[l], f[l], n_l^{i}[l] \right) \)
- BIAS OF nth CONVOLUTION: \( b[l] \)
- ACTIVATION FUNCTION: \( \psi[l] \)
- OUTPUT: \( a[l] \) with size \( \left( n_l^{i}[l], n_l^{o}[l] \right) \)

And we have

\[
\forall n \in [1, 2, ..., n_{l}^{o}]
\]
\[
\text{conv}(a^{l-1}, K^{(o)}), x, y = \psi^{l}\left(\sum_{i=1}^{d^{i-1}} \sum_{j=1}^{d^{j-1}} k_{i,j}^{n_{l}^{i}[l]} a^{l-1}_{i,j} + b_{l}^{n_{l}^{o}[l]} \right)
\]
\[
\text{dim}(\text{conv}(a^{l-1}, K^{(o)})) = \left( n_{l}^{i}[l], n_{l}^{o}[l] \right)
\]

Thus,

\[
a[l] = \left[ \psi[l](\text{conv}(a^{l-1}, K^{[l]})), \text{psi}[l](\text{conv}(a^{l-1}, K^{[l]})), ..., \text{psi}[l](\text{conv}(a^{l-1}, K^{[l]})) \right]
\]

with,

\[
n_{l}^{i}[l] = \left\lfloor \frac{n_{l}^{i}[l-1]}{s[l]} + 1 \right\rfloor ; s > 0
\]
\[
= n_{l}^{i}[l-1] + 2p^{l} - j^{l} ; s = 0
\]

\( n_{l}^{i}[l] \) = number of filters

The learned Parameters at the \( l^{th} \) layer are:
• FILTERS with \( \{f^{(l)} \times f^{(l)} \times n_c^{(l-1)} \times n_c^{(l)} \} \) Parameters

• BIAS with \((1 \times 1 \times 1) \times n_i^{(l)} \) Parameters

Pooling Layer Parameters

\[
d_{x,y,z}^{(l)} = \text{pool}(d^{(l-1)})x, y, z = \phi(d_{x+j-1, y+1}^{(l)}, \text{size}=[1, 2, 3...f^l])
dim(d') = n'H, n'W, n'_C
\]

\[
n_H^{(l)} = \left[ \frac{n_H^{(l-1)}}{p^l} + 1 \right]; s > 0
\]

\[
n_W^{(l)} = n_W^{(l-1)} + 2p^l - f^l; s = 0
\]

\[
n_C^{(l)} = n_C^{(l-1)}
\]

Fully Connected Layer Parameters

• INPUT: \( a^{(l-1)} \) with size \((n_H^{(l-1)}, n_W^{(l-1)}, n_C^{(l-1)})\) \(a^{(l)}\) being the image in the input

• PADDING: \( p' \) rarely used, (stride): \( s^{(l)} \)

• SIZE OF THE POOLING FILTER: \( f^{(l)} \)

• POOLING FUNCTION: \( phi^{(l)} \)

• OUTPUT: \( a^{(l)} \) with size \((n_H^{(l-1)}, n_W^{(l-1)}, n_C^{(l-1)})\)

2.3 DATA SET

Various already existing as well as unique datasets were used to train and test the models. In this section, we’ll go through a quick overview of each dataset we’ll be using. The datasets utilised for Video Deep-Fake Detection are as follows:

1. Deep-Fake Detection Dataset (DFD): This is the Google/Jigsaw Deep-Fake Detection Dataset. It has 3068 Deep-Fake videos, which were created from 363 original videos of 28 persons of various genders, ages, and cultural backgrounds.

2. Deep-Fake Detection Challenge Dataset (DFDC): This is the Deep-Fake Detection Challenge Dataset from Facebook. There are two versions of it. 5K films made by two facial modification algorithms are included in the preview edition. There are 124k videos made by eight facial modification algorithms in the entire dataset version.

3. Custom Dataset: A custom dataset is generated by mixing 936 from existing Video Deep-Fake Detection Datasets, such as DFDC and DFD datasets, to test the generalizability of an algorithm.

3 PERFORMANCE PARAMETER

Performance Parameter To evaluate the outputs of three deep neural networks: CNN, CNN-LSTM, and CNN-GRU, the following metric parameters are used to compare their performance:

• Accuracy:- The most common comparative metric is accuracy. The ratio of the total number of correctly classified examples to the total number of examples classified is known as the correct classification rate. It is the most commonly used metric for comparing models.

• Precision:- The ratio of successfully categorised positive examples divided by the total number of anticipated positive examples is the precision value. A high precision number indicates that an example classified as positive is, in fact, positive.

• Recall:- The ratio of the total number of correctly categorised positive examples divided by the total number of positive examples is known as recall. A high recall rate shows that the class has been appropriately identified.

• F1 score:- The combined effect of precision and recall value is represented by the F1 score. It’s the sum of recall and precision in a harmonic form. summarise the results of the CNN model, CNN-LSTM model, and CNN-GRU model trained and tested on the DFD dataset, DFDC dataset, and Custom dataset, respectively. The variation of the metric parameters described above is obtained over a number of epochs. It’s also worth noting that the outcomes of the models with and without finetuning are nearly identical.

4 OUTPUTS AND RESULT ANALYSIS

In this section results of each model based on their performance on various datasets, various testing parameters, and training parameters is discussed by graphical means. This section also includes some popular video Deep-Fake detection tested using the models described in the report and
comparative analysis between three models: CNN, CNN-LSTM, CNN-GRU. The results of CNN model, CNN-LSTM model and CNN-GRU model trained and tested on DFD dataset, DFDC dataset and Custom dataset. The variation of above mentioned metric parameters are obtained over number of epochs. And it is observed that with and without finetuning the models results are similar.

\[
\begin{array}{ccc}
\text{Source} & \text{Target} & \text{Output} \\
\end{array}
\]

Figure 4. Output

![Output](https://example.com/output.png)

Figure 5. Output

5 CONCLUSION

Seeing is Believing’ was formerly thought to be true. It’s no longer true, given the emergence of Deep-Fake content. Deep-Fake has begun to erode people’s faith in media information. Deep-fake films have the potential to wreak havoc on politics, slander individuals, disseminate hate speech, and the list goes on. As a result of the spread of this, detecting it effectively becomes critical.

This paper describes a method for detecting Deep-Fake videos that employs Transfer Learning. By leveraging pre-trained networks, the work provides valuable insights into the detection (with comparative aspect) of AI-generated bogus videos.

Frame Extraction and detection are used to successfully achieve Deep-Fake detection of videos in this work. Frame detection involves obtaining a sequence of frames, which are then analysed for temporal feature changes using CNN, LSTM, and GRU. Using the CNN model, we retrieved face features from numerous consecutive frames and passed them to the network. The hybrid framework of CNN layer followed by LSTM proves to be relevant for the task of Video Deep-Fake Detection, according to the findings obtained by performing tests on different datasets. In a Deep-Fake Video, the CNN-LSTM pipeline works in the domain of temporal discontinuities in successive frames. The CNN model is followed by the CNN-LSTM model, which also does well in detecting Fake frames since the detection pipeline collects frames from videos and feeds them into the model.

6 FUTURE WORK

According to the literature review, Deep-Fake detection is one of the most popular research topics in the AI sector. To detect alterations in the Deep-Fake media, a variety of methods based on neural networks and biological signals have been used.

So far, the research has concentrated on the visual and audio components separately. So, in order to do the Deep-Fake Video as well as Audio Detection at the same time on the same video, future work will entail implementing Deep Temporal Convolutional Model (TCN) for deeper analysis and testing. Also, rather than using interpretability methods to those models as the methodology utilised, work on developing models that operate as a generalising architectural foundation for the majority of sorts of Deep-Fake techniques.
Although much research has been done on the development and detection of Deep-Fake, not enough has been done on the reversal of those manipulations. As a result, a network may be created that can undo the Deep-Fake video or image, allowing us to reconstruct the original video or image from the Deep-Fake one. Seeing the consequences that Deep-Fakes have on human life it is a good lesson to double check the content before sharing it on any internet platforms.

References


