Configuration Manual

MSc Research Project
Data Analytics

John Maruthukunnel Jacob
Student ID: 21138494

School of Computing
National College of Ireland

Supervisor: Dr Cristina Muntean

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	John Maruthukunnel Jacob
Student ID:	21138494
Programme:	Data Analytics
Year:	2022
Module:	MSc Research Project
Supervisor:	Dr Cristina Muntean
Submission Due Date:	15/12/2022
Project Title:	Configuration Manual
Word Count:	XXX
Page Count:	8

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:	15th December 2022

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

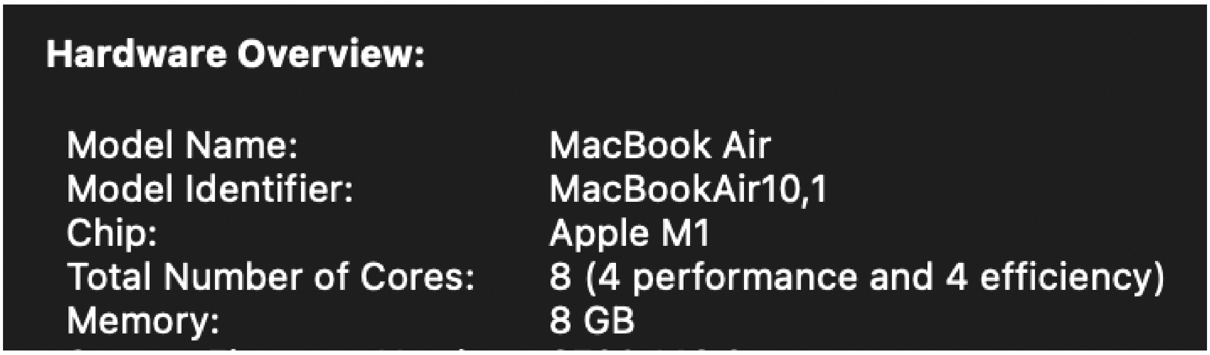
John Maruthukunnel Jacob
21138494

1 Introduction

This configuration manual is created to replicate the research project 'Binary Gender Classification of African Fingerprints using CNN'. In order to reproduce the research, the hardware and software listed in this configuration manual is required. From configuring the execution environment to viewing the model results, the coding procedures required to replicate this study, will be easily done with the aid of this manual. For ease of use, a step-by-step manual is organized into various sections below.

2 Hardware Requirement

The research was done on a MacBook air m1 with 8-core CPU (central process- ing unit) with 4 performance cores and 4 efficiency cores and 8-core integrated GPU (graphics processing unit).

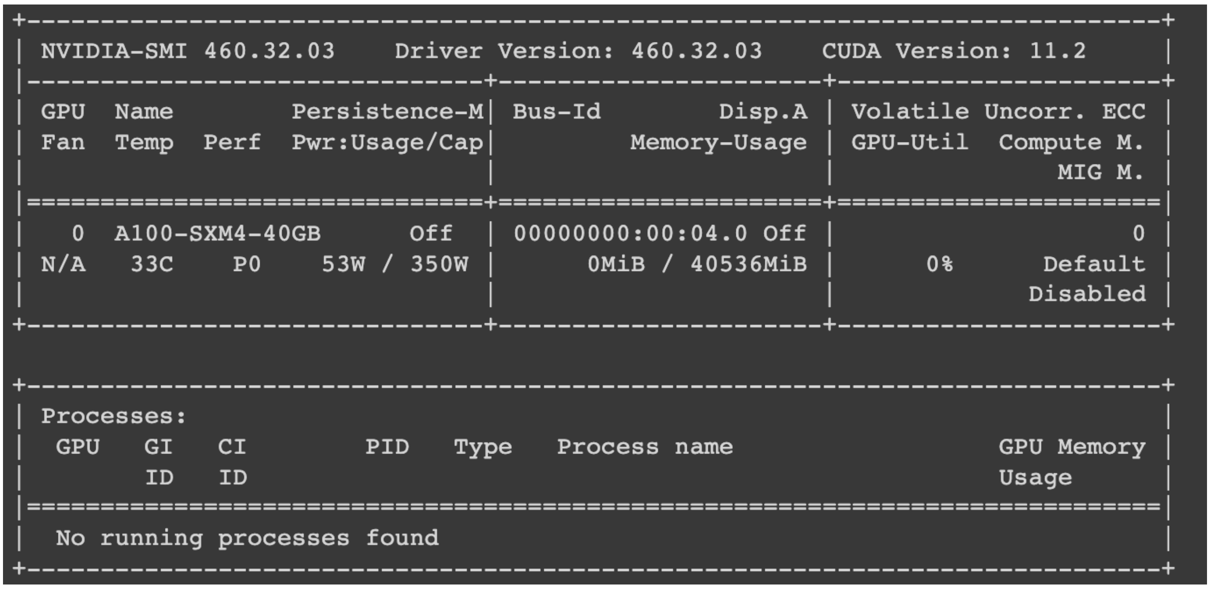


```
Hardware Overview:

  Model Name:                 MacBook Air
  Model Identifier:           MacBookAir10,1
  Chip:                       Apple M1
  Total Number of Cores:      8 (4 performance and 4 efficiency)
  Memory:                     8 GB
```

Figure 1: System Configuration

3 GPU Configuration

The code is implemented using Google Colab Pro. The Colab Pro is a paid subscription with a price of 9.95 euros.The Colab Pro GPU configuration used for this research is shown in Figure 2

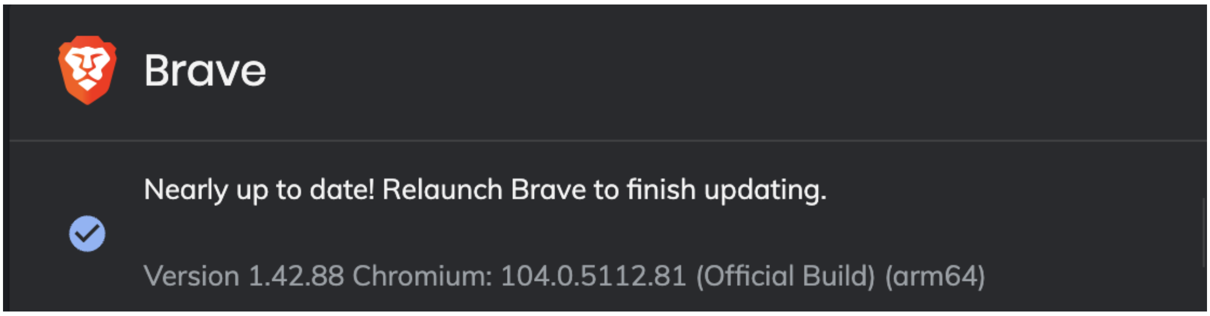


```
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 460.32.03    Driver Version: 460.32.03    CUDA Version: 11.2     |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|                               |                      |               MIG M. |
|===============================+======================+======================|
|   0  A100-SXM4-40GB      Off  | 00000000:00:04.0 Off |                    0 |
| N/A   33C    P0    53W / 350W |      0MiB / 40536MiB |      0%      Default |
|                               |                      |             Disabled |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                                  |
|  GPU   GI   CI        PID   Type   Process name                  GPU Memory |
|        ID   ID                                                   Usage      |
|=============================================================================|
|  No running processes found                                                 |
+-----------------------------------------------------------------------------+
```

Figure 2: GPU Configuration

4 Software Requirement

For the use of Google Colab Pro, Brave web browser was used. The Brave browser version details are shown in Figure 3



Brave

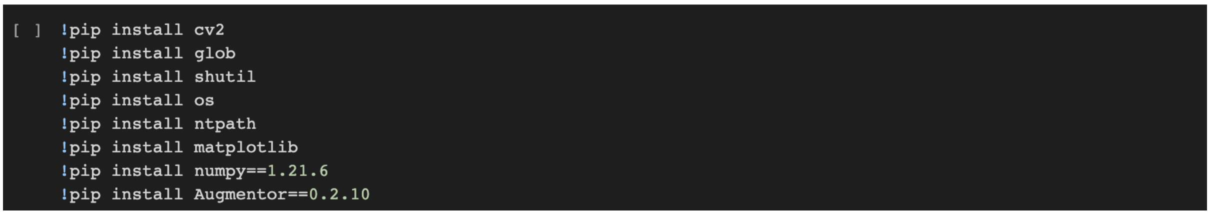
Nearly up to date! Relaunch Brave to finish updating.

Version 1.42.88 Chromium: 104.0.5112.81 (Official Build) (arm64)

Figure 3: Brave browser version

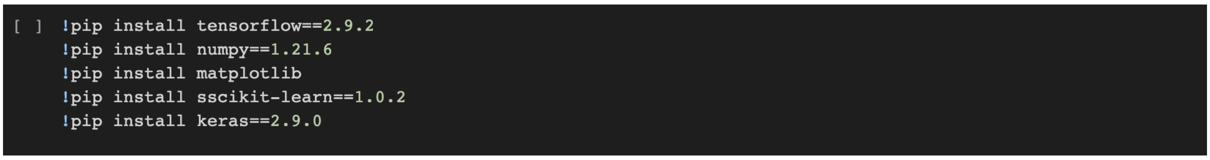
5 Package Installation

The packages necessary for data augmentation,pre-processing and modelling installed using pip command is shown below in 4 and 5.



```
[ ] !pip install cv2
    !pip install glob
    !pip install shutil
    !pip install os
    !pip install ntpath
    !pip install matplotlib
    !pip install numpy==1.21.6
    !pip install Augmentor==0.2.10
```

Figure 4: Data augmentation and pre-proccessing packages



```
[ ] !pip install tensorflow==2.9.2
    !pip install numpy==1.21.6
    !pip install matplotlib
    !pip install sscikit-learn==1.0.2
    !pip install keras==2.9.0
```

Figure 5: Modelling packages

6 Data Collection

The data was sourced from Kaggle data repository [1]. The dataset can be downloaded by pressing on the button as shown in Figure 6.

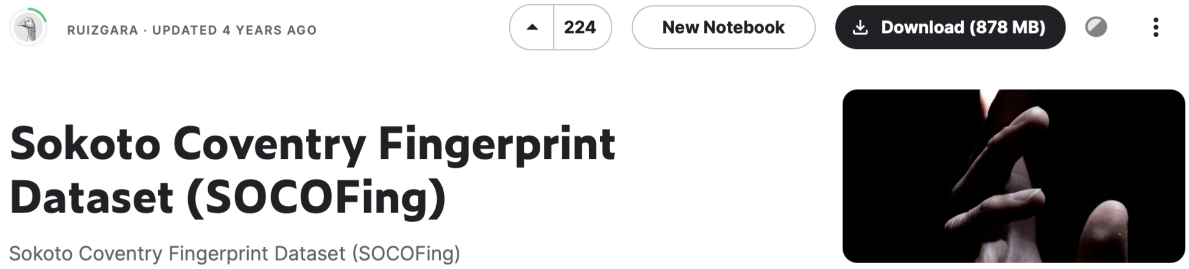
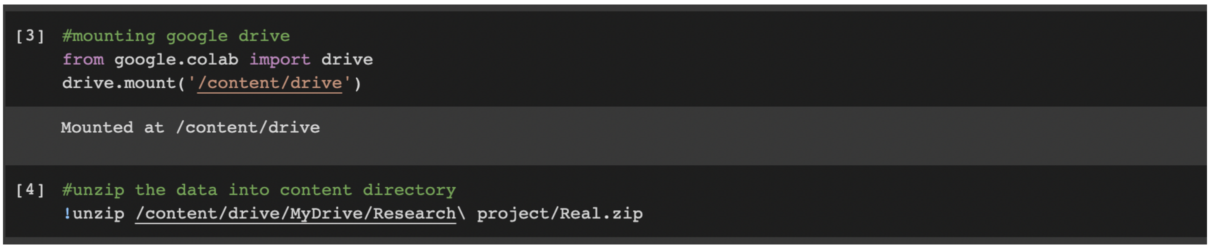

RUIZGARA · UPDATED 4 YEARS AGO
224
New Notebook
Download (878 MB)
Sokoto Coventry Fingerprint Dataset (SOCOFing)
Sokoto Coventry Fingerprint Dataset (SOCOFing)

Figure 6: Kaggle Repository

7 Data Pre-processing

The python file 'Data_pre_process.ipynb' is used for data preparation. From the original dataset available in Kaggle, only the folder named 'Real' is chosen for the research. The folder named 'Real' is zipped and uploaded to google drive under a new folder created named 'Research Project'. The google drive is mounted in Colab notebook and the zip file is unzipped. The data is then converted to jpg format and split into male and female classes.

[1] 'https://www.kaggle.com/datasets/ruizgara/socofing'

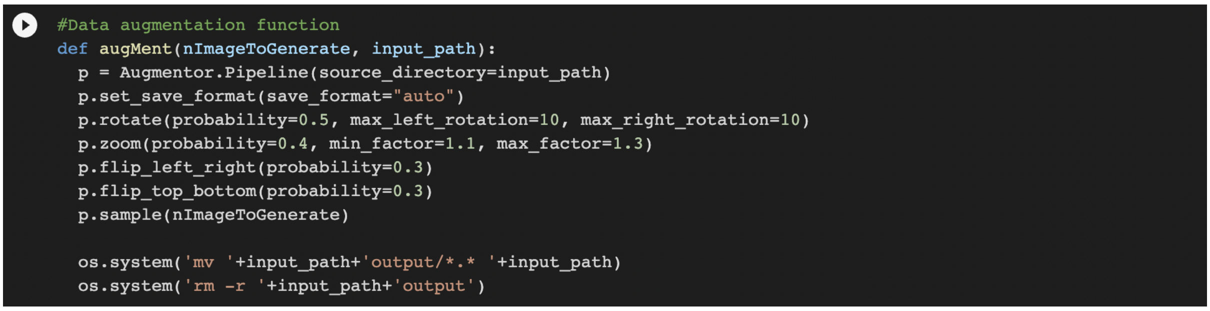


```
[3] #mounting google drive
    from google.colab import drive
    drive.mount('/content/drive')

    Mounted at /content/drive

[4] #unzip the data into content directory
    !unzip /content/drive/MyDrive/Research\ project/Real.zip
```

Figure 7: Mounting drive to Google Colab

8 Data Augmentation

The data augmentation techniques such as rotate, zoom and flip is performed on the dataset using Augmentor function. 2000 images from male class is selected and 1230 images from female class. The female class size is increased to 2000 using Augmentor function.Then the data is split into training and testing data and saved into a folder 'data'. The 'data' folder tree structure is created using code.



```
#Data augmentation function
def augMent(nImageToGenerate, input_path):
  p = Augmentor.Pipeline(source_directory=input_path)
  p.set_save_format(save_format="auto")
  p.rotate(probability=0.5, max_left_rotation=10, max_right_rotation=10)
  p.zoom(probability=0.4, min_factor=1.1, max_factor=1.3)
  p.flip_left_right(probability=0.3)
  p.flip_top_bottom(probability=0.3)
  p.sample(nImageToGenerate)

  os.system('mv '+input_path+'output/*.* '+input_path)
  os.system('rm -r '+input_path+'output')
```

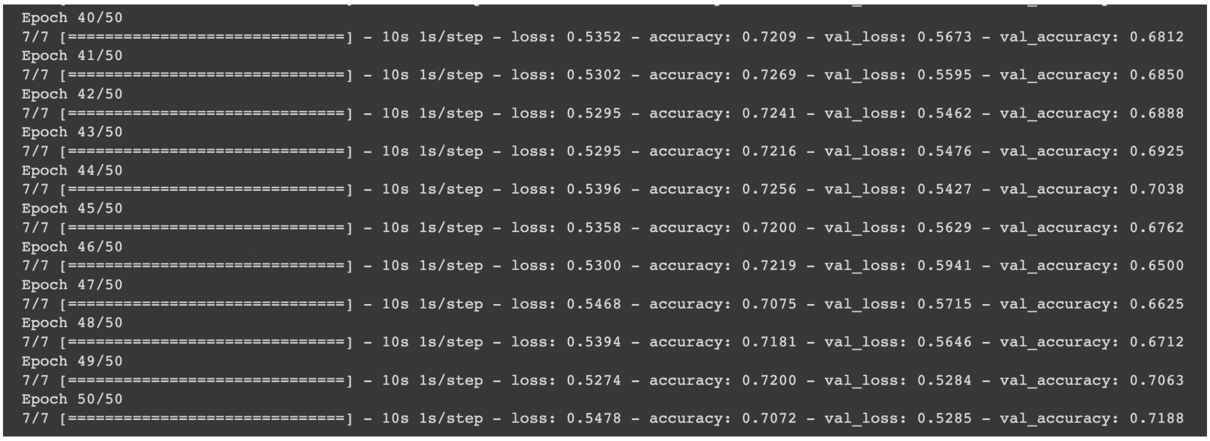
Figure 8: Augmentation function

9 Modelling

The python files 'Resnet50.ipynb', 'vgg19.ipynb', 'vgg16.ipynb', 'inceptionv3.ipynb' is used to implement the models ResNet-50, VGG-19, VGG-16 and InceptionV3 respectively. The final dataset named 'data.zip' is unzipped into Colab notebook.The necessary packages are installed using pip command.The necessary libraries are then imported.The four models implemented in this research are pre-trained using 'imagenet' dataset with the last layer frozen. The parameter values for each model is shown in below sections.

9.1 VGG-19

The final parameters are loss='categorical crossentropy', adam optimizer, dropout value of 0.5, 512 feature selection in dense layer, and sigmoid activation as final layer. Various parameters were modified to obtain the best results. The model consists of two completely linked layers. The training dataset, which includes 3200 images, is used to train the model. Using a fit generator with a batch size of 512 and 50 epochs, the model history is produced. For each training period, the model test and training loss and accuracy are generated. 50 epochs of the model training were completed in 8.4 minutes.

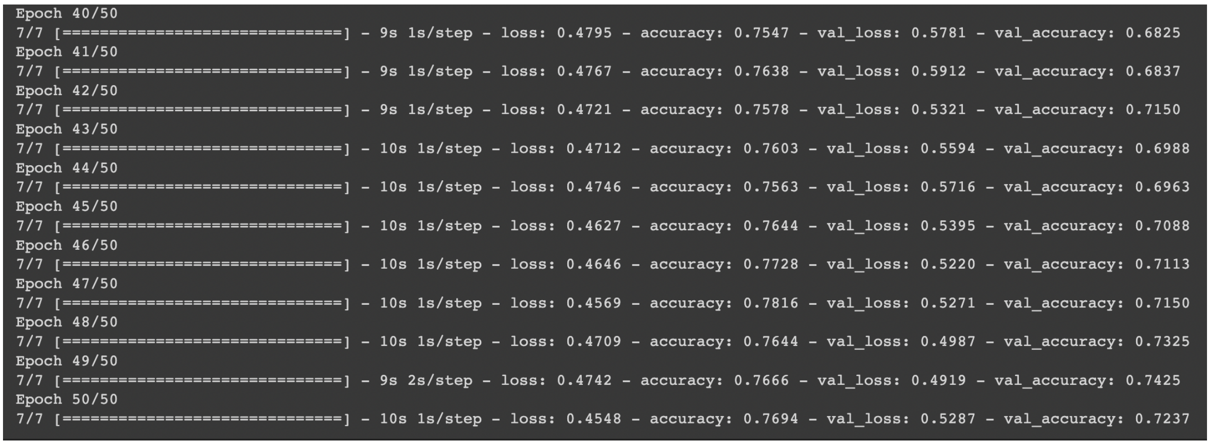


```
Epoch 40/50
7/7 [==============================] - 10s 1s/step - loss: 0.5352 - accuracy: 0.7209 - val_loss: 0.5673 - val_accuracy: 0.6812
Epoch 41/50
7/7 [==============================] - 10s 1s/step - loss: 0.5302 - accuracy: 0.7269 - val_loss: 0.5595 - val_accuracy: 0.6850
Epoch 42/50
7/7 [==============================] - 10s 1s/step - loss: 0.5295 - accuracy: 0.7241 - val_loss: 0.5462 - val_accuracy: 0.6888
Epoch 43/50
7/7 [==============================] - 10s 1s/step - loss: 0.5295 - accuracy: 0.7216 - val_loss: 0.5476 - val_accuracy: 0.6925
Epoch 44/50
7/7 [==============================] - 10s 1s/step - loss: 0.5396 - accuracy: 0.7256 - val_loss: 0.5427 - val_accuracy: 0.7038
Epoch 45/50
7/7 [==============================] - 10s 1s/step - loss: 0.5358 - accuracy: 0.7200 - val_loss: 0.5629 - val_accuracy: 0.6762
Epoch 46/50
7/7 [==============================] - 10s 1s/step - loss: 0.5300 - accuracy: 0.7219 - val_loss: 0.5941 - val_accuracy: 0.6500
Epoch 47/50
7/7 [==============================] - 10s 1s/step - loss: 0.5468 - accuracy: 0.7075 - val_loss: 0.5715 - val_accuracy: 0.6625
Epoch 48/50
7/7 [==============================] - 10s 1s/step - loss: 0.5394 - accuracy: 0.7181 - val_loss: 0.5646 - val_accuracy: 0.6712
Epoch 49/50
7/7 [==============================] - 10s 1s/step - loss: 0.5274 - accuracy: 0.7200 - val_loss: 0.5284 - val_accuracy: 0.7063
Epoch 50/50
7/7 [==============================] - 10s 1s/step - loss: 0.5478 - accuracy: 0.7072 - val_loss: 0.5285 - val_accuracy: 0.7188
```

Figure 9: VGG-19 training

9.2 VGG-16

The final parameters are loss='categorical crossentropy', adam optimizer, dropout value of 0.2, 512 feature selection in dense layer, and sigmoid activation as final layer. Various parameters were modified to obtain the best results. The model consists of two completely linked layers. The training dataset, which includes 3200 images, is used to train the model. Using a fit generator with a batch size of 512 and 50 epochs, the model history is produced. For each training period, the model test and training loss and accuracy are generated. 50 epochs of the model training were completed in 8.6 minutes.

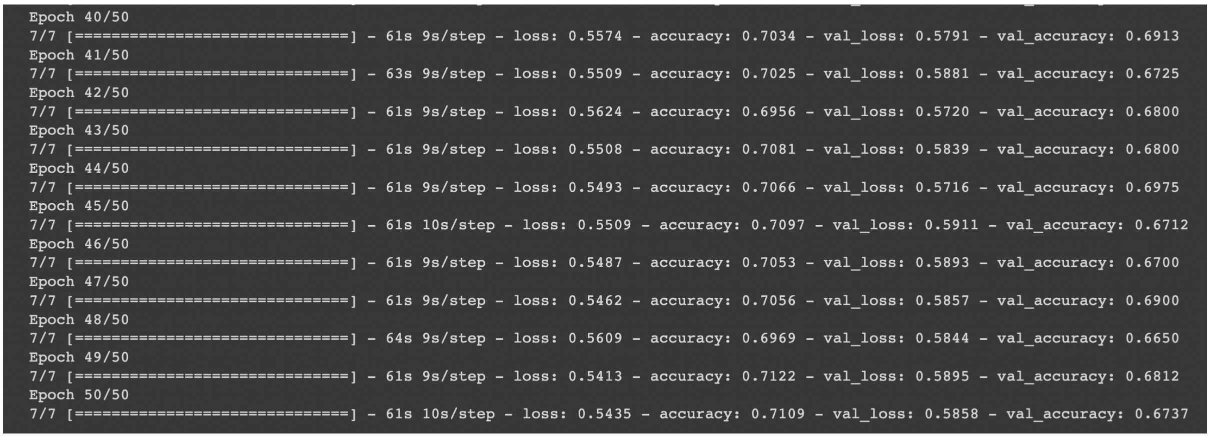


```
Epoch 40/50
7/7 [==============================] - 9s 1s/step - loss: 0.4795 - accuracy: 0.7547 - val_loss: 0.5781 - val_accuracy: 0.6825
Epoch 41/50
7/7 [==============================] - 9s 1s/step - loss: 0.4767 - accuracy: 0.7638 - val_loss: 0.5912 - val_accuracy: 0.6837
Epoch 42/50
7/7 [==============================] - 9s 1s/step - loss: 0.4721 - accuracy: 0.7578 - val_loss: 0.5321 - val_accuracy: 0.7150
Epoch 43/50
7/7 [==============================] - 10s 1s/step - loss: 0.4712 - accuracy: 0.7603 - val_loss: 0.5594 - val_accuracy: 0.6988
Epoch 44/50
7/7 [==============================] - 10s 1s/step - loss: 0.4746 - accuracy: 0.7563 - val_loss: 0.5716 - val_accuracy: 0.6963
Epoch 45/50
7/7 [==============================] - 10s 1s/step - loss: 0.4627 - accuracy: 0.7644 - val_loss: 0.5395 - val_accuracy: 0.7088
Epoch 46/50
7/7 [==============================] - 10s 1s/step - loss: 0.4646 - accuracy: 0.7728 - val_loss: 0.5220 - val_accuracy: 0.7113
Epoch 47/50
7/7 [==============================] - 10s 1s/step - loss: 0.4569 - accuracy: 0.7816 - val_loss: 0.5271 - val_accuracy: 0.7150
Epoch 48/50
7/7 [==============================] - 10s 1s/step - loss: 0.4709 - accuracy: 0.7644 - val_loss: 0.4987 - val_accuracy: 0.7325
Epoch 49/50
7/7 [==============================] - 9s 2s/step - loss: 0.4742 - accuracy: 0.7666 - val_loss: 0.4919 - val_accuracy: 0.7425
Epoch 50/50
7/7 [==============================] - 10s 1s/step - loss: 0.4548 - accuracy: 0.7694 - val_loss: 0.5287 - val_accuracy: 0.7237
```

Figure 10: VGG-16 training

9.3 InceptionV3

The final parameters are loss='categorical crossentropy', adam optimizer, dropout value of 0.2, 512 feature selection in dense layer, and sigmoid activation as final layer. Various parameters were modified to obtain the best results. The model consists of two completely linked layers. The training dataset, which includes 3200 images, is used to train the model. Using a fit generator with a batch size of 512 and 50 epochs, the model history is produced. For each training period, the model test and training loss and accuracy are generated. 50 epochs of the model training were completed in 50 minutes.

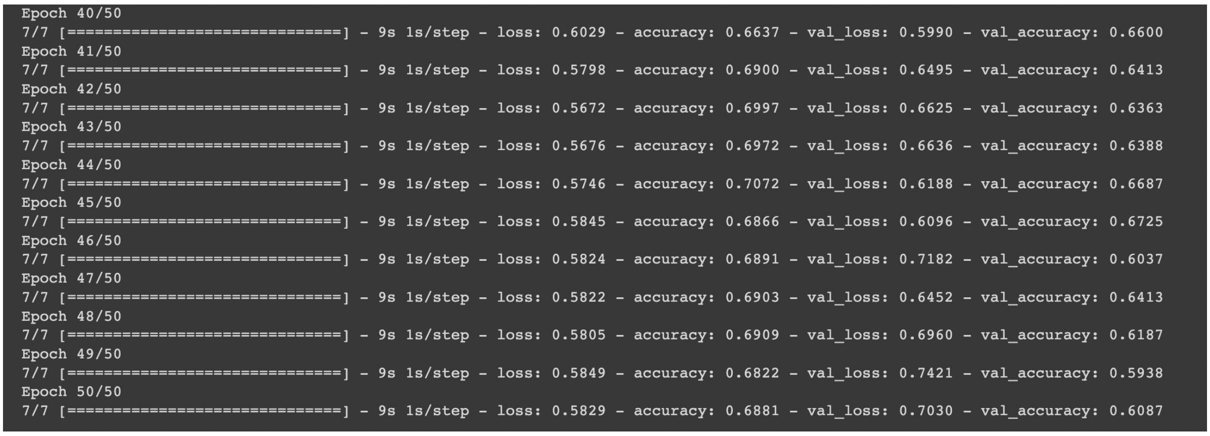


```
Epoch 40/50
7/7 [==============================] - 61s 9s/step - loss: 0.5574 - accuracy: 0.7034 - val_loss: 0.5791 - val_accuracy: 0.6913
Epoch 41/50
7/7 [==============================] - 63s 9s/step - loss: 0.5509 - accuracy: 0.7025 - val_loss: 0.5881 - val_accuracy: 0.6725
Epoch 42/50
7/7 [==============================] - 61s 9s/step - loss: 0.5624 - accuracy: 0.6956 - val_loss: 0.5720 - val_accuracy: 0.6800
Epoch 43/50
7/7 [==============================] - 61s 9s/step - loss: 0.5508 - accuracy: 0.7081 - val_loss: 0.5839 - val_accuracy: 0.6800
Epoch 44/50
7/7 [==============================] - 61s 9s/step - loss: 0.5493 - accuracy: 0.7066 - val_loss: 0.5716 - val_accuracy: 0.6975
Epoch 45/50
7/7 [==============================] - 61s 10s/step - loss: 0.5509 - accuracy: 0.7097 - val_loss: 0.5911 - val_accuracy: 0.6712
Epoch 46/50
7/7 [==============================] - 61s 9s/step - loss: 0.5487 - accuracy: 0.7053 - val_loss: 0.5893 - val_accuracy: 0.6700
Epoch 47/50
7/7 [==============================] - 61s 9s/step - loss: 0.5462 - accuracy: 0.7056 - val_loss: 0.5857 - val_accuracy: 0.6900
Epoch 48/50
7/7 [==============================] - 64s 9s/step - loss: 0.5609 - accuracy: 0.6969 - val_loss: 0.5844 - val_accuracy: 0.6650
Epoch 49/50
7/7 [==============================] - 61s 9s/step - loss: 0.5413 - accuracy: 0.7122 - val_loss: 0.5895 - val_accuracy: 0.6812
Epoch 50/50
7/7 [==============================] - 61s 10s/step - loss: 0.5435 - accuracy: 0.7109 - val_loss: 0.5858 - val_accuracy: 0.6737
```

Figure 11: InceptionV3 training

9.4 ResNet-50

The final parameters are loss='categorical crossentropy', adam optimizer, dropout value of 0.2, 512 feature selection in dense layer, and sigmoid activation as final layer. Various parameters were modified to obtain the best results. The model consists of two completely linked layers. The training dataset, which includes 3200 images, is used to train the model. Using a fit generator with a batch size of 512 and 50 epochs, the model history is produced. For each training period, the model test and training loss and accuracy are generated. 50 epochs of the model training were completed in 7.7 minutes.

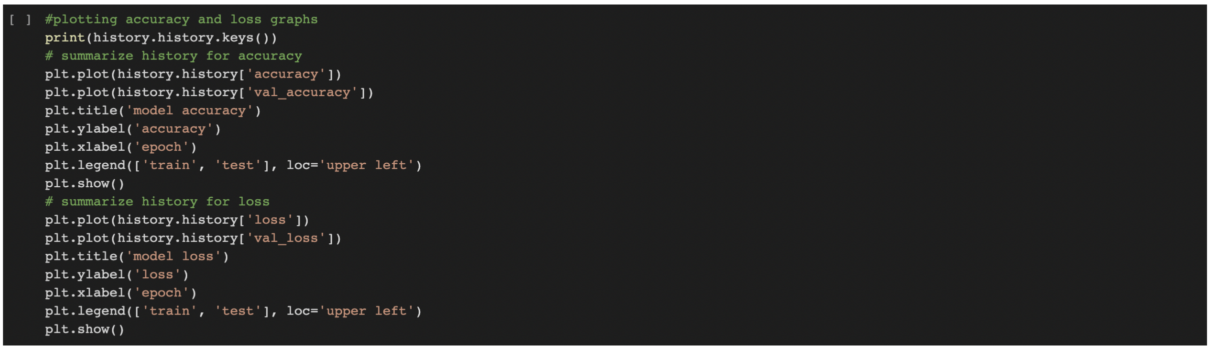


```
Epoch 40/50
7/7 [==============================] - 9s 1s/step - loss: 0.6029 - accuracy: 0.6637 - val_loss: 0.5990 - val_accuracy: 0.6600
Epoch 41/50
7/7 [==============================] - 9s 1s/step - loss: 0.5798 - accuracy: 0.6900 - val_loss: 0.6495 - val_accuracy: 0.6413
Epoch 42/50
7/7 [==============================] - 9s 1s/step - loss: 0.5672 - accuracy: 0.6997 - val_loss: 0.6625 - val_accuracy: 0.6363
Epoch 43/50
7/7 [==============================] - 9s 1s/step - loss: 0.5676 - accuracy: 0.6972 - val_loss: 0.6636 - val_accuracy: 0.6388
Epoch 44/50
7/7 [==============================] - 9s 1s/step - loss: 0.5746 - accuracy: 0.7072 - val_loss: 0.6188 - val_accuracy: 0.6687
Epoch 45/50
7/7 [==============================] - 9s 1s/step - loss: 0.5845 - accuracy: 0.6866 - val_loss: 0.6096 - val_accuracy: 0.6725
Epoch 46/50
7/7 [==============================] - 9s 1s/step - loss: 0.5824 - accuracy: 0.6891 - val_loss: 0.7182 - val_accuracy: 0.6037
Epoch 47/50
7/7 [==============================] - 9s 1s/step - loss: 0.5822 - accuracy: 0.6903 - val_loss: 0.6452 - val_accuracy: 0.6413
Epoch 48/50
7/7 [==============================] - 9s 1s/step - loss: 0.5805 - accuracy: 0.6909 - val_loss: 0.6960 - val_accuracy: 0.6187
Epoch 49/50
7/7 [==============================] - 9s 1s/step - loss: 0.5849 - accuracy: 0.6822 - val_loss: 0.7421 - val_accuracy: 0.5938
Epoch 50/50
7/7 [==============================] - 9s 1s/step - loss: 0.5829 - accuracy: 0.6881 - val_loss: 0.7030 - val_accuracy: 0.6087
```

Figure 12: ResNet-50 training

10 Evaluation

The model evaluation was done using testing accuracy and loss. The training and testing accuracy is plotted along with the loss graphs. The VGG-19 and VGG-16 models obtained the best accuracy of 72% followed by InceptionV3 (67%) and, ResNet-50(60%). The graphs are plotted using the matplotlib as shown in 13



```
#plotting accuracy and loss graphs
print(history.history.keys())
# summarize history for accuracy
plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy'])
plt.title('model accuracy')
plt.ylabel('accuracy')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper left')
plt.show()
# summarize history for loss
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('model loss')
plt.ylabel('loss')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper left')
plt.show()
```

Figure 13: Barplot code

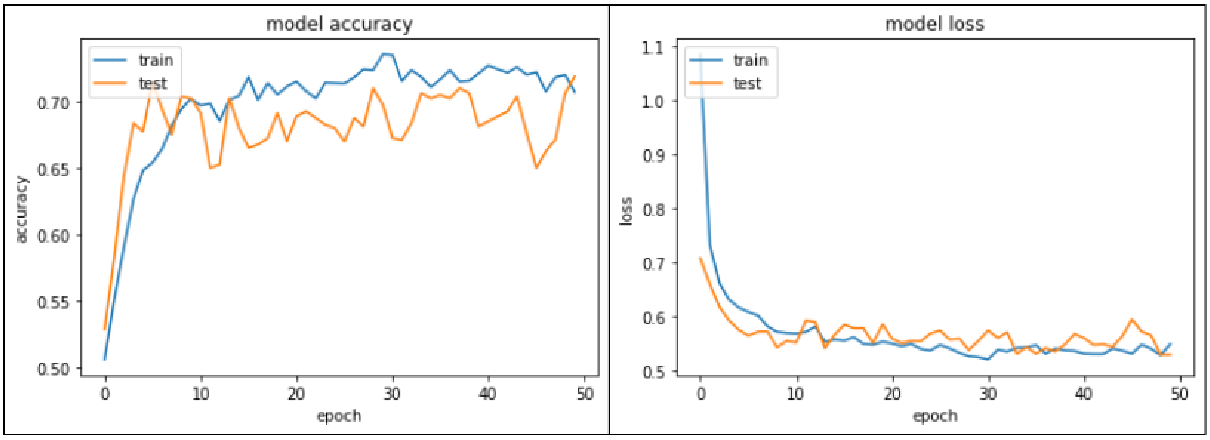


Figure 14: VGG-19 accuracy and loss graph

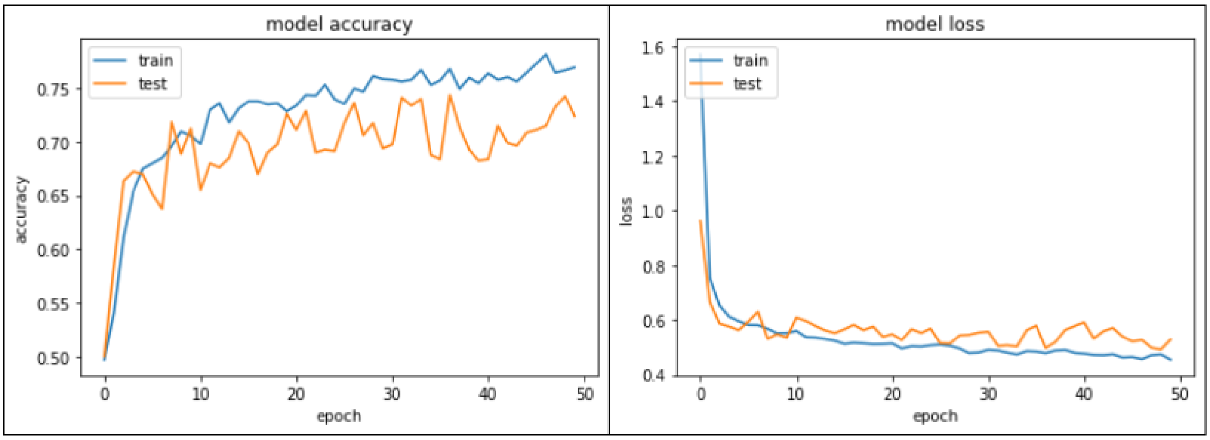


Figure 15: VGG-16 accuracy and loss graph

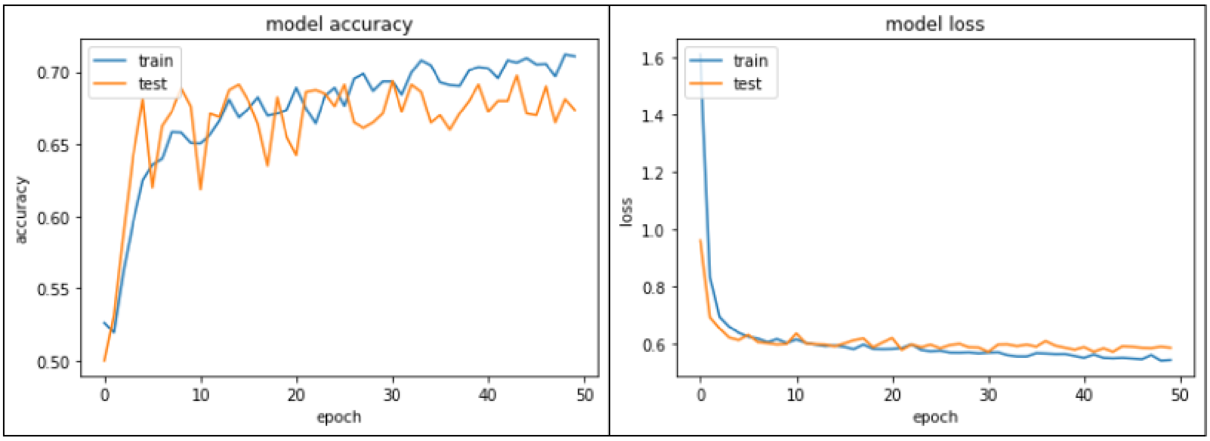


Figure 16: InceptionV3 accuracy and loss graph

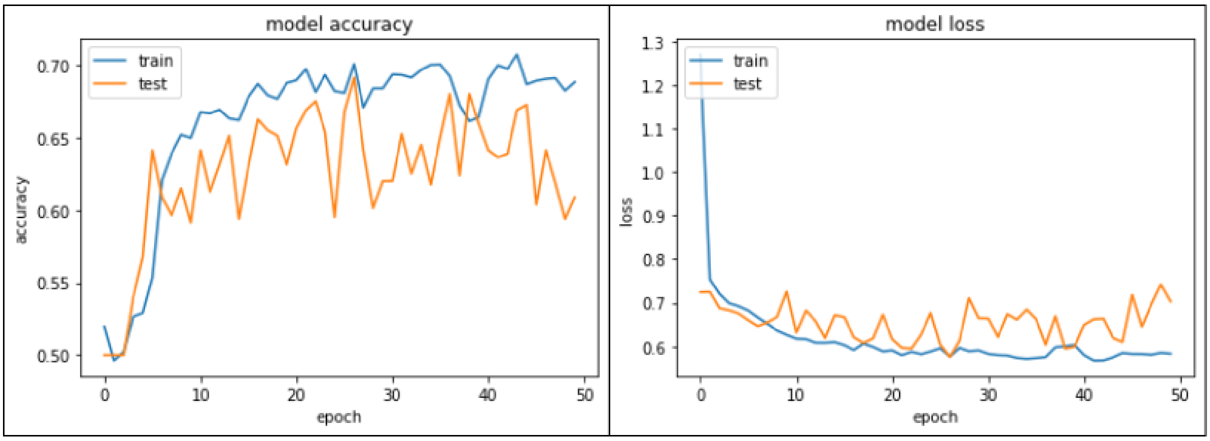


Figure 17: ResNet-50 accuracy and loss graph