National College of Ireland

Configuration Manual

MSc Research Project
Data Analytics

Sruthi Chandrasekaran
Student ID: x19233159

School of Computing
National College of Ireland

Supervisor: Christian Horn

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Sruthi Chandrasekaran
Student ID:	x19233159
Programme:	Data Analytics
Year:	2021
Module:	MSc Research Project
Supervisor:	Christian Horn
Submission Due Date:	16/8/2021
Project Title:	Configuration Manual
Word Count:	XXX
Page Count:	5

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):	

Configuration Manual

Sruthi Chandrasekaran
x19233159

1 Introduction

The purpose of this document is to provide all of the instructions needed to replicate the implementation of the American Sign Language Recognition and Translation.

2 Hardware Requirements

Windows 10 with 64-bit operating system, an 11th Gen Intel(r) Core (TM) i7 Processor and 8GB of RAM were the configuration of the computer that is used for project implementation.

Figure 1: Hardware Configuration

3 Software Requirement

Through out the research, Python 3 is used for all coding section. Anaconda Navigator platform is installed, where it has Jupyter Lab, Jupyter Notebook, in that a Python 3

file is opened to run and execute code. The 64bit version of Anaconda for Windows 10 needs to be installed.

After Successful installation, to open Anaconda Navigator(Fig. 3), **Start -¿ Anaconda Navigator**, then launch juper lab or jupyter notebook. Automatically it will pop up in chrome browser once we click launch.

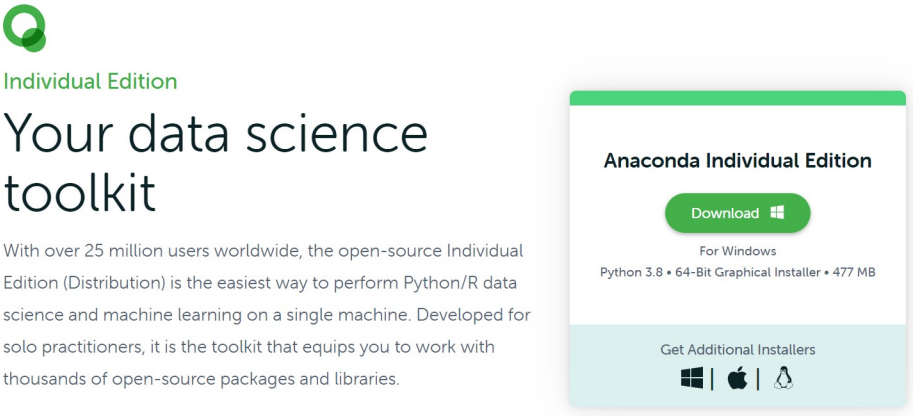

Figure 2: Download Anaconda

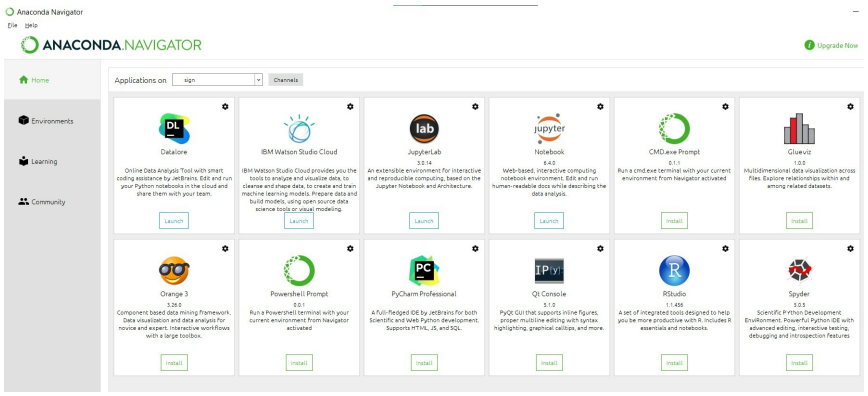

Figure 3: Anaconda Navigator

4 Library Package Requirements

The required python packages installed in the environment is listed below. 'pip' command is used to install all packages.

- Keras
- Tensorflow
- Numpy

- Matplotlib
- Open CV

```
import cv2
import matplotlib.pyplot as plt
import numpy as np

import tensorflow as tf
from tensorflow import keras
from tensorflow.keras import layers
from tensorflow.keras import Model
from tensorflow.keras.layers import Conv2D, MaxPool2D, Flatten, Dense, Dropout, BatchNormalization
from tensorflow.keras.models import Sequential
from tensorflow.keras import regularizers
```

Figure 4: Libraries Required

5 Dataset Description

- The dataset is extracted from Kaggle , "ASL Alphabet". There are 87, 000 images in the dataset. It has 29 folders, 26 ASL letters and 3 special characters.

  https://www.kaggle.com/grassknoted/asl-alphabet

- Dataset contains two folders, one is test and other is train.

6 Dataset Preparation and Pre-processing

- The JupyterNotebook file train_model_final.ipynb available in the artefacts zip file ,contains the step by step coding done in order to prepare, process and transform data as required by the research objective.
- The Image Augmentation is carried out.

```
from tensorflow.keras.preprocessing.image import ImageDataGenerator

train_dir = 'C:/Users/chand/train'

train_datagen = ImageDataGenerator(rotation_range = 10,
width_shift_range = 0.2,
height_shift_range = 0.2,
shear_range = 0.2,
zoom_range = 0.2,
horizontal_flip = True,
fill_mode = 'nearest')

train_generator = train_datagen.flow_from_directory(train_dir,
                                                    class_mode = 'categorical',
                                                    target_size = (64, 64),
                                                    subset = 'training')

validation_generator = train_datagen.flow_from_directory(train_dir,
                                                    class_mode = 'categorical',
                                                    target_size = (64, 64),
                                                    subset = 'validation')
Found 78300 images belonging to 29 classes.
Found 8700 images belonging to 29 classes.
```

Figure 5: Data Pre-Processing

7 Model Preparation

In the notebook train_model_final.ipynb in folder 'train' and final.py in folder 'code' are the files that contains code for this research. These files are available in artefacts. train_model_final.ipynb file contains code for training CNN model and final.py contains code for real-time implementation of the system.

7.1 CNN Model

- Data is extracted, and Data Augmentation is done
- Convolution Neural Network model is trained and build
- The model is trained with 10 epochs
- Trained model is saved as "gesture_model.h5" in folder 'code'.
- Accuracy of this model is 94%

```
model = Sequential()

model.add(Conv2D(16, kernel_size = [3,3], padding = 'same', activation = 'relu', input_shape = (64,64,3)))
model.add(Conv2D(32, kernel_size = [3,3], padding = 'same', activation = 'relu'))
model.add(MaxPool2D(pool_size = [3,3]))

model.add(Conv2D(32, kernel_size = [3,3], padding = 'same', activation = 'relu'))
model.add(Conv2D(64, kernel_size = [3,3], padding = 'same', activation = 'relu'))
model.add(MaxPool2D(pool_size = [3,3]))

model.add(Conv2D(128, kernel_size = [3,3], padding = 'same', activation = 'relu'))
model.add(Conv2D(256, kernel_size = [3,3], padding = 'same', activation = 'relu'))
model.add(MaxPool2D(pool_size = [3,3]))

model.add(BatchNormalization())

model.add(Flatten())
model.add(Dropout(0.5))
model.add(Dense(512, activation = 'relu', kernel_regularizer = regularizers.l2(0.001)))
model.add(Dense(29, activation = 'softmax'))

model.summary()
```

Figure 6: CNN Model

8 American Sign Language Translation

- The CNN model with Image Augmentation has an accuracy of 94% . Hence it is used for Ameerican Sign Language Recognition and Translation.
- final.py contains the code for the real-time implementation of the ASL Recognition and Translation.
- A frame comes in, where the hand gesture have to be shown in the rectangle and once recognized it will translate and display text below.
- To convert to audio, move hand towards the top right corner, where it is written 'tap to speech'
- The user can show any hand gesture and translate to both text and audio.

8.1 Steps to trigger the ASL Recognition and Translation System

- Open Anaconda prompt with admin rights
- Set the path according to the file location

- Activate conda by 'conda activate'
- Execute the command 'python final.py'
- It will fetch the model automatically and starts the execution.
- Now the UI of ASL recognition and translation system will appear as shown in fig. 7
- 'WARNING : The background should be white.'

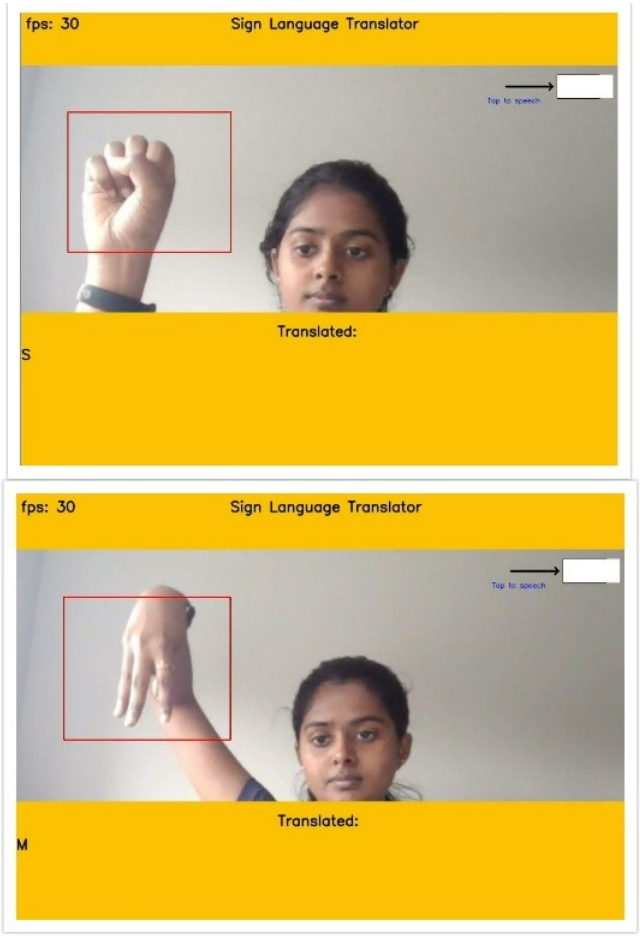

Figure 7: Real-time ASL Recognition and Translation