National College of Ireland

Configuration Manual

DeepFakeCNN: Deep Fake Image detection and Security Breach detection using Convolutional Neural Networks

MSc Research Project
MSc in Cybersecurity

Darshan Siddaiah
Student ID: x22187456

School of Computing
National College of Ireland

Supervisor: Prof. Vikas Sahni

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Darshan Siddaiah
Student ID:	22187456
Programme:	MSc in Cybersecurity **Year:** 2024
Module:	Practicum
Lecturer:	Prof. Vikas Sanhi
Submission Due Date:	16-09-2024
Project Title:	DeepFakeCNN: Deep Fake Image and video detection using Convolutional Neural Networks
Word Count:	1111 **Page Count:** 6

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Darshan Siddaiah

Date: 12-08-2024

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

DeepFakeCNN: Deep Fake Image and video detection using Convolutional Neural Networks

Darshan Siddaiah
Student ID: 22187456

1 Introduction

This document provides all the hardware and software required to carry on this research successfully. All the steps involved in accomplishing this research are mentioned logically to be repeated easily by anyone. This research compares different CNN (D**enseNet121, InceptionResNetV2, VGG16 and EfficientNetB7**) and RNN (**Gated Recurrent Unit (GRU), LSTM and SimpleRNN**) models, to accurately identify the fake videos from real videos. Fake videos are videos which can be AI generated or used AI tools to alter the video.

2 System configuration

This section focuses on hardware and software required to execute this research successfully.

2.1 Hardware configuration

This research can be conducted on a personal computer/laptop h configuration as mentioned in the table 1.

Hardware	Configuration
System	Asus rog flow x13
System Type	64 bit
RAM	32 GB
Graphics	none
SSD Memory	500 GB
Processor	AMD Ryzen 7 6800HS 3.20 GHz

Table1: Hardware Configuration

2.2 Software configuration

All the requirements related to software configuration are mentioned in this section. The details of the operating system required for the successful implementation of this research is mentioned in the Table2:

Specification	Value
Edition	Windows 11 Home Single Language
Version	23H2
OS Build	22631.3880

Table2: Software Configuration

The dataset used in this research consists of high-quality videos, which require GPU processing. Given this need, Google Colab is the ideal platform for running Python code. To access Google Colab, you just need a Gmail account, and it can be opened in any web browser. In this research, Google Chrome was used to access Google Colab.

Google Colab is particularly convenient because it comes with all the necessary libraries pre-installed and offers code suggestions as you write, making the coding process smoother and more efficient. Figure 1 illustrates the version of Chrome used in this research.

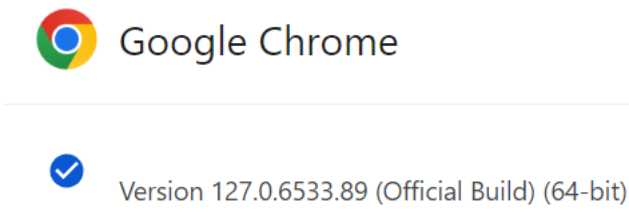

Figure1: Browser Specification

2.3 Google Colab Configuration

We will discuss the steps to execute the python code in Google Colab

- We need to have an active Gmail account access Google Colab.
- Visit the website[1].
- Login to Google Colab with your Gmail account shown in Figure 2.
- You can open a new notebook or upload an existing one to the Google Colab to execute python code

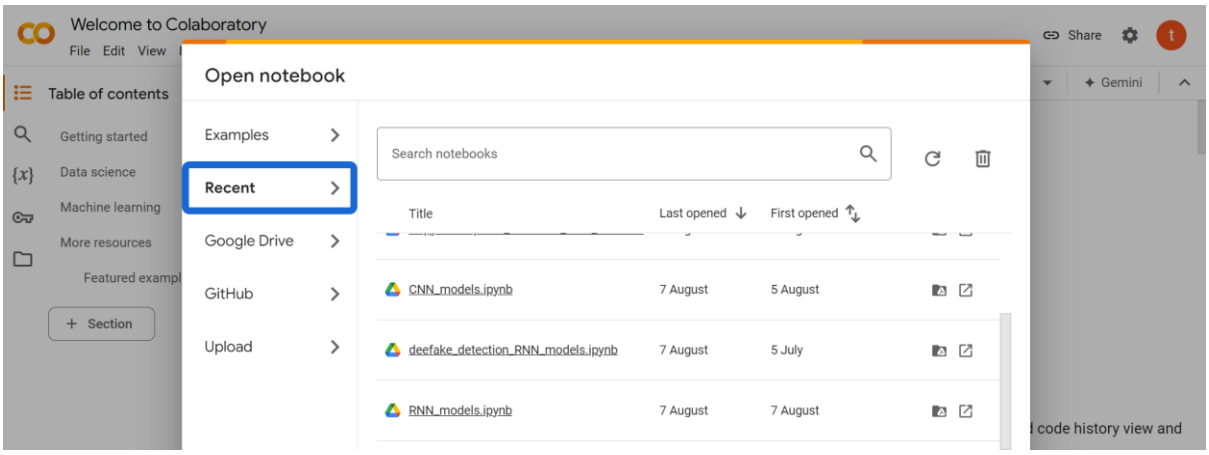

Figure 2: Google Colab

2.4 Libraries

The programming language used is **Python** version **3.10.12**. The libraries used are mentioned below with their version.

[1] https://colab.research.google.com/?utmsource

Library	Version
Pandas[2]	2.1.4
TensorFlow[3]	2.17.0
Keras[4]	3.4.1
Numpy[5]	1.26.4
imageio[6]	2.34.2
OpenCV2[7]	4.10.0

Table 3: Libraries and their version

3 Project Implementation

3.1 Data Collection

The Deepfake Detection Challenge (DFDC) dataset[8] is publicly available and can be downloaded from Kaggle. The dataset is a novel dataset for the research. Dataset contains high quality videos, and the size of the dataset is around 4.44 GB as shown in Figure 3 below.

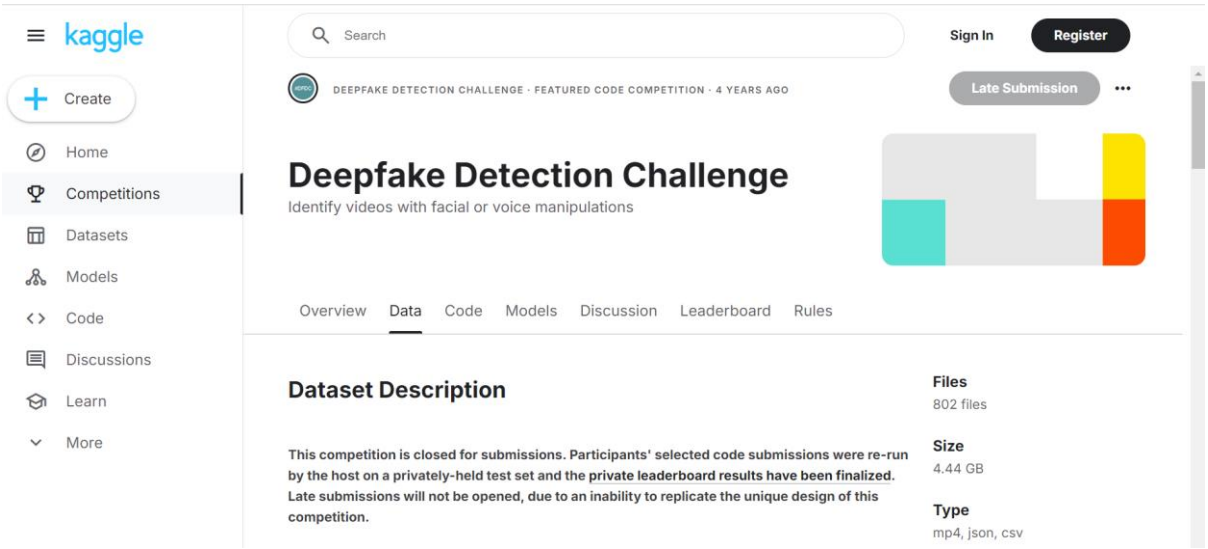

Figure 3

The DFDC dataset contains 2 main folders train_sample_videos (401 videos) and test_videos (400 videos). The dataset was made up of video files, each of which had labels that indicated whether the video was a fake or a real one. We have a Jason file named metadata JSON which has three columns it includes:

Label: Indicates whether the video is a fabrication or a genuine one.

[2] https://pandas.pydata.org
[3] https://www.tensorflow.org
[4] https://keras.io
[5] https://numpy.org
[6] https://pypi.org/project/imageio
[7] https://opencv.org
[8] https://www.kaggle.com/c/deepfake-detection-challenge/data

Split: It is a parameter that indicates whether the video is a part of the training set, or the validation set.

Original: When referring to fraudulent videos, this term refers to the original video file

3.2 Model Building

This research relies on Python as the scripting language to develop deep learning models. For the models to run successfully, several key libraries and packages are essential:

- **NumPy and pandas**: These are crucial for manipulating and analysing data.
- **Keras**: This library is vital for importing tools needed to build deep learning models, including DenseNet121, InceptionResNetV2, VGG16, and EfficientNetB7.
- **TensorFlow**: Used to create large-scale neural networks, TensorFlow is essential for tasks such as classification and prediction.
- **Imageio**: This library provides a simple way to read and write various types of image data.
- **OpenCV**: OpenCV is necessary for image processing tasks.

Having these packages in place is critical for the successful execution of the deep learning models in this research.

3.3 Model Evaluation and Visualization

After training all the models, the next crucial step is to evaluate their performance using specific metrics. To effectively compare the models, it's important to generate visualizations like plots, confusion matrices, and heatmaps. To accomplish this, you'll need to install the following Python packages:

- **sklearn**: This library is essential for generating confusion matrices and calculating evaluation metrics such as recall and precision.
- **matplotlib**: To gain a clearer understanding of the model's performance, you'll use this library to plot graphs comparing metrics like loss, accuracy, and recall across training and testing data.
- **seaborn**: This package is specifically needed to create heatmaps for visualizing the confusion matrix.

These tools are key to thoroughly evaluating and comparing the models.