**Problem Statement and Requirements:**

**Static Vs Dynamic Input :**

* Static input - images/ frames of video (CNN)
* ~~Dynamic input - live video and live output~~

For now static input

**Whether Pose name is provided by the user?**

~~Yes: Input = Video, Pose name~~

~~Processing = Using pose estimation we get the position of joints.~~

~~Output = Error between predicted pose and ground truth pose.~~

No : Input = Video

Processing = Model try to classify yoga pose and get

The predicted position of joints

Output = Error between predicted pose and ground truth

Pose of predicted.

**Target device**

~~Mobile Phone - Deploy using Tflite model~~

Backend Server

~~WebApp ????~~

**Pose Detection Model: (Pretrained or from scratch)**

Pretrained - Tensorflow Hub(Movenet), Open Pose, e.t.c

**\*\*\*(Most Important)Method for judging Pose:**

After getting key points from the model-

1. Predefined Geometric heuristic (R4/R5)
2. ML way?? (R4/R6) - Comparator Net

**Meeting and Progress:**

**Date: 07/12/2021**

1. Analyzed the result of pose correction with the team.
2. We also discussed openpose output and the algorithm it uses.

**Date: 29/11/2021**

1. Tested the dataset with the old comparator net model on the new test dataset.
2. Compare the result between the newly trained model of comparator net and the old comparator net model.

**Date: 25/11/2021**

1. Change the comparator net model to the original model.
2. Created a new small dataset of the same poses which the classification model contains.
3. Passes this dataset through the pipeline and stores the result.

**Date: 24/11/2021**

1. Discuss the result of the new pipeline.
2. Openpose causes a significant error by not detecting the accurate key points.
3. Try to improve the result

**Date: 23/11/2021**

1. Created a new dataset for testing purposes consisting of random yoga poses.
2. Run the pipeline through a new test dataset as all the previous dataset was used for training the comparator net model.

**Date: 22/11/2021**

1. The training was completed with 97.5% accuracy on the test dataset.
2. Integrated the new classification model in the main pipeline.

**Weekend...**

**Date: 19/11/2021**

1. Tried with a different optimizer for training the classification model- Adam.
2. Training going on for the classification model.

**Date: 18/11/2021**

1. Treat different variants of the same pose as different classes.
2. Changed script for training with the new number of classes.
3. New model training causes overfitting as there is a difference of 10% in test and train accuracy.
4. Changed hyperparameter for training.

**Date: 17/11/2021**

1. Discussed result of comparator net.
2. The result of the new comparator net shows better performance than previous shows.
3. Some example results show low similarity scores even when there are little differences like hand posture is different.

**Date: 16/11/2021**

1. Training is completed for 2 different sets of hyperparameters.
2. The test accuracy of the selected model is 96%.
3. Now comparator net similarity scores show a variety of output between 0 and 1.

**Date: 15/11/2021**

1. Disable the downsampling in the training dataset of comparator net.
2. Trained with different hyperparameters of comparator net.

**Weekend**

**Date: 11/11/2021**

1. The main pipeline is completed.
2. Try different hyperparameters for comparator net models for more insightful results.
3. Discussed second approach for judging the pose - Angle method

**Date: 10/11/2021**

1. Saved the classification model for the 99th epoch.
2. All the models are integrated into the main pipeline.

Action:

1. Test the model with different inputs when the colab restriction renews.

**Date: 9/11/2021**

1. The issue with openpose is resolved now. After a series of changes in the build script, it is finally working.
2. Changed CMake version from 3.12 to 3.13
3. CUDA version from 10.0 to 11.1
4. Pose key coordinates are being stored in .npy files for each image.
5. Dataset is created by taking each possible pair and then downsampling the class which is in majority.
6. Created the model for comparator net.
7. Training is being done for 100 epochs. The best validation accuracy is at the 90th epoch.

Action:

1. Create the model in .h5 format with the checkpoint file.
2. Integrate it with the main pipeline.

**Date: 8/11/2021**

1. Exploring different pose estimation models of TensorFlow and media pipe.
2. Trying to resolve the openpose issue parallelly.

**Diwali weekend**

**Date: 3/11/2021:**

1. Tried the previous version of openpose. Still the same issue.

**Date: 2/11/2021**

1. Found a few posts in the Github issue regarding this. It seems to be a bug in the python wrapper of openpose.

**Weekend**

**Date: 29/10/2021**

1. Python wrapper of Openpose giving None output in the function datum object.
2. <https://github.com/CMU-Perceptual-Computing-Lab/openpose/issues/1855>

**Date: 28/10/2021**

1. Preprocessing is completed without error, but the output is faulty.
2. Some error from the openpose end.

**Date:27/10/2021**

1. Preprocessing script is written to save the coordinates of every image in the dataset in .npy format.
2. Google colab limit reaches.

Action:

1. Start the preprocessing on google colab.

**Date:26/10/2021**

1. Manual subclassification is completed for all classes along with filtration of images where only a single object is present.
2. The training script is completed.

Action:

1. Complete preprocessing script to save the coordinates dataset.

**Date:25/10/2021**

1. Manual sub-classification of the total of 5 classes out of 7 classes is completed.
2. These are - cobra, down dog, tree, plank, goddess.

Action:

1. Complete the sub-classification by tomorrow.
2. Complete the training script.

**Date:22/10/2021**

1. Manual sub-classification of cobra pose is completed into 2 subclasses.
2. Written code to preprocess the dataset for comparator net model.

**Date:21/10/2021**

1. Manual fine classification of tree pose is completed into 5 subclasses.
2. Openpose and classification models are merged completely.

**Date:20/10/2021**

1. Further classifying the tree yoga pose into 3 classes.
2. Merged the open pose with the classification model.

Action

1. Write a script to preprocess input and to use openpose python API for coordinates prediction.

**Date:19/10/2021**

1. Discussed using different Pose estimation models like Openpose, Media pipe, e.t.c.
2. Discussed different approaches to judge a user pose (angle method and Neural network method)

Action:

1. Merge the classification model with the OpenPose model.
2. Further, classify the dataset into different yoga variants.

**Date:18/10/2021**

1. The training script is written completely.
2. Using different hyperparameter sets achieved the train classification accuracy of 99.84% and test accuracy of 95.29%

**WEEKEND...**

**Date:14/10/2021**

1. Dataset is doubled by horizontal flip using Keras API.
2. Written the code till preprocessing for training the classification model.
3. Solved the bug during the normalization of the dataset.

Action:

1. Complete the training script and start the training.

**Date:13/10/2021**

1. Used a different method to download original images using google search API. Bing API is only able to download 60-70 relevant images.
2. Created a raw dataset of 5 classes (tree, warrior 2, down dog,

planks, goddess ) and further added 2 classes (cobra, triangle).

Action:

1. Increase the dataset using augmentation techniques(horizontal flip).

**Date:12/10/2021**

1. Images from the chrome extension contain garbage images like low resolution-thumbnail, banners and other non-image files.
2. Dataset for 2 classes is complete.
3. Trying the python api of google and bing api to download only original images.

Action:

1. Complete the dataset by tomorrow of 5 classes.

**Date:11/10/2021**

1. Increasfilterfiltering them manually.
2. Decided to use 2 models ( 1 medium weight and 1 lightweight)- Xception and mobilenetv2.

**BottleNeck - Creating the dataset manually and filtering.**

**Weekend...**

**Date:8/10/2021**

1. Discussed measuring the angle with horizontal for judging any pose.
2. Downloaded a yoga dataset which consists of around 200 images per class which we can use.
3. Explored Yoga-82 dataset which provides a set of links for all images. But most of the links are broken. So, I decided not to use it.

**Date: 7/10/2021**

1. Discussion about integrating a classification model in System design.
2. Decided to use image input for the classification model instead of coordinate inputs.

Action:

1. Find a dataset for yoga classification.
2. Come up with a classification model.

**Date: 6/10/2021**

1. Two new mentors joined the project. Explained to them what has been done till now.

**Date: 5/10/2021**

1. Discussed how to integrate our classification model in system design given in paper <http://jasonswee.com/wp-content/uploads/2019/05/CV_Report.pdf>.
2. Discussed 2 methods of judging any pose - Cosine similarity and neural network method.

Action:

1. We need to finalize the system design of our model flow from end to end.

**Date: 4/10/2021**

1. Downloaded dataset from Kaggle and inference on one more asana (Warrior pose).

**Date: 1/10/2021**

1. Completed inference on tree poses asana concerning 1 target image. And stored the result in google drive.

**Date: 30/09/2021**

1. Finding bugs in YoPoCo github and syncing it with a colab page.
2. Modified demo.py to correctly run it on GCP.

Action:

1. Use demo.py to run on two asanas.

**Date: 29/09/2021**

1. Open Pose Installed correctly on Windows and working file in Visual Studio(By using the latest version of cuda and cudnn). But the Python API is not working.
2. In GCP, I tried multiple ways to install openpose. Finally, one method worked with the python API.
3. demo.py of Pose Correction github runs without any error but the output video is empty.

**Date: 28/09/2021**

1. OpenPose successfully builds on Windows but while running the demo Unhandled exception at 0x00007FF8FEDF286E (ucrtbase.dll) error is coming.

Action:

1. Trying to build with latest version

**Date: 22/09/2021**

1. In GCP, CUDA is Downgraded to 10.1 from 11.2 and Python to 3.5. While building openpose from the source the same error occurs of CUDA\_cublas\_device\_LIBRARY. Many threads say that it is because of the cmake version. Cmake >= 3.12.2 is required but gcp is providing cmake 3.12.0.
2. While building a later version of cmake, the gcp session is crashing repeatedly because of RAM getting full.
3. Switched to Windows environment, Installed Visual studio and cmake gui. Building failed because CUDA is not getting recognized.

Action:

1. Need to Install CUDA 10.1 outside anaconda on my laptop and retry the building process. (23/09)
2. Figure out any other way to install cmake in gcp.(23/09)

**Date: 21/09/2021**

Discussion points:

1. Discussed about set-up environment using google collab page for enabling <https://github.com/nosyarlin/YoPoCo>
2. Discussed two environments ( Windows and google collab page (GCP) ).
3. Installation on the google collab page continued smoothly but encountered problems while building the openposecode.
4. Installation on windows continued smoothly but encountered problems while building the openpose code. It requires visual studio IDE

Action items:

1. Install visual studio and build openpose - Mention the due date for completion.(23/09)
2. downgrade cudnn, cuda and nvidia driver and build openpose in GCP -(22/09)
3. Build complete https://github.com/nosyarlin/YoPoCo with test verification of two basic aasana's - Mention the due date for completion.(24/09)

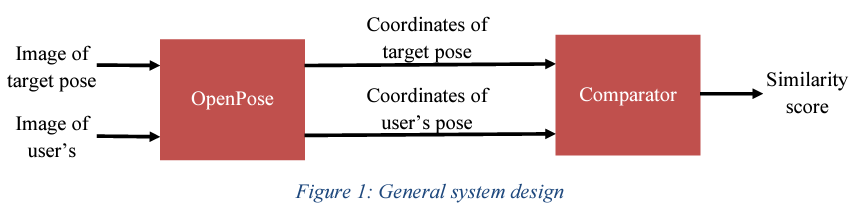
Blocker

1. Installation of python API of openpose

**Related work:**

**Resource6**

Implementing this github to check the performance of this model.

****

They are using 2 method for judging pose:

1. Cosine Similarity
2. Using a Neural network, they are calling it Comparator Net.

Github - <https://github.com/nosyarlin/YoPoCo>

**Current Work ----->>>** Installation: (Stuck on Open Pose installation)

* ~~Pytorch~~
* ~~Python3.5~~
* ~~Cuda 10.1~~
* ~~Cudnn 7.5~~
* ~~Ubuntu 16.04~~
* ~~OpenPose~~

**Resources:**

1. Rating yoga pose

<https://medium.com/@hollhc17/using-computer-vision-to-rate-yoga-poses-90a5dcb9aabe>

1. Tensorflow lite Model for pose estimation

[MoveNet: Ultra fast and accurate pose detection model.](https://www.tensorflow.org/hub/tutorials/movenet)

1. Yoga correction project when pose name is provided by user

<https://github.com/skhaniyur/aligned-yoga-app>

1. Pose Correction paper

<https://arxiv.org/abs/2006.11718>

1. Pose correction paper 2\*\*\*

<https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=9509937>

1. Pose correction paper3

<http://jasonswee.com/wp-content/uploads/2019/05/CV_Report.pdf>