Configuration Manual

MSc Research Project
MSc. in Cloud Computing

Sanal Sudhakaran
Student ID: 21125775

School of Computing
National College of Ireland

Supervisor: Dr. Rashid Mijumbi

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: Sanal Sudhakaran … ……………………………………………………………………………………

Student ID: 21125775 ……………………………………………………………………………………………..……

Programme: MSc. in Cloud Computing …………………………………………………………… **Year:** 2022 …………………………..

Module: Research Project ……………………………………………………………………………………………….………

Lecturer: Dr. Rashid Mijumbi ……………………………………………………………………………………………….………

Submission Due Date: 15/12/2022 ……………………………………………………………………………………………….………

Project Title: Facial Emotion Detection using Deep Learning for Psychometric Assessment in a Cloud Environment ……………………………………………………………………………………………….………

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: Sanal Sudhakaran ……………………………………………………………………………………………………

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

Forename Surname
Student ID:

1 Introduction

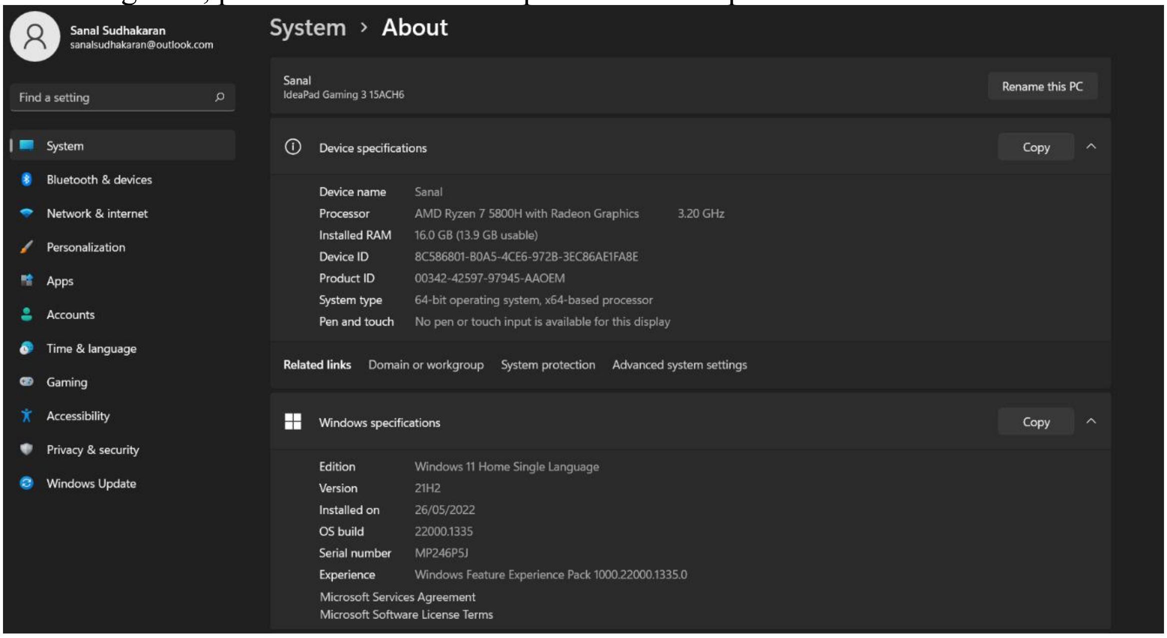
All the requirements for replicating the study and its results on a personal environment are listed in this Configuration Manual. Data Import and Exploratory Data Analysis, data augmentation, all models built, and software and hardware requirements are included.

2 System Specifications

This section provides the details of Software and Hardware requirements to implement the research done.

2.1 Hardware Requirements

Below Figure 1, provides the hardware specifications required.



Sanal
IdeaPad Gaming 3 15ACH6

Device specifications

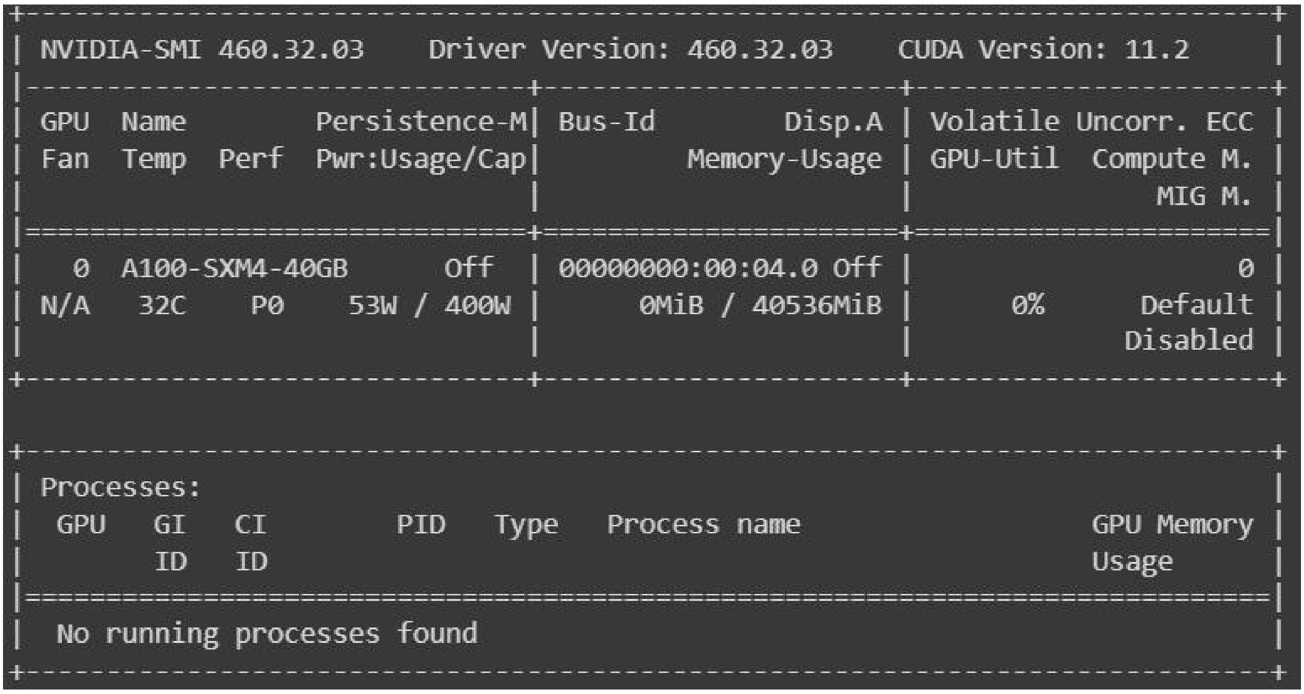
Device name	Sanal
Processor	AMD Ryzen 7 5800H with Radeon Graphics 3.20 GHz
Installed RAM	16.0 GB (13.9 GB usable)
Device ID	8C586801-B0A5-4CE6-972B-3EC86AE1FA8E
Product ID	00342-42597-97945-AAOEM
System type	64-bit operating system, x64-based processor
Pen and touch	No pen or touch input is available for this display

Windows specifications

Edition	Windows 11 Home Single Language
Version	21H2
Installed on	26/05/2022
OS build	22000.1335
Serial number	MP246P5J
Experience	Windows Feature Experience Pack 1000.22000.1335.0

Microsoft Services Agreement
Microsoft Software License Terms

Figure 1: Hardware Requirements



```
+-----------------------------------------------------------------------------+
| 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   32C    P0    53W / 400W |      0MiB / 40536MiB |      0%      Default |
|                               |                      |             Disabled |
+-------------------------------+----------------------+----------------------+

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

Figure 2: Hardware Requirements for google Colab

2.2 Software Requirements

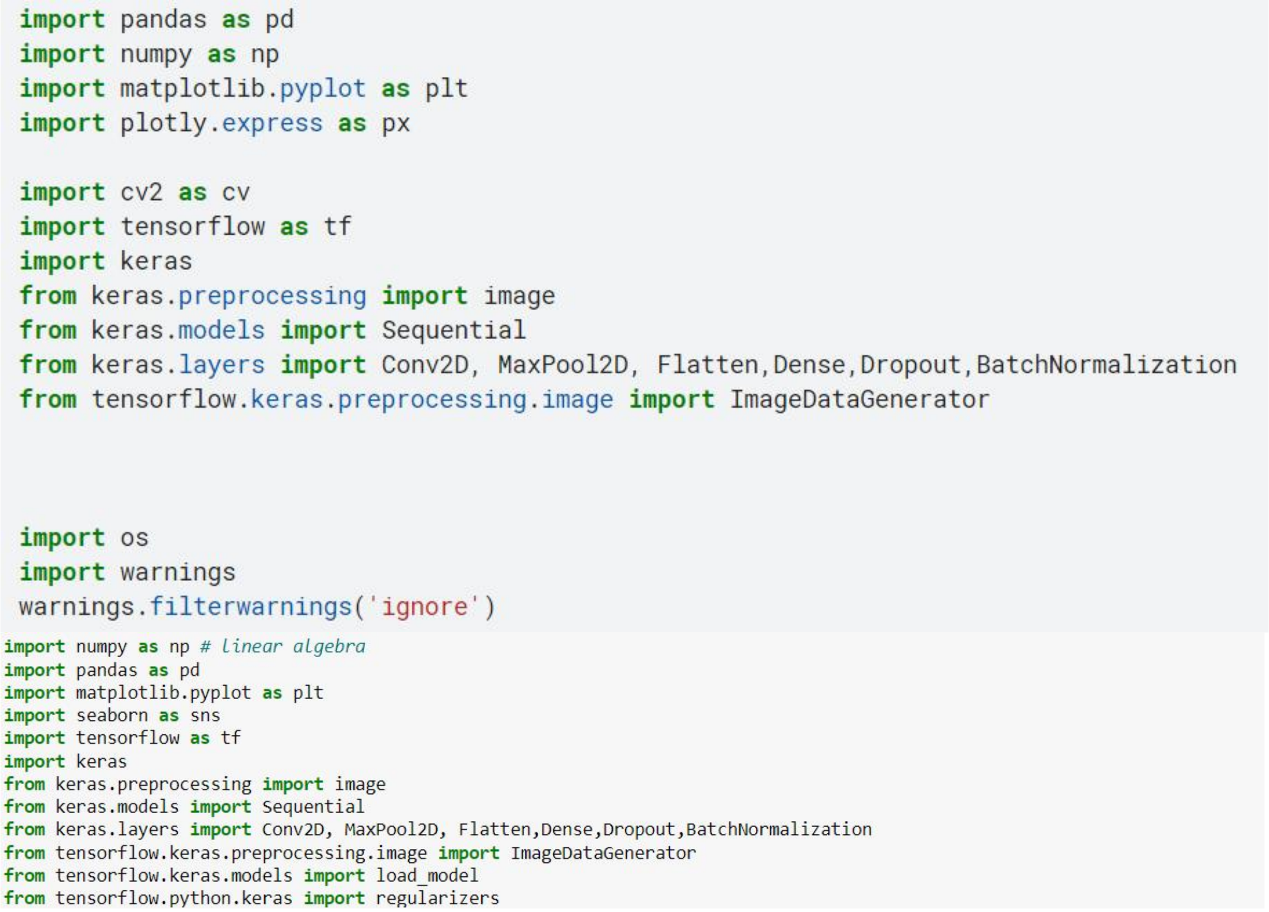
- Jupyter Notebook (Version 6.0.3) or Google Colab
- Python (Version 3.7.6)

3 Data Collection

The dataset is collected from Kaggle
https://www.kaggle.com/datasets/ananthu017/emotion-detection-fer

4 Data Exploration

All the Python libraries required to implement the entire project are listed in Figure 2.

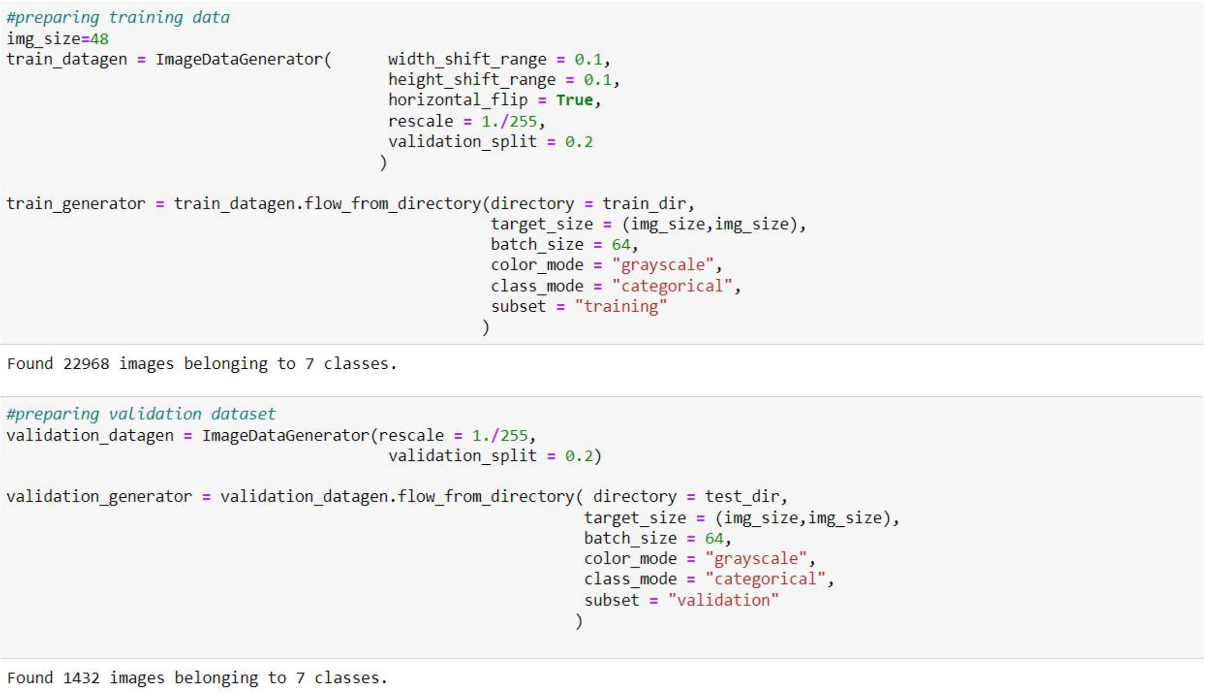


```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import plotly.express as px

import cv2 as cv
import tensorflow as tf
import keras
from keras.preprocessing import image
from keras.models import Sequential
from keras.layers import Conv2D, MaxPool2D, Flatten,Dense,Dropout,BatchNormalization
from tensorflow.keras.preprocessing.image import ImageDataGenerator



import os
import warnings
warnings.filterwarnings('ignore')
```

```python
import numpy as np # linear algebra
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import tensorflow as tf
import keras
from keras.preprocessing import image
from keras.models import Sequential
from keras.layers import Conv2D, MaxPool2D, Flatten,Dense,Dropout,BatchNormalization
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.models import load_model
from tensorflow.python.keras import regularizers
```

Figure 3: Required Python Libraries

5 Data Pre-processing:



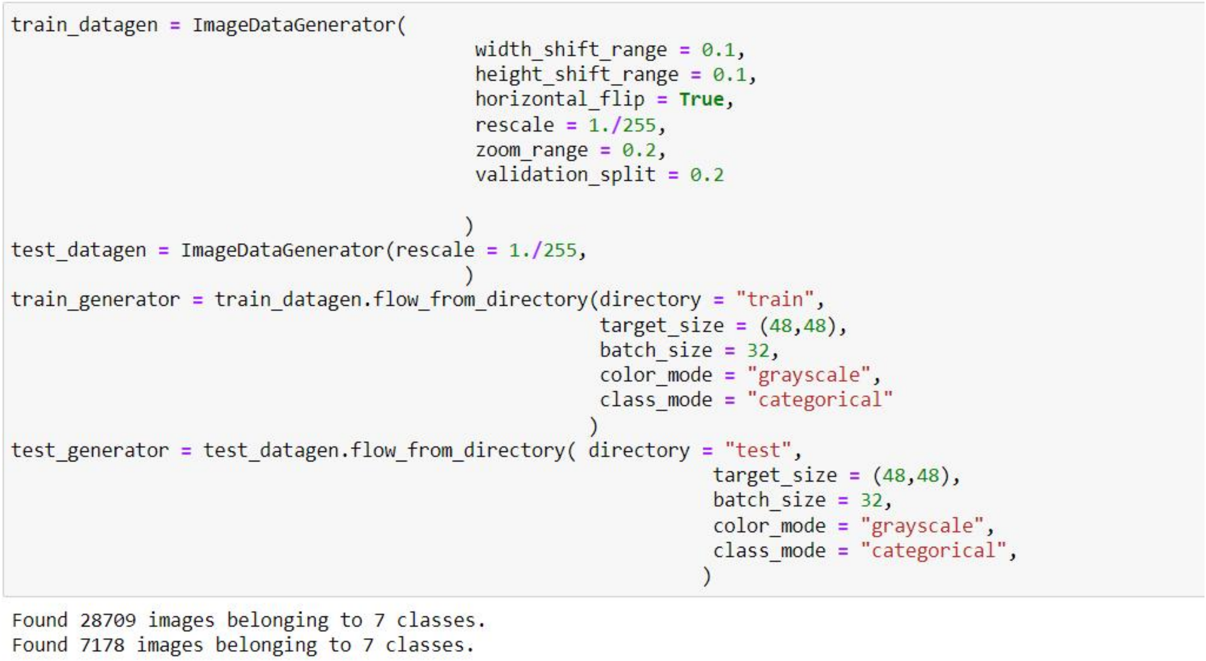
```
#preparing training data
img_size=48
train_datagen = ImageDataGenerator(      width_shift_range = 0.1,
                                         height_shift_range = 0.1,
                                         horizontal_flip = True,
                                         rescale = 1./255,
                                         validation_split = 0.2
                                        )

train_generator = train_datagen.flow_from_directory(directory = train_dir,
                                                    target_size = (img_size,img_size),
                                                    batch_size = 64,
                                                    color_mode = "grayscale",
                                                    class_mode = "categorical",
                                                    subset = "training"
                                                   )
```

```
Found 22968 images belonging to 7 classes.
```

```
#preparing validation dataset
validation_datagen = ImageDataGenerator(rescale = 1./255,
                                         validation_split = 0.2)

validation_generator = validation_datagen.flow_from_directory( directory = test_dir,
                                                              target_size = (img_size,img_size),
                                                              batch_size = 64,
                                                              color_mode = "grayscale",
                                                              class_mode = "categorical",
                                                              subset = "validation"
                                                             )
```

```
Found 1432 images belonging to 7 classes.
```

Figure 4: Data pre-processing and augmentation for Custom CNN Model (3-Layers)

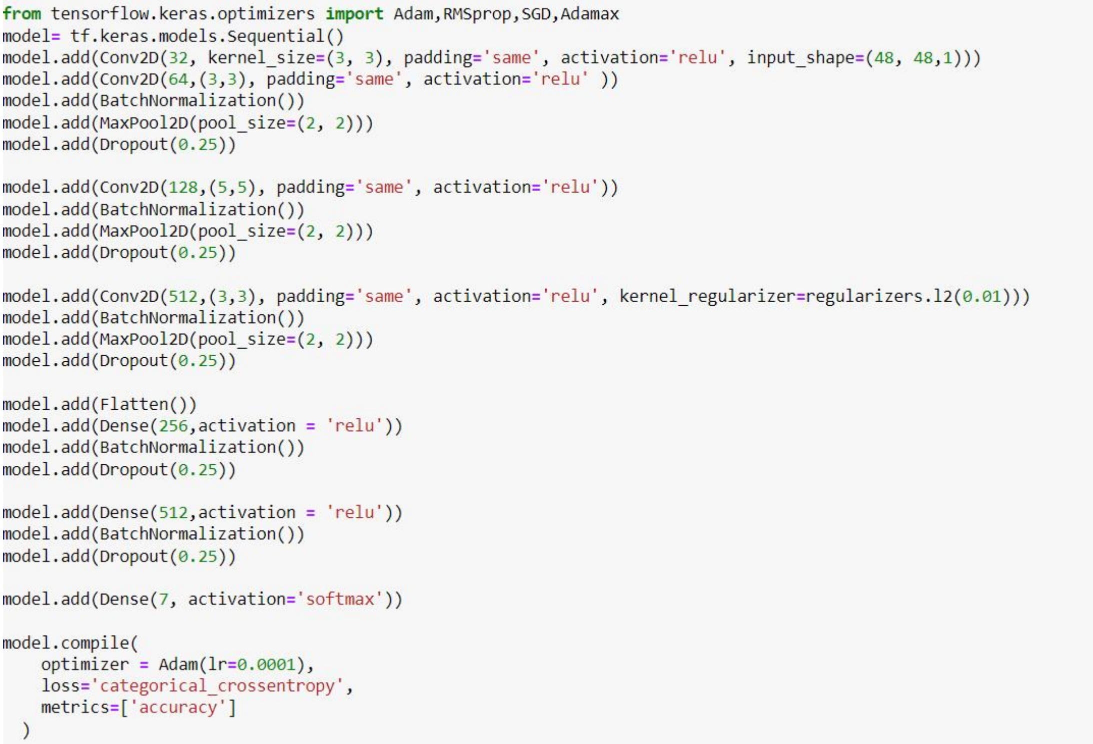


```
train_datagen = ImageDataGenerator(
                                         width_shift_range = 0.1,
                                         height_shift_range = 0.1,
                                         horizontal_flip = True,
                                         rescale = 1./255,
                                         zoom_range = 0.2,
                                         validation_split = 0.2

                                        )
test_datagen = ImageDataGenerator(rescale = 1./255,
                                        )
train_generator = train_datagen.flow_from_directory(directory = "train",
                                                    target_size = (48,48),
                                                    batch_size = 32,
                                                    color_mode = "grayscale",
                                                    class_mode = "categorical"
                                                   )
test_generator = test_datagen.flow_from_directory( directory = "test",
                                                              target_size = (48,48),
                                                              batch_size = 32,
                                                              color_mode = "grayscale",
                                                              class_mode = "categorical",
                                                             )
```

```
Found 28709 images belonging to 7 classes.
Found 7178 images belonging to 7 classes.
```

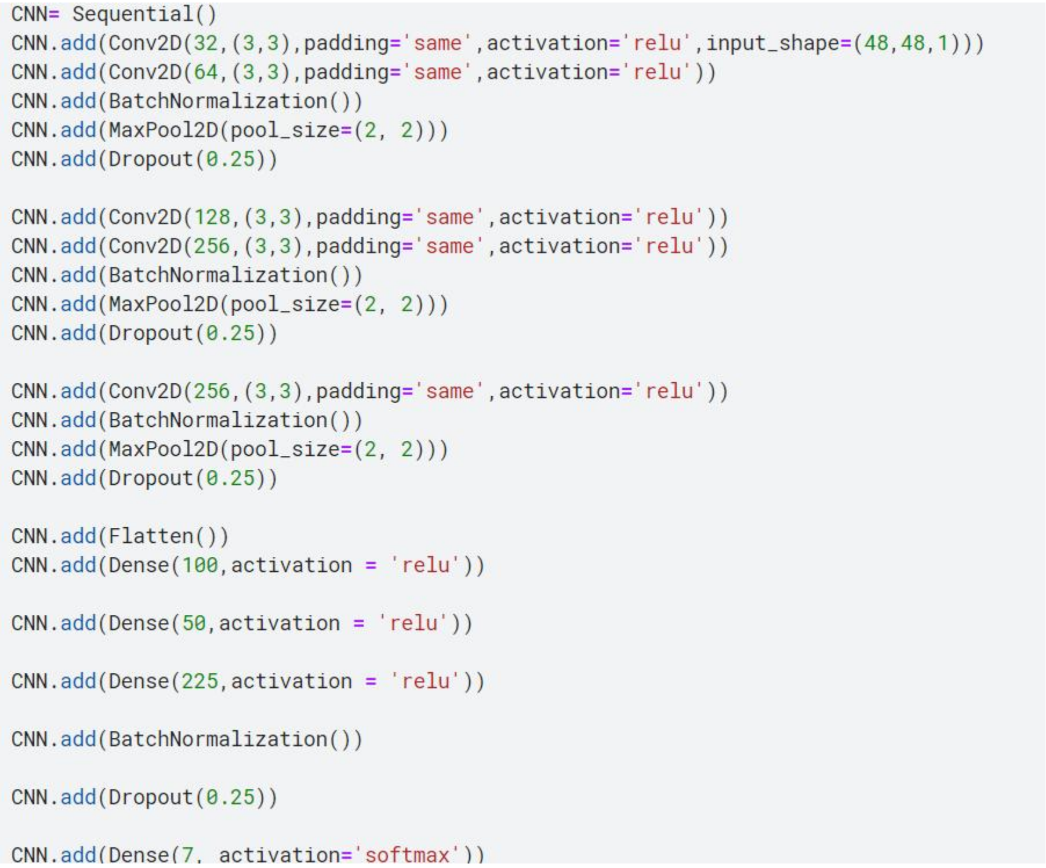
Figure 5: Data pre-processing and augmentation for Custom CNN Model (5-Layers)

6 Model Preparation:



```
from tensorflow.keras.optimizers import Adam,RMSprop,SGD,Adamax
model= tf.keras.models.Sequential()
model.add(Conv2D(32, kernel_size=(3, 3), padding='same', activation='relu', input_shape=(48, 48,1)))
model.add(Conv2D(64,(3,3), padding='same', activation='relu' ))
model.add(BatchNormalization())
model.add(MaxPool2D(pool_size=(2, 2)))
model.add(Dropout(0.25))

model.add(Conv2D(128,(5,5), padding='same', activation='relu'))
model.add(BatchNormalization())
model.add(MaxPool2D(pool_size=(2, 2)))
model.add(Dropout(0.25))

model.add(Conv2D(512,(3,3), padding='same', activation='relu', kernel_regularizer=regularizers.l2(0.01)))
model.add(BatchNormalization())
model.add(MaxPool2D(pool_size=(2, 2)))
model.add(Dropout(0.25))

model.add(Flatten())
model.add(Dense(256,activation = 'relu'))
model.add(BatchNormalization())
model.add(Dropout(0.25))

model.add(Dense(512,activation = 'relu'))
model.add(BatchNormalization())
model.add(Dropout(0.25))

model.add(Dense(7, activation='softmax'))

model.compile(
    optimizer = Adam(lr=0.0001),
    loss='categorical_crossentropy',
    metrics=['accuracy']
  )
```

Figure 6: Custom CNN(3 Layer) Model Preparation



```
CNN= Sequential()
CNN.add(Conv2D(32,(3,3),padding='same',activation='relu',input_shape=(48,48,1)))
CNN.add(Conv2D(64,(3,3),padding='same',activation='relu'))
CNN.add(BatchNormalization())
CNN.add(MaxPool2D(pool_size=(2, 2)))
CNN.add(Dropout(0.25))

CNN.add(Conv2D(128,(3,3),padding='same',activation='relu'))
CNN.add(Conv2D(256,(3,3),padding='same',activation='relu'))
CNN.add(BatchNormalization())
CNN.add(MaxPool2D(pool_size=(2, 2)))
CNN.add(Dropout(0.25))

CNN.add(Conv2D(256,(3,3),padding='same',activation='relu'))
CNN.add(BatchNormalization())
CNN.add(MaxPool2D(pool_size=(2, 2)))
CNN.add(Dropout(0.25))

CNN.add(Flatten())
CNN.add(Dense(100,activation = 'relu'))

CNN.add(Dense(50,activation = 'relu'))

CNN.add(Dense(225,activation = 'relu'))

CNN.add(BatchNormalization())

CNN.add(Dropout(0.25))

CNN.add(Dense(7, activation='softmax'))
```

Figure 7: Custom CNN(5 Layer) Model Preparation

References:

https://keras.io/api/layers/convolution_layers/

https://www.tensorflow.org/tutorials/images/cnn