YOGAI

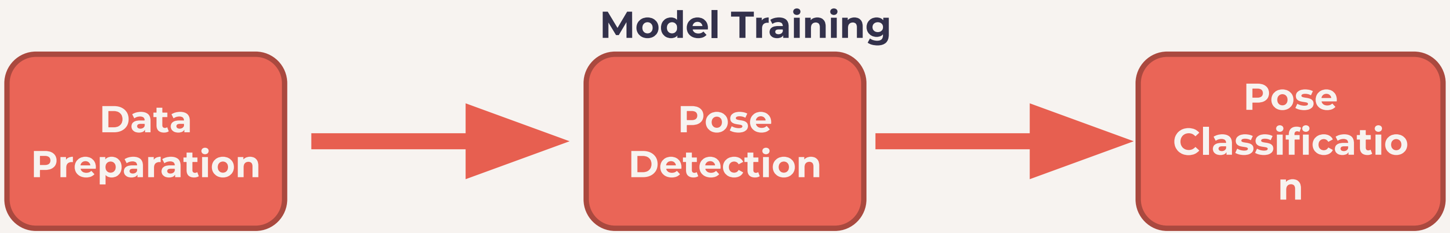
YOGAI

Where health is a priority!

Problem Statement

To build an application that aims to make use of Artificial Intelligence to create an e-yoga instructor that detects correct yoga poses and helps improve the posture for its users.

Stage (1/3)



Model Training

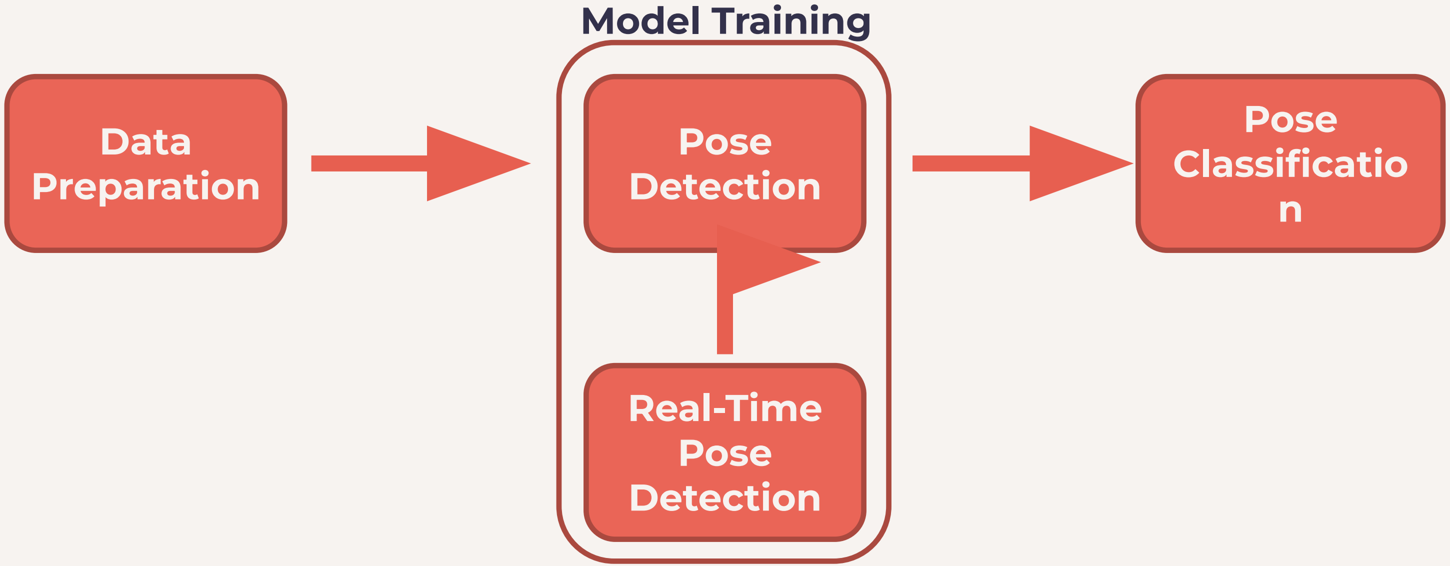
Data Preparation

Pose Detection

Pose Classificatio n

Yoga Pose Classification through Images

Stage (2/3)



Yoga Pose Classification through Realtime WebCam

Stage (3/3)

1. Pose Detection on Live Video
2. Web Application Development
3. Adding more poses..

https://indatalabs.com/wp-content/uploads/2020/05/pose-estimation-project-t.png

Software and Libraries

python

Jupyter

MediaPipe

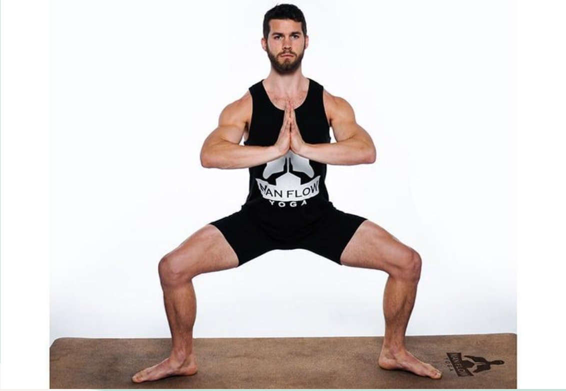
OpenCV

Implementation

Dataset:- https://www.kaggle.com/datasets/niharika41298/yoga-poses-dataset

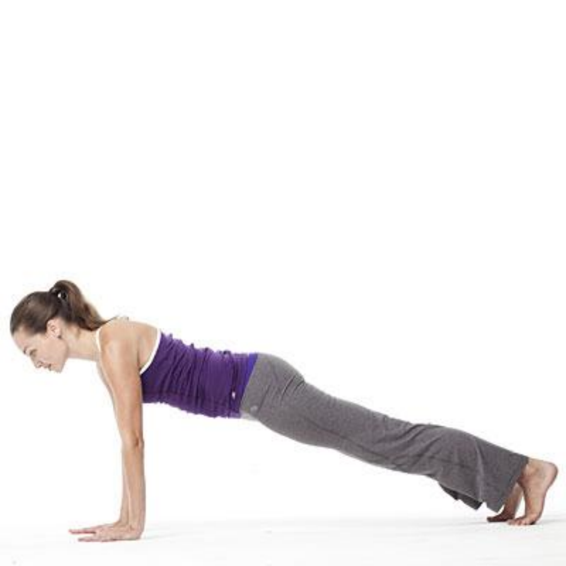
Poses in the Dataset

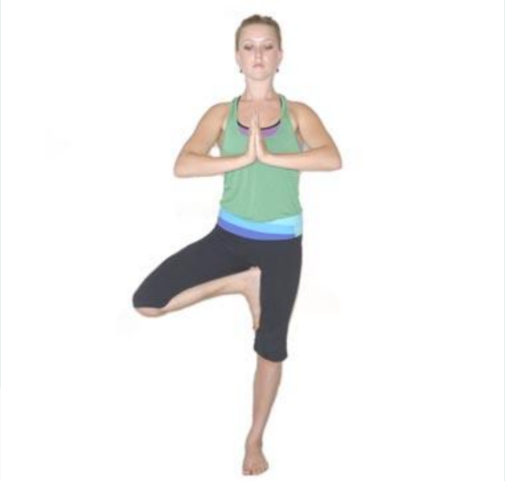
Downdog Pose



Goddess pose

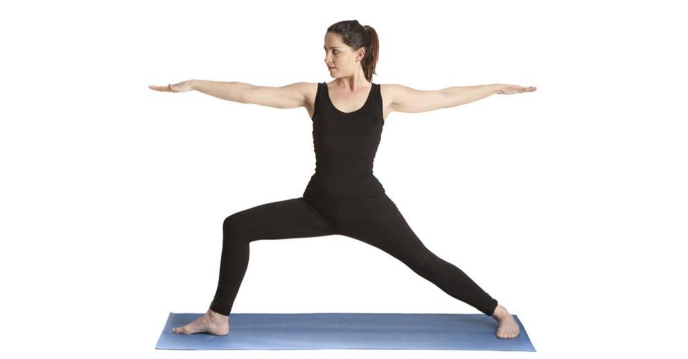
Plank pose



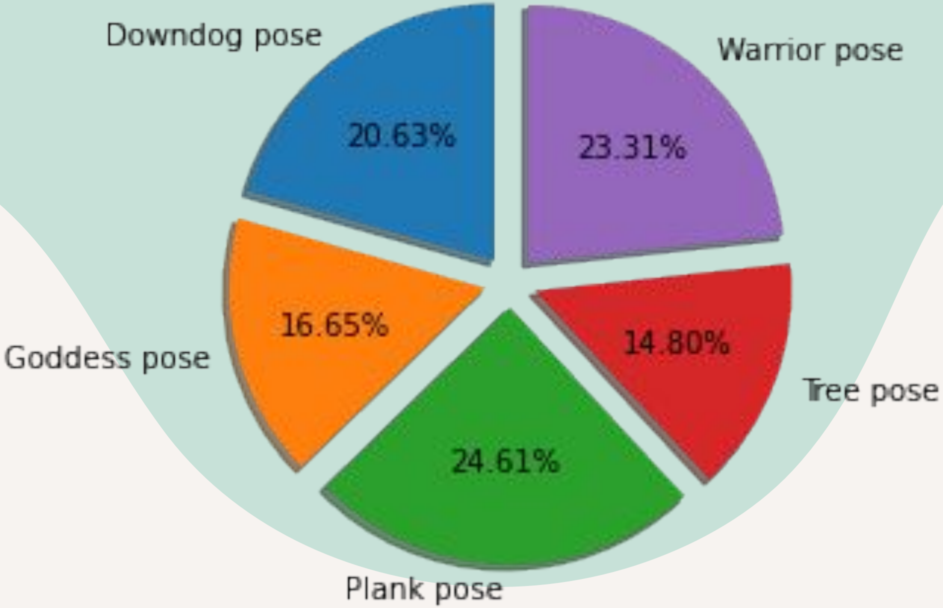


Tree pose

Warrior pose



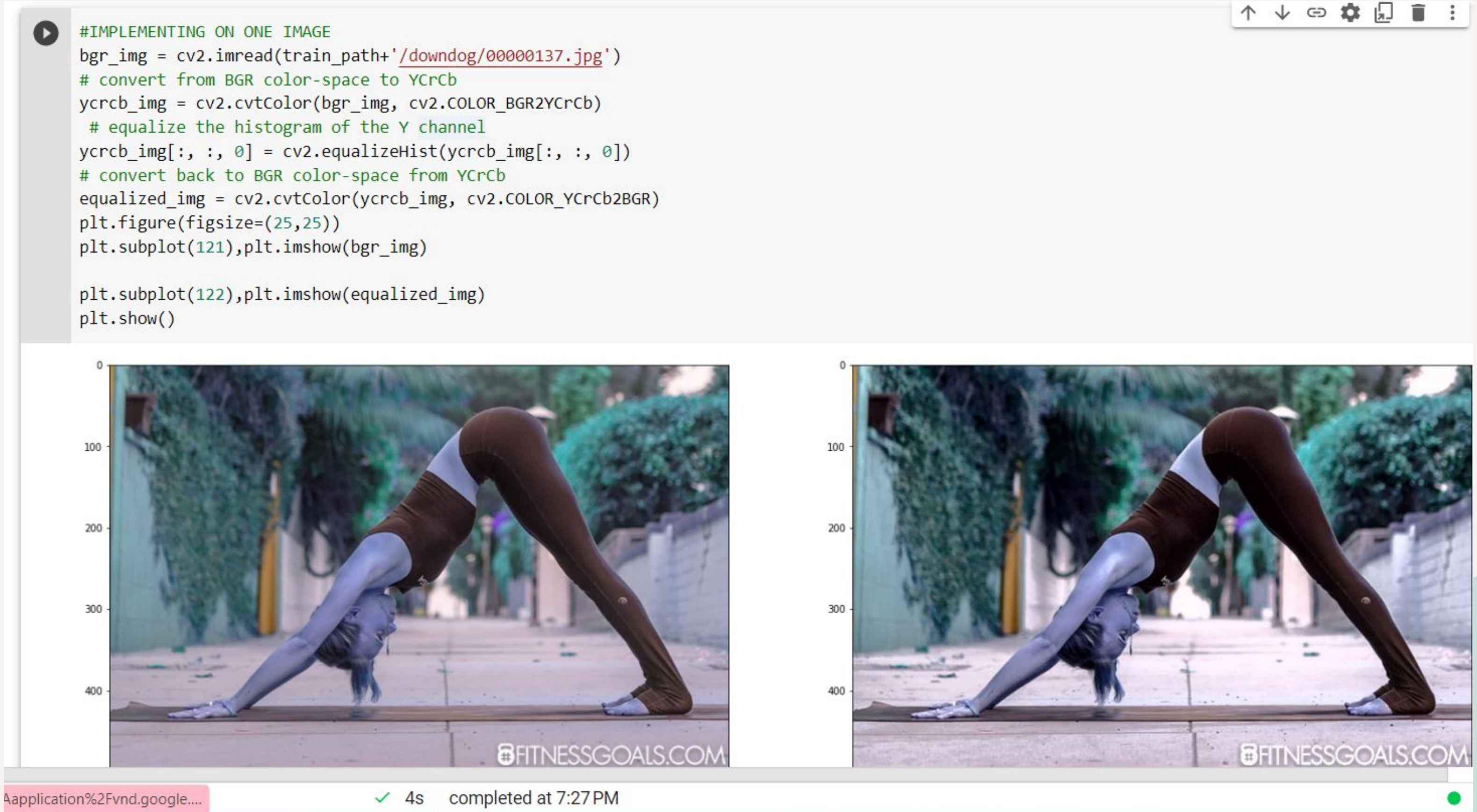
Data distribution in dataset : -
Total 1081 records



DATA PREPROCESSING

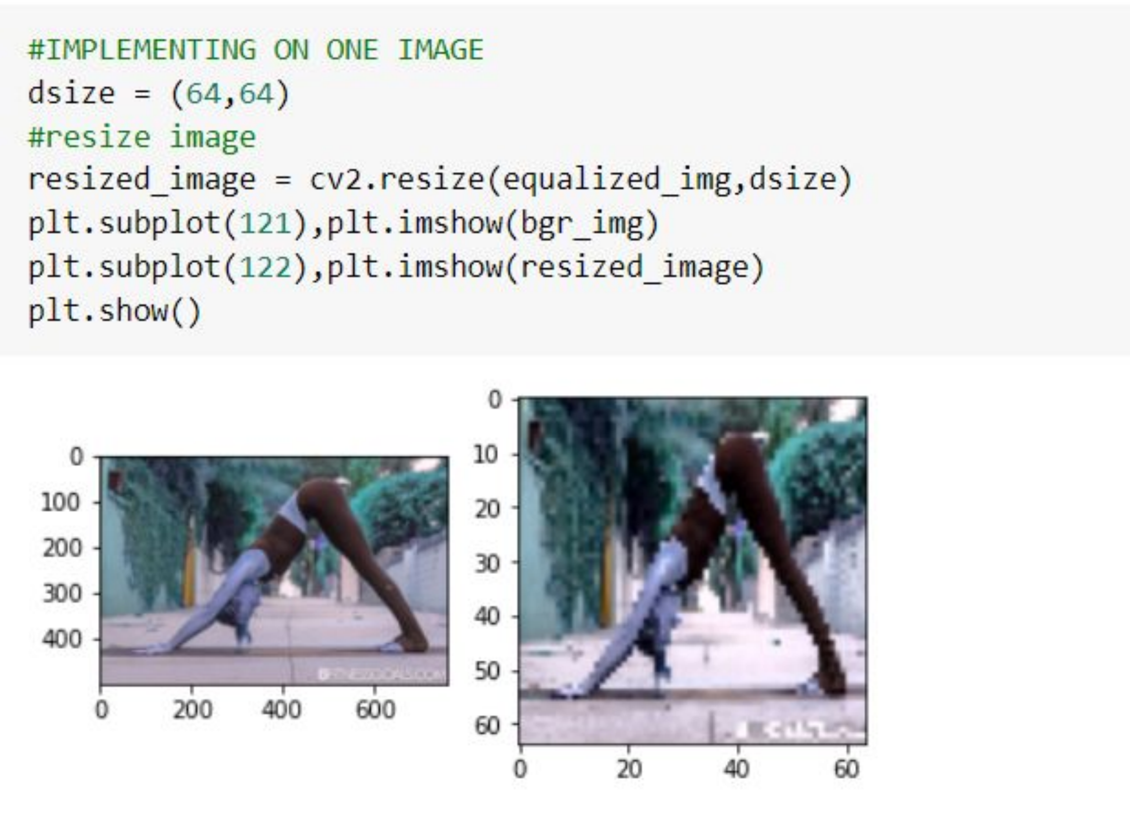
HISTOGRAM EQUALIZATION

Histogram Equalization is used for improving the contrast of any image, that is- to make the dark portion darker and the bright portion brighter.



```
#IMPLEMENTING ON ONE IMAGE
bgr_img = cv2.imread(train_path+'/downdog/00000137.jpg')
# convert from BGR color-space to YCrCb
ycrcb_img = cv2.cvtColor(bgr_img, cv2.COLOR_BGR2YCrCb)
 # equalize the histogram of the Y channel
ycrcb_img[:, :, 0] = cv2.equalizeHist(ycrcb_img[:, :, 0])
# convert back to BGR color-space from YCrCb
equalized_img = cv2.cvtColor(ycrcb_img, cv2.COLOR_YCrCb2BGR)
plt.figure(figsize=(25,25))
plt.subplot(121),plt.imshow(bgr_img)

plt.subplot(122),plt.imshow(equalized_img)
plt.show()
```

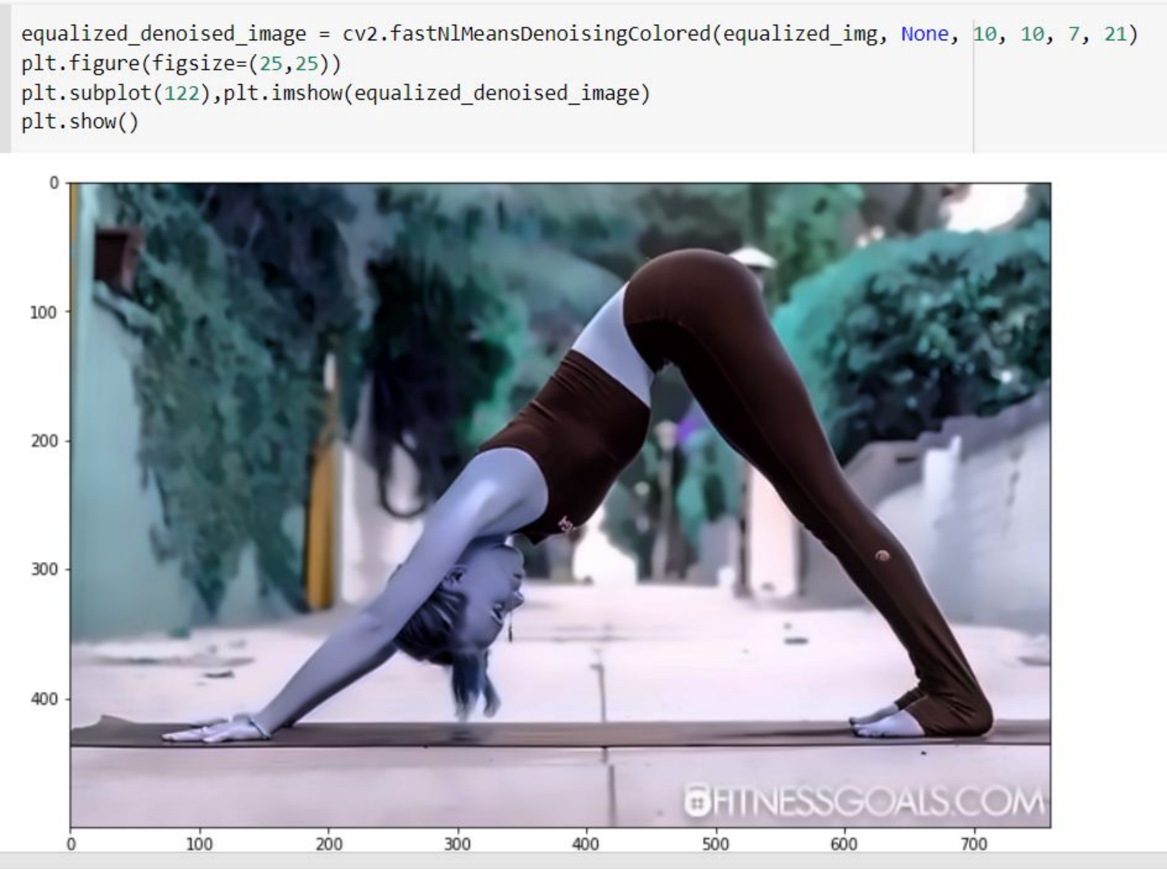
RESIZING



```
#IMPLEMENTING ON ONE IMAGE
dsize = (64,64)
#resize image
resized_image = cv2.resize(equalized_img,dsize)
plt.subplot(121),plt.imshow(bgr_img)
plt.subplot(122),plt.imshow(resized_image)
plt.show()
```

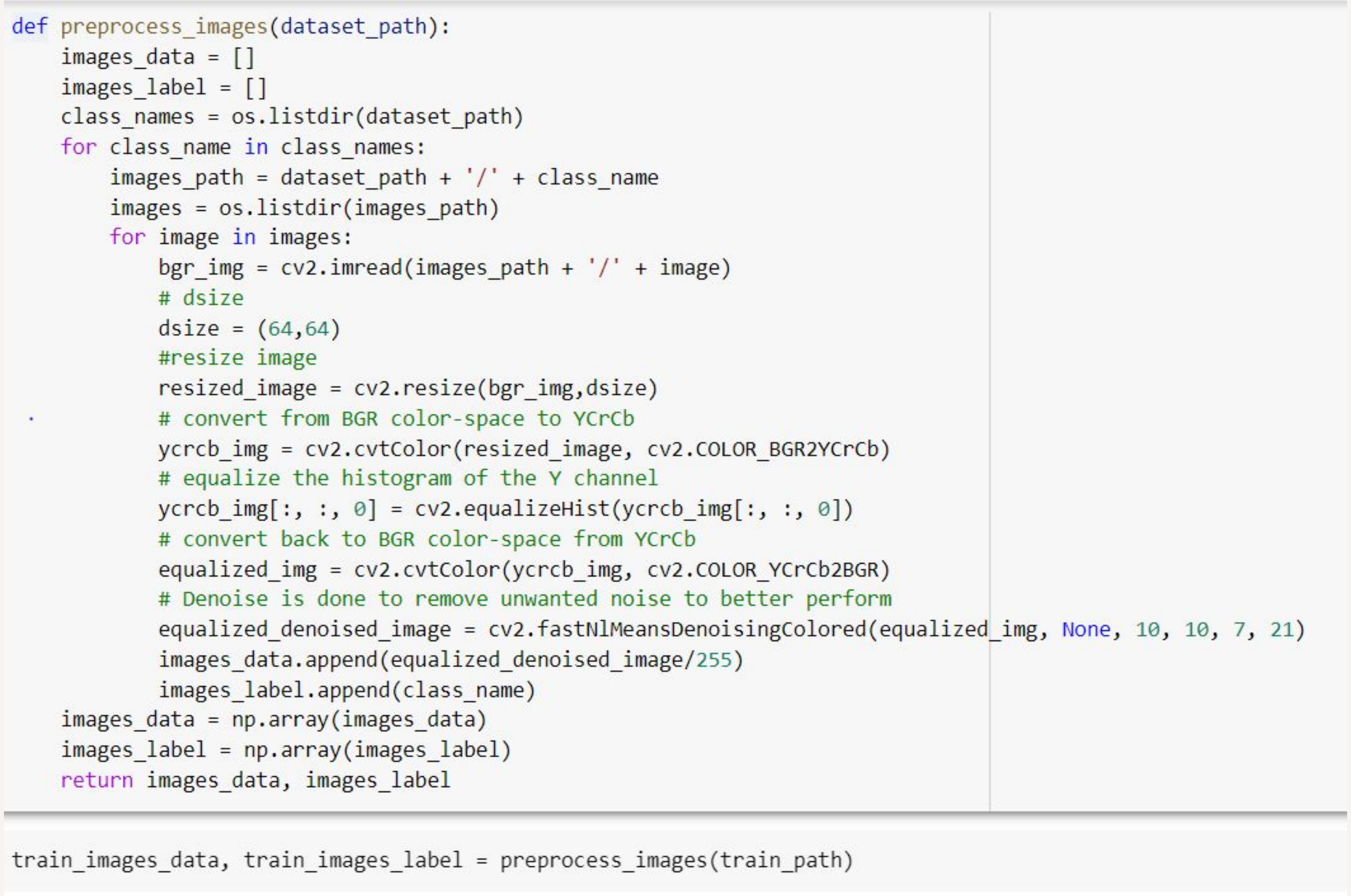
DENOISING

Denoise is done to remove unwanted noise so that model performs better



```
equalized_denoised_image = cv2.fastNlMeansDenoisingColored(equalized_img, None, 10, 10, 7, 21)
plt.figure(figsize=(25,25))
plt.subplot(122),plt.imshow(equalized_denoised_image)
plt.show()
```

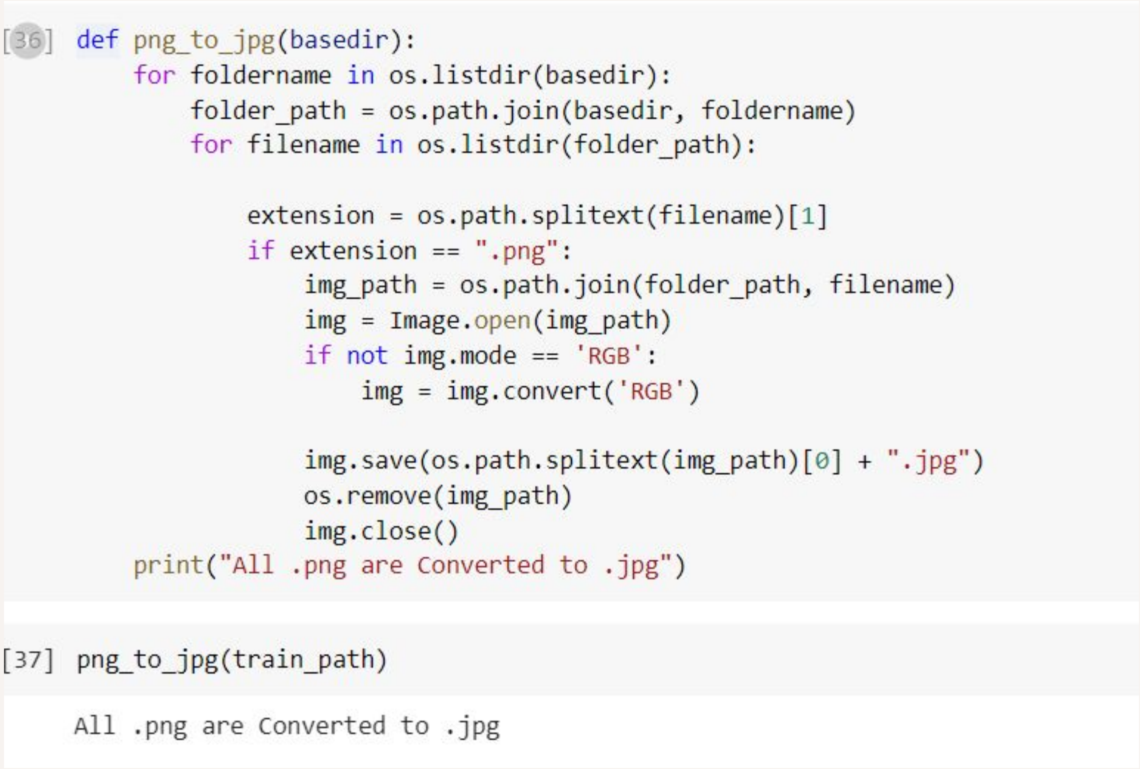
APPLYING TO ALL IMAGES



```
def preprocess_images(dataset_path):
    images_data = []
    images_label = []
    class_names = os.listdir(dataset_path)
    for class_name in class_names:
        images_path = dataset_path + '/' + class_name
        images = os.listdir(images_path)
        for image in images:
            bgr_img = cv2.imread(images_path + '/' + image)
            # dsize
            dsize = (64,64)
            #resize image
            resized_image = cv2.resize(bgr_img,dsize)
            # convert from BGR color-space to YCrCb
            ycrcb_img = cv2.cvtColor(resized_image, cv2.COLOR_BGR2YCrCb)
            # equalize the histogram of the Y channel
            ycrcb_img[:, :, 0] = cv2.equalizeHist(ycrcb_img[:, :, 0])
            # convert back to BGR color-space from YCrCb
            equalized_img = cv2.cvtColor(ycrcb_img, cv2.COLOR_YCrCb2BGR)
            # Denoise is done to remove unwanted noise to better perform
            equalized_denoised_image = cv2.fastNlMeansDenoisingColored(equalized_img, None, 10, 10, 7, 21)
            images_data.append(equalized_denoised_image/255)
            images_label.append(class_name)
    images_data = np.array(images_data)
    images_label = np.array(images_label)
    return images_data, images_label
```

```
train_images_data, train_images_label = preprocess_images(train_path)
```

Converting .png images to jpg

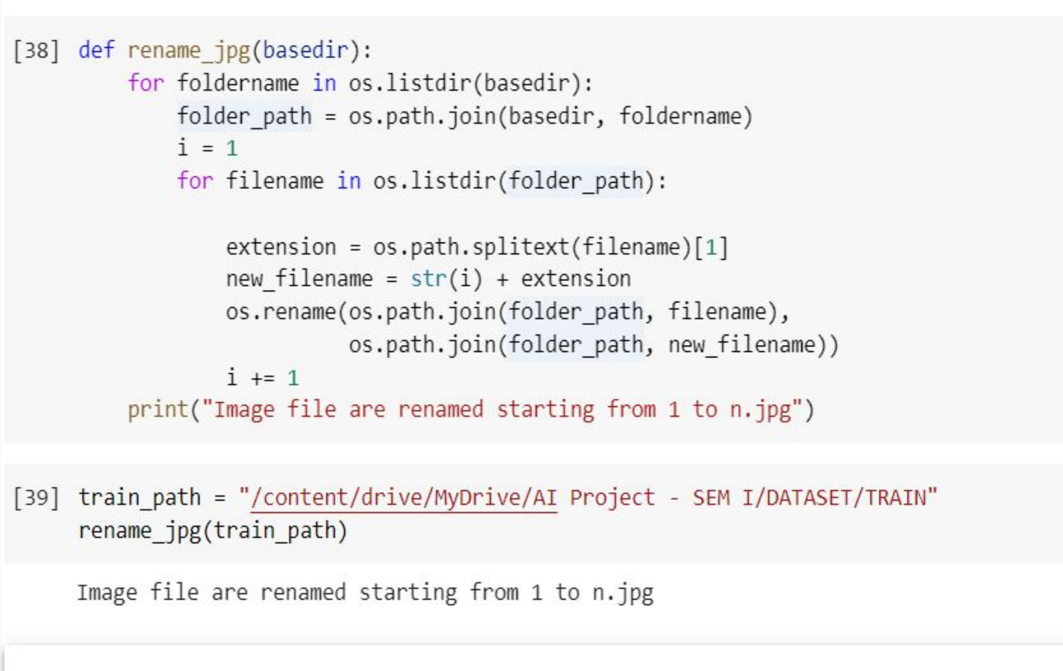


```
[36] def png_to_jpg(basedir):
         for foldername in os.listdir(basedir):
             folder_path = os.path.join(basedir, foldername)
             for filename in os.listdir(folder_path):

                 extension = os.path.splitext(filename)[1]
                 if extension == ".png":
                     img_path = os.path.join(folder_path, filename)
                     img = Image.open(img_path)
                     if not img.mode == 'RGB':
                         img = img.convert('RGB')

                     img.save(os.path.splitext(img_path)[0] + ".jpg")
                     os.remove(img_path)
                     img.close()
         print("All .png are Converted to .jpg")
```

```
[37] png_to_jpg(train_path)
```

```
All .png are Converted to .jpg
```

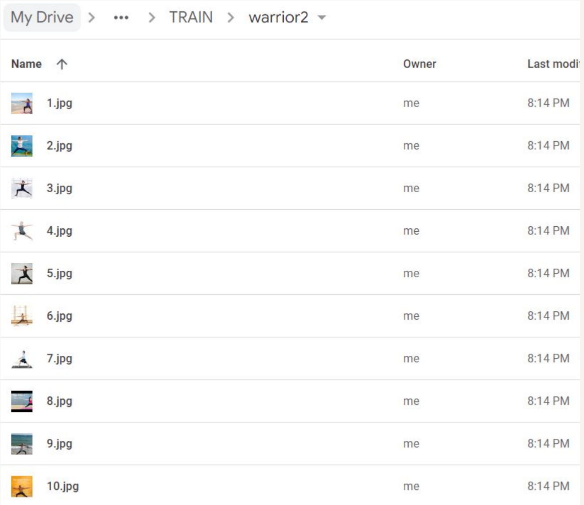
Renaming .jpg files from 1 to n



```
[38] def rename_jpg(basedir):
         for foldername in os.listdir(basedir):
             folder_path = os.path.join(basedir, foldername)
             i = 1
             for filename in os.listdir(folder_path):

                 extension = os.path.splitext(filename)[1]
                 new_filename = str(i) + extension
                 os.rename(os.path.join(folder_path, filename),
                           os.path.join(folder_path, new_filename))
                 i += 1
         print("Image file are renamed starting from 1 to n.jpg")
```

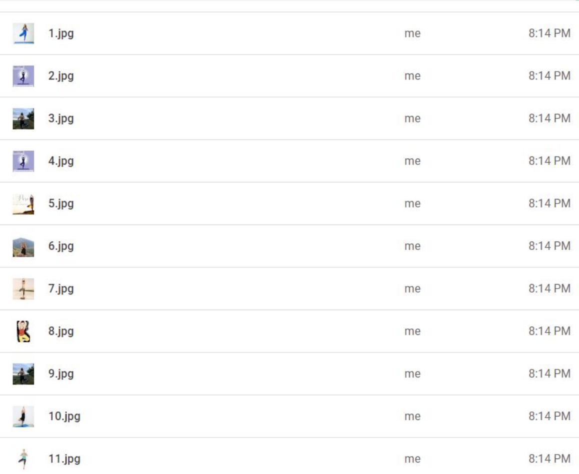
```
[39] train_path = "/content/drive/MyDrive/AI Project - SEM I/DATASET/TRAIN"
     rename_jpg(train_path)
```

```
     Image file are renamed starting from 1 to n.jpg
```



My Drive > ••• > TRAIN > warrior2

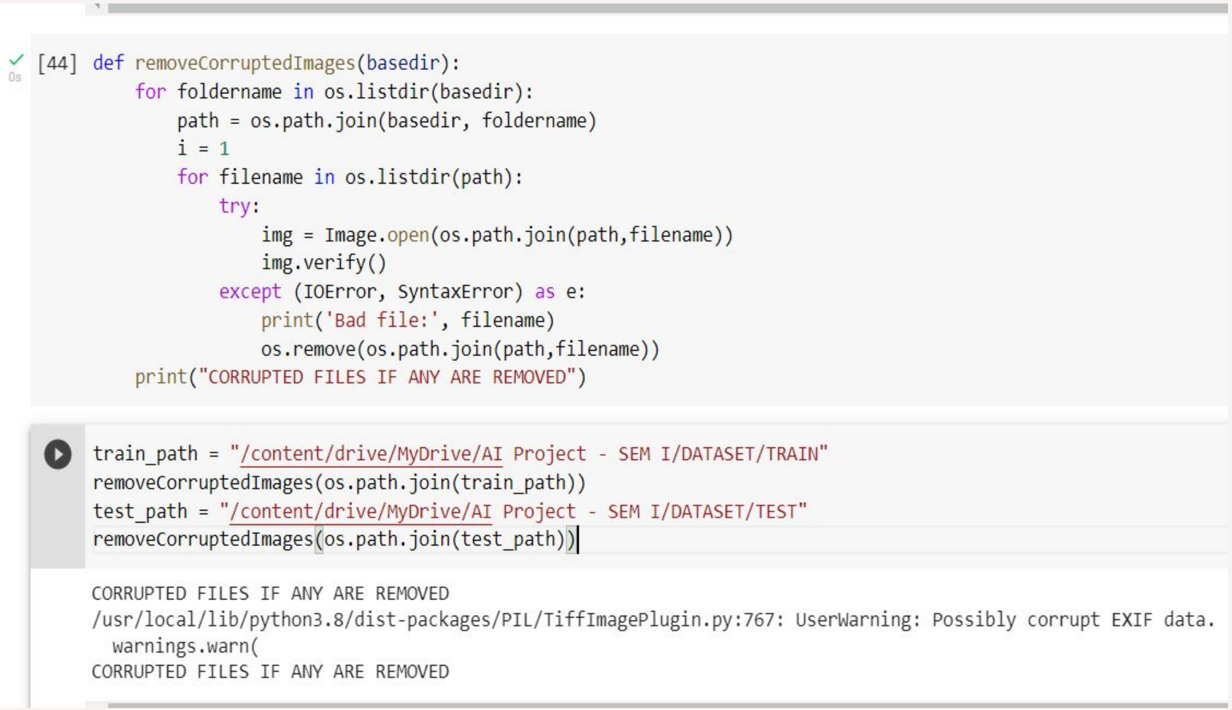
Name ↑	Owner	Last modi
1.jpg	me	8:14 PM
2.jpg	me	8:14 PM
3.jpg	me	8:14 PM
4.jpg	me	8:14 PM
5.jpg	me	8:14 PM
6.jpg	me	8:14 PM
7.jpg	me	8:14 PM
8.jpg	me	8:14 PM
9.jpg	me	8:14 PM
10.jpg	me	8:14 PM



1.jpg	me	8:14 PM
2.jpg	me	8:14 PM
3.jpg	me	8:14 PM
4.jpg	me	8:14 PM
5.jpg	me	8:14 PM
6.jpg	me	8:14 PM
7.jpg	me	8:14 PM
8.jpg	me	8:14 PM
9.jpg	me	8:14 PM
10.jpg	me	8:14 PM
11.jpg	me	8:14 PM

REMOVING CORRUPTED FILES

It is necessary to remove to corrupted images to reduce training error



```
def removeCorruptedImages(basedir):
    for foldername in os.listdir(basedir):
        path = os.path.join(basedir, foldername)
        i = 1
        for filename in os.listdir(path):
            try:
                img = Image.open(os.path.join(path,filename))
                img.verify()
            except (IOError, SyntaxError) as e:
                print('Bad file:', filename)
                os.remove(os.path.join(path,filename))
    print("CORRUPTED FILES IF ANY ARE REMOVED")
```

```
train_path = "/content/drive/MyDrive/AI Project - SEM I/DATASET/TRAIN"
removeCorruptedImages(os.path.join(train_path))
test_path = "/content/drive/MyDrive/AI Project - SEM I/DATASET/TEST"
removeCorruptedImages(os.path.join(test_path))
```

```
CORRUPTED FILES IF ANY ARE REMOVED
/usr/local/lib/python3.8/dist-packages/PIL/TiffImagePlugin.py:767: UserWarning: Possibly corrupt EXIF data.
  warnings.warn(
CORRUPTED FILES IF ANY ARE REMOVED
```

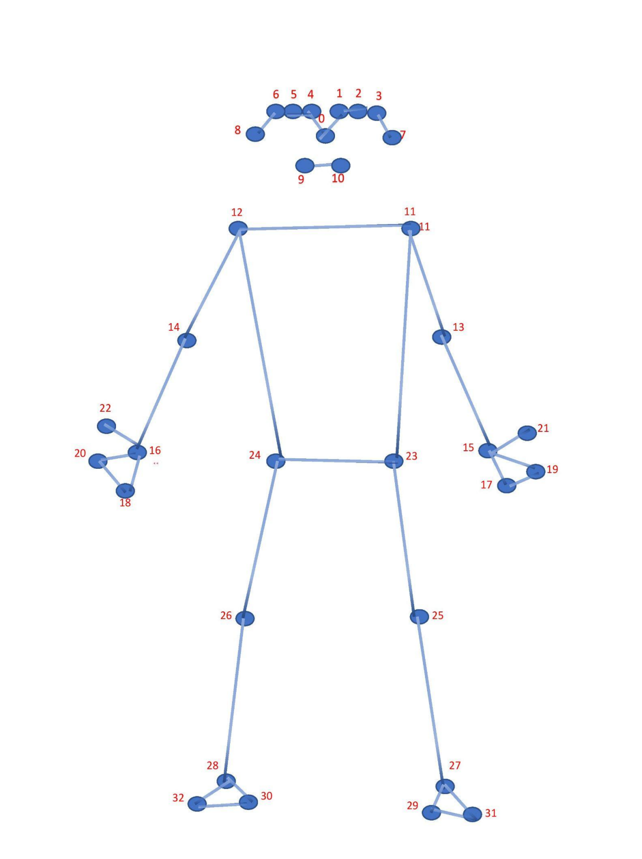
IMPLEMENTATION

Website :- https://ai-project-four.netlify.app/

Thankyou!

Akanksha Goel
Akshita Gupta
Ayushi Sood
Rachit Mittal

METHODOLOGY



FEATURE EXTRACTION

Initially there are ***33*** key points that the model will detects using MediaPipe Library.

0.nose

1. left_eye_inner
2. left eye
3. left eye_outer
4. right_eye_inner
5. right eye
6. right_eye_outer

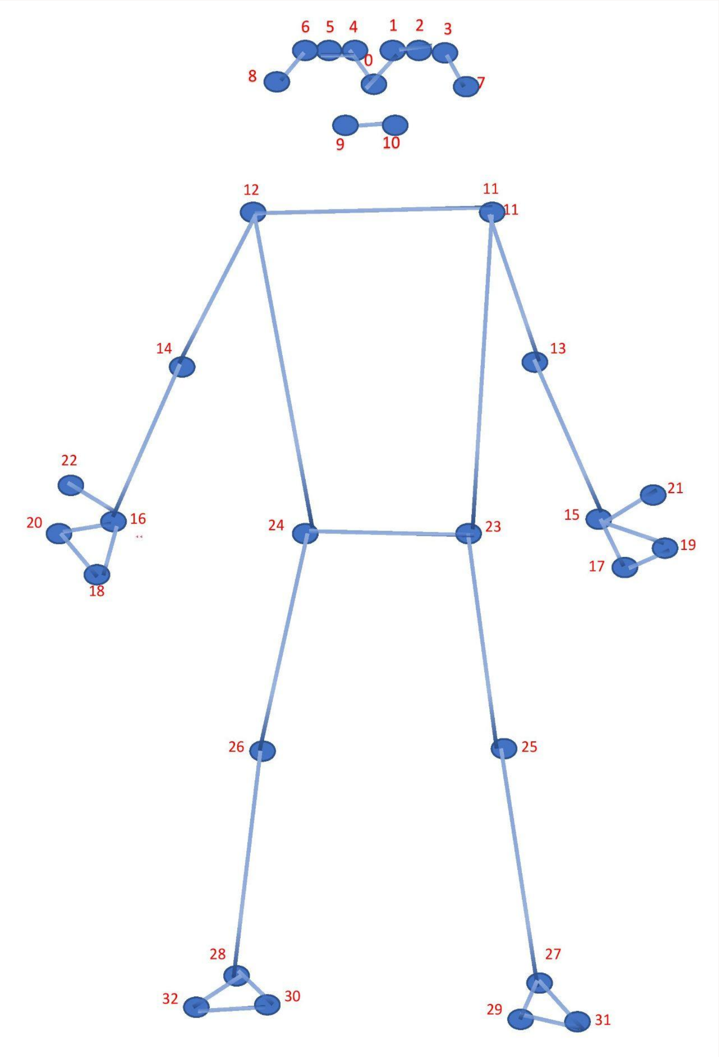
7. left_ear
8. right_ear

9. mouth_left
10. mouth_right

11. left_shoulder
12. right_shoulder

13. left elbow
14. right elbow

15. left wrist
16 right_wrist



0.nose

1. left_eye_inner
2. left eye
3. left eye_outer
4. right_eye_inner
5. right eye
6. right_eye_outer

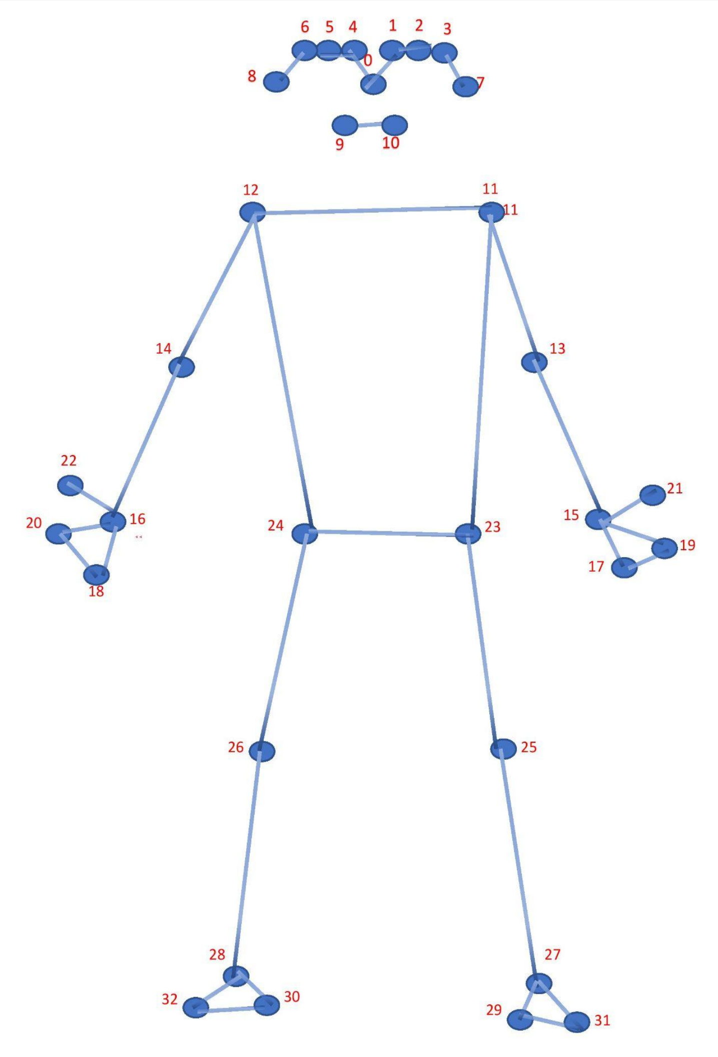
7. left_ear
8. right_ear

9. mouth_left
10. mouth_right

11. left_shoulder
12. right_shoulder

13. left elbow
14. right elbow

15. left wrist
16 right_wrist



17. left pinky
18. right pinky

19. left_index
20.right_index

21.left_thumb
22.right_thumb

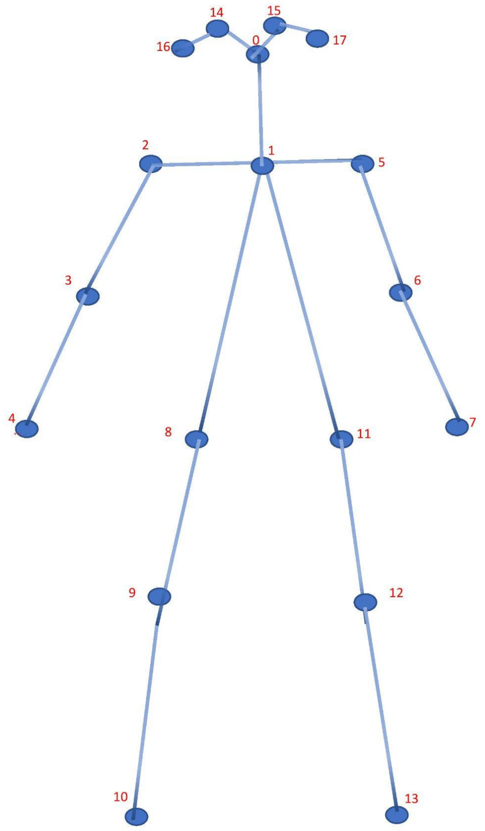
23. left hip
24.right hip

25.left knee
26. right knee

27.left ankle
28.right ankle

29. left heel
30.right heel

31. left foot_index
32 right foot index



Now,the key points are reduced to *18*.

CLASSIFICATION

Some key points will not contribute for classification.
With these 18 joint vectors are calculated with different confidence intervals,
Further,

For all of these key points angle between them and ground are estimated ; then depending on their ranges different yoga poses will be classified.