

## **DEEP FAKE IMAGES AND VIDEOS DETECTION USING DEEP LEARNING TECHNIQUES**

B.S.Murthy Sir<sup>1</sup> Kolukuluri Sunitha<sup>2</sup>,

<sup>1</sup>Assistant professor , M.Sc DEPT, Dantuluri Narayana Raju College, **Bhimavaram, Andharapradesh**

**Email:-** suryanarayanamurthy.b@gmail.com

<sup>2</sup>PG Student of M.Sc, Dantuluri Narayana Raju College, **Bhimavaram, Andharapradesh**

**Email:-** kolukulurisunitha521@gmail.com

### **ABSTRACT**

Now-a-days biometric systems are useful in recognizing person's identity but criminals change their appearance in behavior and psychological to deceive recognition system. To overcome from this problem we are using new technique called Deep Texture Features extraction from Videos and then building train machine learning model using CNN (Convolution Neural Networks) algorithm. This technique refer as LBP or NLBP as this technique heavily dependent on features extraction using LBP (Local Binary Pattern) algorithm. In this project LBP Based machine learning Convolution Neural Network called LBP to detect fake face Videos.

Here first we will extract LBP from Videos and then train LBP descriptor Videos with Convolution Neural Network to generate training model. Whenever we upload new test image then that test image will be applied on training model to detect whether test image contains fake image or non-fake image.

### **1 INTRODUCTION**

#### **WAYS TO DETECTION OF FAKE FACES**

Detecting fake faces can be a challenging task, as fake technology has become increasingly sophisticated. However, there are several methods and techniques that researchers and technology companies have developed to help identify fake faces. Here are some common approaches to detecting fake faces:

#### **Inconsistencies in Facial Features:**

Fake faces often exhibit subtle inconsistencies that may not be present in real faces. These can include unusual lighting and shading, misaligned facial features, or artefacts around the eyes, mouth, or hairline. Deep learning models can be trained to spot these irregularities.

### Blinking and Facial Expressions:

Fake videos may lack natural blinking patterns and facial expressions. Some fake algorithms struggle to generate convincing eye movements and subtle facial changes, which can be detected through careful analysis of the video frames.

## **2 RELEATED WORK**

The broad adoption of Deep Fakes is attributable to the high quality of the faked movies and the ease with which their programmes may be used by a wide variety of users, from professionals to novices with varied degrees of programming ability. The creation of these apps typically involves the use of deep learning methods. It is well-established that deep learning can successfully represent complex and high-dimensional data. For dimensionality reduction, a specific type of deep network called deep autoencoders has been frequently used and image compression . The first effort at deep-fake creation was FakeApp, developed by an Internet user utilising the auto encoder-decoder pairing structure

### **3 implementation study**

#### **Existing System:**

Deep fake technology leverages advanced deep learning algorithms to create convincing but fake images and videos, often used maliciously for spreading misinformation or committing fraud. Current detection methods rely on forensic analysis to identify inconsistencies, feature-based approaches using crafted features like facial landmarks, and machine learning models for classification based on predefined characteristics. However, these methods struggle due to rapid technological advancements that outpace detection capabilities, high computational demands for real-time detection, and the adaptability of deep fake techniques to evade detection.

Detecting sophisticated deep fakes that mimic human behavior closely remains a significant challenge, exacerbated by vulnerabilities to adversarial attacks. Overcoming these hurdles requires enhanced detection techniques capable of addressing evolving deep fake methods and improving resistance against deceptive tactics.

#### **LIMITATIONS:**

- Difficulty in detecting highly sophisticated deep fake creations that mimic human behavior and appearance closely.
- Lack of robustness against adversarial attacks that specifically aim to bypass detection systems.

#### **PROPOSED SYSTEM:**

The proposed system aims to enhance deep fake detection using advanced deep learning techniques. It leverages Convolutional Neural Networks (CNNs) for automated feature extraction from images and video frames, crucial for capturing subtle artifacts indicative of fake content. Additionally, Generative Adversarial Networks (GANs) are utilized to generate realistic but fake images and videos, simultaneously training discriminators to distinguish between authentic and fabricated media.

## **IMPLEMENTATION**

### **INTRODUCTION TO TECHNOLOGY**

Python is a general-purpose interpreted, interactive, object-oriented, and high-level programming language. An interpreted language, Python has a design philosophy that emphasizes code readability (notably using whitespace indentation to delimit code blocks rather than curly brackets or keywords), and a syntax that allows programmers to express concepts in fewer lines of code than might be used in languages such as C++ or Java. It provides constructs that enable clear programming on both small and large scales. Python interpreters are available for many operating systems. CPython, the reference implementation of Python, is open source software and has a community-based development model, as do nearly all of its variant implementations. CPython is managed by the non-profit Python Software Foundation. Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library.

## **5 RESULTS AND DISCUSSION**

### **5.1 SCREENSHOTS**

#### **Fake Image Identification**

Now-a-days biometric systems are useful in recognizing person's identity but criminals change their appearance in behaviour and psychological to deceive recognition system. To overcome from this problem we are using new technique called Deep Texture Features extraction from Videos and then building train machine learning model using CNN (Convolution Neural Networks) algorithm. This technique refer as LBPNet or NLBPNet as this technique heavily dependent on features extraction using LBP (Local Binary Pattern) algorithm.

In this project we are designing LBP Based machine learning Convolution Neural Network called

LBPNET to detect fake face Videos. Here first we will extract LBP from Videos and then train LBP descriptor Videos with Convolution Neural Network to generate training model. Whenever we upload new test image then that test image will be applied on training model to detect whether test image contains fake image or non-fake image. Below we can see some details on LBP.

Local binary patterns (LBP) is a type of visual descriptor used for classification in computer vision and is a simple yet very efficient texture operator which labels the pixels of an image by thresholding the neighborhood of each pixel and considers the result as a binary number. Due to its discriminative power and computational simplicity, LBP texture operator has become a popular approach in various applications. It can be seen as a unifying approach to the traditionally divergent statistical and structural models of texture analysis. Perhaps the most important property of the LBP operator in real-world applications is its robustness to monotonic gray-scale changes caused, for example, by illumination variations. Another important property is its computational simplicity, which makes it possible to analyze Videos in challenging real-time settings.

The LBP feature vector, in its simplest form, is created in the following manner:

Divide the examined window into cells (e.g. 16x16 pixels for each cell).

For each pixel in a cell, compare the pixel to each of its 8 neighbors (on its left-top, left-middle, left- bottom, right-top, etc.). Follow the pixels along a circle, i.e. clockwise or counter-clockwise.

Where the center pixel's value is greater than the neighbor's value, write "0". Otherwise, write "1". This gives an 8-digit binary number (which is usually converted to decimal for convenience).

Compute the histogram, over the cell, of the frequency of each "number" occurring (i.e., each combination of which pixels are smaller and which are greater than the center). This histogram can be seen as a 256-dimensional feature vector.

Optionally normalize the histogram.

Concatenate (normalized) histograms of all cells. This gives a feature vector for the entire window. The feature vector can now be processed using the Support vector machine, extreme learning

machines, or some other machine learning algorithm to classify Videos. Such classifiers can be used for face recognition or texture analysis.

A useful extension to the original operator is the so-called uniform pattern,[8] which can be used to reduce the length of the feature vector and implement a simple rotation invariant descriptor. This idea is motivated by the fact that some binary patterns occur more commonly in texture Videos than others. A local binary pattern is called uniform if the binary pattern contains at most two 0-1 or 1-0 transitions. For example, 00010000 (2 transitions) is a uniform pattern, but 01010100 (6 transitions) is not. In the computation of the LBP histogram, the histogram has a separate bin for every uniform pattern, and all non-uniform patterns are assigned to a single bin. Using uniform patterns, the length of the feature vector for a single cell reduces from 256 to 59. The 58 uniform binary patterns correspond to the integers 0, 1, 2, 3, 4, 6, 7, 8, 12, 14, 15, 16, 24, 28, 30, 31, 32, 48, 56, 60, 62, 63, 64, 96, 112, 120, 124, 126, 127, 128, 129, 131, 135, 143, 159, 191, 192, 193, 195, 199, 207, 223, 224, 225, 227, 231, 239, 240, 241, 243, 247, 248, 249, 251, 252, 253, 254 and 255.

#### CNN working procedure

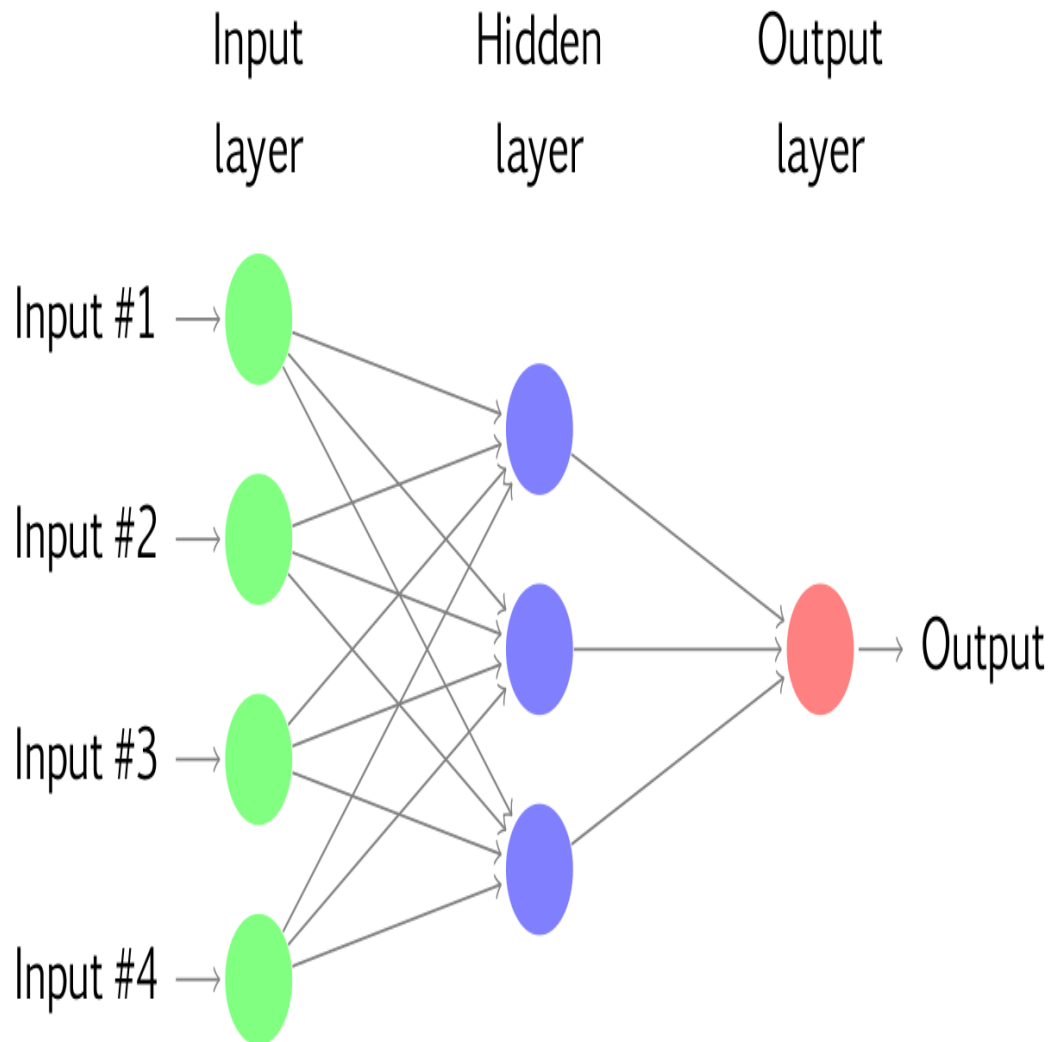
To demonstrate how to build a convolutional neural network based image classifier, we shall build a 6 layer neural network that will identify and separate one image from other. This network that we shall build is a very small network that we can run on a CPU as well. Traditional neural networks that are very good at doing image classification have many more parameters and take a lot of time if trained on normal CPU. However, our objective is to show how to build a real-world convolutional neural network using TENSORFLOW.

Neural Networks are essentially mathematical models to solve an optimization problem. They are

made of neurons, the basic computation unit of neural networks. A neuron takes an input (say  $x$ ), do some computation on it (say: multiply it with a variable  $w$  and adds another variable  $b$ ) to produce a value (say;  $z = wx + b$ ). This value is passed to a non-linear function called activation function ( $f$ ) to produce the final output (activation) of a neuron. There are many kinds of activation functions. One of the popular activation function is Sigmoid. The neuron which uses

sigmoid function as an activation function will be called sigmoid neuron. Depending on the activation functions, neurons are named and there are many kinds of them like RELU, TanH.

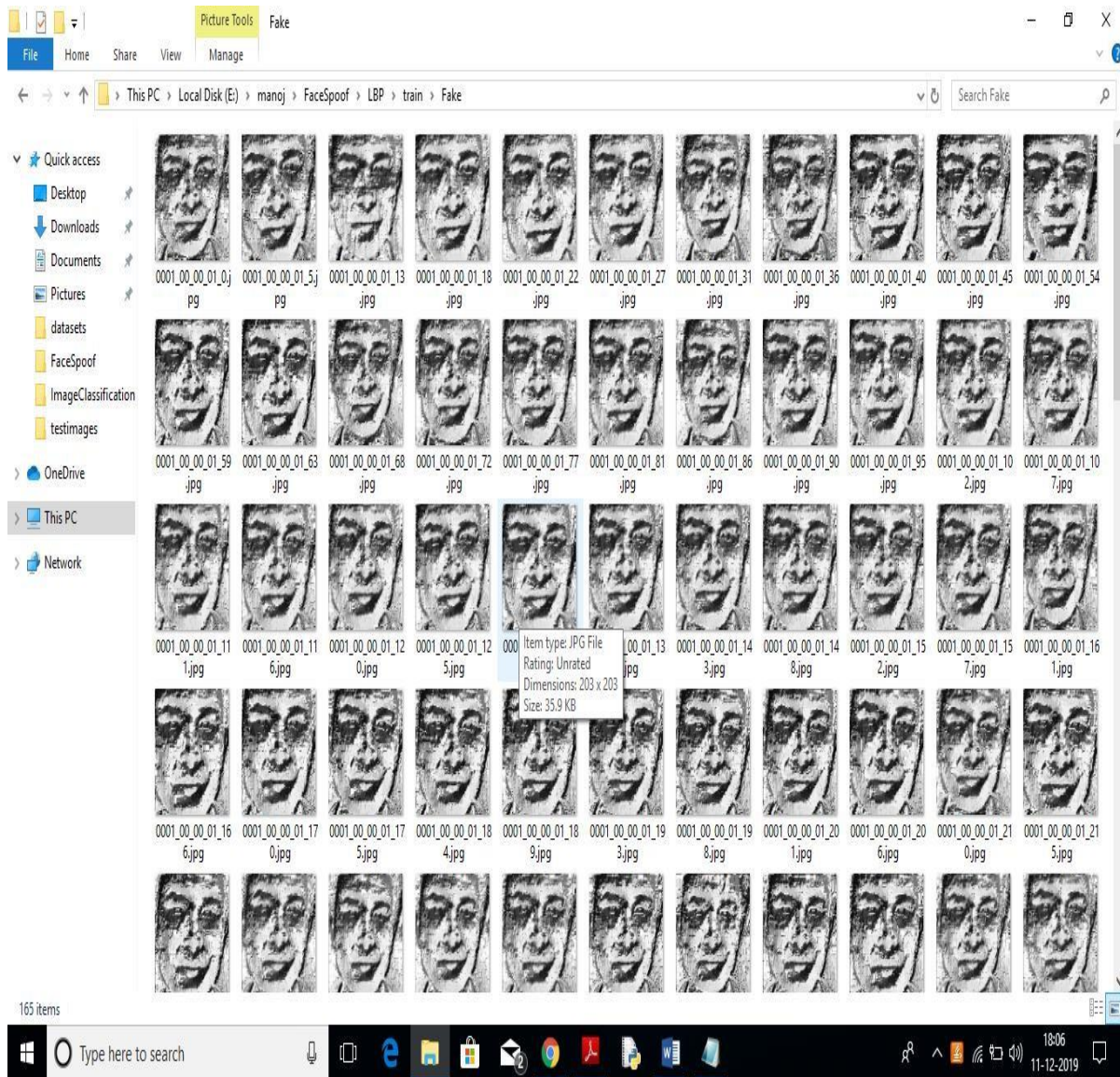
If you stack neurons in a single line, it's called a layer; which is the next building block of neural networks. See below image with layers

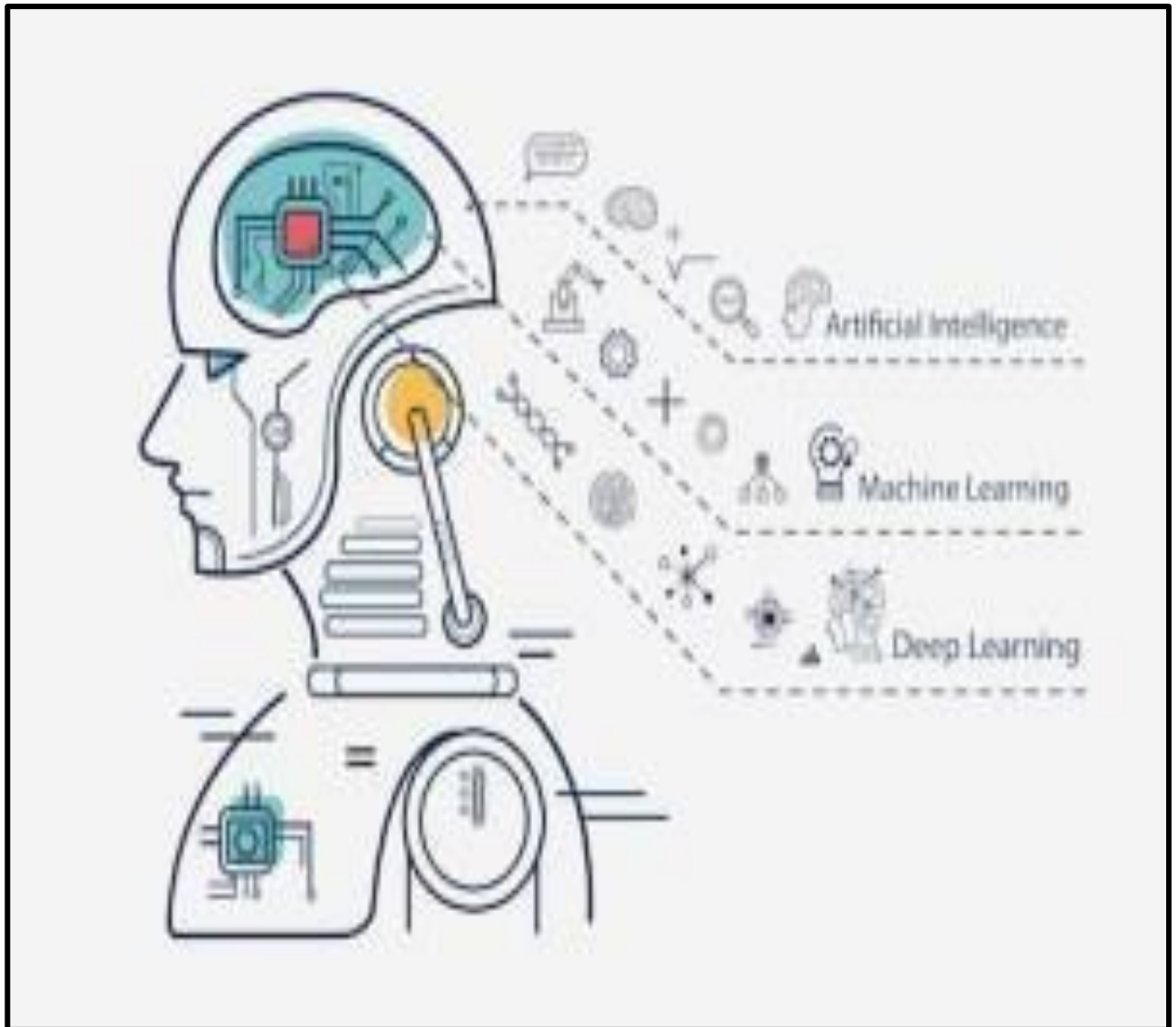


To predict image class multiple layers operate on each other to get best match layer and this process continues till no more improvement left.

Dataset Details:

In this paper author has used NUAA Photograph Imposter (fake) Database with Videos obtained from real and fake faces. We also used Videos and convert that image into LBP format. Below are some Videos from LBP folder



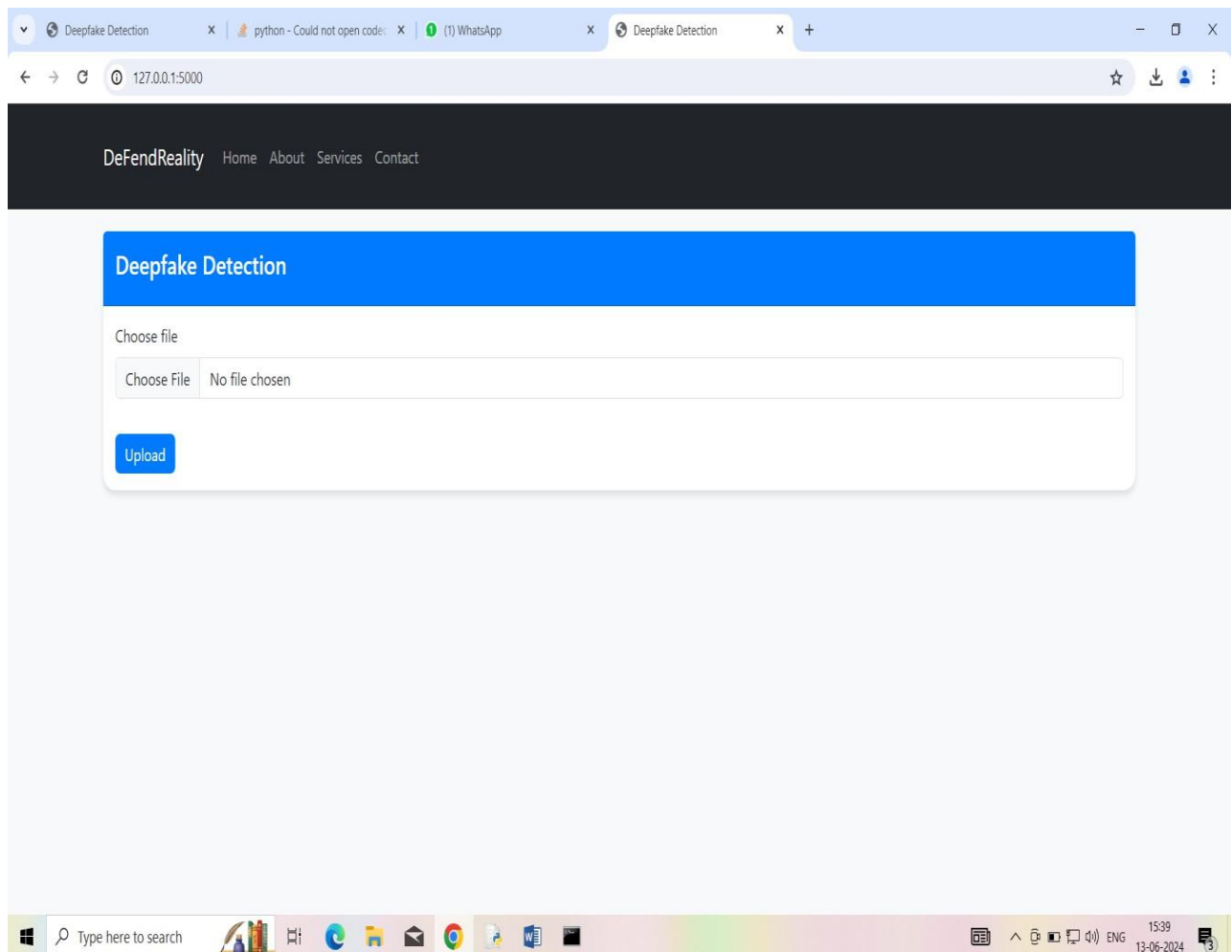


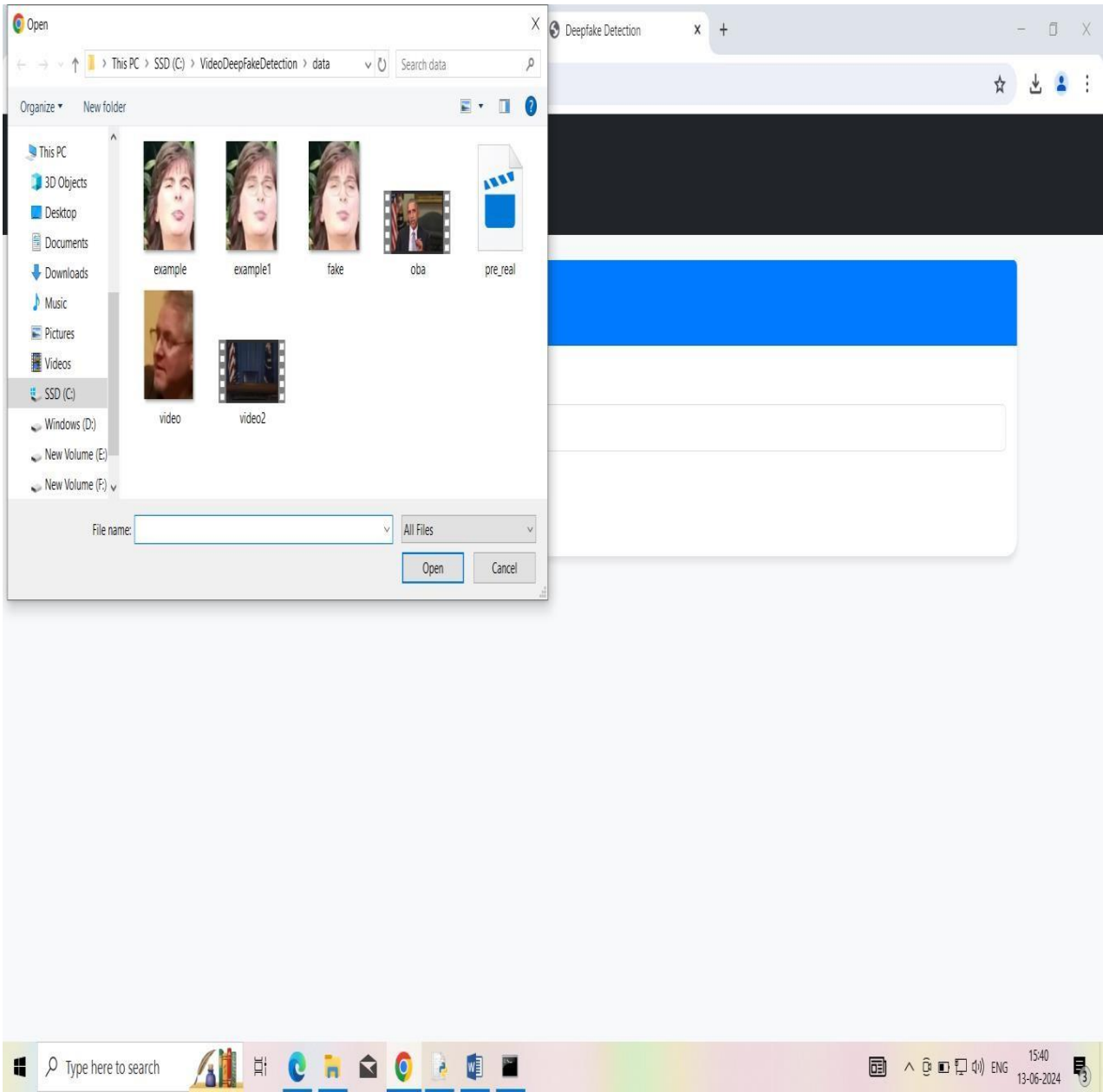
All this fake and real Videos you can see inside ‘ LBP/train’ folder.

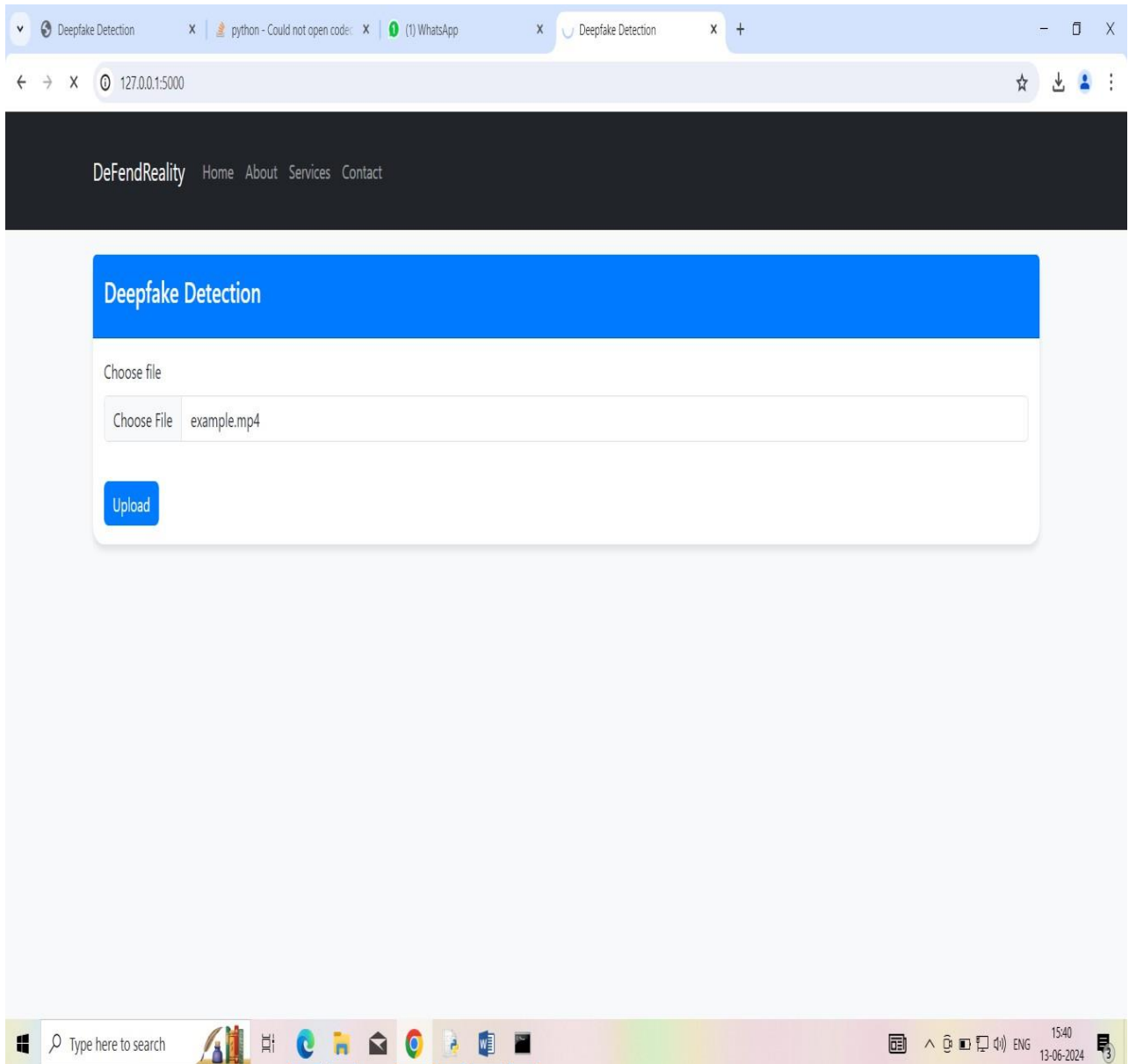


This project consists of following modules:

- 1) Generate NLBPNet Train & Test Model: in this module we will read all LBP Videos from LBP folder and then train CNN model with all those Videos.
- 2) Upload Test Image: In this module we will upload test image from ‘testVideos’ folder. Application will read this image and then extract Deep Textures Features from this image using LBP algorithm.
- 3) Classify Picture In Image: This module apply test image on CNN train model to predict whether test image contains spoof or non-spoof face.







Deepfake Detection

python - Could not open code: X


(1) WhatsApp

Deepfake Detection

127.0.0.1:5000/result?video\_info=%7B%22name%22%3A%22example.mp4%22%2C%22size%22%3A%22141.64+KB%22%2C%22user%22%3A%22Guest%22%2C%22source%22%3A%222024-06-13+10%3A38+UTC%22%2C%22per%22%3A%2260%7D&video\_path2=static/videos%5C1u...

DeFendReality Home About Services Contact

### Deepfake Detection Result



**Video Information:**

**Name:** example.mp4

**Size:** 141.64 KB

**User:** Guest

**Date:** 2024-06-13 10:10:38 UTC

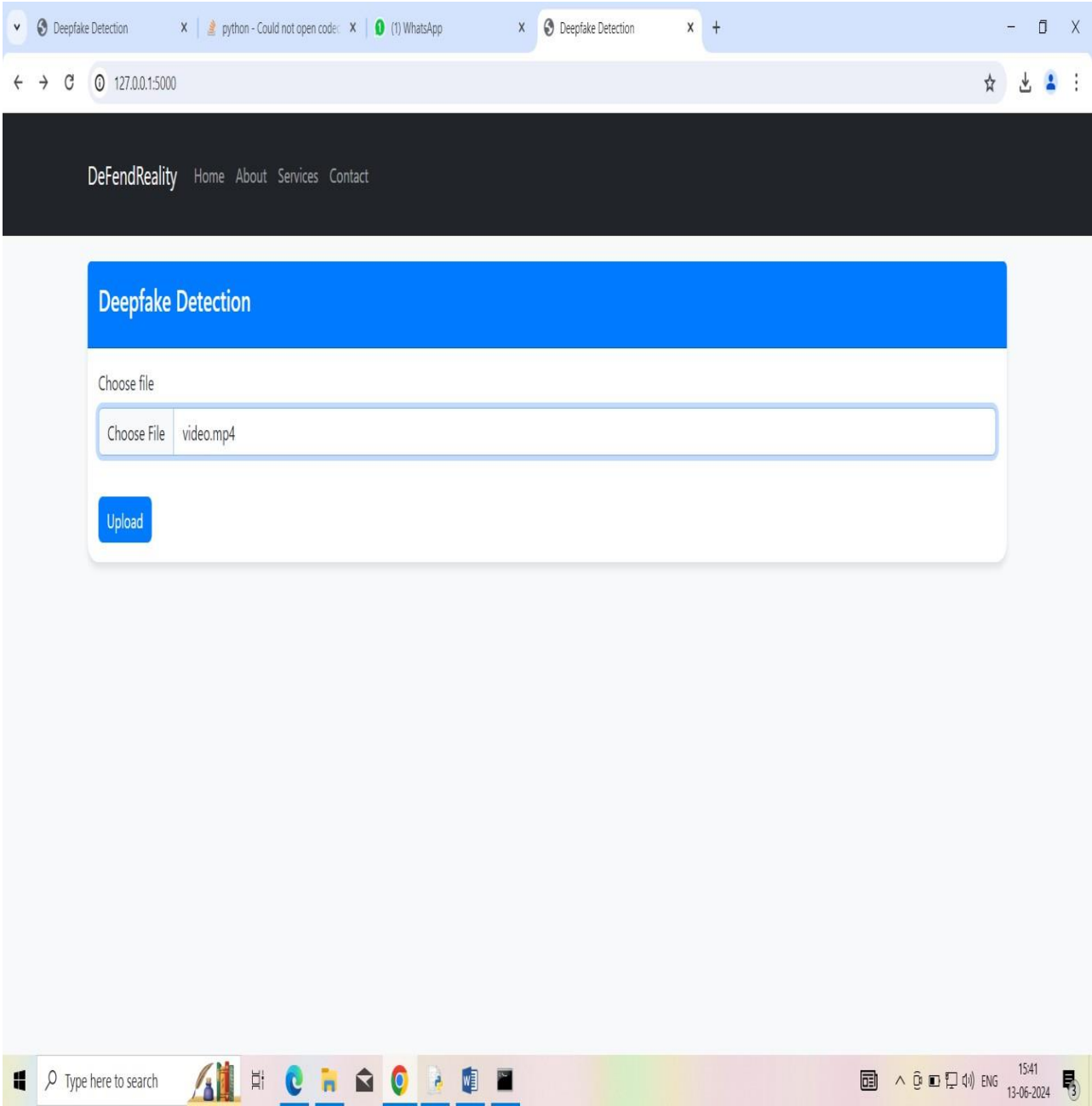
**Deepfake Detection Rate:**

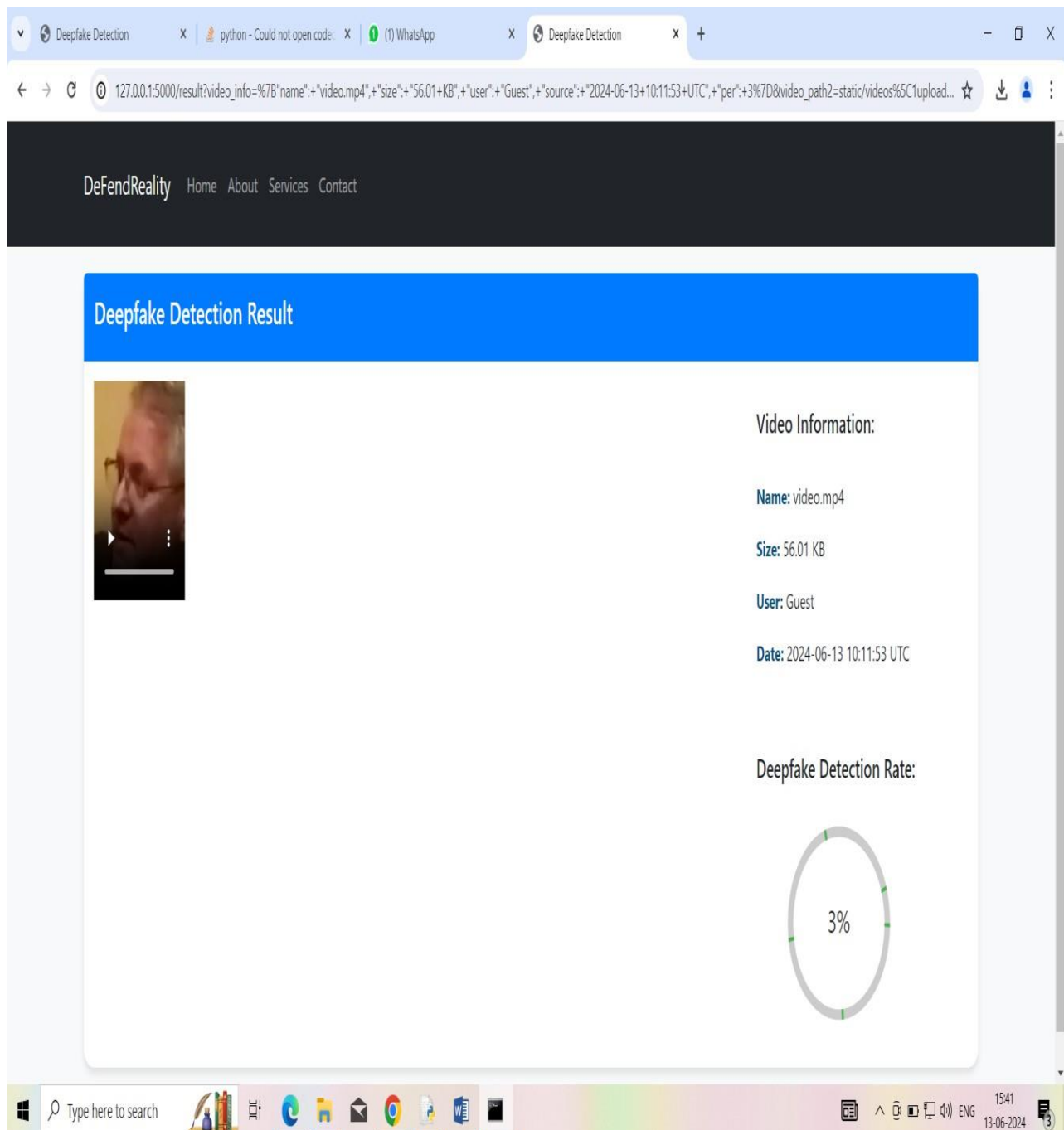
60%

© 2024 Deepfake Detection - Tüm hakları saklıdır.  
Sosyal medyada bizi takip edin: [Twitter](#) | [Instagram](#)

Type here to search

15:41  
13-06-2024





## 6. CONCLUSION AND FUTURE WORK

In this project, we have proposed a novel common fake feature network based the pairwise learning, to detect the fake face/general Videos generated by state-of-the-art GANs successfully. The proposed CFFN can be used to learn the middle- and high-level and discriminative fake feature by aggregating the cross-layer feature representations into the last fully connected layers.

The proposed pairwise learning can be used to improve the performance of fake image detection further. With the proposed pairwise learning, the proposed fake image detector should be able to have the ability to identify the fake image generated by a new GAN. Our experimental results

demonstrated that the proposed method outperforms other state-of-the-art schemes in terms of precision and recall rate.

## 7. REFERENCES

- [1]Karras, T.; Aila, T.; Laine, S.; Lehtinen, J. Progressive growing of gans for improved quality, stability, and variation. arrive Preprint, arXiv:1710.10196 2017. 256
  - [2]Brock, A.; Donahue, J.; Simonyan, K. Large scale gan training for high fidelity natural image synthesis. arXiv Preprint, arXiv:1809.11096 2018.
  - [3]Zhu, J.Y.; Park, T.; Isola, P.; Efros, A.A. Unpaired image-to-image translation using cycle- consistent 259 adversarial networks. arXiv Preprint, 2017.
  - [4]AI can now create fake porn, making revenge porn even more complicated., <http://theconversation.com/ai-can-now-create-fake-porn-making-revenge-porn-even-more-complicated-92267>, 262 2018.
  - [5]Hsu, C.; Lee, C.; Zhuang, Y. Learning to detect fake face Videos in the Wild. 2018 International Symposium 264 on Computer, Consumer and Control (IS3C), 2018, pp. 388– 391. doi:10.1109/IS3C.2018.00104.
  - [6]H.T. Chang, C.C. Hsu, C.Y.a.D.S. Image authentication with tampering localization based on watermark 266 embedding in wavelet domain. Optical Engineering 2009, 48, 057002.
  - [7]Hsu, C.C.; Hung, T.Y.; Lin, C.W.; Hsu, C.T. Video forgery detection using correlation of noise residue. Proc. of the IEEE Workshop on Multimedia Signal Processing. IEEE, 2008, pp. 170– 174. [8]Farid, H. Image forgery detection. IEEE Signal Processing Magazine 2009, 26, 16– 25.
  - [9]Huaxiao Mo, B.C.; Luo, W. Fake Faces Identification via Convolutional Neural Network. Proc. of the ACM Workshop on Information Hiding and Multimedia Security. ACM, 2018, pp. 43– 47. [10]Marra, F.; Gragnaniello, D.; Cozzolino, D.; Verdoliva, L. Detection of GAN-Generated Fake Videos over Social Networks. Proc. of the IEEE Conference on Multimedia Information Processing and Retrieval, 2018, 274 pp. 384– 389. doi:10.1109/MIPR.2018.00084.
  - [11]Chollet, F. Xception: Deep learning with depthwise separable convolutions. Proc. of the IEEE conference on 276 Computer Vision and Pattern Recognition 2017, pp. 1610– 02357.
-