2021 IEEE International Conference on Computer Science, Artificial Intelligence and Electronic Engineering (CSAIEE)

2021 IEEE International Conference on Computer Science, Artificial Intelligence and Electronic Engineering (CSAIEE) | 978-1-6654-2204-8/21/$31.00 ©2021 IEEE | DOI: 10.1109/CSAIEE54046.2021.9543264

DeepfakeVideo Detection by Combining Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN)

Yunes Al-Dhabi
College of Software, Northeastern University
Shenyang, Liaoning, 110819, China
yusoft@stumail.neu.edu.cn

Shuang Zhang*
College of Software, Northeastern University
Shenyang, Liaoning, 110819, China
*Corresponding author: zhangs@swc.neu.edu.cn

***Abstract*— Nowadays, people are facing an emerging problem called deepfake videos. These videos were created using deep learning technology. Some are created just for fun, while others are trying to manipulate your opinions, cause threats to your privacy, reputation, and so on. Sometimes, deepfake videos created using the latest algorithms can be hard to distinguish with the naked eye. That's why we need better algorithms to detect deepfake. The system we are going to present is based on a combination of CNN and RNN, as research shows that using CNN and RNN combined achieve better results. We are going to use a pre-trained CNN model called Resnext50. Using this, we save the time of training the model from scratch. The proposed system uses Resnext pretrained model for Feature Extraction and these extracted features are used to train the Long short-term memory (LSTM). Using CNN and RNN combined, we capture the inter frames as well as intra frames features which will be used to detect if the video is real or fake. We evaluated our method using a large collection of deepfake videos gathered from a variety of distribution sources. We demonstrate how our system may obtain competitive results while utilizing a simplistic architecture.**

Keywords*— *Deep learning, Deepfake Detection, convolutional Neural Network (CNN), Recurrent Neural Network (RNN)

I. INTRODUCTION

The "Deepfake" word is a combination of "Deep learning" and "fake". These are AI-generated videos in which a person in a video or image is replaced by someone else. The idea of swapping faces in photos is not new. We can find examples of such photos made in the 19th century. Back then, these photos were made by hand. However, when the idea of neural networks became popular and with advancements in the computational field, people began to use this technology to create such deepfake videos and images. Nowadays, we can download and run such programs which can swap our faces with others. Today, none of us will be surprised by apps like Face App [1], Snapchat that have the ability to swap faces in good quality and make us funny. In January 2018, a desktop application called Fake App was launched [2]. This app allows users to easily create and share videos with their faces swapped with each other. After this app, many such apps were launched, like Face Swap and DeepFaceLab [3]. Larger companies have also started to use deepfakes. The Japanese AI company DataGrid made a full-body deepfake that can create a person from scratch. They intend to use them for fashion and apparel. A mobile deepfake app, Impressions, was launched in March 2020 [4]. It was the first app for the creation of celebrity deepfake videos from mobile phones. Today, we can find various desktop as well as mobile apps which we can use to create a good quality deepfake. One of the more infamous examples is from President Obama and his video posted online [5] where Jordan Peele shares the grim reality and the ease of creating a deepfake video that can make politicians such as the former president of The United States say offensive and inciting statements, as seen in Figure 1. Furthermore, deepfakes can get more detailed as more data is available to train the deep learning algorithms. Deepfakes will not disappear any time soon, and their impact is already being felt on a global scale. It is essential to develop countermeasures to protect individuals and organizations from the harmful applications of deepfakes.

Figure 1. Screenshot taken from the controversial Obama deepfake video in which the former president makes disparaging remarks about President Trump, leaving the audience with the unsettling truth that not everything they see and hear on the internet can be trusted. The screenshot is captured by Fagan, with Business Insider (Fagan 2018).

II. RELATED WORKS

A. *Use of CNN and RNN Based Hybrid Model for Deepfake Detection.)*

Recently, a proposed method of CNN and LSTM based deepfake detection by (M. F. Hashmi, B. K. K. Ashish et al., 2020) this method uses convolutional Neural Network (CNN) and stores the abnormal features for training [6]. 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. Instead of storing frames, main features extracted from CNN are stored in LSTM using mean pooling. The Recurrent Neural Network (RNN) pipeline learns based on these features-fed inputs from CNN and then evaluates results. The model was trained with the network of videos consisting of their real and fake, collected from multiple websites. This training compares real video with its deepfake video and CNN extract the abnormal features only. 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. Problem with this method is that without real video first it is challenging to extract abnormal features. The Methods which use CNN and RNN combined together will use CNN for feature extraction and RNN is used to produce final output. One of such method is presented by (D. Güera, E. J. Delp et al., 2018) the CNN feature extractor used in this method is InceptionV3 and RNN used is LSTM with 0.5 chance of dropout [7], the accuracy achieved is only on 600 video data-set that they have created also it fail to achieve same accuracy when such techniques will be applied on different distribution. Another method which uses CNN and RNN combined is proposed Another method which uses CNN and RNN combined is proposed by (Fei, J., Xia, Z., Yu, P. et al., 2020) this method uses Eulerian Motion Magnification along with InceptionV3 and LSTM. Eu-lerian Motion magnification is used to magnify the facial region of video and then CNN is used to extract inter frame features and LSTM used to extract intra frame features [8]. The accuracy achieved on FaceForensic++ data set fails to achieve the same accuracy when such techniques will be applied on different distribution.

B. *Deepfake Detection Using CNN Based Classifier*

Some proposed method uses CNN classifier with other algorithms to produce the final output, (N. S. Ivanov, A. V. Arzhskov et al., 2020) proposed architecture which uses CNN classifier along with inconsistent head pose estimator and fast super resolution CNN model [9]. This method achieves accuracy of 95.5% but the classifier alone can achieve accuracy of 94.9% on UADFV dataset. Method proposed by (I. Amerini, A. Del Bimbo et al., 2019) the optical flow fields are used with CNN. Optical flow is a vector field which is computed on two consecutive frames to extract apparent motion between the observer and the scene itself [10]. Using this optical flow vector and VGG16 accuracy achieved is 81% when used with ResNet accuracy achieved is 75%. This accuracy achieved is achieved on only FaceForensic++ dataset. (J. Baek, S. Bae et al., 2020) proposed method to detect the deepfake using discriminator used in GAN network [11]. The proposed method uses two discriminators to increase the capability of deepfake detection. The accuracy achieved with this method is less but the results were consistent. Another proposed method [12], which uses CNN to capture distinctive artifacts in deepfake video by (Yuezun Li & Siwei Lyu, 2018) the authors of this paper used image processing for creation of deepfake videos. This method is tested on VGG16, among these ResNet outperforms others. (D. M. Montserrat et al., 2020) proposed a method which uses EfficientNetB5 with MTCNN the accuracy achieved is 92.61% on DFDC dataset [13]. (D. Afchar, V. Nozick, et al., 2019) proposed two deep neural network Meso4 and MesoInception4 both have very number of layers still manages to achieve average classification score of 0.89 for Meso4 [14]. This method mainly focused on two Deepfake video generation techniques Deepfake and Face2Face, so while testing the dataset used was generated using these two methods only.

C. *Other Methods of Deepfake Detection*

There are also methods in deepfake detection which does not include any type of deep neural network, (M. A. Younus & T. M. Hasan, 2020) proposed a method which compares the blur and sharpness of facial region with the background, and based on that it detects the deepfake video [15]. The accuracy achieved with this method on UADFV dataset is 90%. (M. F. Hashmi, B. K. K. Ashish, et al., 2020) provides architecture which uses a smaller number of frames per video to assess its realism [16]. this method uses facenet with the metric learning approach using a triplet network architecture. (T. Jung, S. Kim, et al., 2020) proposed a Deep Vision algorithm which uses frequency of eye blinking in video for detection of deepfake [17]. Various factors which effect the blinking pattern (age, time of day, gender, emotional state etc.) using these factors as an input the proposed algorithm can predict if the video is fake or not. For this to work efficiently the input should be precise. (F. F. Kharbat, T. Elamsy et al., 2020) proposed a method which uses SVM classifier with edge features detection algorithms such as HOG, SURF, KAZE, etc. Using these SVM is trained for classification [18]. Using HOG, the accuracy achieved is maximum i.e. 94%. (J. Baek, Y. Yoo, et al., 2020) proposed a method which uses head position for detection of deepfake [19]. The proposed method first extract 2D facial landmark to create 3D model of head and deepfake videos show inconsistency in 3D head poses, that inconsistency is captured using SVM classifier. These methods which does not use any deep neural network does not achieve the accuracy up to the mark.

III. DEEP FAKE DETECTION METHODOLOGY

This research proposes a method for detecting deepfake videos that is based on the same method that was used to generate the Deepfake by GAN "Generative adversarial network". The system we are going to present is based on a combination of CNN and RNN, as research shows that using CNN and RNN combined achieves better results. We are going to build our model based on the resnext50_32x4d. By using this, we save the time of training the model from scratch. Specifically, our method detects face warping artifacts by comparing the generated face areas to their surrounding regions, which is performed by splitting the video into frames. Then, using a resnext50 pretrained model to extract the features, the LSTM is used to capture the temporal discrepancies between frames introduced by GAN during the Deepfake reconstruction. In order to train our model, we streamlined the process by simulating the resolution inconsistencies in affine face wrappings directly in the simulation environment. The mechanism for deepfake creation is deep learning models such as autoencoders and generative adversarial networks (GAN), which have been applied widely in the computer vision domain.

A. *Autoencoders*

It is well known neural network to create deepfake videos. Autoencoders contains two parts encoder and decoder.

Function of encoder is the compression of input. First the input is represented in smaller latent space using encoder. The function of decoder is opposite of encoder to get original data from compressed data. By using encoder of one image with the decoder of other we can create a deepfake. For deepfake creation this encoder and decoder only applied on facial region. Figure 2 shows the architecture of autoencoder which is used to create deepfake image. As shown in Figure 2 encoder of face A is used with the decoder of face B to get reconstructed face B from A. Some examples which use these techniques to create deepfake are Faceswap, DFaker, DeepFakeLab and etc.

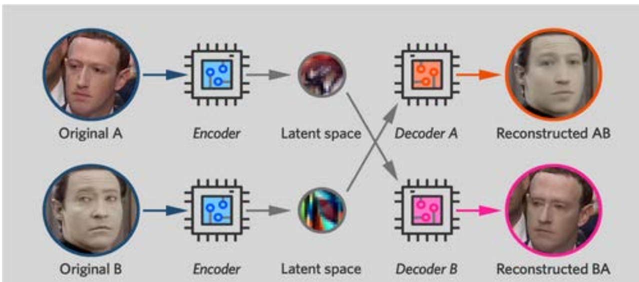

Figure 2. Overview of the Process of Deepfake Creation. The original face is extracted and processed by the Encoder. The final reconstructed face image is produced from both images (original face and latent face) (J. Baek 2020).

B. A generative adversarial network (GAN)

GAN is the advancement in the autoencoder technique to generate deepfake. In GAN there are two different deep neural networks, generator and discriminator. Generator in GAN works similar to autoencoder, but we can achieve better results because discriminator net is rejecting some bad examples. Job of discriminator is to reject bad deepfake example and generator keeps producing the deepfake until it successfully fools the discriminator, that makes such fakes more similar to real videos and also makes them harder to recognize by naked eyes. Some of the open-source projects are using this technique for example, Faceswap-GAN. Figure 3 shows the architecture of GAN network [20].

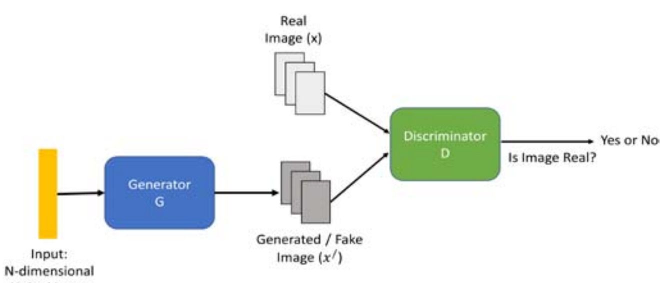

Figure 3. Simplified description of a GAN learning framework (Salvaris 2018).

IV. PROPOSED SYSTEM

There are many tools available for generating DeepFake, but there are few tools available for detecting DeepFake. Our approach to detecting DeepFake would make a contribution to preventing DeepFake from spreading over the internet. Our approach provides a web-based platform that allows users to upload videos and classify them as fake or real. Previously, proposed deepfake detection systems which use CNN and RNN combined managed to achieve an average accuracy above 90%, but the test accuracy was conducted on the same distribution where CNN and RNN are trained. When such techniques are applied to different distributions, they will fail to achieve the same accuracy. The overarching purpose of this research is to build an optimal model to detect deepfake videos and achieve the highest possible accuracy. The above overarching aim can be split into three sub-objectives. Firstly, prepare the dataset, which is a combination of different distributions, because by using videos from different distributions, the results will be consistent. Our second sub-objective is to reduce the gap between the achieved accuracy and the validation accuracy. Previously, proposed deepfake detection systems which use CNN and RNN combined managed to achieve an average accuracy above 90%, but the test accuracy was conducted on the same distribution where the CNN and RNN were trained. When such techniques are applied to a different distribution, they will fail to achieve the same accuracy. Our third sub-objective is to make a real-time web-based python platform named Anti-Deepfake, where the user can upload or select a video containing people on which our model will be applied. It will say if the video is a deepfake or not. Our goal is to make the model return binary values, either deepfake or not, but ideally, we want it to return the probability of the video being a deepfake, for more in-depth results.

A. Dataset

We will perform research on deepfake detection using a mixture dataset that contains an equal number of videos which is a combination of different distributions. Because by using videos from different distributions, the results will be consistent. The dataset includes the DFDC dataset [21], which contains 3000 videos, the FaceForensics++ dataset [22], which contains 2000 videos, and the Celeb-DF dataset [23], which contains 1000 videos. Our newly prepared dataset contains 6000 videos, which are divided into 50% of original videos and 50% of fake videos. Figure 4 represents the datasets of the proposed system: -

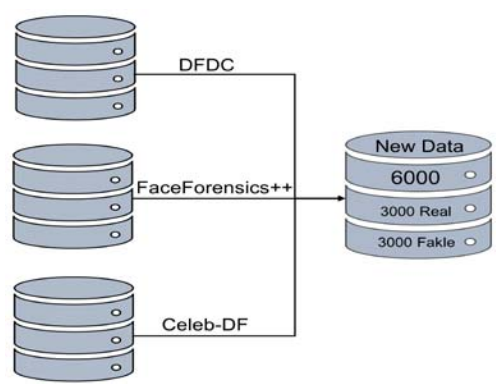

Figure 4. Dataset and Analysis Flowchart which combined of three kinds of videos from different resources [21,22,23].

B. Preprocessing

The data pre-processing begins with the video framing that splits the video into frames. After that, the face detector will be applied to detect the faces and crop the frame with the detected faces only. The mean of the dataset is determined to ensure that the number of frames remains consistent. The new dataset is constructed using frames that are equal to the mean of the previously processed facial frames. During preprocessing, frames that have no faces are disregarded. When analyzing a 10-second video at a frame rate of 30 frames per second, we propose training the model using only the first 100 frames. In terms of video sequence data preprocessing, each frame is resized to 224 x 224 pixels, and a sub-sequence sampling of length N is used to determine the length of the input sequence – N = 10, 20, 40, 60, 80, 100 frames. This helps us to determine the number of frames required per video for accurate detection. The ReLU activation function is used to train the entire model from end-to-end with a learning rate of 1e-5 and decay of 1e-5.

C. *Model*

The proposed system is composed of combining Convolutional Neural Networks (CNN) and Recurrent Neural Network (RNN). The model is composed of resnext50 32x4d and a single LSTM layer with a 2048 shape input vector and 2048 latent features along with 0.4 chance of dropout and ReLU Activation function (Rectified Linear Unit). By using this, we save the time of training the model from scratch. CNN will extract the frames from the videos and then extract the face region from the frames. After extracting faces from the frames, we will fine-tune the network by adding additional required layers and setting an appropriate learning rate to ensure that the gradient descent model converges effectively. The output feature vector of CNN is passed to LSTM for training purposes. After training, LSTM will produce the final output to check the accuracy. Figure 5 represents the model of the proposed system: -

```
==========================================================================================
Layer (type:depth-idx)                        Output Shape              Param #
==========================================================================================
Model                                         --                        --
├─Sequential: 1-1                             [20, 2048, 7, 7]          --
│    └─Conv2d: 2-1                            [20, 64, 112, 112]        9,408
│    └─BatchNorm2d: 2-2                       [20, 64, 112, 112]        128
│    └─ReLU: 2-3                              [20, 64, 112, 112]        --
│    └─MaxPool2d: 2-4                         [20, 64, 56, 56]          --
│    └─Sequential: 2-5                        [20, 256, 56, 56]         --
│    │    └─Bottleneck: 3-1                   [20, 256, 56, 56]         63,488
│    │    └─Bottleneck: 3-2                   [20, 256, 56, 56]         71,168
│    │    └─Bottleneck: 3-3                   [20, 256, 56, 56]         71,168
│    └─Sequential: 2-6                        [20, 512, 28, 28]         --
│    │    └─Bottleneck: 3-4                   [20, 512, 28, 28]         349,184
│    │    └─Bottleneck: 3-5                   [20, 512, 28, 28]         282,624
│    │    └─Bottleneck: 3-6                   [20, 512, 28, 28]         282,624
│    │    └─Bottleneck: 3-7                   [20, 512, 28, 28]         282,624
│    └─Sequential: 2-7                        [20, 1024, 14, 14]        --
│    │    └─Bottleneck: 3-8                   [20, 1024, 14, 14]        1,390,592
│    │    └─Bottleneck: 3-9                   [20, 1024, 14, 14]        1,126,400
│    │    └─Bottleneck: 3-10                  [20, 1024, 14, 14]        1,126,400
│    │    └─Bottleneck: 3-11                  [20, 1024, 14, 14]        1,126,400
│    │    └─Bottleneck: 3-12                  [20, 1024, 14, 14]        1,126,400
│    │    └─Bottleneck: 3-13                  [20, 1024, 14, 14]        1,126,400
│    └─Sequential: 2-8                        [20, 2048, 7, 7]          --
│    │    └─Bottleneck: 3-14                  [20, 2048, 7, 7]          5,550,080
│    │    └─Bottleneck: 3-15                  [20, 2048, 7, 7]          4,497,408
│    │    └─Bottleneck: 3-16                  [20, 2048, 7, 7]          4,497,408
├─AdaptiveAvgPool2d: 1-2                      [20, 2048, 1, 1]          --
├─LSTM: 1-3                                   [1, 20, 2048]             33,554,432
├─Linear: 1-4                                 [1, 2]                    4,098
├─Dropout: 1-5                                [1, 2]                    --
==========================================================================================
Total params: 56,538,434
Trainable params: 56,538,434
```

Figure 5. Screenshot of the proposed Model. ResNext is used for extracting features and the LSTM for sequence processing.

V. EXPERIMENTAL SETUP

All preliminary analyses and sample testing are run using the Graphics Processing Unit (GPU) provided by Google Colab. We used the Pro version of Colab which allow us to access a High-RAM runtime environment. With Colab Pro our notebooks can stay connected for up to 24 hours. We demonstrate our experimental steps as follow: -

1) First, we have to collect the data set of real and fake videos, it should be labelled data set, while collecting dataset we have to make sure that the videos come from different distribution for better results.
2) We have to extract the frames from the videos and then extract faces region from the frames.
3) After extracting faces from the frames, we have to pass that data to The CNN for feature extraction.
4) The output feature vector of CNN is passed to LSTM for training purpose, then the LSTM will produce the final output to check the accuracy. Figure 6 represents the system architecture of the proposed system: -

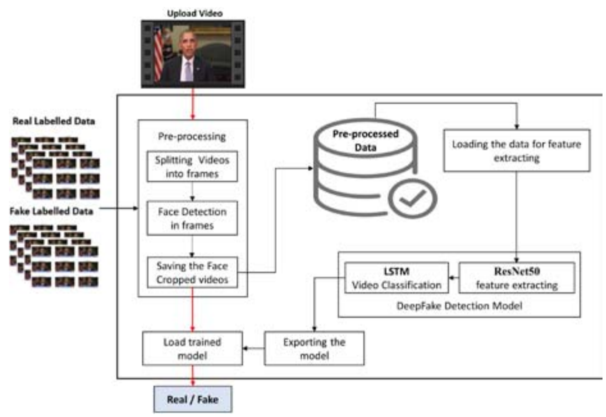

Figure 6. The flowchart of the proposed Deepfake Video detector. The red line represents the Prediction flow and the black line represents the Training Flow.

VI. EXPERIMENTAL RESULS AND EVALUATION

We demonstrate our system's performance in terms of detection accuracy using sub sequences of length N = 10, 20, 40, 60, 80, 100 frames in Table 1. These frame sequences are extracted sequentially from each video (without frame skips). The entire pipeline is trained end-to-end till the validation set reaches a 20-epoch loss plateau.

TABLE 1. CLASSIFICATION RESULTS OF OUR DATASET SPLITS USING VIDEO SUBSEQUENCES WITH DIFFERENT LENGTHS.

Model Name	Training accuracy (%)	Validation accuracy (%)
model_10_frames	86.7515	84.6314
model_20_frames	89.4027	87.2261
model_40_frames	93.0206	89.3658
model_60_frames	96.1385	92.5027
model_80_frames	98.2506	94.2512
model_100_frames	99.8311	95.5424

A. *Accuracy*

In this research, the model with 100 frames obtained an accuracy of 99.83 after 20 epochs. This means that the algorithm correctly distinguishes between deepfake and real videos with 99.8%. Moreover, the model's validation accuracy is 95.5424, meaning 95.5 % accurate classification of deepfakes from the validation dataset. Figure 7 presents the ratios of the training accuracy and validation accuracy. The second goal of this research was to reduce the gap between the achieved accuracy and the validation accuracy and we have achieved small gap between them.

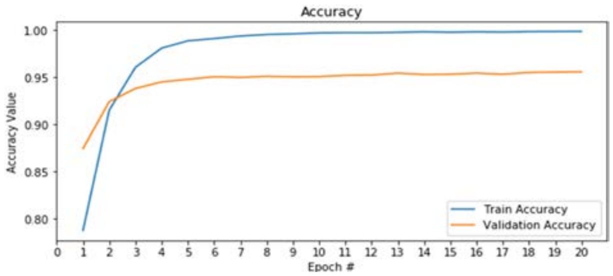

Figure 7.The ratio between training accuracy and validation accuracy

B. Loss

The model result has a loss value of 0.0053. This is an excellent value for the data training model. Inferring that the model is capable of detecting deepfakes. The proposed model produces a validation loss value of 0.2324, which indicates that the model makes some incorrect decisions with the general data in comparison to the training data, but the difference is not that huge. Figure 8 illustrates a comparison of value changes between the training loss and validation loss.

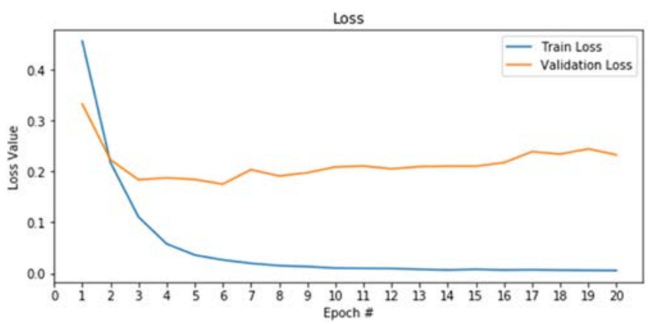

Figure 8. The ratio between training loss and validation loss

C. Confusion Matrix

Figure 9 illustrates the research's confusion matrix. To keep things simple, this confusion matrix is shown with a darker shade of color for each class. More bright values correspond to larger numbers, whereas darker values correspond to lesser numbers. The confusion matrix is quite dark in the False-positive (58) and False-negative (39) sections, indicating that the model is making less errors and predicting more precisely.

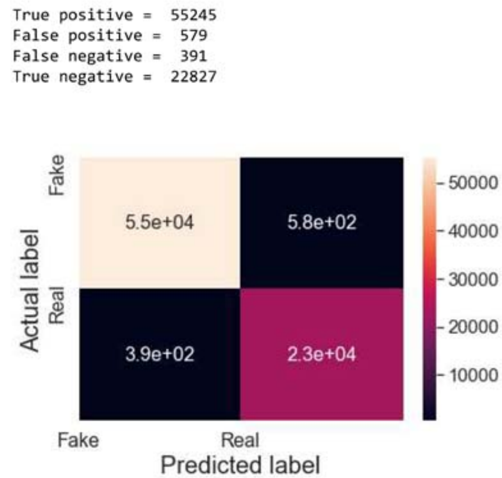

Figure 9. Confusion matrix with the result of model

D. Prediction result

The proposed method is capable to distinguish between real and deepfake videos. The final conceptualization of this research provides users with a web-based python platform, where the user can upload or select a video containing people to which our model will be applied. It will say if the video is a deepfake or real. Our goal is to make the model return binary values, either deepfake or not, but ideally, we want it to return the probability of the video being a deepfake, for more in-depth results. Figure 10 illustrates the final result, on which our model with 100-frames could be applied to obtain extremely high accuracy.

Figure 10. Expected Results

VII. Conclusion

In this paper, we presented a deepfake detection system that can identify deepfake videos automatically. Our experimental results using a large collection of manipulated videos have obtained competitive results while utilizing a simplistic architecture. We believe that our work can be solid enough to be considered significant enough for further research and development. The performance rates will be good enough if someone considering the data provided. We strongly believe that our work can be the basis in case for further improvement in case more data are available.

REFERENCES

[1] Faceapp. https://www.faceapp.com/. (Accessed on 05/29/2018).

[2] Fakeapp. https://www.fakeapp.org/. (Accessed on 05/29/2018).

[3] Ivan Perov, Daiheng Gao, Nikolay Chervoniy, Kunlin Liu, Sugasa Marangonda, Chris Umé, Mr Dpfks, Carl Shift Facenheim, Luis RP, Jian Jiang, et al. Deepfacelab: A simple, flexible and extensible face swapping framework. *arXiv preprint arXiv:2005.05535*, 2020.

[4] Porter, Jon. "Another convincing deepfake app goes viral prompting immediate privacy backlash". *The Verge*. Archived from the original on 3 September 2019. Retrieved 8 November 2019.

[5] Fagan, K. 2018. A viral video that appeared to show obama calling trump a 'dips' shows a disturbing new trend called 'deepfakes'.

[6] M. F. Hashmi, B. K. K. Ashish, A. G. Keskar, N. D. Bokde, J. H. Yoon and Z. W. Geem," An Exploratory Analysis on Visual Counterfeits Using Conv-LSTM Hybrid Architecture," in IEEE Access, vol. 8, pp. 101293-101308, 2020, doi: 10.1109/ACCESS.2020.2998330.

[7] D. Güera and E. J. Delp, "Deepfake Video Detection Using Recurrent Neural Networks," *2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, Auckland, New Zealand, 2018, pp. 1-6, doi: 10.1109/AVSS.2018.8639163.

[8] Fei, J., Xia, Z., Yu, P. *et al.*" Exposing AI-generated videos with motion magnification." *Multimed ToolsAppl* (2020). https://doi.org/10.1007/s11042-020-09147-3.

[9] N. S. Ivanov, A. V. Arzhskov and V. G. Ivanenko, "Combining Deep Learning and Super-Resolution Algorithms for Deep Fake Detection," *2020 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus)*, St. Petersburg and Moscow, Russia, 2020, pp. 326-328, doi:10.1109/EIConRus49466.2020.9039498.

[10] I. Amerini, L. Galteri, R. Caldelli and A. Del Bimbo, "Deepfake Video Detection through Optical Flow Based CNN," *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, Seoul, Korea (South), 2019, pp. 1205-1207, doi: 10.1109/ICCVW.2019.00152

[11] J. Baek, Y. Yoo and S. Bae, "Generative Adversarial Ensemble Learning for Face Forensics," in *IEEE Access*, vol. 8, pp. 45421-45431, 2020, doi: 10.1109/ACCESS.2020.2968612.

[12] Yuezun Li and Siwei Lyu. Exposing deepfake videos by detecting face warping artifacts. arXiv preprint arXiv:1811.00656, 2018

[13] AD. M. Montserrat *et al*., "Deepfakes Detection with Automatic Face Weighting," *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Seattle, WA, USA, 2020, pp. 2851-2859, doi: 10.1109/CVPRW50498.2020.00342.

[14] D. Afchar, V. Nozick, J. Yamagishi and I. Echizen, "MesoNet: a Compact Facial Video Forgery Detection Network," 2018 IEEE International Workshop on Information Forensics and Security (WIFS), Hong Kong, Hong Kong, 2018, pp. 1-7, doi: 10.1109/WIFS.2018.8630761.

[15] M. A. Younus and T. M. Hasan, "Effective and Fast DeepFake Detection Method Based on Haar Wavelet Transform," *2020 International Conference on Computer Science and Software Engineering (CSASE)*, Duhok, Iraq, 2020, pp. 186-190, doi: 10.1109/CSASE48920.2020.9142077

[16] M. F. Hashmi, B. K. K. Ashish, A. G. Keskar, NA. Kumar, A. Bhavsar and R. Verma, "Detecting Deepfakes with Metric Learning," *2020 8th International Workshop on Biometrics and Forensics (IWBF)*, Porto, Portugal, 2020, pp. 1-6, doi: 10.1109/IWBF49977.2020.9107962.

[17] T. Jung, S. Kim and K. Kim, "DeepVision: Deepfakes Detection Using Human Eye Blinking Pattern," in *IEEE Access*, vol. 8, pp. 83144-83154, 2020, doi: 10.1109/ACCESS.2020.2988660.

[18] F. F. Kharbat, T. Elamsy, A. Mahmoud and R. Abdullah, "Image Feature Detectors for Deepfake Video Detection," *2019 IEEE/ACS 16th International Conference on Computer Systems and Applications (AICCSA)*, Abu Dhabi, United Arab Emirates, 2019, pp. 1-4, doi: 10.1109/AICCSA47632.2019.9035360.

[19] J. Baek, Y. Yoo and S. Bae, "Generative Adversarial Ensemble Learning for Face Forensics," in *IEEE Access*, vol. 8, pp. 45421-45431, 2020, doi: 10.1109/ACCESS.2020.2968612.

[20] Salvaris M., Dean D., Tok W.H. (2018) Generative Adversarial Networks. In: Deep Learning with Azure. Apress, Berkeley, CA. https://doi.org/10.1007/978-1-4842-3679-6_8.

[21] https://www.kaggle.com/c/deepfake-detectionchallenge/data

[22] https://github.com/ondyari/FaceForensics.

[23] http://www.cs.albany.edu/˜lsw/celeb-deepfakeforensics.html