A

PROJECT REPORT

on

YOGA POSTURE DETECTION

Submitted in partial fulfilment for the Award of Degree of

BACHELOR OF ENGINEERING

IN

INFORMATION TECHNOLOGY

BY

M. Yogitha Nandini (160117737030)

&

P. Arun Raj (160117737034)

Under the guidance of

Ms. R. Deepa

Assistant Professor
Dept. of IT, CBIT.

DEPARTMENT OF INFORMATION TECHNOLOGY
CHAITANYA BHARATHI INSTITUTE OF TECHNOLOGY (A)
(Affiliated to Osmania University; Accredited by NBA (AICTE) and NAAC (UGC), ISO Certified 9001:2015), Kokapet (V), GANDIPET(M), HYDERABAD – 500 075

Website: www.cbit.ac.in

2020-2021

CHAITANYA BHARATHI INSTITUTE OF TECHNOLOGY (A)
Kokapet (Village), Gandipet, Hyderabad, Telangana-500075. www.cbit.ac.in
ISO Certified 9001:2015

COMMITTED TO RESEARCH, INNOVATION AND EDUCATION
42 years

CERTIFICATE

This is to certify that the project work entitled "**Yoga Posture Detection**" submitted by **M. Yogitha Nandini (160117737030)** and **P. Arun Raj (160117737034)** in partial fulfilment of the requirements for the award of the degree of **BACHELOR OF ENGINEERING in INFORMATION TECHNOLOGY to CHAITANYA BHARATHI INSTITUTE OF TECHNOLOGY(A),** affiliated to **OSMANIA UNIVERSITY**, Hyderabad, is a record of bonafide work carried out by him under my supervision and guidance. The results embodied in this report have not been submitted to any other University or Institute for the award of any other Degree or Diploma.

Project Guide
Ms. R. Deepa
Assistant Professor, Dept. of IT,
CBIT, Hyderabad.

Head of the Department
Dr. K. Radhika.
Professor & Head, Dept. of IT,
CBIT, Hyderabad.

DECLARATION

I declare that the project report entitled **“Yoga Posture Detection”** is being submitted by me in the Department of Information Technology, Chaitanya Bharathi Institute of Technology (A), Osmania University.

This is record of bonafide work carried out by me under the guidance and supervision of **Ms. R.Deepa**, Assistant Professor, Dept. of IT, C.B.I.T.

No part of the work is copied from books/journals/internet and wherever the portion is taken, the same has been duly referred in the text. The reported are based on the project work done entirely by me and not copied from any other source.

M. Yogitha Nandini (160117737030)
P. Arun Raj (160117737034)

ACKNOWLEDGEMENTS

It is my privilege to acknowledge with deep sense of gratitude and devotion for keen personal interest and invaluable guidance rendered by our Project Guide **Ms. R.Deepa**, Assistant Professor, Department of Information, Chaitanya Bharathi Institute of Technology.

I take the opportunity to express my thanks to **Dr K. Radhika**, Professor & Head of IT Department, CBIT for her valuable suggestions and moral support. I am grateful to my Principal **Dr. G. P. Saradhi Varma,**, Chaitanya Bharathi Institute of Technology, for his cooperation and encouragement. Finally, I also thank all the staff members, faculty of Dept. of IT, CBIT, and my friends, who with their valuable suggestions and support, directly or indirectly helped me in completing this project work.

ABSTRACT

Yoga is an ancient Indian art, but from the last decade, a large number of people are adopting yoga as part of their life. This is due to the health benefits. It is important to do this exercise in right way especially in right posture. Yoga should be done under the guidance of a trainer but it is also not affordable for all the peoples. So, we came up with this project which helps people to gain some knowledge about the yoga posture which they do and the correct position of it. In this project, we train a model which helps to classify the name of the yoga posture. we take the input of image posture from the user and output the name of pose and give the text of steps to be performed for the yoga posture and voice for the yoga posture steps. Finally gives the visual view of his/her input posture and the relevant output posture from the training set.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

INTRODUCTION

1.1 OVERVIEW

Yoga is originated in ancient India and it is a group exercise associated with mental, physical and spiritual strength. Yoga has been attracting peoples from so many years but from the last decade, a large number of people are adopting yoga as part of their life. This is due to the health benefits. It is important to do this exercise in right way especially in right posture. It has been observed that sometime due to lack of assistance or knowledge people don’t know the correct method to do yoga and start doing yoga without any guidance, thus they injure themselves during self-training due to improper posture. Yoga should be done under the guidance of a trainer but it is also not affordable for all the peoples. Nowadays people use their mobile phones to learn how to do yoga poses and start doing that but while doing that they don’t even know that the yoga pose they are doing is in the right way or not. To overcome these limitations, many works have been done.

We take the input of image posture from the user and output the name of pose and give the text of steps to be performed for the yoga posture and voice for the yoga posture steps. Finally gives the visual view of his/her input posture and the relevant output posture from the training set.

We train a model which helps to classify the name of the yoga posture. Similarity functions to give the relevant images for the user inputted image from the training set.

1.2 APPLICATIONS

These are the existing applications in which they have pre-recorded videos. Every app below need a subscription plan for which users must pay according to the plan.

1. Yoga Studio
2. Down Dog
3. Alo Moves
4. Find What Feels Good
5. Daily Yoga
6. Yoga For Beginners
7. Gaia
8. Minute Yoga

1.3 PROBLEM DEFINITION

In our project we wanted the user to do yoga in a proper manner without hurting themselves. This model helps to know the name of posture performed by the user and guides the user with the steps to be performed for the yoga posture to be perfect. Both steps and the visual view of his/her input, the relevant output postures are given in order to adjust their posture.

1.4 AIM OF THE PROJECT

The main objective of this project is the user must know the mistakes of the yoga posture he/she is doing. It must be available and affordable for every user who wants to adopt yoga as their part of life. Yoga should be done under the guidance of a trainer; this project will guide the user in easy and understandable way.

2. LITERATURE SURVEY

2.1 PAPER 1

Title: Yoga posture recognition by detecting human joint points in real time using Microsoft Kinect.

Author: Muhammad Usama Islam, Hasan Mahmud, Faisal Bin Ashraf, Iqbal Hossain and Md. Kamrul Hasan.

Year of Publication: Year 2017.

Introduction:

Human by nature is vulnerable and subject to wide ranging health diseases of which musculoskeletal disorders is an important arena and needs urgent attention. Every year a wide range of people are affected from various types of musculoskeletal disorders due to accidents or aging problem. Yoga can promote positive physical change. Studies have shown that yoga is effective in managing symptoms associated with musculoskeletal disorders including osteoarthritis carpal tunnel syndrome, hyper kyphosis and low back pain. Additionally, Yoga shows significant improvements in motor skills and physiological methods that includes blood pressure, heart speed rate and human body weight have been noted. Research also suggests that cardiopulmonary benefits of yoga include improvement of cardiorespiratory wellness as well as improved forced expiratory volume and increased vital capacity. Traditionally, yoga is done in a yoga center in the presence of a yoga trainer who can guide the patient through therapeutic assistance. Our solution suggests a distant yoga training approach without the trainer where the patient can stand in front of a device and perform yoga poses correctly without the need of trainer or being present at the yoga training center. Real time human pose detection is a growing and important area which specializes on realizing and understanding the human pose from depth images. With the introduction of Microsoft Kinect it became easier to understand human pose. Till now a wide range of work is done on human pose detection, physical rehabilitation using Kinect. But, Yoga an important aspect of physical rehabilitation remained untouched. In our work we want to detect yoga poses by human from Microsoft Kinect and cross check the poses against true data . We have set a reference model of true data for certain poses with respect to which we have detected the poses.The main challenge was the detection of body co-ordinate joints. Using the co-ordinates of Kinect capture, the poses are detected. The purpose is to remotely monitor the yoga poses for patients. The system will work as a personal guidance system. The purpose of remotely monitoring yoga poses is to increase fitness with minimal monetary investment. As, at a certain time doctors or physicians can’t look after all the patients, the will benefit both the patients and doctors in terms of efficiency.

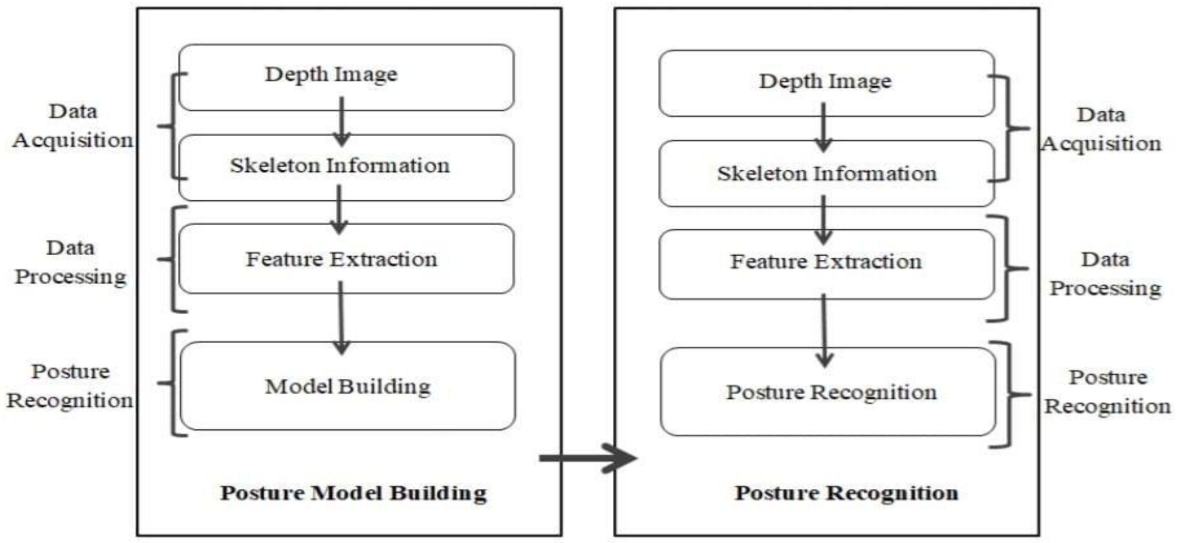


Figure 2.1.1: Steps of the proposed system

In this paper, we have proposed a system to recognize three major yoga pose by detecting of human joint points using Microsoft Kinect. We have detected the yoga poses considering above 97% accuracy in every angle between different body parts. Our system can also be used to recognize other yoga poses from the reference model of each pose. Therefore, we hope our proposed approach will help to practise yoga without trainer.

Advantages:

Graphical analysis of the deviation of angles with reference model.

Disadvantages:

Kinect device is expensive and not user friendly.

2.2 PAPER 2

Title: Computer-assisted yoga training system.

Author: Hua-Tsung Chen & Yu-Zhen He & Chun-Chieh Hsu.

Year of Publication: Year 2018.

Introduction:

For most sports players/practitioners, it is essential to spend time exercising on their own, in addition to the regular training courses given by a coach or instructor. However, if not under the instruction of a coach/instructor, a sports player/practitioner may make progress only to a limited extent, and even may get injured during self-training due to improper postures or training ways. Thus, a great number of sports players/practitioners expect the development of computer-assisted training systems to assist them in improving their performance and protecting them from injury. Accordingly, the topic of computer-assisted sport or exercise training system is attracting more and more research attention. Numerous automatic or semi-automatic training systems have been developed for many sports or exercises and so on. Since the ancient Indian art, yoga, not only promotes physical health but also helps to purge the body, mind, and soul, it gains growing popularity nowadays. When practicing yoga, the practitioner has to align body positions in a special way. If the practitioner does not perform a yoga pose correctly, improper postures may cause serious harm to muscles and ligaments of the body. Thus, we are motivated to develop a computer vision-based yoga self-training system for assisting practitioners in exercising by themselves. Postural instructions and feedback can be provided automatically to help practitioners to adjust their poses.

In this paper, we develop a computer assisted yoga training system, aiming at instructing the practitioner to perform yoga pose correctly and preventing injury caused by improper postures. Integrating computer vision techniques, the proposed system analyzes the practitioner’s posture from front and side views by extracting the contour, skeleton, dominant axes, and feature points of the human body. Then, based on the domain knowledge of yoga training, visualized instructions for posture rectification are presented so that the practitioner can easily understand how to adjust his/her posture.

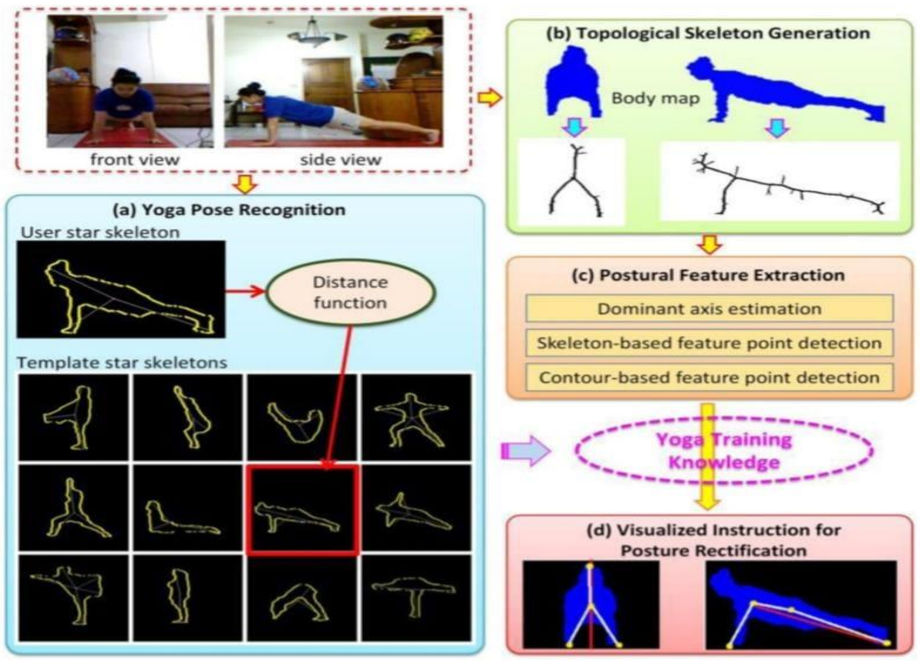

front view
side view
(b) Topological Skeleton Generation
Body map
(a) Yoga Pose Recognition
User star skeleton
Distance function
Template star skeletons
(c) Postural Feature Extraction
Dominant axis estimation
Skeleton-based feature point detection
Contour-based feature point detection
Yoga Training Knowledge
(d) Visualized Instruction for Posture Rectification

Figure 2.2.1: Schematic diagram of the proposed yoga training system

Advantages:

Takes front view and side view. Uses distance function and gives visualized view for posture rectification.

Disadvantages:

It just gives similar yoga postures body map and skeleton map.

2.3 PAPER 3

Title: Machine learning gesture analysis of yoga for exergame development.

Author: Paula Pullen , William Seffens.

Year of Publication: Year 2018.

Introduction:

Benefits and physiological mechanisms has become an active area of study. To decrease the disparity between populations who can readily access yoga classes and therapies, benefits of yoga could be implemented in an exergame format in clinical or home environments. This platform could be installed with low-cost hardware using the cloud for analytics and data collection. We analyzed yoga posture alignment using a 3D room sensor to produce a physical activity exergame for specific groups, such as young adults. This research utilises gesture analysis software to provide skill improvement feedback to students in a yoga course setting. We positioned a yoga mat two metres in front of a Kinect sensor in both sagittal or perpendicular and frontal view orientations. Sagittal view orientation was slightly more accurate than frontal, but not significantly different. Inaccuracy was measured as the number of imputed joint positions observed during the pose, out of a total of 20 joints. Standing poses were significantly more accurate than seated or supine body orientations. The Kinect skeleton algorithm becomes confused when the subjects head is below the waist. This is presumably why no yoga commercial product exists in contrast to exergames such as bowling, dance, Zumba or Chopra meditation. Instability of joint positions in the skeleton data stream was more severe for seated poses as evidenced by large fluctuations between image frames that resulted in visible displacements of joints that are described as jitter.

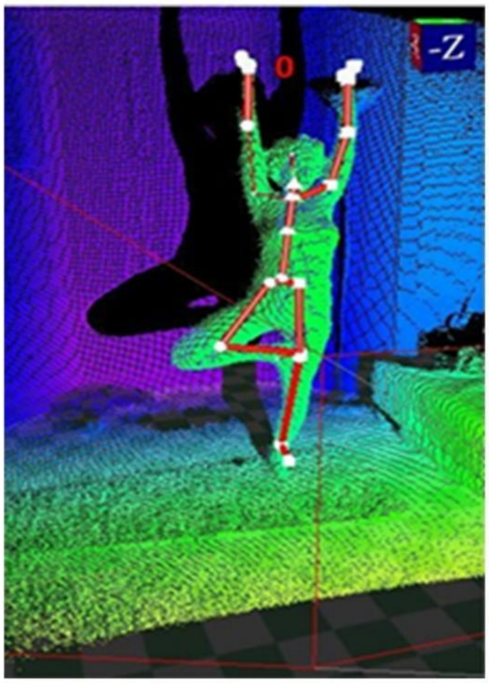


Figure 2.3.1: Skeleton capture of the proposed yoga system

Gesture analysis for yoga alignment training may be a useful tool for the development of home and clinical YT for hard to reach populations. The experimental exergame developed here provides a tool that scores the performance of yoga postures and provides improvement metrics. Statistical measurements have been shown to be able to track changes of yoga posture learning in young adults over a 10-week course.

This could be useful for detecting adherence in home-based YT. Prior research by others has shown that even short-term yoga-based lifestyle interventions were efficacious in weight loss, inflammation and stress and positively influenced cardiovascular risk factors. Our plans are to target special populations with YT and study the potential effects of body mass and age on posture alignment and limb stretch.

Advantages:

Build the exergameusing Adaboost gesture analysis.

Disadvantages:

Kinect device is expensive and not user friendly.

2.4 PAPER 4

Title: Recognition of yoga poses using EMG signals from lower limb muscles.

Author: Pradchaya Anantamek , Narit Hnoohom.

Year of Publication: Year 2019.

Introduction:

The objective was to search for a solution to check the accuracy of the postures from the lower-limb muscles. Electromyography (EMG) is widely used to investigate muscle activity in various fields. EMG is an electrical activity that occurs in the muscle layer during active motion. This EMG signal can be used for many purposes, including the diagnosis of muscle disease, studying the biomechanics of human movement, or other muscular disorders. However, the reliability of detection in this type of EMG technique depends on the placement of the electrodes. If several electrodes are placed tightly next to each other, it generates severe technical problems in signal imaging. Moreover, the impedances of electrode-skin contact must be low, due to the similarity to each other, and it should also be stable over time. Moreover, the crosstalk interference must be reduced, and the artifacts should be avoided. Implementation modalities and performance of electrode grids should be progressively improved. For dry electrode systems, the conductive gel is employed for enhancing time-stability and reducing artifacts.

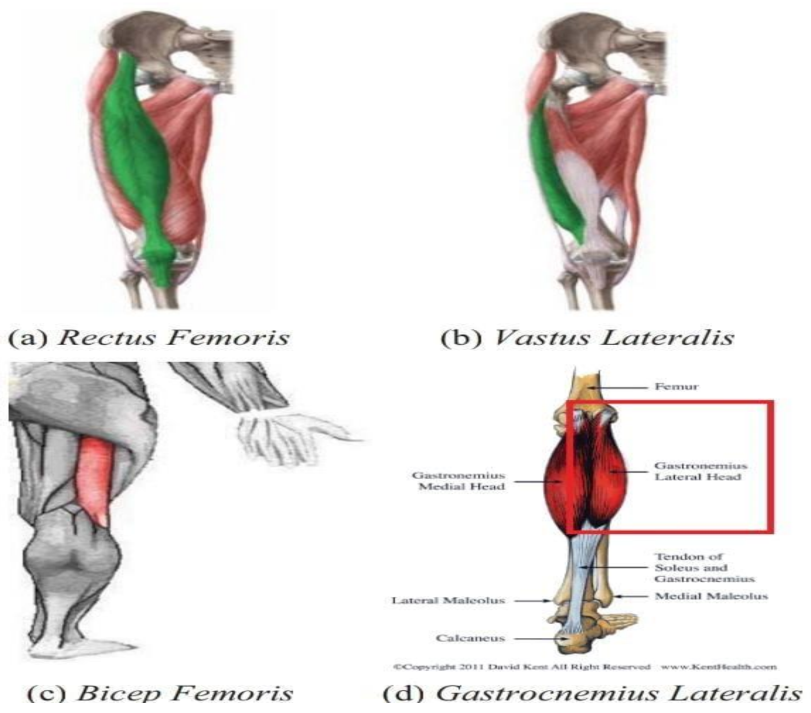

(a) Rectus Femoris
(b) Vastus Lateralis
Femur
Gastronemius Medial Head
Gastronemius Lateral Head
Tendon of Soleus and Gastrocnemius
Lateral Maleolus
Medial Maleolus
Calcaneus
©Copyright 2011 David Kent All Right Reserved www.KentHealth.com
(c) Bicep Femoris
(d) Gastrocnemius Lateralis

Figure 2.4.1: Selection of lower limb muscles of the right leg

This paper presented the yoga posture recognition for determining the correctness of the postures during yoga exercises using machine learning techniques. The EMG signals were used to analyze the motion of four lower-limb muscles during yoga exercises. Feature extraction was used to extract the EMG data before entering it into machine learning techniques for posture recognition, which was performed with SMO, J48, and Random Forest. The experimental results showed that the Random Forest Decision Tree algorithm was able to achieve 87.43 percent classification accuracy. In future research work, more subjects should be involved in this study so that the classification process can achieve a higher level of accuracy.

Advantages:

Myoware muscle sensor, Embedded system are used and weka to classify yoga postures. SMO, J48, RF algorithms gives accuracy.

Disadvantages:

There is discomfort while needle electrodes inserted. Afterwards, muscle may feel a little sore for few days. Experts are required to perform EMG.

2.5 PAPER 5

Title: Implementation of Machine Learning Technique for Identification of Yoga Poses.

Author: Yash Agrawal, Yash Shah, Abhishek Sharma.

Year of Publication: Year 2020.

Introduction:

Yoga is originated in ancient India and it is a group exercise associated with mental, physical and spiritual strength. Yoga and sports have been attracting peoples from so many years but from the last decade, a large number of people are adopting yoga as part of their life. This is due to the health benefits. It is important to do this exercise in right way especially in right posture. It has been observed that sometime due to lack of assistance or knowledge people don't know the correct method to do yoga and start doing yoga without any guidance, thus they injure them-self during self-training due to improper posture. Yoga should be done under the guidance of a trainer but it is also not affordable for all the peoples. Nowadays people use their mobile phones to learn how to do yoga poses and start doing that but while doing that they don't even know that the yoga pose they are doing is in the right way or not. To overcome these limitations, many works have been done. Computer vision and data science techniques have been used to build AI software that works as a trainer. This software tells about the advantages of that pose. It also tells about the accuracy of the performance. Using this software one can do yoga without the guidance of a trainer. To use machine learning and Deep learning modules a large number of image dataset has been created which contain 10 yoga poses. Features have been extracted using computer vision and tf-pose Algorithm. This Algorithm draws a skeleton of a human body by marking all the joint of a body and connects all the joints which give a stick diagram known as the skeleton of a body. Coordinates and the angles made by the joints can be extracted using this algorithm and then used those angles as features for machine learning models. Several machine learning models has been used to calculate the test accuracy of the model. Random Forest classifier gives the best accuracy among all the models.

In this paper, a system is suggested that classify ten yoga poses and the dataset upholds on six classification models of machine learning. The yoga pose is detected based on the angles extracted from the Skeleton joints of TF pose estimation algorithm. 94.28% accuracy altogether was attained of all machine learning models. The data preprocessing and model training was done on Google Colab and Ubuntu 18.04.4 LTS terminal. Future ideas also includes expansion of YOGI dataset on more yoga poses and implement deep learning modules for better performance. In addition to that an audio guidance system will also be included.

Advantages:

TF pose estimation algorithm is used. 94.28% accuracy altogether was attained of all machine learning models.

Disadvantages:

No visual view to guide the user.

3. SYSTEM REQUIREMENT SPECIFICATION

3.1 SOFTWARE REQUIREMENTS

3.1.1 Colab

Google Colaboratory is a free online cloud-based Jupyter notebook environment that allows us to train our machine learning and deep learning models on CPUs, GPUs, and TPUs.

Colaboratory is a Google research project created to help disseminate machine learning education and research. It's a Jupyter notebook environment that requires no setup to use and runs entirely in the cloud.

Colab notebooks allow you to combine executable code and rich text **i**n a single document, along with images**,** HTML**,** LaTeX and more. When you create your own Colab notebooks, they are stored in your Google Drive account. You can easily share your Colab notebooks with co-workers or friends, allowing them to comment on your notebooks or even edit them. To find out more, see Overview of Colab. To create a new Colab notebook you can use the File menu.

Data science with Colab you can harness the full power of popular Python libraries to analyse and visualise data. The code cell below uses numpy to generate some random data, and uses matplotlib to visualise it. To edit the code, just click the cell and start editing.

You can import your own data into Colab notebooks from your Google Drive account, including from spreadsheets, as well as from GitHub and many other sources.

Machine learning with Colab you can import an image dataset, train an image classifier on it, and evaluate the model, all in just a few lines of code. Colab notebooks execute code on Google's cloud servers, meaning you can leverage the power of Google hardware, including GPUs and TPUs, regardless of the power of your machine. All you need is a browser.

Colab is used extensively in the machine learning community with applications including:

- Getting started with TensorFlow
- Developing and training neural networks
- Experimenting with TPUs
- Disseminating AI research
- Creating tutorials

3.1.2 Python

Python is an interpreter, high-level and general-purpose programming language. Python's design philosophy emphasizes code readability with its notable use of significant whitespace. Its language constructs and objectoriented approach aim to help programmers write clear, logical code for small and large-scale projects.

Python is dynamically typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly, procedural), object-oriented, and functional programming. Python is often described as a "batteries included" language due to its comprehensive standard library.

Key advantages of learning Python

- Python is Interactive − You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.
- Python is Object-Oriented − Python supports Object-Oriented style or technique of programming that encapsulates code within objects.
- Python is a Beginner's Language − Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

Characteristics of Python

- Following are important characteristics of Python Programming –
- It can be used as a scripting language or can be compiled to bytecode for building large applications.
- It provides very high-level dynamic data types and supports dynamic type checking.
- It supports automatic garbage collection.
- It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.

Applications of Python:

- Easy-to-learn − Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.
- Easy-to-read − Python code is more clearly defined and visible to the eyes.
- Easy-to-maintain − Python's source code is fairly easy-to-maintain.
- A broad standard library − Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.
- Interactive Mode − Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.
- Portable − Python can run on a wide variety of hardware platforms and has the same interface on all platforms.

3.1.3 Keras

Keras is one of the leading high-level neural networks APIs. It is written in Python and supports multiple back-end neural network computation engines.

Keras contains numerous implementations of commonly used neural-network building blocks such as layers, objectives, activation functions, optimizers, and a host of tools to make working with image and text data easier to simplify the coding necessary for writing deep neural network code.

Keras was created to be user friendly, modular, easy to extend, and to work with Python. The API was “designed for human beings, not machines,” and “follows best practices for reducing cognitive load.”

Neural layers, cost functions, optimizers, initialization schemes, activation functions, and regularization schemes are all standalone modules that you can combine to create new models. New modules are simple to add, as new classes and functions. Models are defined in Python code, not separate model configuration files.

The Model is the core Keras data structure. There are two main types of models available in Keras: the Sequential model, and the Model class used with the functional API.

3.1.4 OpenCV

OpenCV (Open Source Computer Vision Library) is a library of programming functions mainly aimed at real-time computer vision. Originally developed by Intel, it was later supported by Willow Garage then Itseez. The library is cross-platform and free for use under the open-source Apache 2 License. Starting with 2011, OpenCV features GPU acceleration for real-time operations.

In the early days of OpenCV, the goals of the project were described as:

- Advance vision research by providing not only open but also optimized code for basic vision infrastructure.
- No more reinventing the wheel.
- Disseminate vision knowledge by providing a common infrastructure that developers could build on, so that code would be more readily readable and transferable.
- Advance vision-based commercial applications by making portable, performance-optimized code available for free – with a lice.

Applications: OpenCV's application areas include:

- 2D and 3D feature toolkits
- Egomotion estimation
- Facial recognition system

- Gesture recognition
- Human–computer interaction (HCI)
- Mobile robotics
- Motion understanding

3.1.5 Google Text-to-Speech

Python library and CLI tool to interface with Google Translate's text-to-speech API. Writes spoken mp3 data to a file, a file-like object (bytestring) for further audio manipulation, or stdout. It features flexible pre-processing and tokenizing.

Features

- Customizable speech-specific sentence tokenizer that allows for unlimited lengths of text to be read, all while keeping proper intonation, abbreviations, decimals and more.
- Customizable text pre-processors which can, for example, provide pronunciation corrections.

3.2 HARDWARE REQUIREMENTS

Table 3.2: PC/Laptop hardware specifications

Operating System	Windows Operating System
Hard Drive Minimum	Minimum 32 GB; Recommended 64 GB or more
RAM	Minimum 1 GB; Recommended 4 GB or more
Input Speakers	For giving output

4. METHODOLOGY

4.1 SYSTEM ARCHITECTURE

BLOCK DIAGRAM OF SYSTEM

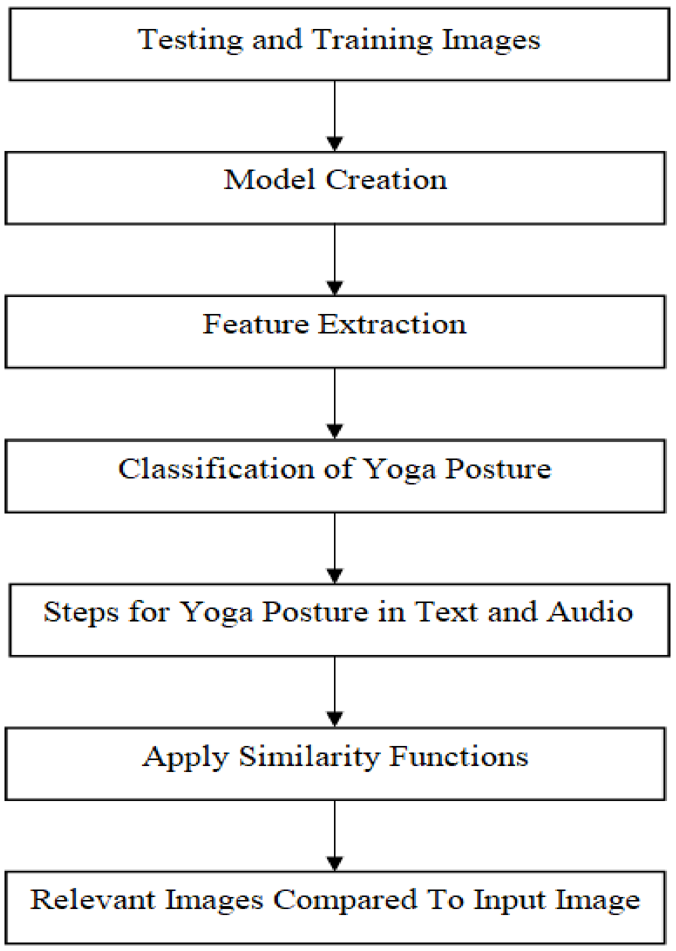

Testing and Training Images
Model Creation
Feature Extraction
Classification of Yoga Posture
Steps for Yoga Posture in Text and Audio
Apply Similarity Functions
Relevant Images Compared To Input Image

Figure 4.1.1: Flow chart of the proposed system

A dataset is created for different number of yoga poses and testing and training of the images is done. We will now preprocess the images using Keras ImageDataGenerator class which will convert the images into an array of vectors that can be fed to the neural network. A set of features or parameters can be initialized to the ImageDataGenerator. These parameters help in extracting maximum features from an image.

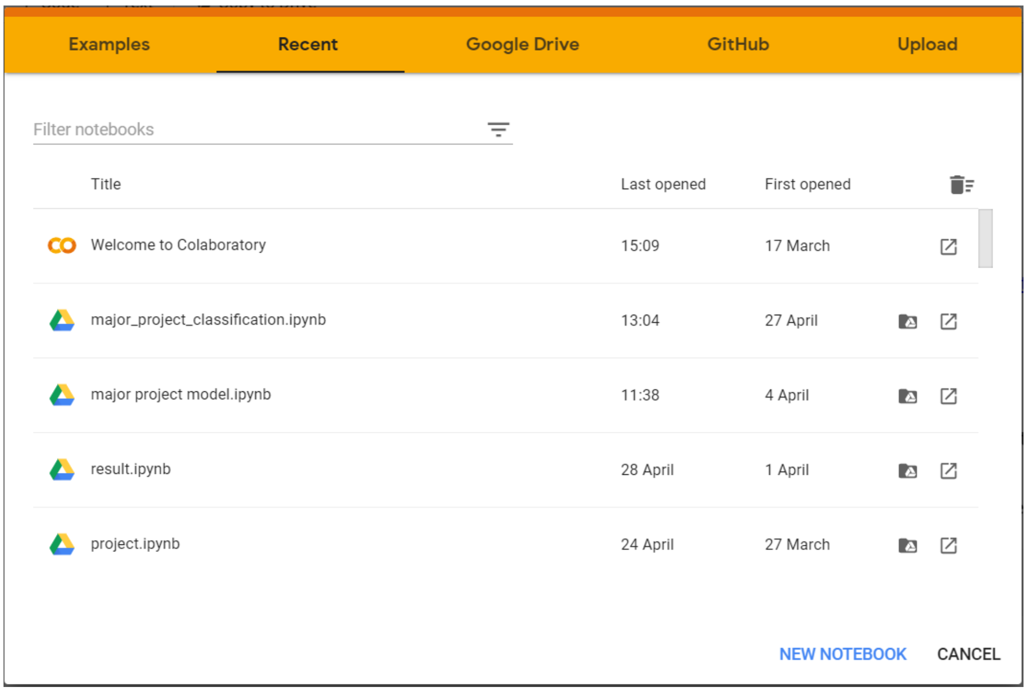
Now we can proceed to build a simple convolution neural network. Keras allows us to build neural networks effortlessly with a couple of classes and methods. The Sequential class initializes a network to which we can add layers and nodes. After compiling the network we get the model.

The model helps us to classify the name of the yoga posture based on the class. According to the class name it also gives the step to be followed to achieve perfect pose in terms of text and audio format.

The similarity functions helps us to predict the most relevant images compared with the user input image from the training set. So, the user gets to know about the name, steps to follow for the yoga pose he/she is doing. The visual view of user input image and relevant images helps to correct them easily.

5. IMPLEMENTATION

Initially, open Google and search Google Colab or use this link. Below is the screen you'll get when you open Colab.



Title	Last opened	First opened
Welcome to Colaboratory	15:09	17 March
major_project_classification.ipynb	13:04	27 April
major project model.ipynb	11:38	4 April
result.ipynb	28 April	1 April
project.ipynb	24 April	27 March

Figure 5.1: To create a notebook in colab

Click on the **NEW NOTEBOOK** button to create a new Colab notebook. You can also upload your local notebook to Colab by clicking the upload button.

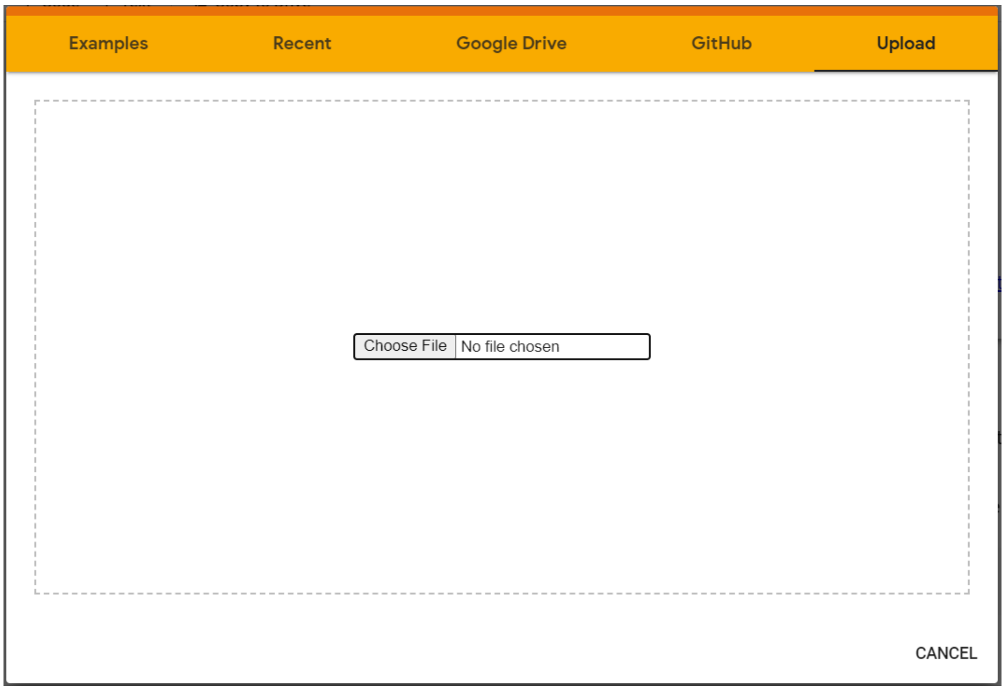


Figure 5.2: To upload a notebook in colab

You can also import your notebook from Google Drive or GitHub, but they require an authentication process.

You can rename your notebook by clicking on the notebook name and change it to anything you want.

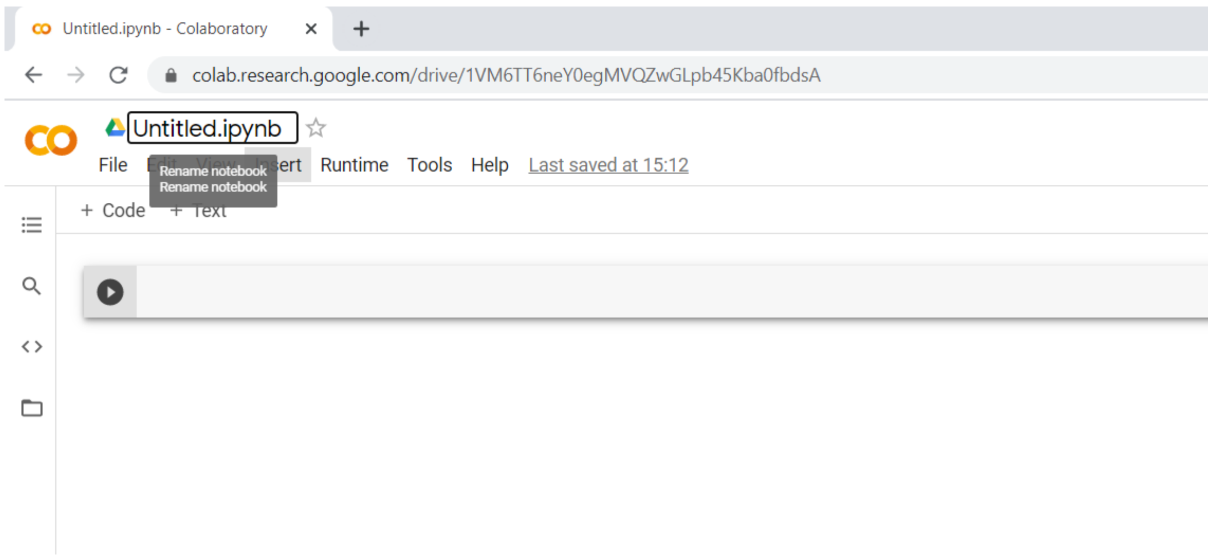
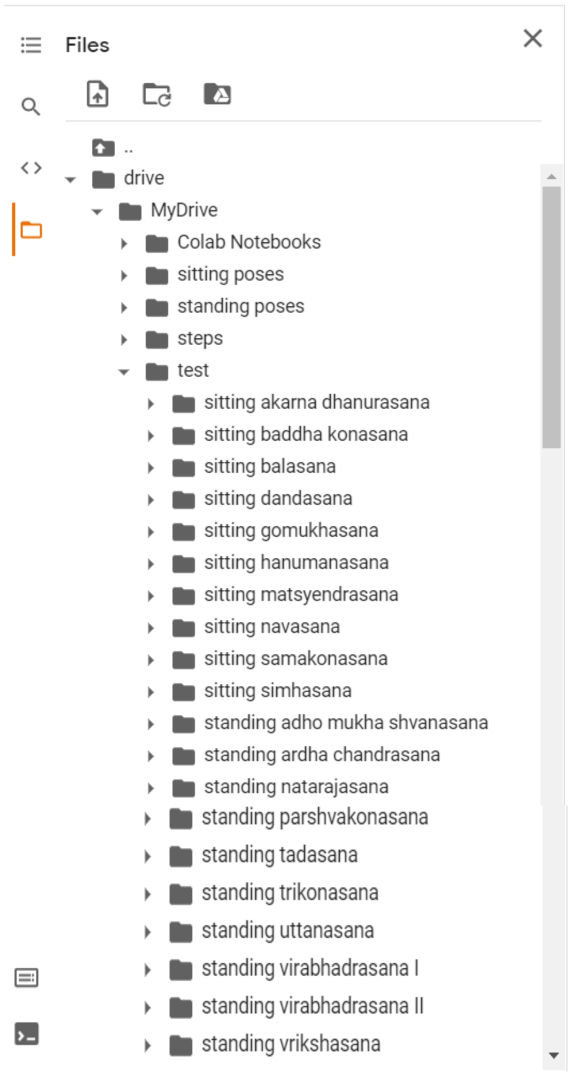

Untitled.ipynb - Colaboratory
colab.research.google.com/drive/1VM6TT6neY0egMVQZwGLpb45Kba0fbdsA
Untitled.ipynb
File
Edit
View
Insert
Runtime
Tools
Help
Last saved at 15:12
Rename notebook
+ Code
+ Text

Figure 5.3: To rename your notebook in colab

We can use the Colab cell for running terminal commands. Most of the popular libraries come installed by default on Google Colab. Python libraries like Pandas, NumPy, scikit-learn are all pre-installed.

Initially collected the images of yoga poses from Google and created dataset. The dataset the splited into training set, testing set and validation set. We have uploaded all the training set, testing set and validation set to Google drive for easy usage of them when required. We are provided with a folders consisting of the images with yoga pose names and along with the respective categories.

We will now preprocess the images using Keras' ImageDataGenerator class which will convert the images into an array of vectors that can be fed to the neural network. A set of features or parameters can be initialized to the ImageDataGenerator. These parameters help in extracting maximum features from an image. We get the information of number of images considered in all the training set, testing set and validation set. We are done processing the image data.



Files

..
drive
MyDrive
Colab Notebooks
sitting poses
standing poses
steps
test
sitting akarna dhanurasana
sitting baddha konasana
sitting balasana
sitting dandasana
sitting gomukhasana
sitting hanumanasana
sitting matsyendrasana
sitting navasana
sitting samakonasana
sitting simhasana
standing adho mukha shvanasana
standing ardha chandrasana
standing natarajasana
standing parshvakonasana
standing tadasana
standing trikonasana
standing uttanasana
standing virabhadrasana I
standing virabhadrasana II
standing vrikshasana

Figure 5.4: List of uploaded folders of dataset

Now we can proceed to build a simple convolutional neural network. Keras allows us to build neural networks effortlessly with a couple of classes and methods. The Sequential class initializes a network to which we can add layers and nodes. The add method allows us to add layers of nodes to the initialized network. We added a Convolutional layer to the network. The convolution will be performed using a 3×3 matrix as specified with the kernel _size parameter. The activation parameter sets the activation function for the nodes. The input size should be same as the size of the outputs from the ImageDataGenerator. We have added a pooling layer to the network. We can add as many layers as we want however, this puts a lot of pressure on the system resources. Choose the layers and nodes based on the capability of the machines. We have added a Flattening layer to the network. Next, we added a hidden layer and an output layer to complete the network.

We will now compile the network to initialize the metrics, loss and weights for the network. The fit method will train the model for a fixed number of epochs. The epochs are the number of times the cycle of training repeats. Steps-per-epoch determines the number of times the weights of each node should be updated for decreasing the loss. Save the model into Google drive for further usage.

We have declared a dictionary containing class number and name of yoga posture. Load the model and then compile it based on required metrics. Take the input of image posture from the user either by uploading it or by providing the link. Load the image with image path and target size. Now, apply array_to_img() function that can be used for converting a NumPy array of pixel data into a PIL image. This can be useful if the pixel data is modified while the image is in array format and can then be saved or viewed. Apply expand_dims() function that expand the shape of an array. Insert a new axis that will appear at the axis position in the expanded array shape. Apply vstack() function makes stack arrays in sequence vertically (row wise). Finally, apply predict_classes() function generates class probability predictions for the input sample.

Our model will predict the labels in the range 0 to 19 based on the above discussed dictionary. Based on the class label the name of the yoga posture is resulted. We created a folder containing the group of multiple folders with the text formatted files of pose name containing the steps to be performed for the respective yoga posture. Based on the class label the step to perform the yoga posture will be shown. Google text to speech is installed to convert the text file content of steps to be performed to output as audio.

Now we wanted to create an image classifier that can tell how similar two images are. For that, there is no need for any complicated libraries like TensorFlow or image classification models. There are two ways to find if an image is similar to another image. First is to look at Mean Square Error (MSE) and the second is Structural Similarity Index (SSIM). MSE can be calculated fairly easily because of NumPy and SSIM is a built in method part of SciKit’s image library so we can just load it up. MSE will calculate the mean square error between each pixels for the two images we are comparing. Whereas, SSIM will do the opposite and look for similarities within pixels; i.e. if the pixels in the two images line up and or have similar pixel density values.

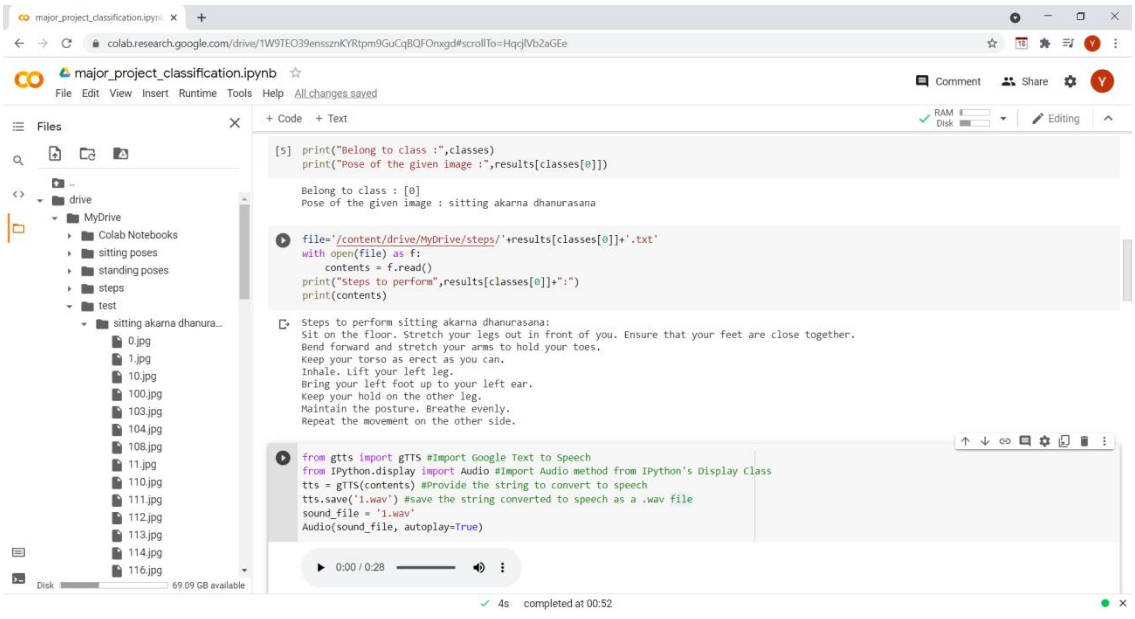
To calculate me Mean Square Error, First we convert the images from unsigned 8-bit integers to floating point, that way we don’t run into any problems with modulus operations “wrapping around”. We then take the difference between the images by subtracting the pixel intensities. Next up, we square these difference (hence mean squared error, and finally sum them up. We are dividing our sum of squares by the total number of pixels in the image. Finally, we return Mean Square Error value.

To calculate Structural Similarity Index, we have imported necessary packages and simply called the function and return the Structural Similarity Index value.

Using for loop we are checking Mean Square Error and Structural Similarity Index of the user input image and all the images of training set with the specific folder of yoga pose to store the values of them in the lists to compare most relevant images based on two functions. Mean Square Error lower the value the better and 0 means the model is perfect and Structural Similarity Index value can vary between -1 and 1, where 1 indicates perfect similarity. Based up on retrieving the index path of the minimum mean square error value or zero value and maximum structural similarity index value. Now using OpenCV, we display the images of the user input image and relevant images of training images obtained from the similarity functions which help user to look at the visual view of the pose he/she performed and the relevant image to correct them.

6. RESULTS

These were the results obtained after taking the input image posture from the user. The below image gives the name of the yoga pose as ‘sitting akarna dhanurasana’ for the user inputed posture based on running with the model. The content of the file which have steps to perform the yoga pose is shown and even the audio file of the content is shown.



```
[5] print("Belong to class :",classes)
    print("Pose of the given image :",results[classes[0]])

    Belong to class : [0]
    Pose of the given image : sitting akarna dhanurasana
```

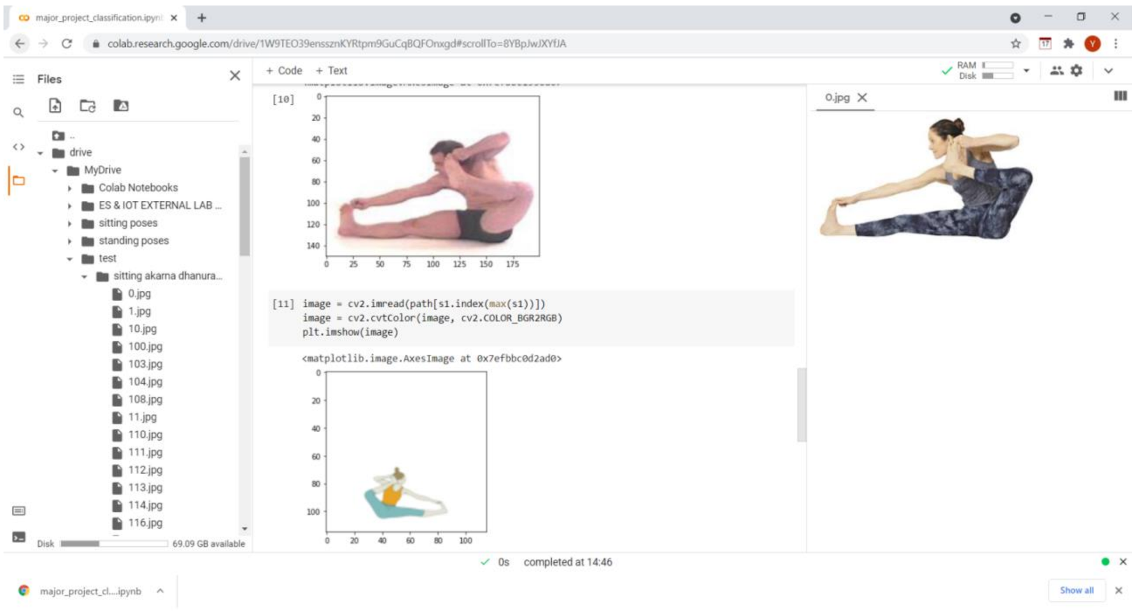
```
file='/content/drive/MyDrive/steps/'+results[classes[0]]+'.txt'
with open(file) as f:
    contents = f.read()
print("Steps to perform",results[classes[0]]+":")
print(contents)
```

```
Steps to perform sitting akarna dhanurasana:
Sit on the floor. Stretch your legs out in front of you. Ensure that your feet are close together.
Bend forward and stretch your arms to hold your toes.
Keep your torso as erect as you can.
Inhale. Lift your left leg.
Bring your left foot up to your left ear.
Keep your hold on the other leg.
Maintain the posture. Breathe evenly.
Repeat the movement on the other side.
```

```
from gtts import gTTS #Import Google Text to Speech
from IPython.display import Audio #Import Audio method from IPython's Display Class
tts = gTTS(contents) #Provide the string to convert to speech
tts.save('1.wav') #save the string converted to speech as a .wav file
sound_file = '1.wav'
Audio(sound_file, autoplay=True)
```

Figure 6.1: Output of yoga pose name along with steps to perform in text and audio format

On the right side is the user input images pose and the middle two are the relevant images of similarity functions from the training set.

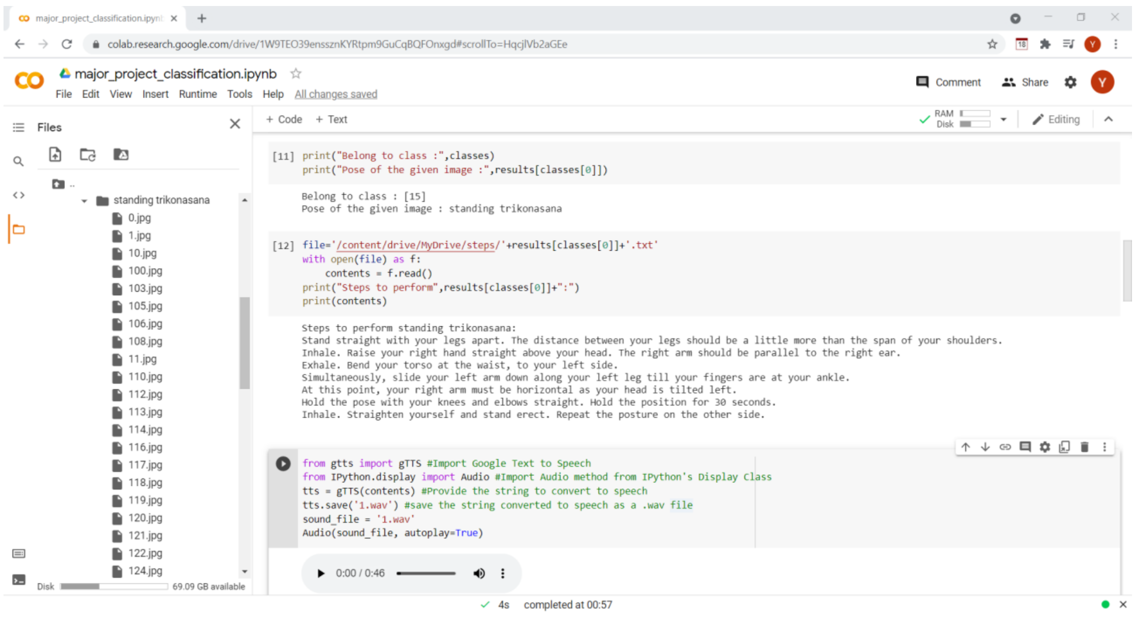


```
[11] image = cv2.imread(path[s1.index(max(s1))])
     image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
     plt.imshow(image)

     <matplotlib.image.AxesImage at 0x7efbbc0d2ad0>
```

colab.research.google.com/drive/1W9TEO39enssznKYRtpm9GuCqBQFOnxgd#scrollTo=8YBpJwJXYfJA
Files
drive
MyDrive
Colab Notebooks
ES & IOT EXTERNAL LAB ...
sitting poses
standing poses
test
sitting akarna dhanura...
[10]
0.jpg
0s completed at 14:46
major_project_cl....ipynb
Show all

Figure 6.2: User input image along with relevant images from training set

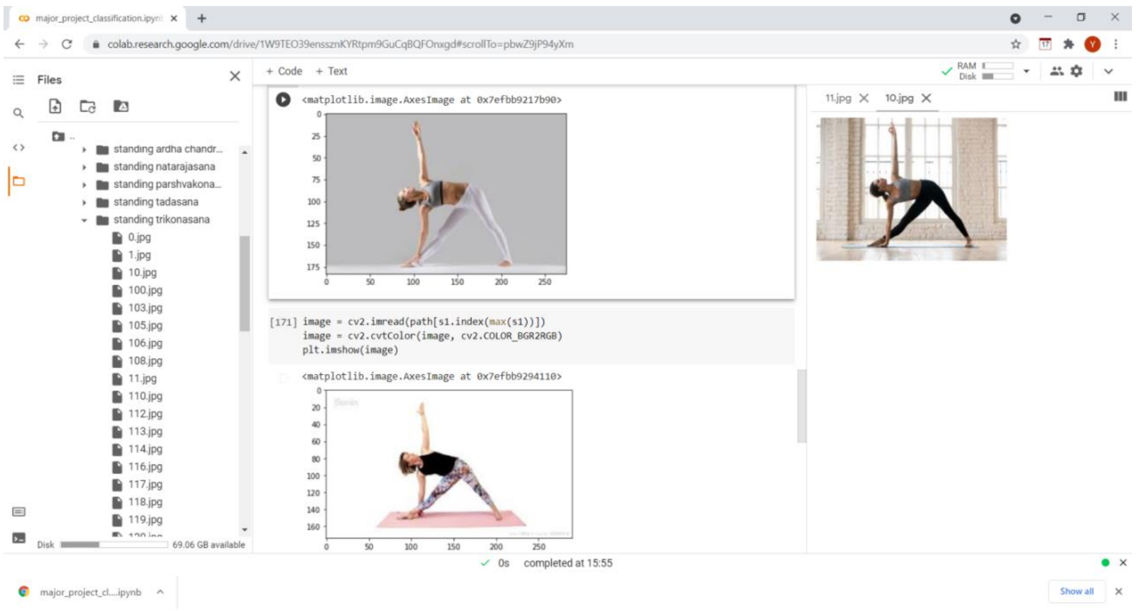
The below image gives the name of the yoga pose as ‘standing trikonasana’ for the user inputed posture based on running with the model. The content of the file which have steps to perform the yoga pose is shown and even the audio file of the content is shown. Which helps to guide for better performing experience.



```
[11] print("Belong to class :",classes)
     print("Pose of the given image :",results[classes[0]])

     Belong to class : [15]
     Pose of the given image : standing trikonasana

[12] file='/content/drive/MyDrive/steps/'+results[classes[0]]+'.txt'
     with open(file) as f:
         contents = f.read()
     print("Steps to perform",results[classes[0]]+":")
     print(contents)

     Steps to perform standing trikonasana:
     Stand straight with your legs apart. The distance between your legs should be a little more than the span of your shoulders.
     Inhale. Raise your right hand straight above your head. The right arm should be parallel to the right ear.
     Exhale. Bend your torso at the waist, to your left side.
     Simultaneously, slide your left arm down along your left leg till your fingers are at your ankle.
     At this point, your right arm must be horizontal as your head is tilted left.
     Hold the pose with your knees and elbows straight. Hold the position for 30 seconds.
     Inhale. Straighten yourself and stand erect. Repeat the posture on the other side.

     from gtts import gTTS #Import Google Text to Speech
     from IPython.display import Audio #Import Audio method from IPython's Display Class
     tts = gTTS(contents) #Provide the string to convert to speech
     tts.save('1.wav') #save the string converted to speech as a .wav file
     sound_file = '1.wav'
     Audio(sound_file, autoplay=True)

     0:00 / 0:46

4s   completed at 00:57
```

Figure 6.3: Output of yoga pose name along with steps to perform in text and audio format

On the right side is the user input images pose and the middle two are the relevant images of similarity functions from the training set. Which helps to make adjustments by looking at the visual views.



```
<matplotlib.image.AxesImage at 0x7efbb9217b90>

[171] image = cv2.imread(path[s1.index(max(s1))])
      image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
      plt.imshow(image)

<matplotlib.image.AxesImage at 0x7efbb9294110>

0s   completed at 15:55
```

Figure 6.4: User input image along with relevant images from training set

7. CONCLUSION & FUTURE SCOPE

During these long months of lockdown, it's difficult for people to go out to exercise, run, and do things they can to stay fit. Yoga is something that can be done indoors easily and has a lot of benefits. But it can be tricky to get any yoga pose correctly if a person doesn't do yoga regularly. With the help of yoga posture detection, people can not only classify the pose but also check how well their posture as compared to a perfect yoga poses.

In this project, we train a model which helps to classify the name of the yoga posture. We take the input of image posture from the user and output the name of pose and give the text of steps to be performed for the yoga posture and voice for the yoga posture steps. Finally we give the visual view of his/her input posture and the relevant output posture from the training set.

Future scope can be Adding real-time prediction feature, so that person can perform yoga in front of the webcam and yoga posture detection can classify the pose in real-time. It can also be extended by adding scoring features to tell how well your yoga posture is comparing the perfect pose with the user's pose will help the user to improve by correcting their posture.

BIBLIOGRAPHY

[1] Muhammad Usama Islam, Hasan Mahmud, Faisal Bin Ashraf, Iqbal Hossain, Md. Kamrul Hasan “Yoga posture recognition by detecting human joint points in real time using Microsoft Kinect.” IEEE Region 10 Humanitarian Technology Conference (R10-HTC). pp.1-5, 2017.

[2] Hua-Tsung Chen & Yu-Zhen He & Chun-Chieh Hsu. “Computer-assisted yoga training system.” Springer, 2018.

[3] Pullen, Paula, and William Seffens. “Machine learning gesture analysis of yoga for exergame development.” IET Cyber-Physical Systems: Theory Applications, vol.3, no.2, pp.106-110, 2018.

[4] Pradchaya Anantamek , Narit Hnoohom. “Recognition of yoga poses using EMG signals from lower limb muscles.”, IEEE, 2019.

[5] Yash Agrawal, Yash Shah, Abhishek Sharma. “Implementation of Machine Learning Technique for Identification of Yoga Poses.”, IEEE, 2020.