DETECTION ENHANCEMENT FOR VARIOUS DEEPFAKE TYPES BASED ON RESIDUAL NOISE AND MANIPULATION TRACES

Mrs.K.ANU M.E, Assistant Professor of Computer science Department,

Department of Computer Science and Engineering

Mr.S.DEEPAK B.E, Student of Computer Science Engineering

Mr. S. JOTHISHWARAN B.E, Student of Computer science and Engineering

.Joseph College of Engineering, Sriperumbudur, Chennai.

Abstract

As deepfake techniques become more sophisticated, the demand for fake facial video detection continues to increase. Various deepfake detection techniques have been introduced but detecting all types of deepfake videos with a single model remains challenging. We propose a technique for detecting various types of deepfake videos using resnet pretrained model. We adopted a network designed for steganalysis to detect pixel-wise residual-noise traces. We also consider landmarks, which are the primary parts of the face where unnatural deformations often occur in deepfake videos, to capture high-level features. Finally, because the effect of a deepfake is similar to that of blurring, we apply features from various video quality measurement tools that can capture traces of blurring. The results demonstrate that each detection strategy is efficient, and that the performance of the proposed network is stable and superior to that of existing detection networks on datasets of various deep fake types.

Advances in Artificial Intelligence and Image Processing are changing the way people interacts with digital images and video. Widespread mobile apps like FACEAPP make use of the most advanced Generative Adversarial Networks (GAN) to produce extreme transformations on human face photos such gender swap, aging, etc. The results are utterly realistic and extremely easy to be exploited even for non-experienced users. This kind of media object took the name of Deepfake and raised a new challenge in the multimedia forensics field: the Deepfake detection challenge. Indeed, discriminating a Deepfake from a real image could be a difficult task even

for human eyes but recent works are trying to apply the same technology used for generating images for discriminating them with preliminary good results but with many limitations: employed Convolutional Neural Networks are not so robust, demonstrate to be specific to the context and tend to extract semantics from images. In this paper, a new approach aimed to extract a Deepfake fingerprint from images is proposed. The method is based on the Expectation-Maximization algorithm trained to detect and extract a fingerprint that represents the Convolutional Traces (CT) left by GANs during image generation. The CT demonstrates to have high discriminative power achieving better results than state-of-the-art in the Deepfake detection task also proving to be robust to different attacks. Achieving an overall classification accuracy of over 98%, considering Deepfakes from 10 different GAN architectures not only involved in images of faces, the CT demonstrates to be reliable and without any dependence on image semantic. Finally, tests carried out on Deepfakes generated by FACEAPP achieving 93% of accuracy in the fake detection task, demonstrated the effectiveness of the proposed technique on a real-case scenario.

Introduction

The increasing sophistication of mobile camera technology and the ever-growing reach of social media and media sharing portals have made the creation and propagation of digital videos more convenient than ever before. Until recently, the number of fake videos and their degrees of realism has been limited by the lack of sophisticated editing tools, the high demand on domain expertise, and the complex and time-consuming process involved. However, the time of fabrication and manipulation of videos has decreased significantly in recent years, thanks to the accessibility to large-volume training data and high-throughput computing power, but more to the growth of machine learning and computer vision techniques that eliminate the need for manual editing steps. In particular, a new vein of AI-based fake video generation methods known as Deep Fake has attracted a lot of attention recently. It takes as input a video of a specific individual ('target'), and outputs another video with the target's faces replaced with those of another individual ('source'). The back bone of deep fake is deep neural networks trained on face images to automatically map the facial expressions of the source to the target. With proper postprocessing, the resulting videos can achieve a high level of realism.

Literature Survey

Techniques for creating and manipulating multimedia information have progressed to the point where they can now ensure a high degree of realism. Deep Fake is a generative deep learning algorithm that creates or modifies face features in a super realistic form, in which it is difficult to distinguish between real and fake features. This technology has greatly advanced and promotes a wide range of applications in TV channels, video game industries, and cinema, such as improving visual effects in movies, as well as a variety of criminal activities, such as misinformation generation by mimicking famous people. To identify and classify Deep Fakes, research in Deep Fake detection using deep neural networks (DNNs) has attracted increased interest. Basically, Deep Fake is the regenerated media that is obtained by injecting or replacing some information within the DNN model. In this survey, we will summarize the Deep Fake detection methods in face images and videos on the basis of their results, performance, methodology used and detection type.

We will review the existing types of Deep Fake creation techniques and sort them into five major categories. Generally, Deep Fake models are trained on Deep Fake datasets and tested with experiments. Moreover, we will summarize the available Deep Fake dataset trends, focusing on their improvements. Additionally, the issue of how Deep Fake detection aims to generate a generalized Deep Fake detection model will be analyzed. Finally, the challenges related to Deep Fake creation and detection will be discussed. We hope that the knowledge encompassed in this survey will accelerate the use of deep learning in face image and video Deep Fake detection methods

System Design

Designing a deepfake system involves a wide range of technical considerations, including data collection, model selection, training, and deployment. Here are some steps to consider when designing a deepfake system:

The first step in designing a deepfake system is to collect the necessary data to train the models. This may involve collecting images, videos, and audio recordings of the target person. It's important to have a diverse range of data to ensure that the model can learn to accurately mimic the target's appearance and voice.

There are many deepfake models available, and the choice of model will depend on the specific requirements of the project. Popular models include Deep Face Lab, Face swap, and First Order Motion Model. Factors to consider when selecting a model include the accuracy of the model, its speed, and its ease of use.

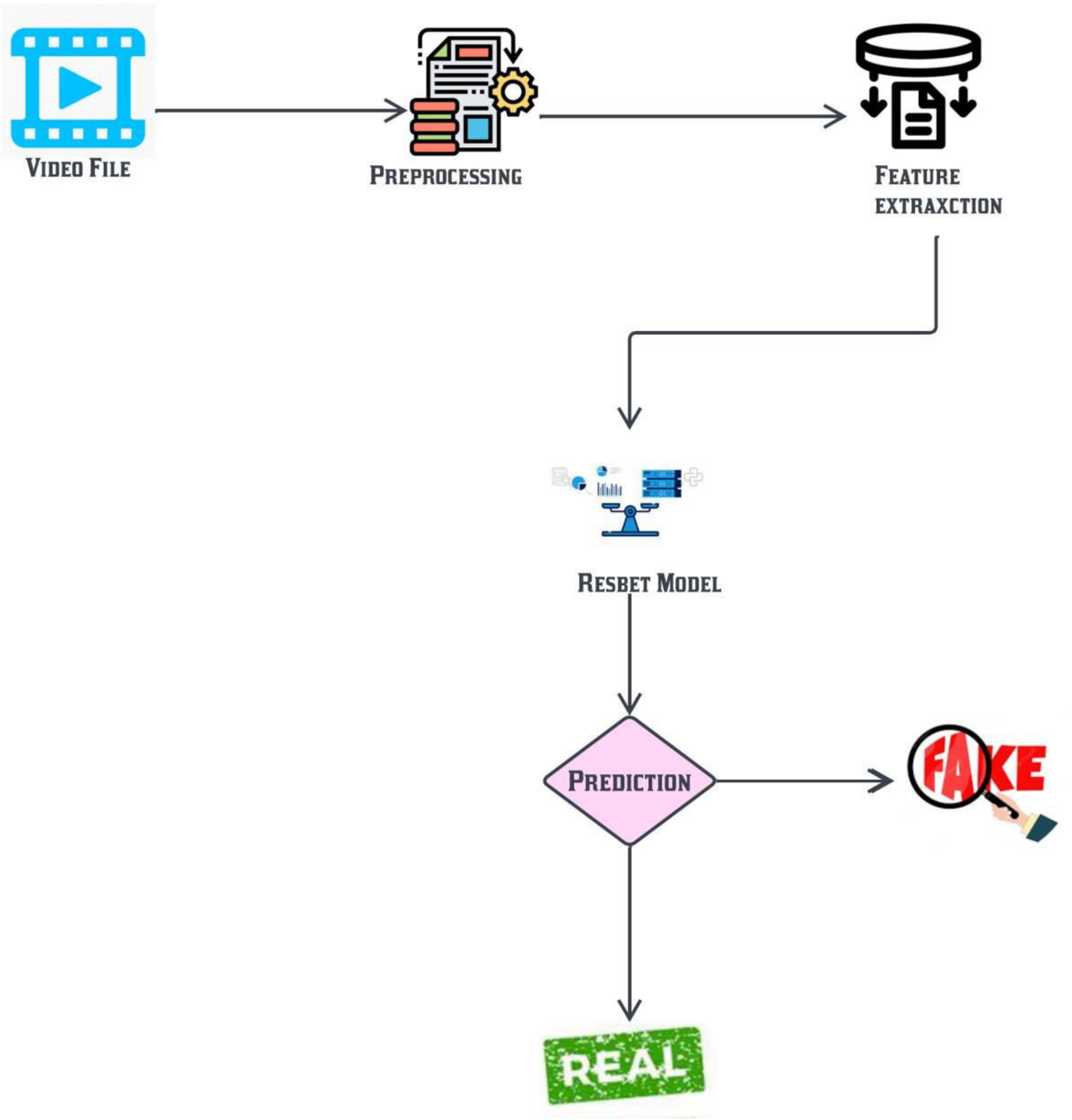
Once the data and model have been selected, it's time to train the model. This typically involves using a large amount of computing power to train the model on the collected data. It's important to ensure that the model is trained on a diverse range of data to ensure that it can accurately mimic the target person.

Once the model has been trained, it can be deployed for use. This may involve integrating the model into a larger application or system. It's important to ensure that the system is secure and that the deepfake technology is used ethically and responsibly.

It's important to evaluate the performance of the deepfake system regularly. This may involve testing the system on new data or in different environments. Regular evaluation can help identify issues and ensure that the system continues to function as intended.

Deepfake technology is constantly evolving, and it's important to design the system with continual improvement in mind. This may involve regularly updating the system with new data or improving the model with new techniques and algorithms.

Deepfake systems may involve collecting and processing sensitive data, such as images or audio recordings of individuals. It's important to ensure that the system is designed with privacy and security considerations in mind, such as encrypting data and restricting access to sensitive information.



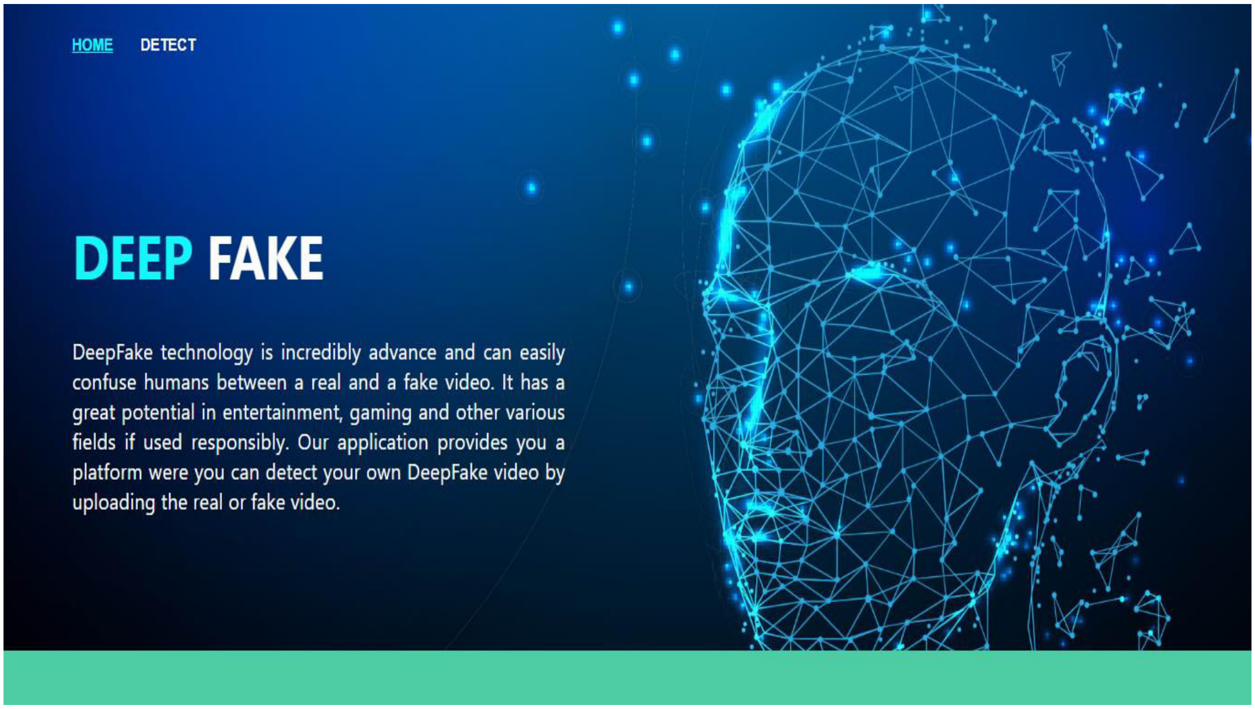
IMPLEMENTTION

```
from flask import Flask, render_template, redirect, request, url_for, send_file
from flask import jsonify, json
from werkzeug.utils import secure_filename
# Interaction with the OS
import os
os.environ['KMP_DUPLICATE_LIB_OK']='True'
```

```
# Used for DL applications, computer vision related processes
import torch
import torchvision
# For image preprocessing
from torchvision import transforms
# Combines dataset & sampler to provide iterable over the dataset
from torch.utils.data import DataLoader
from torch.utils.data.dataset import Dataset
 import numpy as np
import cv2
# To recognise face from extracted frames
import face_recognition
# Autograd: PyTorch package for differentiation of all operations on Tensors
# Variable are wrappers around Tensors that allow easy automatic differentiation
from torch.autograd import Variable
  import time
import sys
# 'nn' Help us in creating & training of neural network
from torch import nn
# Contains definition for models for addressing different tasks i.e. image classification, object detection e.t.c.
from torchvision import models
from skimage import img_as_ubyte
import warnings
warnings.filterwarnings("ignore")
UPLOAD_FOLDER = 'Uploaded_Files'
video_path = "“
detectOutput = []
app = Flask("__main__", template_folder="templates")
```

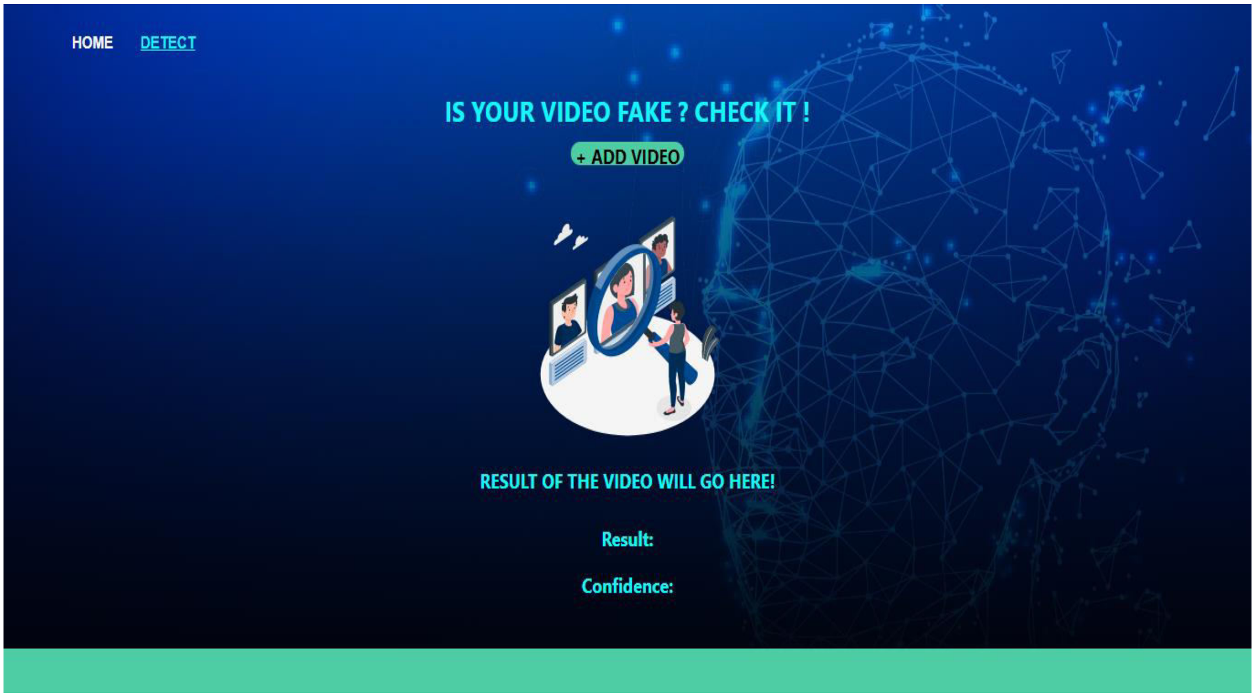
```
app.config['UPLOAD_FOLDER'] = UPLOAD_FOLDER
# Creating Model Architecture
class Model(nn.Module):
 def __init__(self, num_classes, latent_dim= 2048, lstm_layers=1,
hidden_dim=2048, bidirectional=False):
    super(Model, self).__init__()
    # returns a model pretrained on ImageNet dataset
    model = models.resnext50_32x4d(pretrained= True)
     # Sequential allows us to compose modules nn together
    self.model = nn.Sequential(*list(model.children())[:-2])
    # RNN to an input sequence
    self.lstm = nn.LSTM(latent_dim, hidden_dim, lstm_layers, bidirectional)
    # Activation function
    self.relu = nn.LeakyReLU()
 # Dropping out units (hidden & visible) from NN, to avoid overfitting
    self.dp = nn.Dropout(0.4)
    # A module that creates single layer feed forward network with n inputs and m
outputs
    self.linear1 = nn.Linear(2048, num_classes)
    # Applies 2D average adaptive pooling over an input signal composed of
several input planes
    self.avgpool = nn.AdaptiveAvgPool2d(1)
 def forward(self, x):
    batch_size, seq_length, c, h, w = x.shape
    # new view of array with same data
    x = x.view(batch_size*seq_length, c, h, w)
    fmap = self.model(x)
    x = self.avgpool(fmap)
    x = x.view(batch_size, seq_length, 2048)
    x_lstm,_ = self.lstm(x, None)
    return fmap, self.dp(self.linear1(x_lstm[:,-1,:]))
    @app.route('/Detect', methods=['POST', 'GET'])
def DetectPage():
    if request.method == 'GET':
        return render_template('index.html')
    if request.method == 'POST':
        video = request.files['video']
        print(video.filename)
        video_filename = secure_filename(video.filename)
        video.save(os.path.join(app.config['UPLOAD_FOLDER'],
video_filename))
        video_path = "Uploaded_Files/" + video_filename
        prediction = detectFakeVideo(video_path)
 print(prediction)
        if prediction[0] == 0:
              output = "FAKE"
        else:
              output = "REAL"
        confidence = prediction[1]
        data = {'output': output, 'confidence': confidence}
        data = json.dumps(data)
        os.remove(video_path);
        return render_template('index.html', data=data)
app.run(port=3000);
```

SNAPSHOT



DEEP FAKE

DeepFake technology is incredibly advance and can easily confuse humans between a real and a fake video. It has a great potential in entertainment, gaming and other various fields if used responsibly. Our application provides you a platform were you can detect your own DeepFake video by uploading the real or fake video.

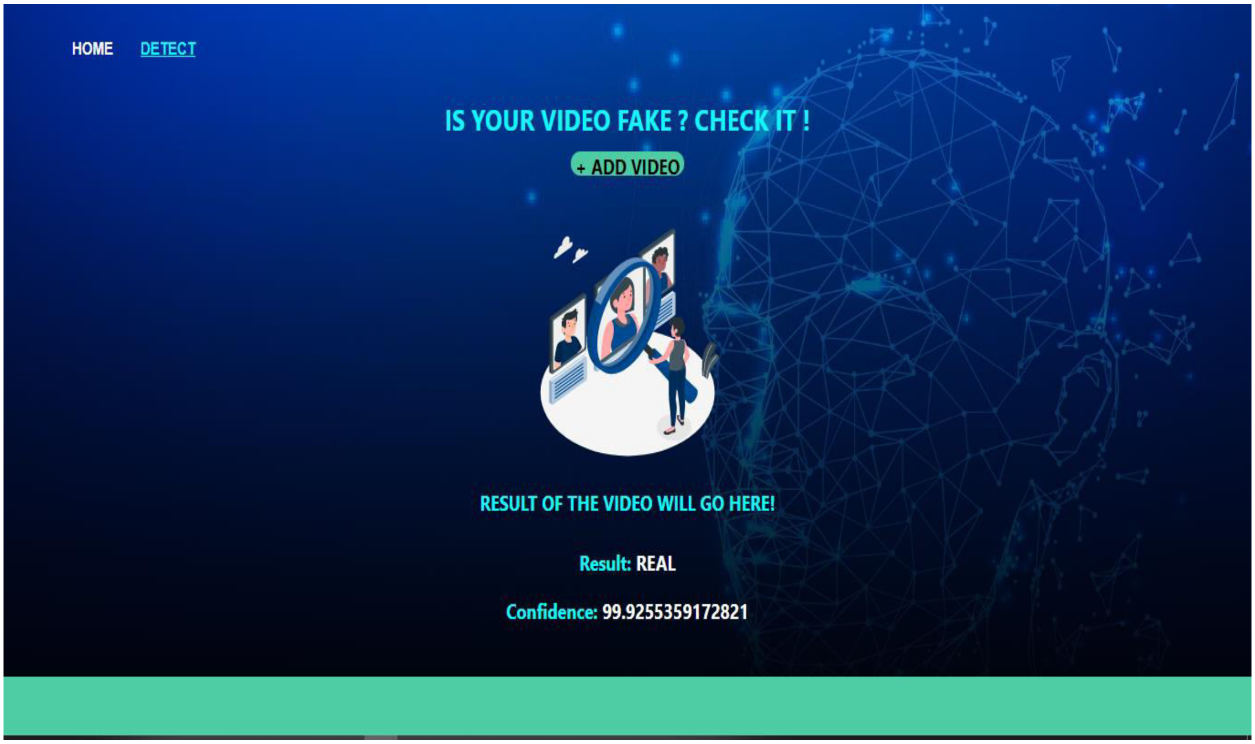


IS YOUR VIDEO FAKE ? CHECK IT !

RESULT OF THE VIDEO WILL GO HERE!

Result:

Confidence:

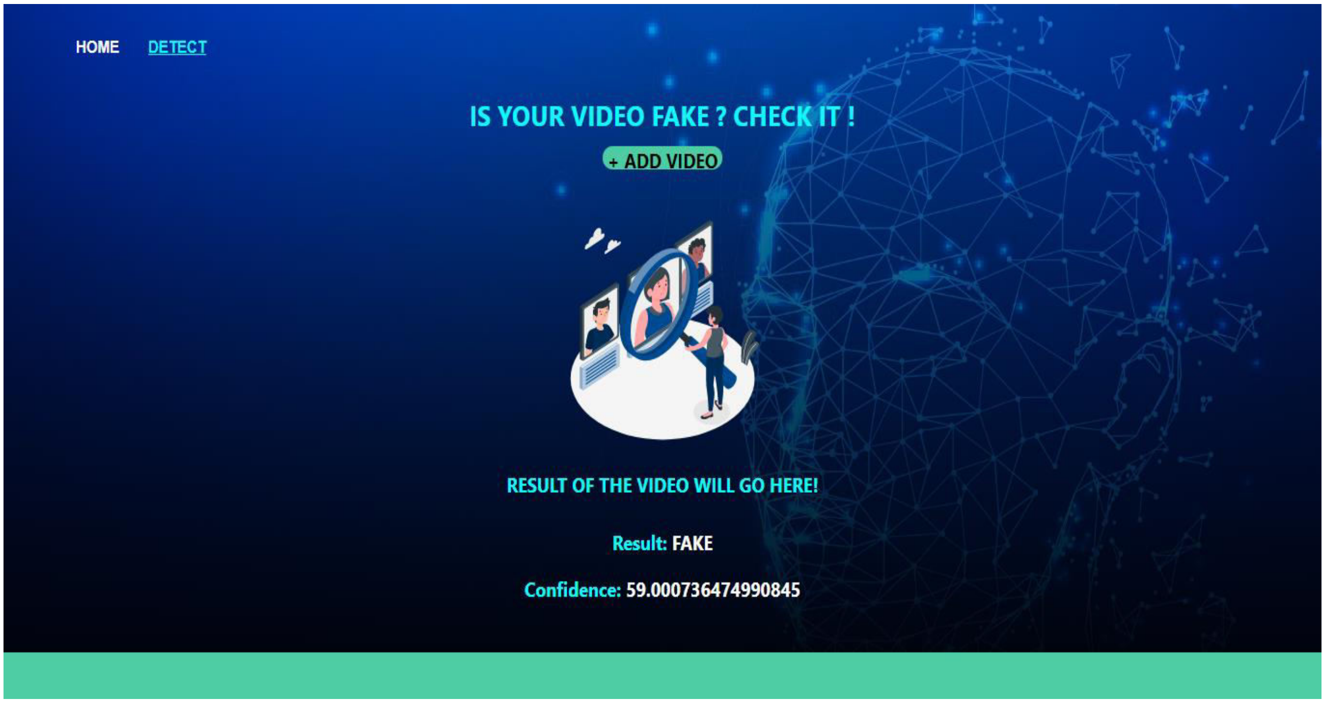


IS YOUR VIDEO FAKE ? CHECK IT !

RESULT OF THE VIDEO WILL GO HERE!

Result: REAL

Confidence: 99.9255359172821



IS YOUR VIDEO FAKE ? CHECK IT !

RESULT OF THE VIDEO WILL GO HERE!

Result: FAKE

Confidence: 59.000736474990845

CONCLUSION

This project is used as a deepfake creation, and detection methods. Deepfake creates forged images or videos that persons cannot differentiate from real images or videos. Deepfakes are created using generative adversarial networks, in which two machine learning models exit. One model trains on a dataset and the other model tries to detect the deepfakes. The forger creates fakes until the other model can't detect the forgery. Deepfakes creating fake news, videos, images, and terrorism events that can cause social and financial fraud. It is increasing affects religions, organizations, individuals and communities', culture, security, and democracy. When deepfake videos and images increase on social media people will ignore to trust the truth. So, deepfake datasets and cross-platform detection techniques need to be developed in the future. This needs efficient, reliable and robust mobile detectors to detect deepfakes in widely used mobile devices. Moreover, will improve deepfake detection by integrating deepfake detection and object detection algorithms.

FUTURE ENHANCEMENTS

This project helps to predict fake face detection networks that exploits macroscopic features are mentioned in this project. In future the project can be extended to give the update in fake detection percentage using the output of current project. As deepfake technology continues to advance, it is essential to develop more sophisticated methods for detecting fake videos. This project can also be used to decrease the privacy problems for famous people faces and morphing videos by detection enhancement in future. Image-based deepfakes involve creating realistic images of individuals, which can be used for a variety of purposes, such as impersonation, entertainment, or artistic expression. Possible future enhancements for image-based deepfakes include: Improving the quality of generated images by using advanced techniques, such as generative adversarial networks (GANs) or attention mechanisms. Increasing the speed of image generation, so that images can be generated in real-time or near real-time.Enhancing ethical considerations, such as developing technologies to detect and prevent deepfakes from being used maliciously.

REFERENCES:

[1] Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, ‘‘Generative adversarial nets,’’ in Proc. Adv. Neural Inf. Process. Syst., 2021, pp. 2672–2680.

[2] S. Wen, W. Liu, Y. Yang, T. Huang, and Z. Zeng, ‘‘Generating realistic videos from keyframes with concatenated GANs,’’ IEEE Trans. Circuits Syst. Video Technol., vol. 29, no. 8, pp. 2337–2348, Aug. 2019.

[3] A. Malik, M. Kuribayashi, S. M. Abdullahi, and A. N. Khan, ‘‘DeepFake detection for human face images and videos: A survey,’’ IEEE Access, vol. 10, pp. 18757–18775, 2022.

[4] J. Damiani, ‘‘A voice deepfake was used to scam a CEO out of $243,000,’’ Forbes, Sep. 2019.

BIOGRAPHY

AUTHOR 1

Mrs. ANU.K,M.E, Assistant professor of Computer Science and Engineering Department in St. Joseph College of Engineering, Sriperumbudur, Chennai, Tamil Nadu. She has completed B.E and M.E in the Anna University Chennai.

AUTHOR 2

Mr. S DEEPAK,B.E Student of Computer Science and Engineering at St.Joseph College of Engineering, Sriperumbudur, Chennai, TamilNadu. Completed data analytical course.

AUTHOR 3

Mr. S.JOTHISHWARAN ,B.E Student of Computer Science and Engineering at St. Joseph College of Engineering, Sriperumbudur, Chennai, Tamil Nadu. I had attended many Workshops, Seminars in Python, Machine Learning. Completed data analytical course.