12/04/2022

Deepfake Detection Tool

Functional Specification

Student: Marcin Kaczmarek - C00242533

Supervisor: Richard Butler - BSc (Hons), MSc, CISSP

# Abstract

It is easy to change the format of any image or video today. But, in addition, there may be a lot of different reasons or goals for doing so, too.

When you start looking for ways to improve your image, it goes from simple to more sinister. For example, you might try to fool someone or change their opinion or perception of reality.

**Contents**

[Abstract 1](#__RefHeading___Toc670_1365044094)

[Introduction 3](#__RefHeading___Toc672_1365044094)

[The scope of functional specifications 3](#__RefHeading___Toc674_1365044094)

[Requirements 3](#__RefHeading___Toc676_1365044094)

[Technologies 4](#__RefHeading___Toc678_1365044094)

[Python Components 6](#__RefHeading___Toc680_1365044094)

[Functional Requirements 7](#__RefHeading___Toc682_1365044094)

[Application components 7](#__RefHeading___Toc684_1365044094)

[The project core feauture 7](#__RefHeading___Toc686_1365044094)

[Things I've Run Into 10](#__RefHeading___Toc688_1365044094)

[Python 10](#__RefHeading___Toc690_1365044094)

[Learning Outcomes 11](#__RefHeading___Toc692_1365044094)

[Conclusion 13](#__RefHeading___Toc2088_237122963)

[References 13](#__RefHeading___Toc2090_237122963)

# Introduction

As part of my studies in Cybercrime and IT Security, I created this project.

This document will include the research and development of software that can tell if the images have been modified.

This research will be based on the Benford-Newcomb law, an exciting way to look at the world around us and how things work.

This work will talk about the phenomenon of deepfake, which is all around us in media. This new technology, which has some advantages, has caused a lot of buzz in the news.

It is no longer possible to see the changes that have been made to images, videos, or audio at first glance.

Even if you use commercial deepfake detection tools, it's usually hard to tell if something is fake. They don't always guarantee 100% originality.

My project will try to make a tool based on Benford's prime number law that might be perfect for my project.

# The scope of functional specifications

How the Deepfake detection tool will work is explained in this document

He will talk about how it works, and the technologies used to make it.

A detailed description of how the system will be used and how users will be able to use all of its features.

It will show how users can interact with the system and what they can expect to happen.

If you don't know how to use the tool or what it can do, this will be an easy-to-understand description.

# Requirements

This is what your computer needs.

My app will look at JPEG files and apply Benford's Law to them.

The database will keep the data that was found this way safe.

Users will get a description, a chart that shows the test results and a description of how likely it is that the image being looked at was changed. Deepfake might own the picture, but it could also be accurate.

This app should be able to run on many different devices, like Android, PC, and Mac.

The system will come with software that can handle multiple user accounts.

Relevant and valuable information will be taken from the raw data and shown in a way that is easy for people to understand.

# Technologies

* Linux, Windows or Mac
* Python, as well as any other programming language, could be used
* Flask



 Python is a powerful, object-oriented, high-level programming language with dynamic semantics that may be executed on a computer's desktop. Python's straightforward, simple-to-learn syntax promotes readability while simultaneously lowering the expense of programme maintenance. Debugging Python applications is straightforward since a bug or erroneous input will never result in a segmentation fault.

[1](Python.org, 2019)

 Flask is a microframework for web developers that allows you to rapidly and easily design and grow online applications. It is based on the Python programming language.

What is the role of Python in web development?

Flask is a Python web framework that does not require the use of any additional tools or libraries in order to function. It is written in Python.

Because of its flexibility of use and limited dependencies, it can continue to operate efficiently even as the number of users grows. Even more versatile than Django itself is Flask. This is because Flask is designed to make things as simple as possible and to be able to make smaller apps.

[2](careerfoundry.com, n.d.)



 Ubuntu Linux is a Linux distribution produced by Canonical, and it is one of the most popular distributions due to the simplicity with which it can be used. People who are just starting started with Linux will find it to be one of the best alternatives as well. The server version, which we will not discuss in this article, is likewise installed on the majority of web servers.

It also serves as my primary operating system for day-to-day operations.

[3](Abubakar, n.d.)



 Anaconda is a free and open-source distribution of the computer languages Python and R. It is available on the internet for download. The Python interpreter, as well as a large number of machine learning and data science tools, are included in the download. Because of Anaconda, it is simple for users who are interested in certain subjects to install all (or almost all) of the packages they require with a single installation.

[4](Venture Lessons, 2019)



The Anaconda Repository is a repository for Anaconda software.

We have over 8,000 open-source data science and machine learning packages in our repository, which is constantly growing. These packages, which have been created and compiled using Anaconda for all major operating systems and architectures, are accessible in our repository.

[4](Venture Lessons, 2019)



 Bootstrap is an open-source framework that allows you to build websites that are responsive and mobile-first. That's a lot, so let's take it one section at a time and go through each part of the term.

Bootstrap was originally developed as an internal tool, but it soon gained popularity and was made available as open source. Bootstrap is a collection of pre-written code chunks in CSS, HTML, and JavaScript that may be used to create dynamic web pages. It enables developers to design websites more rapidly than they would be able to if they had to write each and every line of code themselves.

[5](Codecademy News, 2021)

# Python Components

**DASH**

 Derived from Python, Dash is a framework for developing web-based analytical applications. Dash assists in the development of flexible online dashboards that are both visually appealing and extremely quick, without the need to be familiar with difficult front-end frameworks or languages such as HTML, CSS, or JavaScript. Let's use Dash to create our very first web-based dashboard.

[6](GeeksforGeeks, 2020)

**OpenCV**

 OpenCV is a software library for computer vision and machine learning that is free and open source. More than 2500 optimised algorithms, both traditional and state-of-the-art, are available in the collection. It simplifies hard tasks such as face recognition and identification, classifying human activities in films, and extracting 3D object models from 3D objects.

[7](Treehozz.com, 2020)

**NumPy**

 NumPy is the fundamental package for scientific computing in the Python programming language. Python library that provides a multidimensional array object, various derived objects (such as masked arrays and matrices), and an assortment of routines for performing fast operations on arrays, including mathematical and logical operations, shape manipulation, sorting and selecting, I/O, discrete Fourier transforms, basic linear algebra, fundamental statistical operations, random simulation, and much more. It is available as a Python package.

[8](numpy.org, n.d.)

**Matplotlib**

The pyplot package contains a number of functions that allow matplotlib to behave similarly to MATLAB. It is possible to make changes to a figure using each of the pyplot functions. For example, it is possible to build a figure, create a plotting area in a figure, draw some lines in a plotting area, adorn the plot with labels, and so on.

Various states are preserved across function calls in matplotlib.pyplot, so that it can keep track of things like the current figure and plotting area, and the plotting functions are directed to the current axes (please note that "axes" refers to the axes part of a figure, rather than the strict mathematical term for more than one axis).

[9](matplotlib.org, n.d.)

**SciPy**

Optimizing, integrating, interpolating, solving eigenvalue issues, dealing with algebraic equations, dealing with differential equations, dealing with statistics, and solving many more types of problems are all possible using SciPy.

[10](Scipy.org, 2020)

**Pandas library**

Python with Pandas is a Python library that is free and open-source under the BSD licence. Pandas is a collection of data structures and analytical tools for the Python programming language that are high-performance and simple to use. In this video, we will learn how to put Pandas to work in a real-world setting.

[11](www.tutorialspoint.com, n.d.)

These are the most important tools and libraries that I used when I was creating my project.

# Functional Requirements

## Application components

Here I will introduce, step by step, user interactions with my application.

### The project core feauture

My web application is easy to use and intuitive.

On the main page we have a place where you can click and upload our photo for testing.

**Home page**



|  |  |
| --- | --- |
| Upload JPEG file | The user should be able to upload a picture (JPEG format) |

Users will upload an image file (JPEG) to test the file.

|  |  |
| --- | --- |
| File conversion | Convert the JPEG file to a format usable by Benford's law - DCT (cosine functions)  |

All data manipulations will take place here. First, the JPEG format will be converted to binary numbers. Then using Cosine Functions (DCT conversion), we will get data suitable for use and testing with Benford Law.



|  |  |
| --- | --- |
| Result Chart | Display a chart on which the Benford curve and the curve of the image you're looking at will be shown, and how they line up or don't line up on the same chart. This will make any discrepancies more obvious. |



The processed image is presented in the first step in a graph showing the reproductions of the original Benford's law and the image under test.

Both curves are easy to compare and notice any differences.



Because the original and changed images aren't always clear enough, I added a result table showing any changes and the difference in numbers.

|  |  |
| --- | --- |
| Result Table | In this table, we can see the difference in numbers between both Benford distributions.Original and tested image curve. |



This table will show the test result in numerical form. This allows you to oversee the numbers and calculate the differences based on the figures.

# Things I've Run Into

## Python

As a result of Python's organisational differences from other programming languages, there was a significant learning curve to overcome. During the process of installing packages, it became apparent that the files were not being downloaded to the correct location. And that's exactly what occurred. I was able to fix the difficulty while working on the project after conducting some study and making a few adjustments. A pip install command would occasionally fail to execute correctly and would not function properly. Consequently, certain packages might be installed successfully without encountering any issues with the Python set-up command that had been left out of the package installation process. If you're using Python 2.7, you may have problems with the version you're using. Even though I ran across a few problems, the majority of them were straightforward fixes, such as renaming a package folder or altering the approach I was using to get the code to run.

As a result, Benford's Law was applied to JPEG files, which was the purpose of the analysis carried out.

When Python utilised the DCT function, it was able to deal with the quantization of the grey number without issue. Putting the DCT figures on a graph and making it simple to examine things like that was a crucial part of my effort, which included the graphical interface.

**Users will like the GUI.**

People all over the world see the project as a way to help. The buttons are significant, the images are enormous, and the window is a fixed size to avoid scaling issues and keep things the same.

**Then, make a live graph.**

I found matplotlib to be very useful. It was easy to set up, but there were some placement and live display problems. People who work with images have to wait a long time to see the live graphs. That's what I think. It's because it's updating all numbers from 1 to 9 on the canvas and not just one point. People who don't know a lot about Benford's curve and how to tell if something is real or fake might not be able to follow a number array.

**The chart**

As I was expecting to see much more significant differences in the chart, I decided to add a table that shows the differences in numbers as another way to show that the chart is unique.

**What I have failed to do with the planned utilities**

1. The process for logging in was almost complete. However, there was a problem with it, and I was unable to log in as a new user on sometimes because of it. As a result, I had to disable all of these methods in order to maintain the primary component fully working and ready for presentation at the conference.
2. Unfortunately, the mechanism for producing NTFs has not even reached the point of a more in-depth investigation into the possibilities and ways of putting it into action.

# Learning Outcomes

**Benford's Law**

A Benford Law was unknown to me before this project, so I didn't know about it. So how did my search help me? First, I learned a lot about it. In the past, I've always thought math was the most potent kind of science, and Benford's Law proves it even more now.

**Compression methods for JPEG**

JPEG compression is the key to verifying an image because it uses DCT, which uses cosine functions to find its data. DCT is used to find the data in an image. Quality "Q" will go down as a picture is more compressed, making it look bad (100 meaning full quality). Having a lower number of 1's. JPEG isn't a topic in my field, so I found it hard to understand. This is because image processing isn't a topic in my area.

**Distribution of DCT coefficients**

In a project I worked on, DCT was one of the essential parts. However, I found it hard to understand because a lot of it was math equations that changed data through graphs.

**Testing**



So, this is my flag testing picture.

On top, we can see the original one and below altered.



# Conclusion

As part of my studies in cybercrime and IT security, I created a project called the Deepfake Detection Tool. This publication provided an overview of the research and development of software that could detect whether or not photos had been altered. This investigation was based on the Benford-Newcomb law, which is a fascinating way of looking at the world around us and how things operate in general. This report was discussing the topic of deepfakes, which is prevalent in the media and can be seen everywhere. This new technology, which offers a number of advantages, has generated a great deal of interest in the media. I was speaking about Benford's Law, JPEG compression approaches, and the distribution of DCT coefficients, among other things.

In general, I'm pleased with my final design.

However, despite my conviction that the concept has been executed appropriately, I would not have complete faith in the test findings.

This tool can be useful as one of a number of methods for determining whether or not a picture has been altered in any way.

# References

[1]

Python.org (2019). *What is Python? Executive Summary*. [online] Python.org. Available at: https://www.python.org/doc/essays/blurb/.

[2]

careerfoundry.com. (n.d.). *The Flask Web Framework: A Beginner’s Guide*. [online] Available at: https://careerfoundry.com/en/blog/web-development/what-is-flask/.

[3]

Abubakar, M. (n.d.). *What Is Ubuntu?* [online] How-To Geek. Available at: https://www.howtogeek.com/763775/what-is-ubuntu/.

[4]

Venture Lessons. (2019). *What is Anaconda and how does it relate to Python?* [online] Available at: https://www.venturelessons.com/what-is-anaconda/.

[5]

Codecademy News. (2021). *What Is Bootstrap: A Beginner’s Guide to Bootstrap*. [online] Available at: https://www.codecademy.com/resources/blog/what-is-bootstrap/.

[6]

GeeksforGeeks. (2020). *Introduction to Dash in Python*. [online] Available at: https://www.geeksforgeeks.org/introduction-to-dash-in-python/ [Accessed 13 Apr. 2022].

[7]

Treehozz.com. (2020). *What is cv2 Python?* [online] Available at: https://treehozz.com/what-is-cv2-python [Accessed 13 Apr. 2022].

[8]

numpy.org. (n.d.). *Overview — NumPy v1.19 Manual*. [online] Available at: https://numpy.org/doc/stable/.

[9]

matplotlib.org. (n.d.). *Pyplot tutorial — Matplotlib 3.4.2 documentation*. [online] Available at: https://matplotlib.org/stable/tutorials/introductory/pyplot.html.

[10]

Scipy.org. (2020). *SciPy.org — SciPy.org*. [online] Available at: https://scipy.org/.

[11]

www.tutorialspoint.com. (n.d.). *Python Pandas Tutorial - Tutorialspoint*. [online] Available at: https://www.tutorialspoint.com/python\_pandas/index.htm.