Identification of Acute Lymphocytic Leukemia (Blood Cancer) through microscopic images of blood samples - Configuration Manual

Research In Computing
Msc Data Analytics October 2020/21

Jignesh Anil Waghela
Student ID: x19202024

School of Computing
National College of Ireland

Supervisor: Bharathi Chakravarthi

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Jignesh Anil Waghela
Student ID:	x19202024
Programme:	Msc Data Analytics October 2020/21
Year:	2021
Module:	Research In Computing
Supervisor:	Bharathi Chakravarthi
Submission Due Date:	16/08/2021
Project Title:	Identification of Acute Lymphocytic Leukemia (Blood Cancer) through microscopic images of blood samples - Configuration Manual
Word Count:	462
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:	
Date:	16th August 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):	

Identification of Acute Lymphocytic Leukemia (Blood Cancer) through microscopic images of blood samples - Configuration Manual

Jignesh Anil Waghela
x19202024

Contents

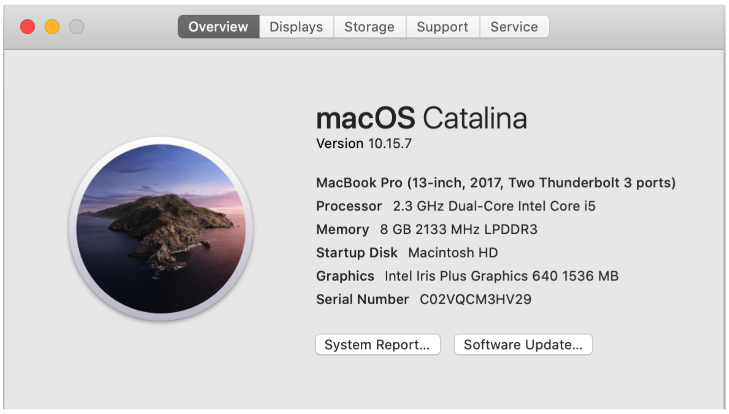
List of Figures

1 Introduction

This configuration handbook explains the requirements for carrying out the experiment. It also serves as a step-by-step guidance for implementing the necessary modifications so that the experiment can be conducted and the desired results can be obtained.

2 Device Specification

- Device Name :- Macbook Pro
- Processor :- 2.3 GHz Dual-Core Intel Core i5
- Memory :- 8 GB RAM 256GB SSD
- System :- macOS Catalina 64 bit
- Graphics :- Intel Iris Plus Graphics 640 1536 MB



macOS Catalina

Version 10.15.7

MacBook Pro (13-inch, 2017, Two Thunderbolt 3 ports)

Processor 2.3 GHz Dual-Core Intel Core i5

Memory 8 GB 2133 MHz LPDDR3

Startup Disk Macintosh HD

Graphics Intel Iris Plus Graphics 640 1536 MB

Serial Number C02VQCM3HV29

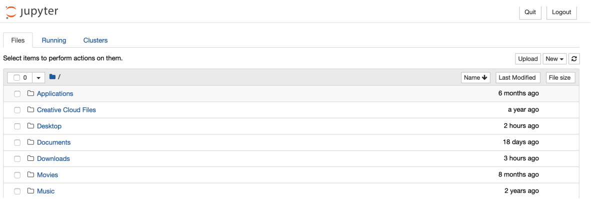
Figure 1: Device Specification

3 Execution Environment in Python

3.1 Setting up Jupyter Notebook

Anaconda Navigator, once installed, enables a Graphical User Interface (GUI) for launching applications. Because of the versatility it offers, Jupyter was chosen.

- For the execution of the work, Python 3 had been used.



Select items to perform actions on them.

Name	Last Modified	File size
Applications	6 months ago	
Creative Cloud Files	a year ago	
Desktop	2 hours ago	
Documents	18 days ago	
Downloads	3 hours ago	
Movies	8 months ago	
Music	2 years ago	

Figure 2: Jupyter Notebook

4 Python packages utilized for the execution

- OS - This package contains a variety of functions for acquiring files and communicating with the operating system [1].

- Glob - This package had been used to locate the names of files within a folder [2].

- OpenCV - This is a Library for python that contains a number of programming tasks. Throughout this research, it was employed for pre-processing processes.

- mlxtend - The confusion matrix was plotted using this term, which represents for machine learning extensions [3].

- Numpy - The Numpy array is a tool for manipulating multidimensional arrays [4].

- Keras - Keras, which itself is built on top of Tensor Flow, is indeed a deep neural networks library that is simple to understand and use. This package had been used the majority in the experiment, for example, to generate image information, import the VGG16 model, and CNN [5].

- Random - To produce a random selection for depiction, this software was utilized. The Random number remains constant if the seed is not updated [6].

- Pandas - Pandas was being used to work with the tags' CSV file. It gives a simple and user-friendly way to interact with structured information [7].

- Matplotlib - This module was used to visualize information [8].

- Sklearn - The classification and regression issues are supported by this library. The confusion matrix and f1 score were generated using this script [9].

- PIL - This package is used to handle and manipulate picture files in various formats [10].

[1] https://www.pythonforbeginners.com/os/pythons-os-module

[2] https://www.oreilly.com/library/view/python-standard-library/0596000960/ch02s15.html

[3] https://pypi.org/project/mlxtend/

[4] https://www.geeksforgeeks.org/numpy-in-python-set-1-introduction/

[5] https://machinelearningmastery.com/tutorial-first-neural-network-python-keras/

[6] https://www.programiz.com/python-programming/modules/random

[7] https://pandas.pydata.org/

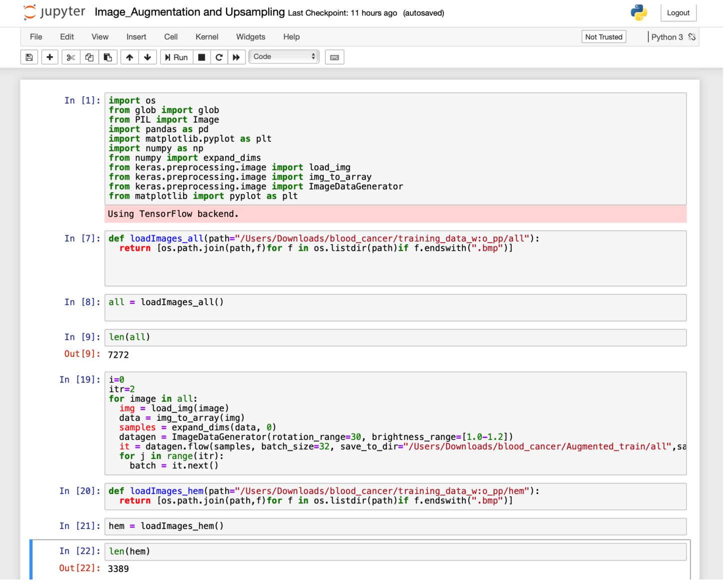
[8] https://en.wikipedia.org/wiki/Matplotlib

[9] https://scikit-learn.org/stable/

[10] https://en.wikipedia.org/wiki/PythonImagingLibrary

5 Code Snippets

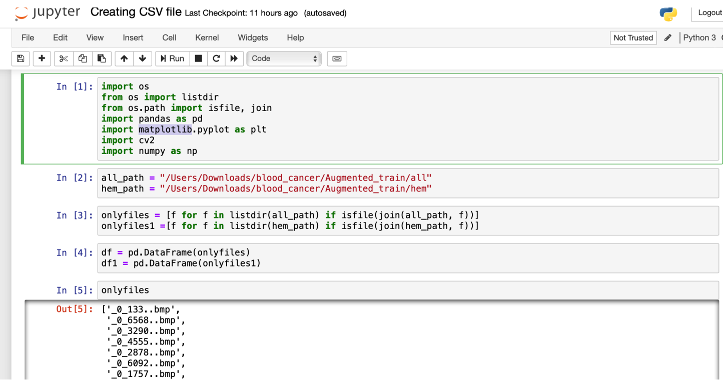
5.1 Data Augmentation and Upsampling


jupyter Image_Augmentation and Upsampling Last Checkpoint: 11 hours ago (autosaved)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Not Trusted
Python 3
Run
Code

```
In [1]: import os
        from glob import glob
        from PIL import Image
        import pandas as pd
        import matplotlib.pyplot as plt
        import numpy as np
        from numpy import expand_dims
        from keras.preprocessing.image import load_img
        from keras.preprocessing.image import img_to_array
        from keras.preprocessing.image import ImageDataGenerator
        from matplotlib import pyplot as plt

        Using TensorFlow backend.

In [7]: def loadImages_all(path="/Users/Downloads/blood_cancer/training_data_w:o_pp/all"):
          return [os.path.join(path,f)for f in os.listdir(path)if f.endswith(".bmp")]

In [8]: all = loadImages_all()

In [9]: len(all)

Out[9]: 7272

In [19]: i=0
         itr=2
         for image in all:
           img = load_img(image)
           data = img_to_array(img)
           samples = expand_dims(data, 0)
           datagen = ImageDataGenerator(rotation_range=30, brightness_range=[1.0-1.2])
           it = datagen.flow(samples, batch_size=32, save_to_dir="/Users/Downloads/blood_cancer/Augmented_train/all",sa
           for j in range(itr):
             batch = it.next()

In [20]: def loadImages_hem(path="/Users/Downloads/blood_cancer/training_data_w:o_pp/hem"):
           return [os.path.join(path,f)for f in os.listdir(path)if f.endswith(".bmp")]

In [21]: hem = loadImages_hem()

In [22]: len(hem)

Out[22]: 3389
```

Figure 3: Data Augmentation and Upsampling

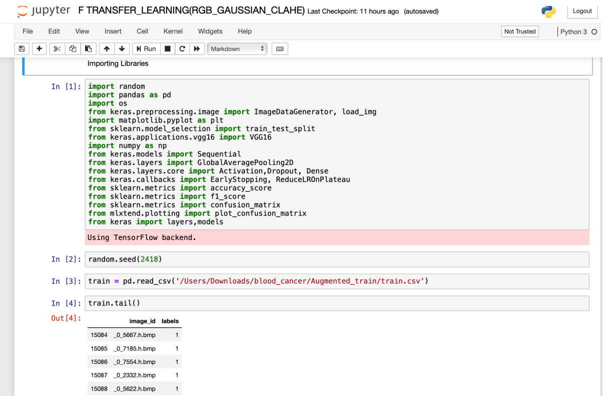
5.2 CSV file Generation



```
In [1]: import os
        from os import listdir
        from os.path import isfile, join
        import pandas as pd
        import matplotlib.pyplot as plt
        import cv2
        import numpy as np

In [2]: all_path = "/Users/Downloads/blood_cancer/Augmented_train/all"
        hem_path = "/Users/Downloads/blood_cancer/Augmented_train/hem"

In [3]: onlyfiles = [f for f in listdir(all_path) if isfile(join(all_path, f))]
        onlyfiles1 =[f for f in listdir(hem_path) if isfile(join(hem_path, f))]

In [4]: df = pd.DataFrame(onlyfiles)
        df1 = pd.DataFrame(onlyfiles1)

In [5]: onlyfiles

Out[5]: ['_0_133..bmp',
         '_0_6568..bmp',
         '_0_3290..bmp',
         '_0_4555..bmp',
         '_0_2878..bmp',
         '_0_6092..bmp',
         '_0_1757..bmp',
```

Figure 4: CSV file generation

5.3 Modelling

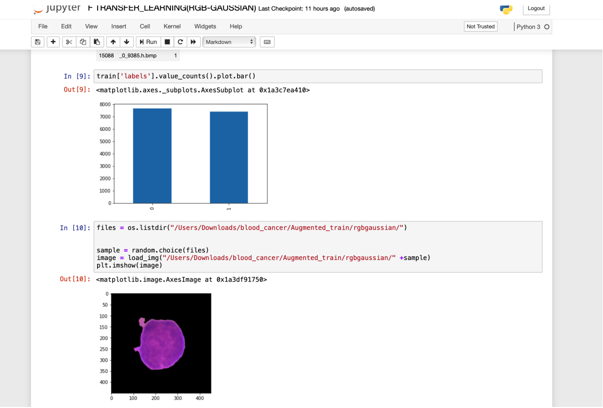

jupyter F TRANSFER_LEARNING(RGB_GAUSSIAN_CLAHE) Last Checkpoint: 11 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3
Run
Markdown

Importing Libraries

```
In [1]: import random
        import pandas as pd
        import os
        from keras.preprocessing.image import ImageDataGenerator, load_img
        import matplotlib.pyplot as plt
        from sklearn.model_selection import train_test_split
        from keras.applications.vgg16 import VGG16
        import numpy as np
        from keras.models import Sequential
        from keras.layers import GlobalAveragePooling2D
        from keras.layers.core import Activation,Dropout, Dense
        from keras.callbacks import EarlyStopping, ReduceLROnPlateau
        from sklearn.metrics import accuracy_score
        from sklearn.metrics import f1_score
        from sklearn.metrics import confusion_matrix
        from mlxtend.plotting import plot_confusion_matrix
        from keras import layers,models

        Using TensorFlow backend.

In [2]: random.seed(2418)

In [3]: train = pd.read_csv('/Users/Downloads/blood_cancer/Augmented_train/train.csv')

In [4]: train.tail()

Out[4]:
```

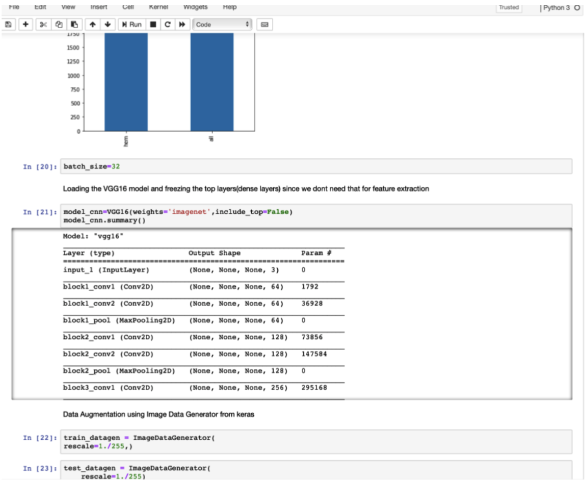
	image_id	labels
15084	_0_5667.h.bmp	1
15085	_0_7185.h.bmp	1
15086	_0_7554.h.bmp	1
15087	_0_2332.h.bmp	1
15088	_0_5622.h.bmp	1

Figure 5: Importing important libraries


jupyter F TRANSFER_LEARNING(RGB-GAUSSIAN) Last Checkpoint: 11 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3
Run
Markdown
8000
7000
6000
5000
4000
3000
2000
1000
0
0
1
0
50
100
150
200
250
300
350
400
0
100
200
300
400

```
15088  _0_9385.h.bmp  1

In [9]: train['labels'].value_counts().plot.bar()

Out[9]: <matplotlib.axes._subplots.AxesSubplot at 0x1a3c7ea410>

In [10]: files = os.listdir("/Users/Downloads/blood_cancer/Augmented_train/rgbgaussian/")


         sample = random.choice(files)
         image = load_img("/Users/Downloads/blood_cancer/Augmented_train/rgbgaussian/" +sample)
         plt.imshow(image)

Out[10]: <matplotlib.image.AxesImage at 0x1a3df91750>
```

Figure 6: Visualization



```
In [20]: batch_size=32
```

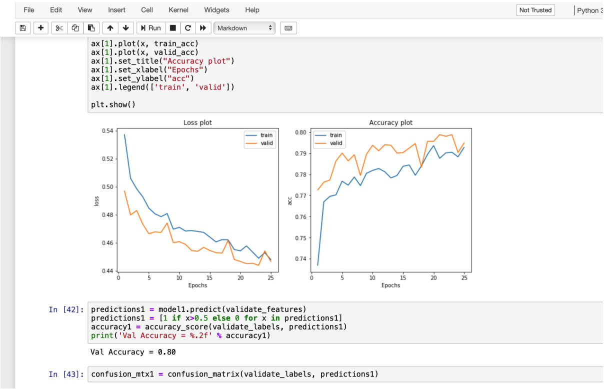
Loading the VGG16 model and freezing the top layers(dense layers) since we dont need that for feature extraction

```
In [21]: model_cnn=VGG16(weights='imagenet',include_top=False)
         model_cnn.summary()

         Model: "vgg16"
         _________________________________________________________________
         Layer (type)                 Output Shape              Param #
         =================================================================
         input_1 (InputLayer)         (None, None, None, 3)     0
         _________________________________________________________________
         block1_conv1 (Conv2D)        (None, None, None, 64)    1792
         _________________________________________________________________
         block1_conv2 (Conv2D)        (None, None, None, 64)    36928
         _________________________________________________________________
         block1_pool (MaxPooling2D)   (None, None, None, 64)    0
         _________________________________________________________________
         block2_conv1 (Conv2D)        (None, None, None, 128)   73856
         _________________________________________________________________
         block2_conv2 (Conv2D)        (None, None, None, 128)   147584
         _________________________________________________________________
         block2_pool (MaxPooling2D)   (None, None, None, 128)   0
         _________________________________________________________________
         block3_conv1 (Conv2D)        (None, None, None, 256)   295168
         _________________________________________________________________
```

Data Augmentation using Image Data Generator from keras

```
In [22]: train_datagen = ImageDataGenerator(
         rescale=1./255,)

In [23]: test_datagen = ImageDataGenerator(
             rescale=1./255)
```

Figure 7: Loading VGG16



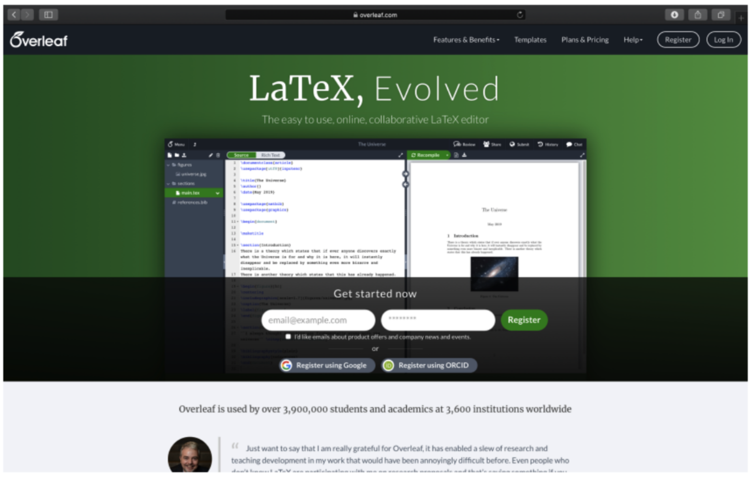
```
ax[1].plot(x, train_acc)
ax[1].plot(x, valid_acc)
ax[1].set_title("Accuracy plot")
ax[1].set_xlabel("Epochs")
ax[1].set_ylabel("acc")
ax[1].legend(['train', 'valid'])

plt.show()
```

```
In [42]: predictions1 = model1.predict(validate_features)
         predictions1 = [1 if x>0.5 else 0 for x in predictions1]
         accuracy1 = accuracy_score(validate_labels, predictions1)
         print('Val Accuracy = %.2f' % accuracy1)

         Val Accuracy = 0.80

In [43]: confusion_mtx1 = confusion_matrix(validate_labels, predictions1)
```

Figure 8: Confusion Matrix

6 Other software that was utilized

Latex is used to write the report. The figures inside the report were created using draw.io.



LaTeX, Evolved

The easy to use, online, collaborative LaTeX editor

Get started now

□ I'd like emails about product offers and company news and events.

Overleaf is used by over 3,900,000 students and academics at 3,600 institutions worldwide

"Just want to say that I am really grateful for Overleaf, it has enabled a slew of research and teaching development in my work that would have been annoyingly difficult before. Even people who

Figure 9: Overleaf

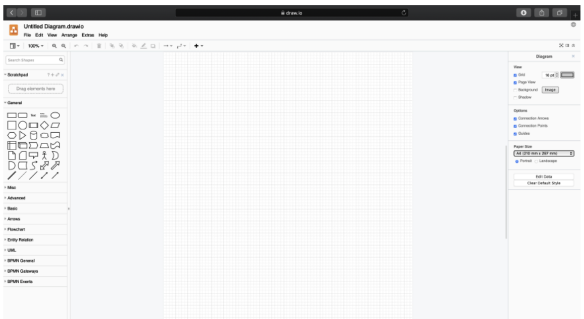


Figure 10: Draw.io