National College of Ireland

Configuration Manual

MSc Research Project
Data Analytics

Aboli Kapadnis
Student ID: x19218451

School of Computing
National College of Ireland

Supervisor: Prof. Hicham Rifai

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Aboli Kapadnis
Student ID:	x19218451
Programme:	MSc Data Analytics
Year:	2020-21
Module:	MSc Research Project
Supervisor:	Prof. Hicham Rifai
Submission Due Date:	16/08/2021
Project Title:	Configuration Manual
Word Count:	405
Page Count:	9

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:	
Date:	17th September 2021

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):	

Configuration Manual

Aboli Kapadnis
x19218451

1 Introduction

The objective for configuration manual is provide details of the system setup, software specification, hardware platforms and steps for the implementing the research project: Brain Tumour Detection using Deep Learning with AlexNet

2 System Configuration

2.1 Hardware

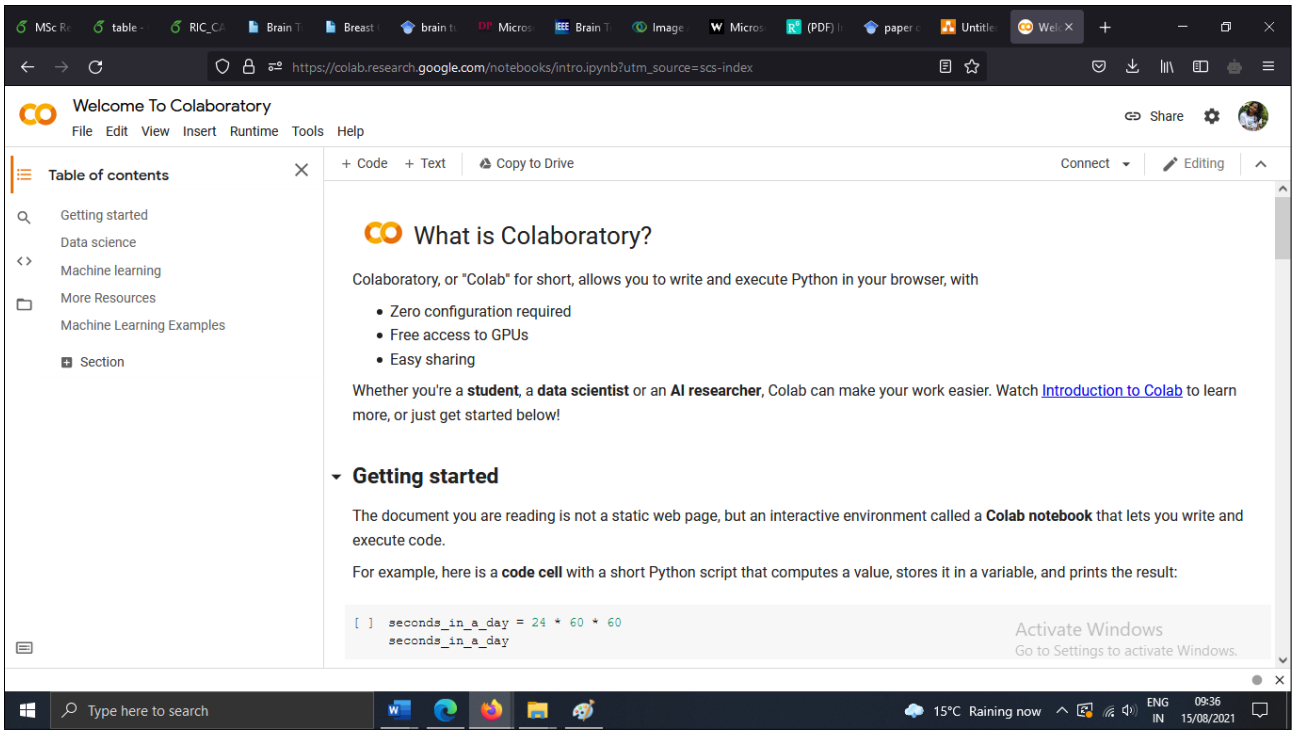
OS	Windows 10
RAM	Minimum 8GB (2.14 GB from Colaboratory)
Hard Disk Space	Minimum 100GB (100 GB drive space)

2.2 Software

Programming Language Tools	Google Colaboratory (Cloud based Jupyter notebook environment), Python version 3, Microsoft Excel, Overleaf
Web Browser	Google Chrome or Mozilla Firefox
Email	Access to a Gmail account

2.2.1 Google colab environment:

CNN and other models consume long time to process the visuals. The experiment is conducted on a Google Collaboratory environment with a 100 GB hard drive, 12.72 GB RAM, and a 48.97 GB run-time GPU. Because all models have additional layers, they take longer to execute on big picture data. The use of a GPU in real time speeds up the execution of CNN and AlexNet. The python libraries Keras and TensorFlow are utilized to implement all three models. On Google Collaboratory Notebook, Python version 3 is used. Google Drive has been used to store the data. Numpy and Keras packages from Python are used for image normalization, argumentation, cropping and up-sampling. 1. Sign in to Google colab with your gmail.



3 Project Development

This project follows a Cross Industry Standard Process for Data Mining (CRISP- DM) Methodology and the details of the research process such as data collection, processing techniques and algorithms are discussed below.

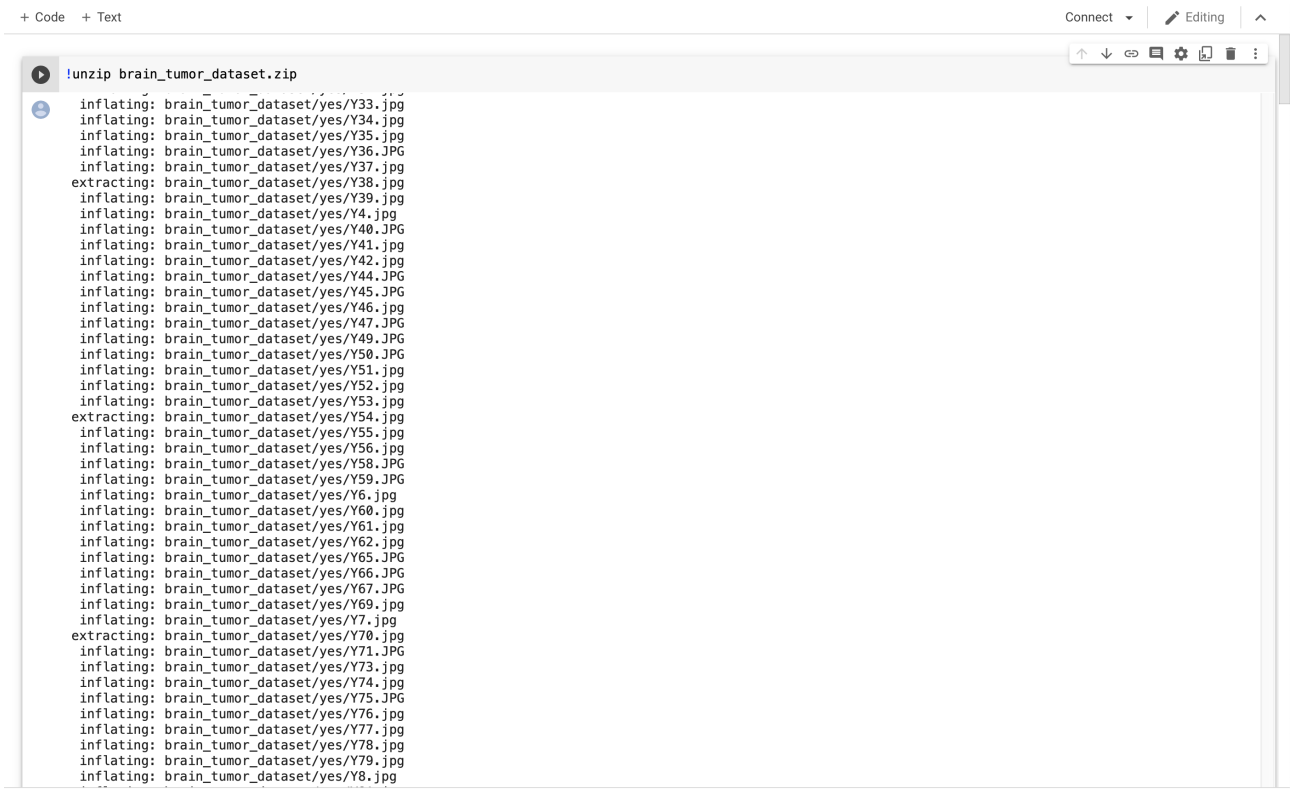
3.1 Google Colaboratory Environment Setup

The Google Colaboratory environment is used for perform the experiments. Google drive access is required for accessing Google Colaboratory and hence a valid Gmail account is essential. Once the URL is followed, a code is visible using which access will be granted to use the Google Drive for mounting the files and data.

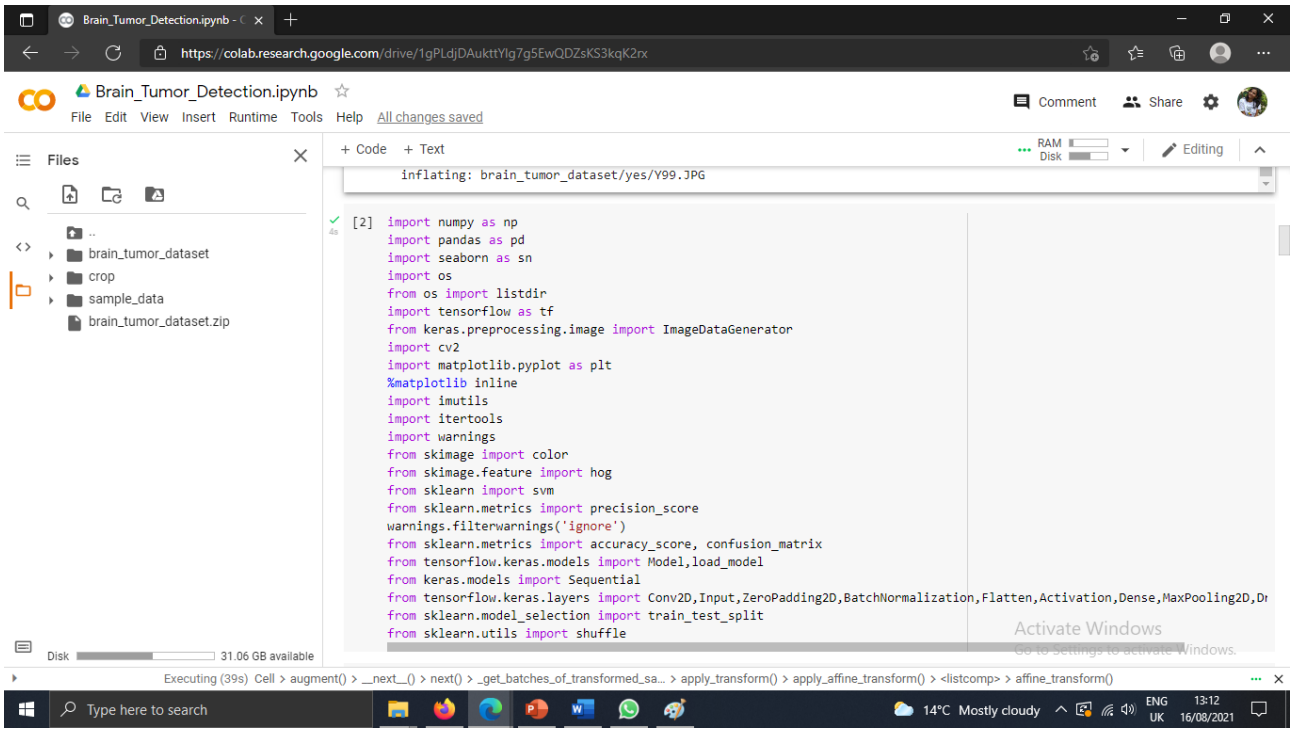
We have used Python 3 notebooks for all Experiments

3.2 Data Preparation

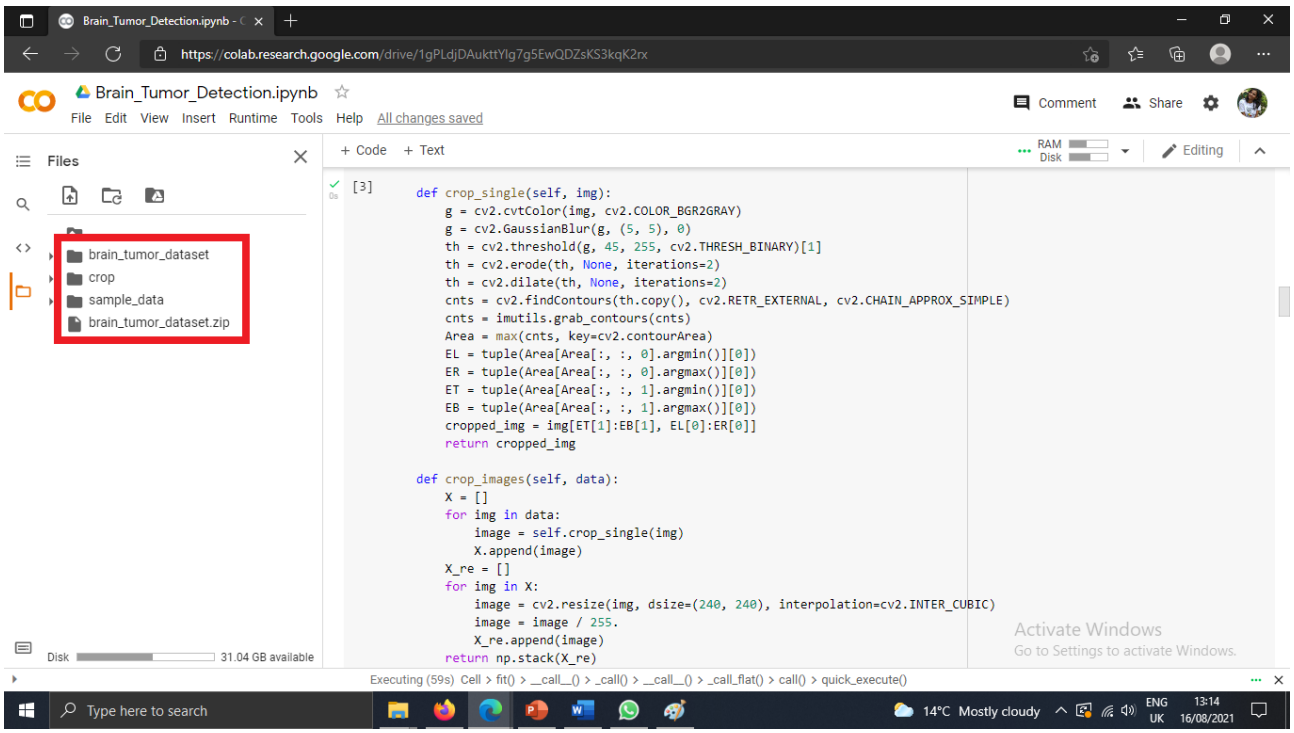
Step 1: Unzipping the dataset



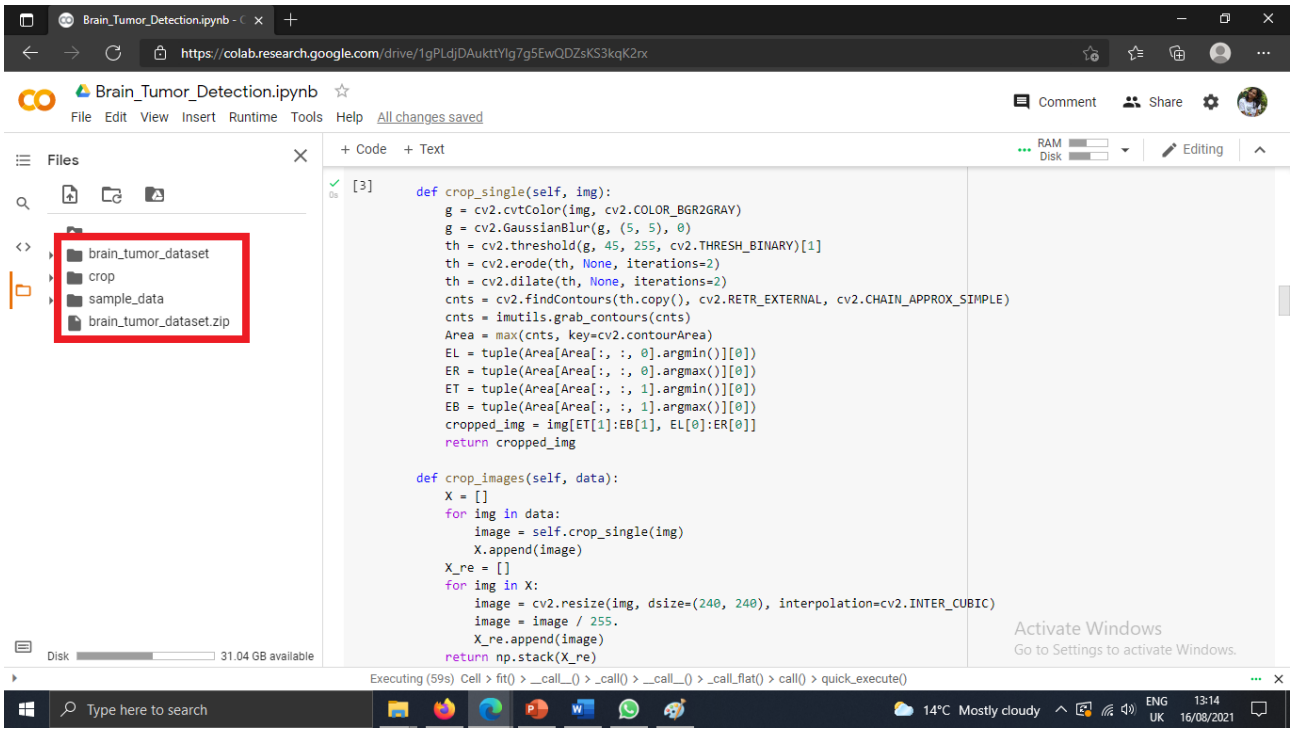
Step 2: Import Libraries



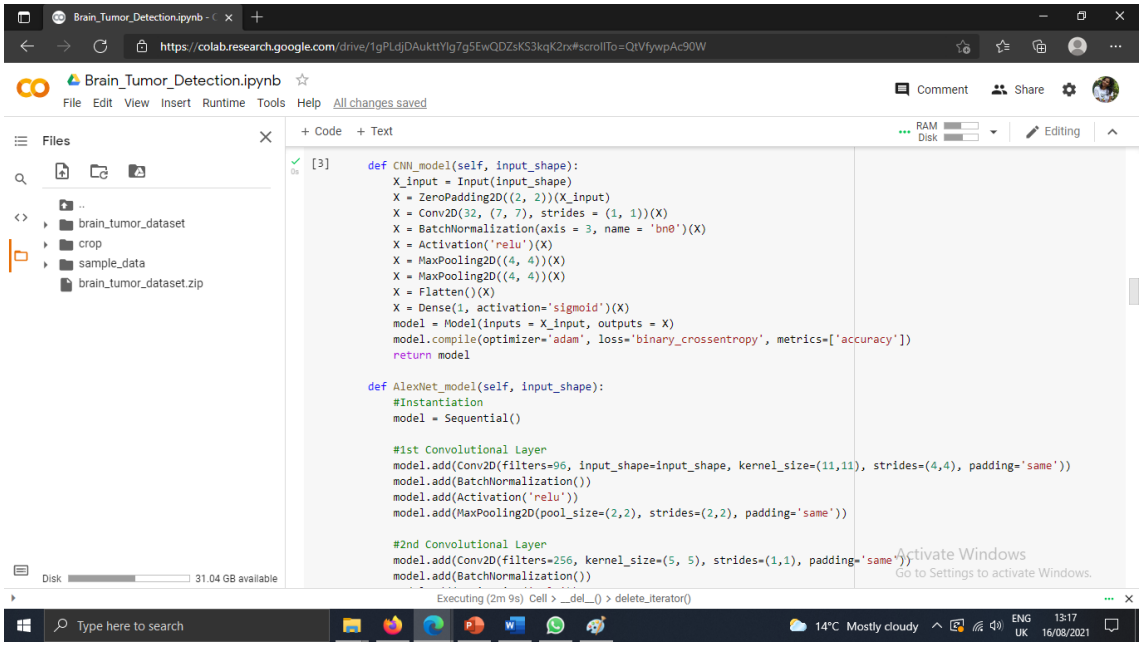
Step 3: Data Augmentation

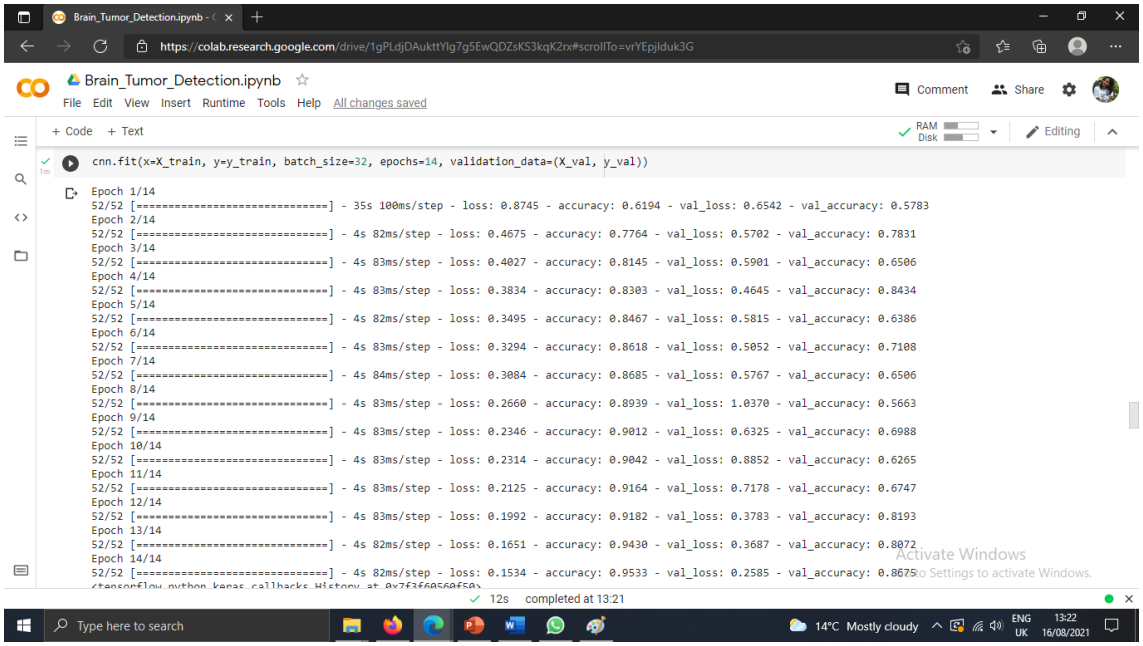


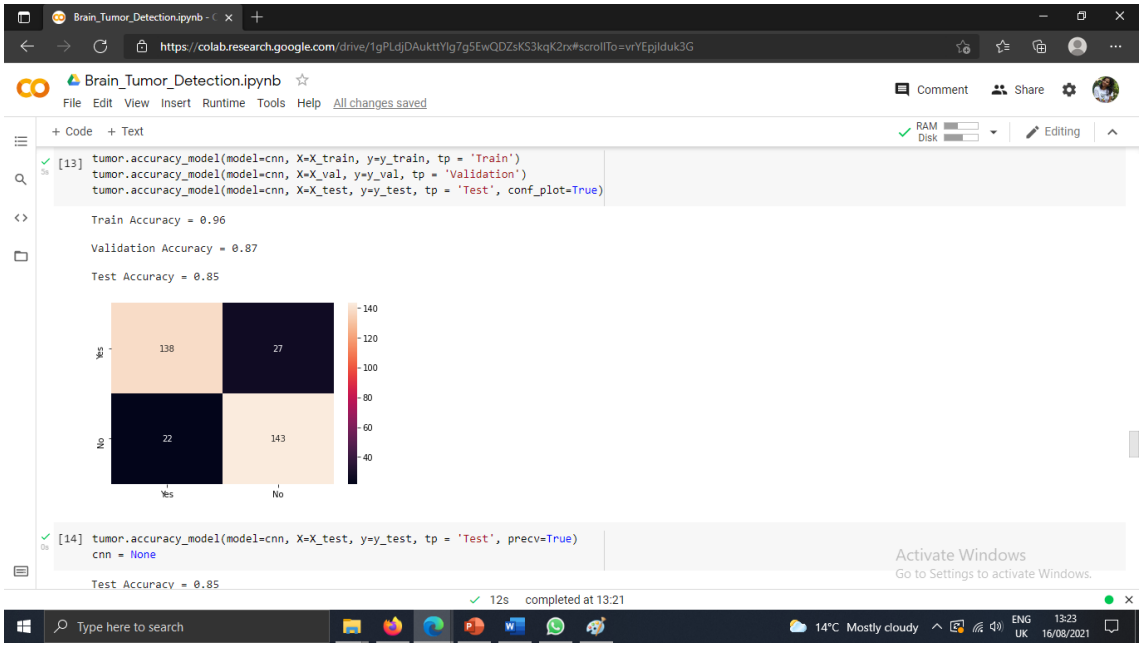
Step 4: Create Crop Folder



Step 5: CNN Model

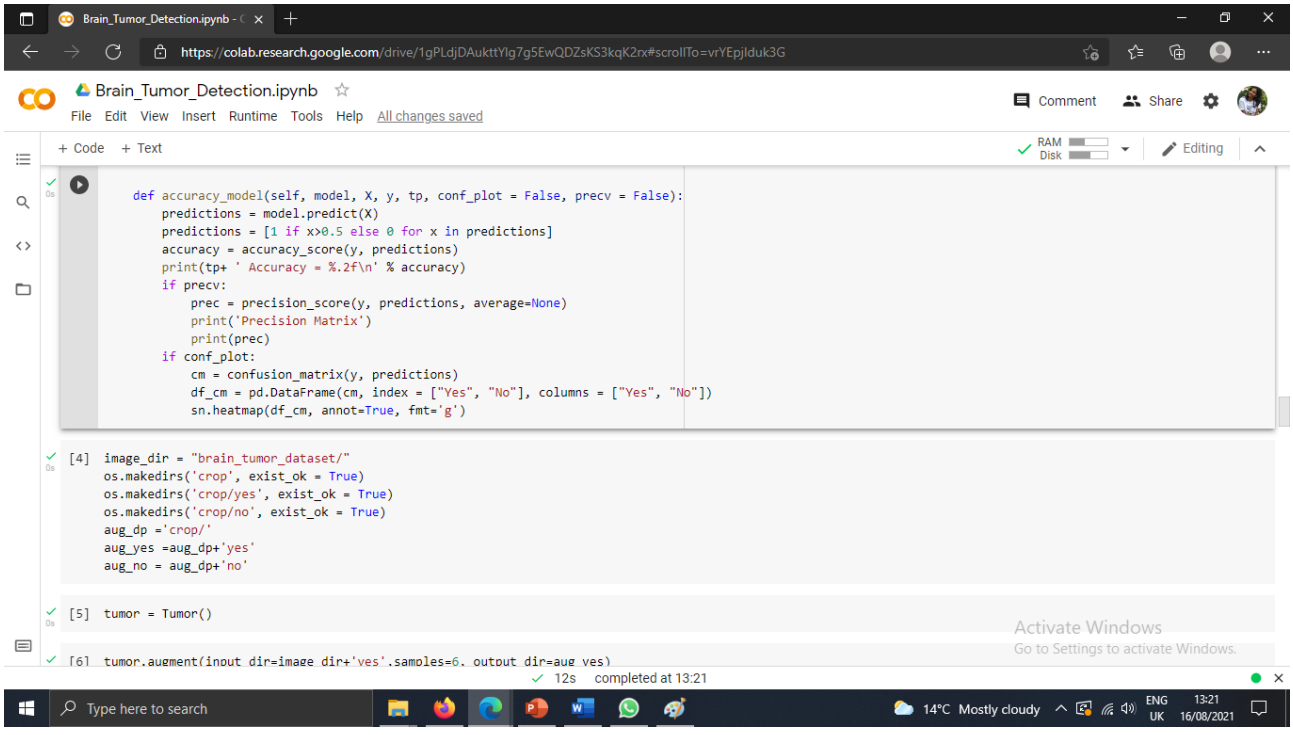


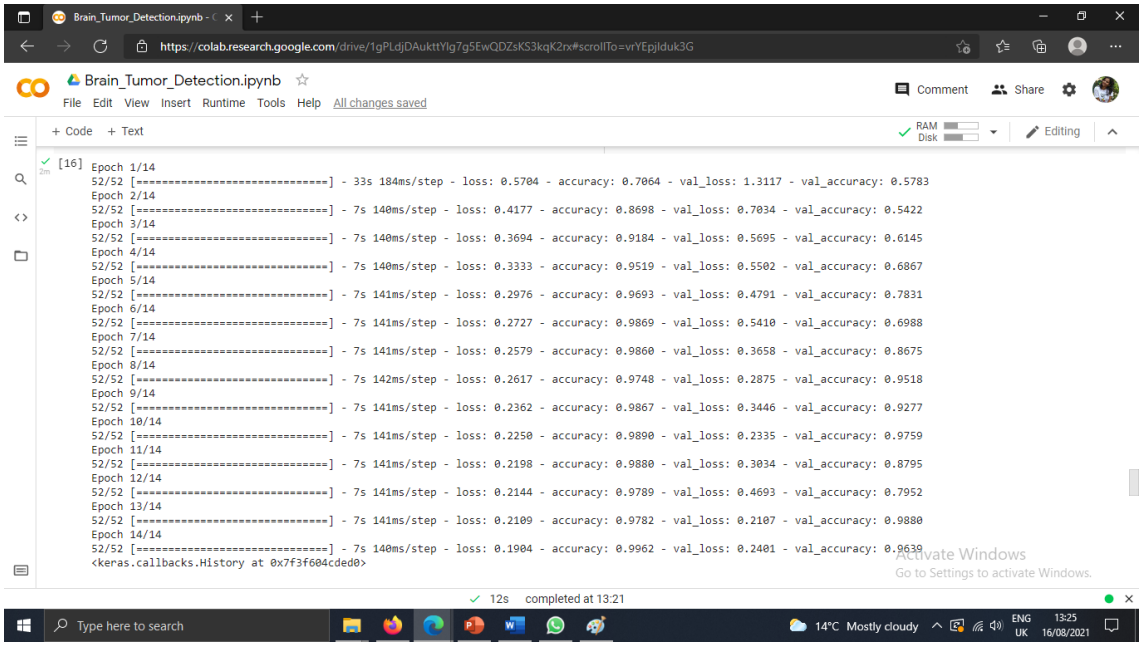




Step 5: AlexNet Model

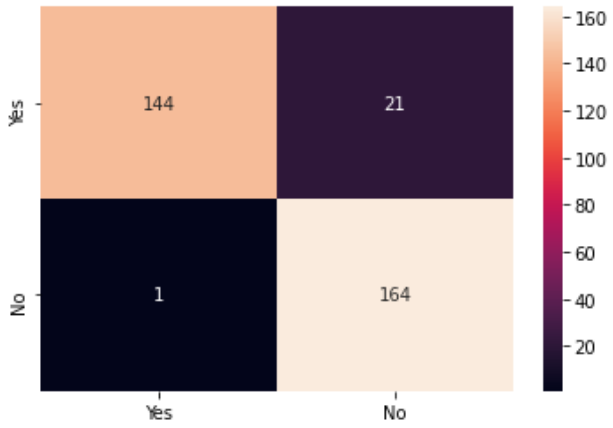
```python
def AlexNet_model(self, input_shape):
    #Instantiation
    model = Sequential()

    #1st Convolutional Layer
    model.add(Conv2D(filters=96, input_shape=input_shape, kernel_size=(11,11), strides=(4,4), padding='same'))
    model.add(BatchNormalization())
    model.add(Activation('relu'))
    model.add(MaxPooling2D(pool_size=(2,2), strides=(2,2), padding='same'))

    #2nd Convolutional Layer
    model.add(Conv2D(filters=256, kernel_size=(5, 5), strides=(1,1), padding='same'))
    model.add(BatchNormalization())
    model.add(Activation('relu'))
    model.add(MaxPooling2D(pool_size=(2,2), strides=(2,2), padding='same'))

    #3rd Convolutional Layer
    model.add(Conv2D(filters=384, kernel_size=(3,3), strides=(1,1), padding='same'))
    model.add(BatchNormalization())
    model.add(Activation('relu'))

    #4th Convolutional Layer
    model.add(Conv2D(filters=384, kernel_size=(3,3), strides=(1,1), padding='same'))
    model.add(BatchNormalization())
```



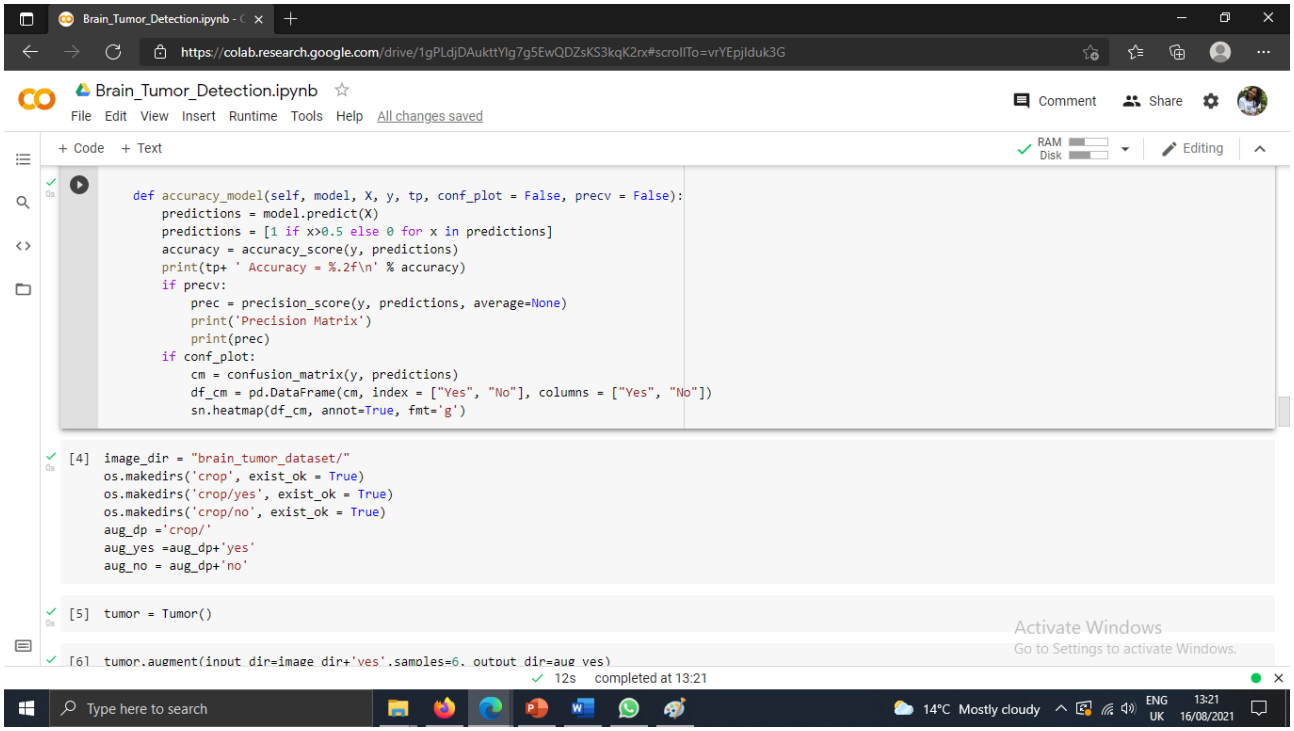



```
tumor.accuracy_model(model=alexnet, X=X_train, y=y_train, tp = 'Train')
tumor.accuracy_model(model=alexnet, X=X_val, y=y_val, tp = 'Validation')
tumor.accuracy_model(model=alexnet, X=X_test, y=y_test, tp = 'Test', conf_plot=True)
```

```
Train Accuracy = 0.97

Validation Accuracy = 0.96

Test Accuracy = 0.93
```



Step 6: Accuracy Model



Step 7: Histogram of Gradient and Support Vector Machine (HOG + SVM)