IsSwap?

Deep Fake Detection using Deep Learning

TABLE OF CONTENTS

ABSTRACT

In this era of technology the intake of information through digital media has grown exponentially and it has provided people with personal motives to spread falsified information among the masses to create biassed opinions and a sense of unrest.

Given that a large part of information consumed by people is in the form of videos, it has become a great target spot for people with malicious intent.

Our project is aimed at overcoming the given challenge by providing a fast and reliable method to determine the authenticity of a given video.

PROJECT OBJECTIVE

isSwap? attempts to protect people from believing in false information which is being spread through these deep-fakes by identifying such videos using different tools and technology. It is a web application which uses deep learning techniques to achieve its goal which is to help the user determine the authenticity of a given video.

INTRODUCTION

In the current world scenario technology has taken over every aspect of human life and has become an essential part of the same. We use technology to complete routine tasks, this has made life simpler in more ways than we can imagine. The sheer amount of information we consume from digital media is enormous.

As our intake of information through digital media has increased, new problems have arisen. Over time with technological advancements people have found ways to use technology to falsify information and spread it to achieve personal vendetta. One of such practices is called ‘Deep-Fake’[1]. Deep Fake is a technique in which a video of a person’s face or body has been digitally altered so that they appear to be saying or doing something which they actually never have said or done. Deep Fakes are created by combining and superimposing existing images or videos using a deep learning technique known as GANs.

MOTIVATION

Information through digital media has grown exponentially and it has provided people with personal motives to spread falsified information specially during elections to create political unrest among the masses or simply to spread a rumor.

Users upload over 500 hours[2] of fresh video content per minute which roughly translate to 7.2 lakhs of new content uploaded everyday and this is just on one platform, namely youtube.

Now the question arises, How do we know that the same technology we love and trust is not being used against us?

The explosive growth in deep fake video and its undetected use is a major threat[3] to democracy, justice, and public trust. Due to this there is an increased demand for fake video analysis, detection and intervention which is the main focus of our project along with spreading awareness about these deep-fakes and letting people know what serious implications these videos have.

BACKGROUND STUDY

1. Numpy: Array-processing package. Provides a high performance multidimensional array object, and tools for working with these arrays.

2. Pandas: Fast, powerful, flexible and easy to use open source data analysis and manipulation tool.

3. Matplotlib: Matplotlib[4] is a plotting library and an extension of numpy. Object Oriented API for embedding plots. Support for custom labels and texts. High quality output in many formats. Very customizable in general.

4. Seaborn: It provides a high level interface to visualize data and is based on matplotlib.

5. Keras: There are two ways to build Keras models: sequential and functional[5].The sequential API allows you to create models layer-by-layer for most problems. It is limited in that it does not allow you to create models that share layers or have multiple inputs or outputs.

Sequential model[6]:-It is a linear stack of layers.

A Sequential model is not appropriate when:

- Your model has multiple inputs or multiple outputs
- Any of your layers has multiple inputs or multiple outputs
- You need to do layer sharing
- You want non-linear topology (e.g. a residual connection, a multi-branch model)

6. Pytorch: It is a python library based on Torch library . it is mainly used in computer vision applications and natural language processing

7. Neural Networks: Neurons in the Neural Network are inspired from biological neurons. This Neural Network would be able to do various tasks like classifying images, prediction, and so on. A Perceptron[7] is an algorithm used for supervised learning of binary classifiers. Binary classifiers decide whether an input, usually represented by a series of vectors, belongs to a specific class. In short, a perceptron is a single-layer neural network.

8. Flask: (Flux Advanced Security Kernel) It is a web framework. It is suitable for the development of web apps. We have chosen flask over django because it is lighter and much more explicit.

9. Tensorflow[8]: It is a python library which has many uses . Its main uses include training and inference of deep neural networks

10. Haar Cascade Classifier: Haar cascade[9] is a program useful for identification of various objects or faces or hand gestures in an image or video.

The algorithm can be explained in four stages:

- Calculating Haar Features
- Creating Integral Images
- Using Adaboost
- Implementing Cascading Classifiers

Long Short Term Memory networks[10] – usually called “LSTMs” are a special kind of

Recurrent neural networks, capable of learning long-term dependencies. They were introduced by Hochreiter & Schmidhuber in 1997, and were refined and popularized by many people. LSTMs are designed to avoid the long term dependency problem. Remembering information for long periods of time is practically their default behavior, not something they struggle to learn!

ResNeXt[11] is a simple, highly modularized network architecture for image classification. It is constructed by repeating a building block that collects a set of transformations with the same topology. It is a simple design which results in a homogeneous, multi-branch architecture that has only a few hyper-parameters to set. This strategy exposes a new dimension, which we call “cardinality” , as an essential factor in addition to the dimensions of depth and width.

TOOLS AND TECHNOLOGIES USED

1. Programming Language
    - a. Python
    - b. JavaScript
    - c. HTML/CSS
2. Programming Framework
    - a. PyTorch
    - b. Flask
3. Neural Networks
    - a. Convolutional Neural Network (CNN)
    - b. Recurrent Neural Network (RNN)
    - c. Long Short Term Memory(LSTM)
4. Libraries
    - a. Numpy
    - b. Pandas
    - c. Matplotlib
    - d. Seaborn
    - e. Keras
    - f. Tensorflow
    - g. Scikit Learn
    - h. OpenCV
5. IDE
    - a. Google Colab
    - b. Jupyter Notebook
6. Version Control
    - a. Git

REQUIREMENT ANALYSIS

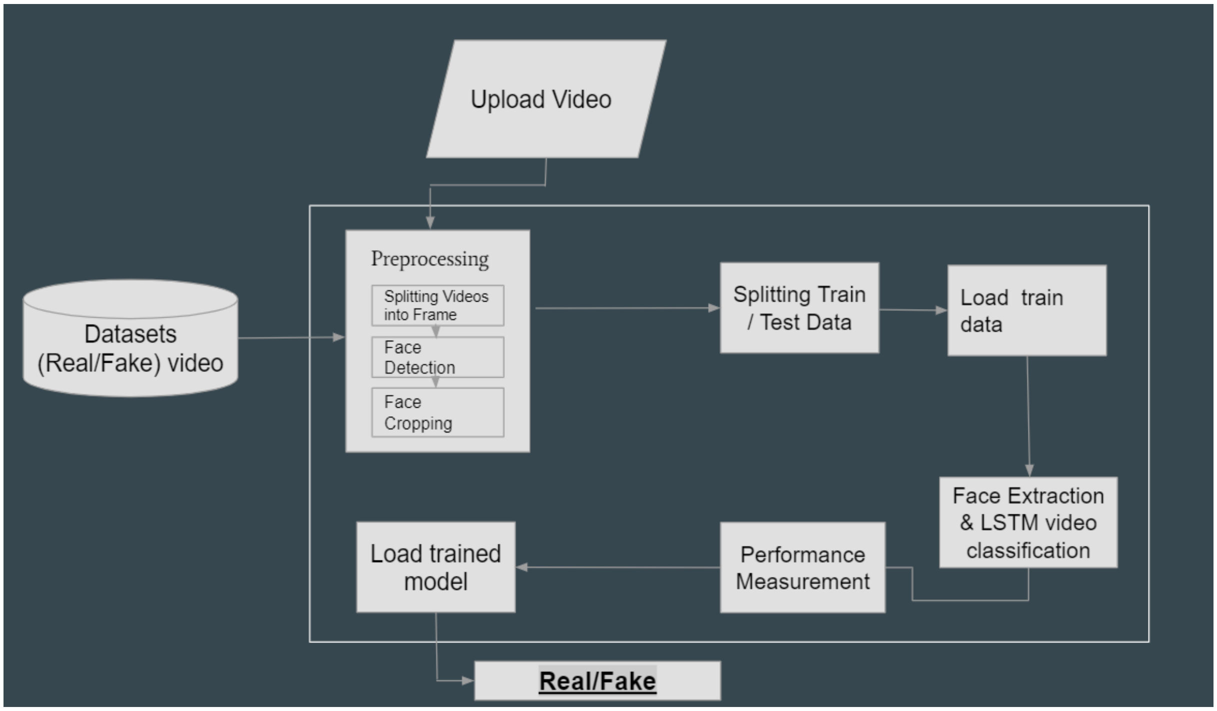
For Deepfake Detection, we will be using Deep Learning, for that Neural Networks, Convolutional Neural Networks and Pytorch will be used. This requires a good amount of computational power and GPU power to do the required ML tasks quickly and efficiently.

We also require these things:-

- For training our model we need training data. For this we need labelled data which tells which data is real and which is fake, so that the model can learn from it and then be able to identify deep fake videos on it's own.
  The data we are using[12] is composed of mp4 files, split into compressed sets of ~10GB apiece. A metadata.json accompanies each set of mp4 files, and contains filename, label (REAL/FAKE), original and split columns, listed below under Columns

- For validation purposes, we also need testing data, so that we can check the performance of our model and see how well it can identify and differentiate deep fake videos from real ones.
  test_videos.zip - a zip file containing a small set of videos to be used as a public validation set.

SYSTEM DESCRIPTION

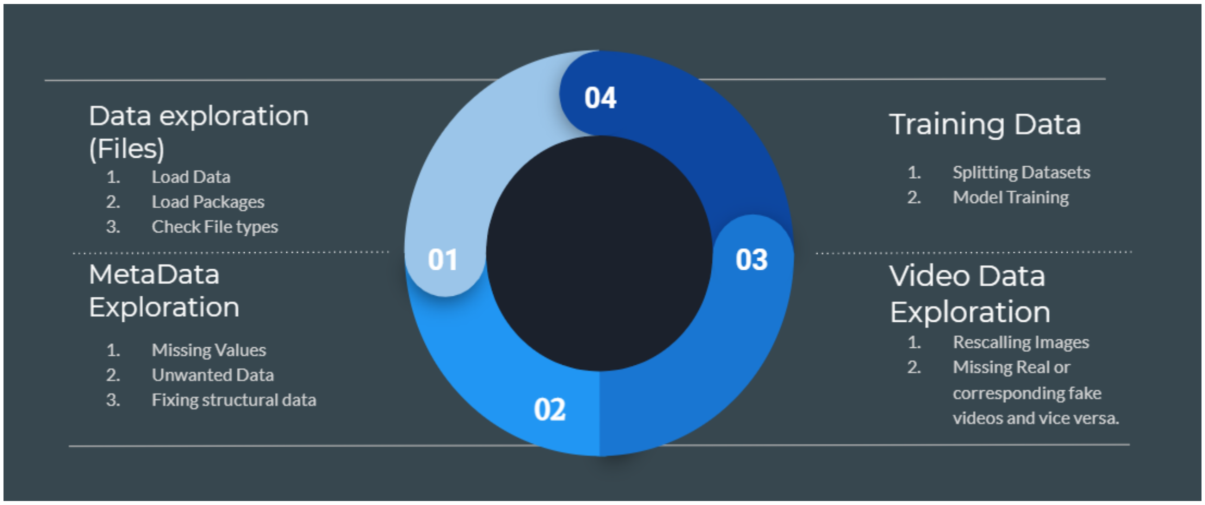

Upload Video
Datasets (Real/Fake) video
Preprocessing
Splitting Videos into Frame
Face Detection
Face Cropping
Splitting Train / Test Data
Load train data
Face Extraction & LSTM video classification
Performance Measurement
Load trained model
Real/Fake

There are two phases of this project. One from the user side i.e. user will upload a video to our model and secondly from the developer side i.e. developer will train the model based on the datasets.

Here we are using a dataset which consists of video and the corresponding labels in a .csv file. This dataset is fed into the model_preprocess notebook. This notebook preprocess the data using a haar cascade frontal face classifier. The preprocessed data will split into training and testing data for measuring the performance in 80:20 ratio.

From here we will train the model for 80% of the training data. We label the train videos with the corresponding label and then feed the data into the model.

ALGORITHM DESIGNED



Data exploration (Files)

1. Load Data
2. Load Packages
3. Check File types

MetaData Exploration

1. Missing Values
2. Unwanted Data
3. Fixing structural data

Training Data

1. Splitting Datasets
2. Model Training

Video Data Exploration

1. Rescalling Images
2. Missing Real or corresponding fake videos and vice versa.

1. Data Preprocessing

a. Split the video into the frame.

b. Validate the video to check if the video is corrupted or not. If it is then delete the video.

c. Haar Cascade frontal face classifier is used to detect the faces in the frame.

d. Remove the face from the frame. (Cropping)

e. Resize the images so as to have fixed pixel size for better output.

2. Model Architecture

Following are the layers available in our model for training a deep neural network.

a. ResNeXt-50 32*4 dimension pre trained model for feature extraction. It consists of 50 layers with 32 nodes in each layer which is capable of learning a large number of parameters.
b. The output of ResNeXt is a pooling layer which gives us a feature vector which is then fed into a sequential layer.
c. Sequential layer fed the input into the LSTM layer. We have used 1 LSTM layer with 2048 hidden dimensions and with the chance of Dropout of 0.4.
d. The output of LSTM is further processed by linear and adaptive layers.
e. Finally the softmax layer is implemented which gives the output whether the video is Real or Fake.
f. Train_epoch trains the model based on the given number of epochs. Epochs is a hyperparameter that defines the number of times that the learning algorithm will work through the entire training dataset.

   Each epoch consists of a number of batches. In our function we are defining a range of parameters for each epoch. It will compute the best possible epoch value and batch size

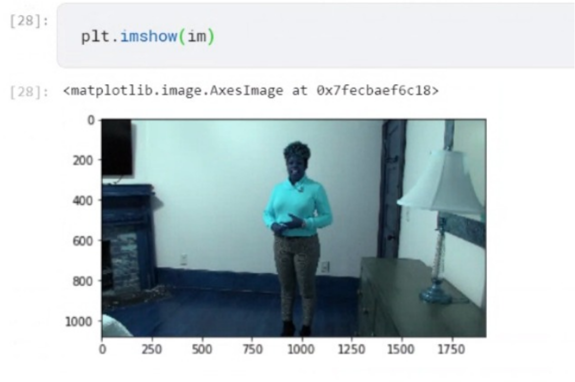
3. Accuracy

a. Confusion metric is used for model evaluation to get the accuracy of the model.

RESULT

Data Preprocessing Result:

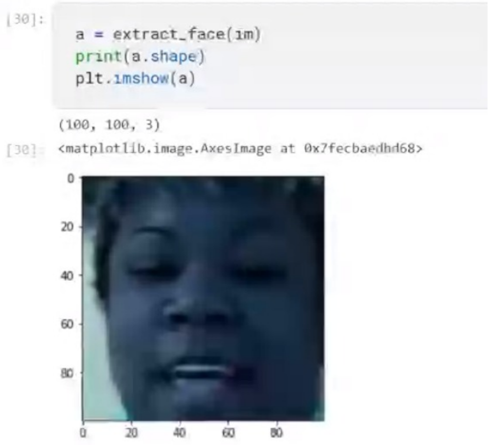
1. Frame Extraction from video



```
[28]: plt.imshow(im)
```

```
[28]: <matplotlib.image.AxesImage at 0x7fecbaef6c18>
```

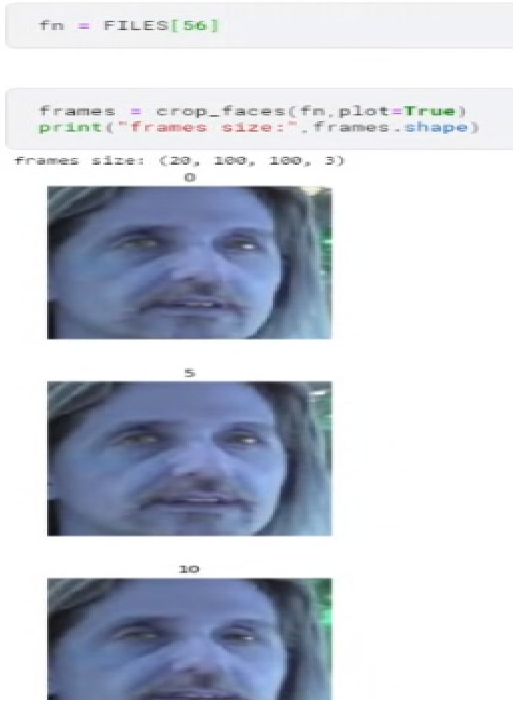
2. Face Extraction from frame.



```
[30]: a = extract_face(im)
      print(a.shape)
      plt.imshow(a)
```

```
(100, 100, 3)
[30]: <matplotlib.image.AxesImage at 0x7fecbaedbd68>
```

3. Cropped faces from the frame and removed an extra part of the image.



```
fn = FILES[56]
```

```
frames = crop_faces(fn,plot=True)
print("frames size:",frames.shape)
```

```
frames size: (20, 100, 100, 3)
```

Data Modelling Result:

The result achieved after training the final model is as we are increasing the number of frames that is sequence length the accuracy also increases. We have checked the sequence length upto 100.

CONCLUSION

IsSwap? is our proposed model to detect the fake faces generated by the state of the art GANs. The proposed model can be used to learn the middle level and high level and discriminative fake features by aggregating the cross-layer feature representations. The proposed feature enables us to do fake feature learning, which allows our trained fake image detector to detect the fake images generated by any GAN, even if it was not included in training. The model developed uses ResNeXt, followed by a sequential layer. This is then passed to an LSTM Layer and softmax layer which predicts whether the video is fake or real. The experimental results demonstrated that the proposed method performed well in predicting whether the video is faker real.

LIMITATIONS

- This technology is not 100% accurate
- It requires access to the internet for uploading video to the web interface
- The file has to be uploaded manually, insertion through URL not available.
- Haar cascades don't seem to be a very convenient tool.
- Setting up the parameters to match various images seems to be nearly manual.
- In case 2 or more faces are present in a given video, the model fails to detect more than 1 face.

REFERENCES

[1]

https://www.researchgate.net/publication/337644519_The_Emergence_of_Deepfake_Technology_A_Review

[2]

https://www.tubefilter.com/2019/05/07/number-hours-video-uploaded-to-youtube-per-minute/#:~:text=The%20platform%27s%20users%20upload%20more,of%20new%20content%20per%20day

[3]

https://www.forbes.com/sites/robtoews/2020/05/25/deepfakes-are-going-to-wreak-havoc-on-society-we-are-not-prepared/?sh=2ab780597494

[4]

https://matplotlib.org/stable/index.html

[5]

https://stackoverflow.com/questions/57751417/what-is-meant-by-sequential-model-in-keras

[6]

https://keras.io/guides/sequential_model/

[7]

https://deepai.org/machine-learning-glossary-and-terms/perceptron#:~:text=A%20Perceptron%20is%20an%20algorithm,a%20single%2Dlayer%20neural%20network

[8]

https://www.tensorflow.org/tutorials

[9]

https://www.cs.cmu.edu/~efros/courses/LBMV07/Papers/viola-cvpr-01.pdf

[10]

https://colah.github.io/posts/2015-08-Understanding-LSTMs/

[11]

https://github.com/facebookresearch/ResNeXt#:~:text=ResNeXt%20is%20a%20simple%2C%20highly,transformations%20with%20the%20same%20topology

[12]

https://www.kaggle.com/c/deepfake-detection-challenge/data

****