Deepfake Detection

Prerna Kumari

Department of Computer Science and Engineering, Chhatrapati Shivaji Maharaj University, Navi Mumbai

Vikas Kumar

Guide, Professor and Head Department of Computer Science and Engineering, Chhatrapati Shivaji Maharaj University, Navi Mumbai

Abstract: ***In this report, we propose another technique to detect fake pictures or facial images, fake audio and fake video which are generated by Artificial Intelligence (AI)- (commonly known as deep fakes). Our technique is based on YOLO deep learning algorithm where a grid is creating from the input picture and predicting bounding boxes and class probabilities for each grid cell. It can recognize multiple faces in an image and can perform real time face recognition.***

Keywords: Deep Fake, Detection, fake picture, artificial intelligence, YOLO

1. Problem Statement

Here Deepfake algorithm is used to examine the facial expression and the body moments, and then it compares the other human facial expression with the same body moments. It also checks the voice that is spreading at the internet with the name of any celebrity is original or not?

Because many celebrities become the victim of such cases where wrong video or voice messages spreading in the wrong way. Worries regarding deepfakes have resulted in a growth of countermeasures, prompting researchers to conduct extensive research. And lectures describing strategies to defend against them abound at computer vision and graphics conferences.

2. Introduction

Now a days fraud activities have become common by implementing Artificial Intelligence (AI). The fraudsters may take the help of fake image, fake audio, and/or fake video with voice. Deepfake video of a celebrity or a reputed person may tarnish their public image in society. Voice cloning with the help of AI has resulted into illegal extortions of money in the form of ransom. Counterfeit 2 cheque leaf of any bank account has been used, in many cases, to withdraw money from others bank accounts, with the help of AI in creating exact image/ replica of the cheque leaf.

In this digital era, vast data is generated every day, necessitating the urgent need of an automated image, audio and video recognition system very important. It will help common people to safeguard themselves from being victim of fraud in their day-to-day life.

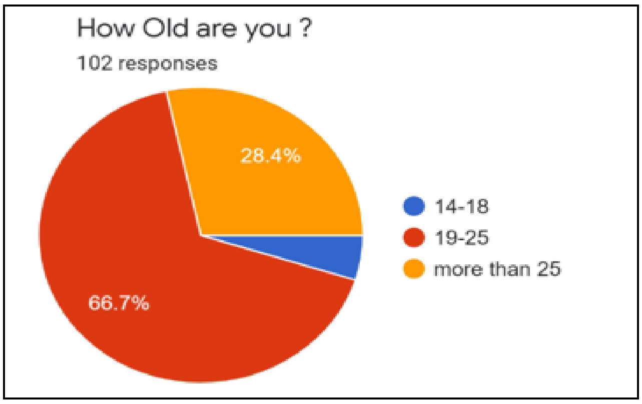
YOLO Deep learning theorem

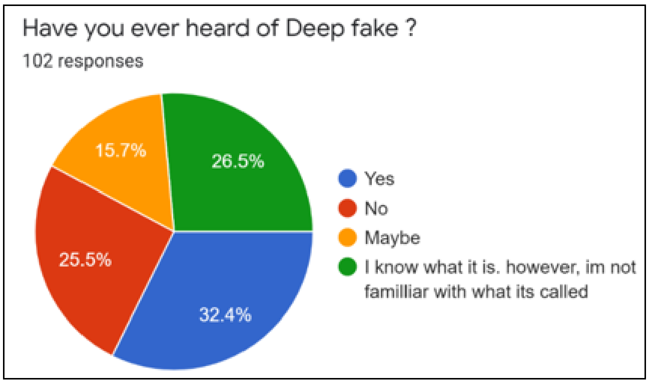
YOLO is a Convolutional Neural Network (CNN), a type of deep neural network, for performing object detection in real-time. CNNs are classifier-based systems that process input images as structured arrays of data and recognize patterns between them. YOLO has the advantage of being much faster than other networks and still maintains accuracy. Read more at: https://viso.ai/deep-learning/yolov3-overview/

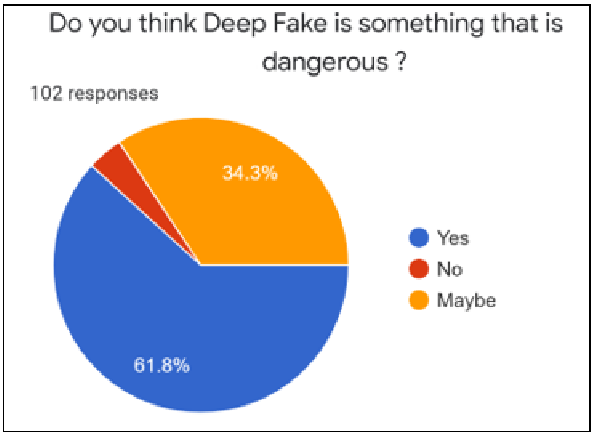
Inception-ResNet-v2 is a convolutional neural network that is trained on more than a million images from the ImageNet database [1]. The network is 164 layers deep and can classify images into 1000 object categories, such as keyboard, mouse, pencil, and many animals. As a result, the network has learned rich feature representations for a wide range of images. The network has an image input size of 299-by-299.

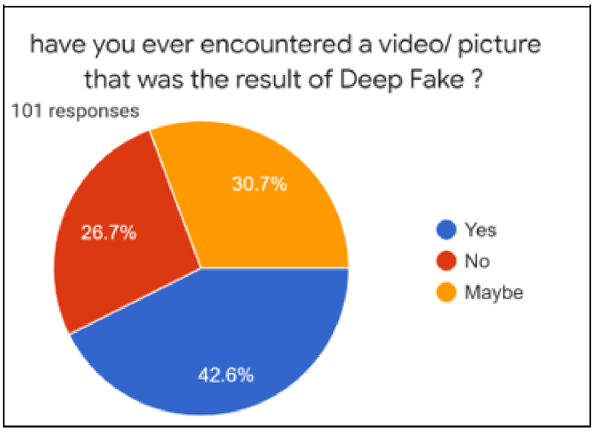
Primary Research: Survey

For our survey and questionnaire, we have decided to create a survey through google forums to help us find out what the common public knows about deepfake, and this could help us use it as proof for what we wrote about deepfake.









3. Proposed Work

The proposed scheme introduces an effective method for detecting deepfakes in videos. Figure 1 shows the system architecture of the suggested deepfake video detection scheme. As seen in Figure 1, the suggested method employed the YOLO face detector to detect faces from video frames. The discriminant spatial-visual features are extracted using the InceptionResNetV2 CNN model. These features help to explore the visual artifacts within video frames and are then distributed into the XGBoost classifier to differentiate between genuine and deepfake videos. The proposed scheme can be explained in detail as follows.

4. Plan of Work

We will Train the selected model using the training dataset.

After that we will Fine-tune the model on the validation set to optimize hyperparameters and improve performance.

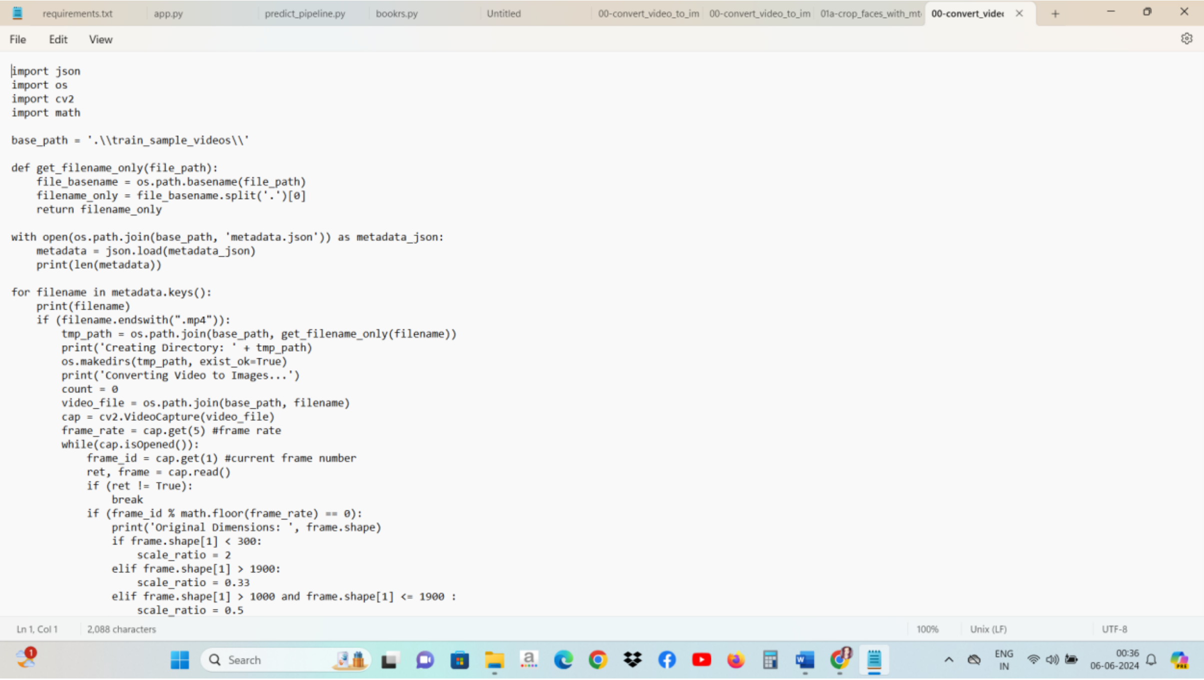
And at the last, we will Implement techniques to prevent overfitting, such as dropout regularization, early stopping, and data augmentation.

The Celeb-DF dataset consists of 408 real and 795. synthesized videos that are made using modified DeepFake generation algorithm.

We will make use of following libraries:

1) Python: Programming language for implementation.
2) TensorFlow or PyTorch: Deep learning frameworks for building and training models.
3) OpenCV: Library for image and video processing.
4) Scikit-learn: Library for machine learning algorithms and evaluation metrics.
5) Flask or Django: Web frameworks for building user interfaces.
6) Docker: Containerization tool for deploying the application.
7) GitHub: Version control system for collaborative development and project management.

Below are some screenshots of the proposed model.



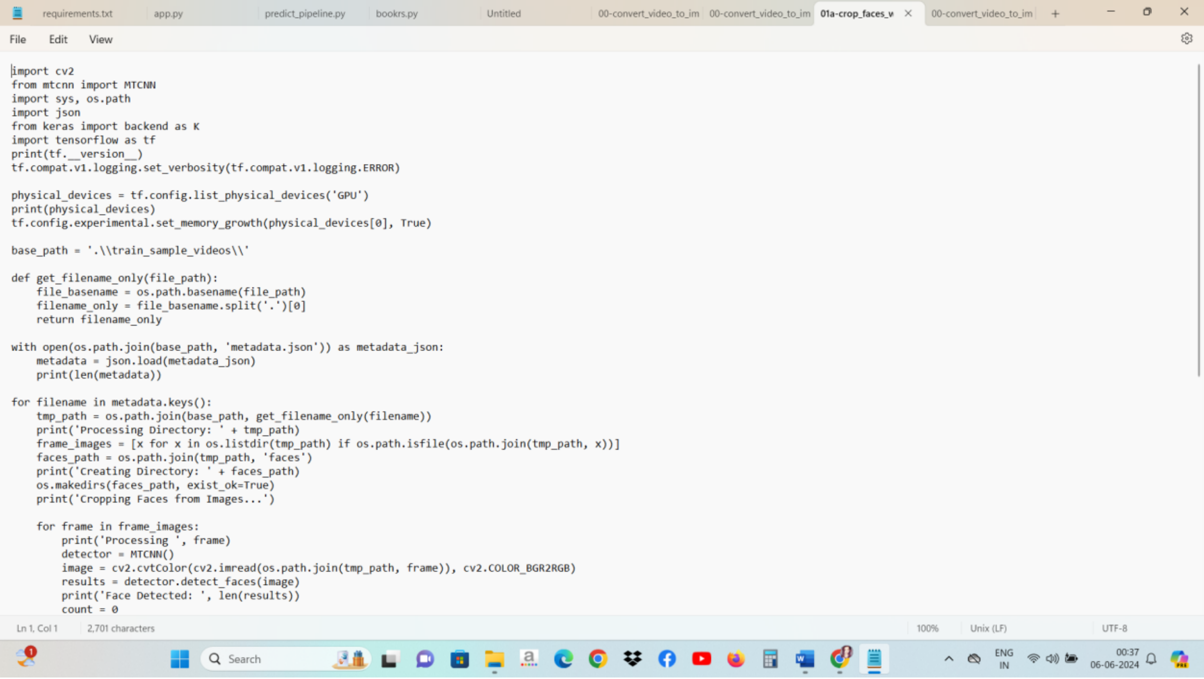
```
import json
import os
import cv2
import math

base_path = '.\\train_sample_videos\\'

def get_filename_only(file_path):
    file_basename = os.path.basename(file_path)
    filename_only = file_basename.split('.')[0]
    return filename_only

with open(os.path.join(base_path, 'metadata.json')) as metadata_json:
    metadata = json.load(metadata_json)
    print(len(metadata))

for filename in metadata.keys():
    print(filename)
    if (filename.endswith(".mp4")):
        tmp_path = os.path.join(base_path, get_filename_only(filename))
        print('Creating Directory: ' + tmp_path)
        os.makedirs(tmp_path, exist_ok=True)
        print('Converting Video to Images...')
        count = 0
        video_file = os.path.join(base_path, filename)
        cap = cv2.VideoCapture(video_file)
        frame_rate = cap.get(5) #frame rate
        while(cap.isOpened()):
            frame_id = cap.get(1) #current frame number
            ret, frame = cap.read()
            if (ret != True):
                break
            if (frame_id % math.floor(frame_rate) == 0):
                print('Original Dimensions: ', frame.shape)
                if frame.shape[1] < 300:
                    scale_ratio = 2
                elif frame.shape[1] > 1900:
                    scale_ratio = 0.33
                elif frame.shape[1] > 1000 and frame.shape[1] <= 1900 :
                    scale_ratio = 0.5
```



```
import cv2
from mtcnn import MTCNN
import sys, os.path
import json
from keras import backend as K
import tensorflow as tf
print(tf.__version__)
tf.compat.v1.logging.set_verbosity(tf.compat.v1.logging.ERROR)

physical_devices = tf.config.list_physical_devices('GPU')
print(physical_devices)
tf.config.experimental.set_memory_growth(physical_devices[0], True)

base_path = '.\\train_sample_videos\\'

def get_filename_only(file_path):
    file_basename = os.path.basename(file_path)
    filename_only = file_basename.split('.')[0]
    return filename_only

with open(os.path.join(base_path, 'metadata.json')) as metadata_json:
    metadata = json.load(metadata_json)
    print(len(metadata))

for filename in metadata.keys():
    tmp_path = os.path.join(base_path, get_filename_only(filename))
    print('Processing Directory: ' + tmp_path)
    frame_images = [x for x in os.listdir(tmp_path) if os.path.isfile(os.path.join(tmp_path, x))]
    faces_path = os.path.join(tmp_path, 'faces')
    print('Creating Directory: ' + faces_path)
    os.makedirs(faces_path, exist_ok=True)
    print('Cropping Faces from Images...')

    for frame in frame_images:
        print('Processing ', frame)
        detector = MTCNN()
        image = cv2.cvtColor(cv2.imread(os.path.join(tmp_path, frame)), cv2.COLOR_BGR2RGB)
        results = detector.detect_faces(image)
        print('Face Detected: ', len(results))
        count = 0
```

5. Conclusion and Future Work

DeepFake detection is a major need in today's world and needs considerable detection techniques as detecting deepfakes will become more challenging in the future. As deepfakes can have major social and political impact improvements should be made continuously in its detection techniques.

Reference

[1] Zhou, Y., & Emil Shi, B. (2021). Photorealistic Facial Expression Synthesis by the Conditional Difference Adversarial Autoencoder. {2}

[2] Foer, F. (2018). The era of the fake video begins. The Atlantic. {3}

[3] Fletcher, J. (2018). Deepfakes, artificial intelligence, and some kind of dystopia: The new faces of online post-fact performance. Theatre Journal, 70(4), 455-471. {4}

[4] Sayler, K. M., & Harris, L. A. (2020). Deep fakes and national security. Congressional Research SVC Washington United States. {5}.