National College of Ireland

Deep Learning for Automated Yoga Practice Assistance

–

MSc Research Project
Data Analytics

Chandana Haluvarthi Prabhudeva
Student ID: x22167099

School of Computing
National College of Ireland

Supervisor: Mayank Jain

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Chandana Haluvarthi Prabhudeva
Student ID:	x22167099
Programme:	Data Analytics
Year:	2023
Module:	MSc Research Project
Supervisor:	Mayank Jain
Submission Due Date:	31/01/2024
Project Title:	**Deep Learning for Automated Yoga Practice Assistance** _
Word Count:	133
Page Count:	3

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:	Chamdana HP
Date:	31th January 2024

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

Chandana Haluvarthi Prabhudeva
X22167099

1 Introduction

The objective of this guide is to provide an explanation of the implementation settings and setup used in this study. The libraries used in this project, as well as the hardware and software settings, are fully described in this documentation. It also explains how to run the code and the steps involved in the coding process.

2 Local Machine System Configuration

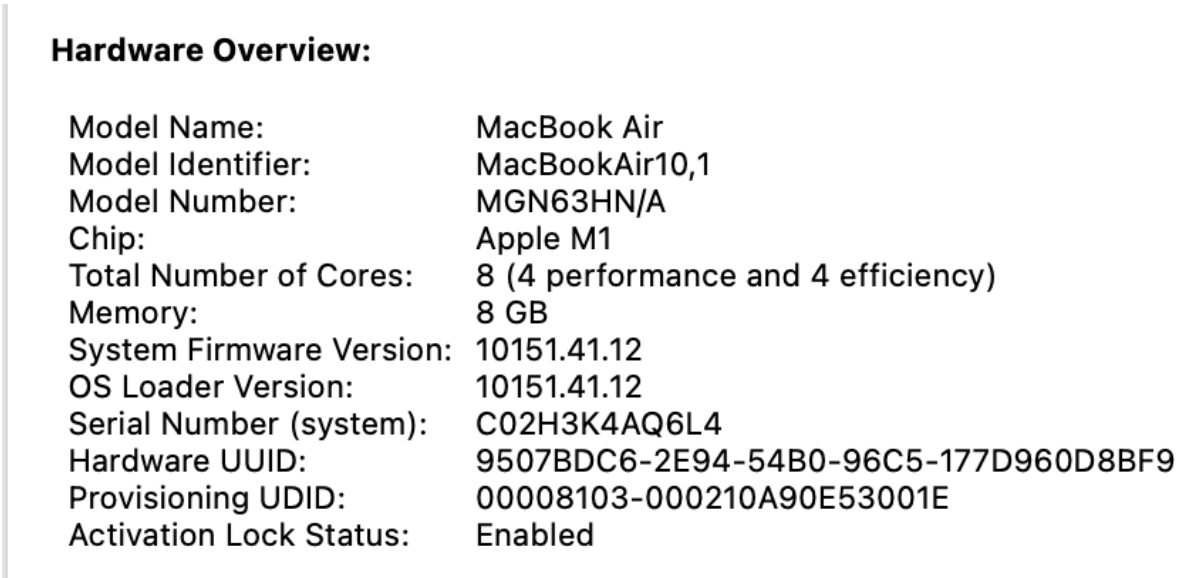

Figure 1: Hardware specification of the system

The above figure is the system configuration used in the reserach.

3 Dataset Collection

The dataset that used in this work for building the classification models is collected from the Kaggle site. The collected dataset have the five categories of the Yoga Pose.

```
(base) chandanahp@Chandanas-MacBook-Air ~ % jupyter --version
Selected Jupyter core packages...
IPython          : 7.31.1
ipykernel        : 6.15.2
ipywidgets       : 7.6.5
jupyter_client   : 7.3.4
jupyter_core     : 4.11.1
jupyter_server   : 1.18.1
jupyterlab       : 3.4.4
nbclient         : 0.5.13
nbconvert        : 6.4.4
nbformat         : 5.5.0
notebook         : 6.4.12
qtconsole        : 5.3.2
traitlets        : 5.1.1
(base) chandanahp@Chandanas-MacBook-Air ~ %
```

Figure 2: Jupyter specification of the system

4 JupyterLab SetUp

The Jupyter lab version 3.4.4 has been used to run the research code and python 7.31.1 has been used all along the research.

5 Importing the Dataset

```
# Importing the libraries to access local folder and importing the dataset
import os
import cv2
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from keras.utils import to_categorical
from PIL import Image
```

Figure 3: Jupyter imports

The below steps are the required steps to run the code:
1. The above libraries are required to run the code efficiently.

2. The dataset is stored in the current directory under the name "Train".

3.The images are converted to rgb if they are in gif and this is done for the whole dataset over a loop

```python
# Step 2: Data Cleaning and Preprocessing

# Load images and labels
images = []
labels = []

for category in categories:
    category_path = os.path.join(dataset_path, category)
    for img_name in os.listdir(category_path):
        img_path = os.path.join(category_path, img_name)

        # Check if the image is a gif
        if img_path.endswith('.gif'):
            img = Image.open(img_path)
            img = img.convert('RGB')
            img = np.array(img)
        else:
            img = cv2.imread(img_path)
            if img is None:
                print(f"Error loading image: {img_path}")
                continue
            img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)

        img = cv2.resize(img, (128, 128))  # Resize to a standard size
        images.append(img)
        labels.append(category)
```

Figure 4: The pre-processing steps that are done

4. The images are plotted using the below set of code, here the actual vs the predicted poses are given.

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

# Function to plot images with labels
def plot_images(images, actuals, preds, class_names):
    plt.figure(figsize=(10, 10))
    for i in range(25):  # Adjust the range as needed
        plt.subplot(5, 5, i + 1)
        plt.xticks([])
        plt.yticks([])
        plt.grid(False)
        plt.imshow(images[i], cmap=plt.cm.binary)
        actual_label = class_names[np.argmax(actuals[i])]
        predicted_label = class_names[np.argmax(preds[i])]
        color = 'blue' if actual_label == predicted_label else 'red'
        plt.xlabel(f"Actual: {actual_label}\nPredicted: {predicted_label}", color=color)
    plt.show()

# Predicting labels on the test set
predicted_labels = model.predict(X_test)

# Assuming 'class_names' is a list of class names in the order they were encoded
plot_images(X_test, y_test, predicted_labels, class_names=label_encoder.classes_)
```

Figure 5: The acutal and the predicted yoga poses