Measuring Workout Effectiveness

Suryatej Reddy (2016102)
Vishaal Udandarao (2016119)
Jigme Lobsang Lepcha (2016045)
Yash Tomar (2016122)

INDRAPRASTHA INSTITUTE *of*
INFORMATION TECHNOLOGY **DELHI**

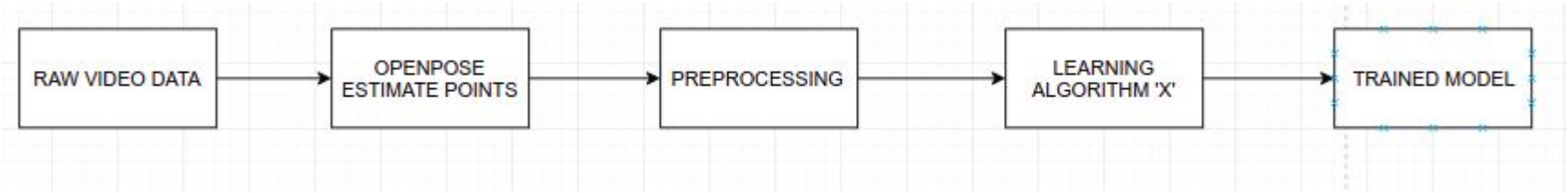
Problem Statement

- Staying fit is an integral part of life in current scenario
- Countless exercises which can be done to keep fit
- We first classify which exercise is being done
- Numerous people are also willing to pay large amount of money for private fitness trainers
- Most people while working out use incorrect technique
- Our project tells us whether the exercise being done is using correct technique or not
- We classify five workouts namely, push ups, pull ups, wall push ups, jumping jacks, squats

Dataset

- We took the activity data from UCF101-Action Recognition Dataset
  - There are a total of 13320 video clips which are divided into 101 action classes
  - The clips of one action class is divided into 25 groups, each group having some common features
  - We are going to use a subset of this dataset comprising of the following 7 exercises: push ups, wall push ups, weight squats, jumping jacks, pull ups, jump rope and boxing punching bags .
- We then map the body keypoints by using Openpose
  - Each video is split into set of n frames (n is dependent on the length of the video)
  - OpenPose returns a 75x1 vector mapping the human skeleton. We preprocess each of these vectors and extract a 28x1 vector.
  - This 28x1 vector represents the entire human skeleton
  - We then fit our classifiers on these 28x1 vectors.

Approach - Learning



Ran openpose on the dataset to get estimate points. An estimate point is an array containing the body part locations

Identified the key features that we would need for training and excluded the rest.

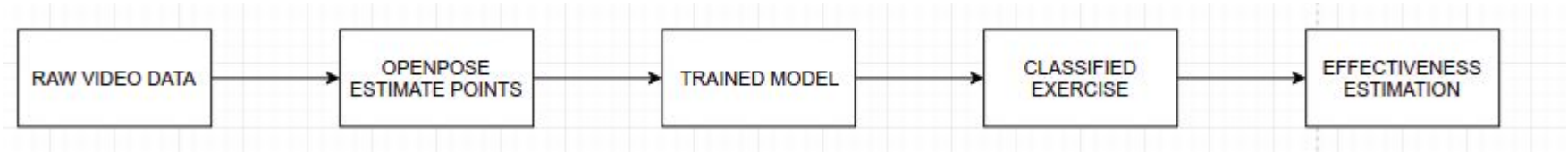
Scaled features to a normal distribution.

Trained various models on the processed data. SVM (Linear and RBF), Decision Trees, Sequential MLP.

Implemented an LSTM but didn't get good accuracy. Trained by taking features from last 2 seconds(60 frames).

Implemented a memory based SVM inspired from an LSTM. Created a modified feature set to incorporate memory of past 30 frames (1s). Saw a **training accuracy** of **1.0** and **test accuracy** of **0.998**.

Approach - Effectiveness

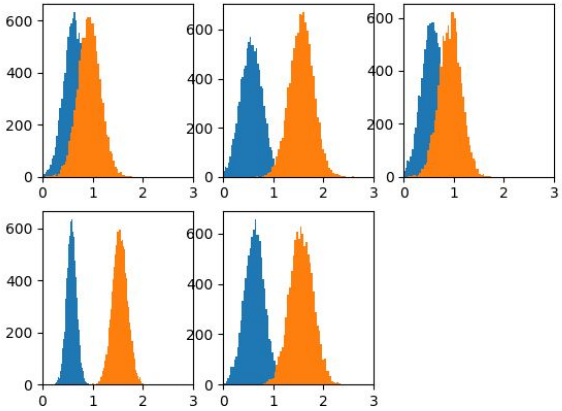


- Used ID3 algorithm to identify most important features(position) that distinguish an exercise from the rest (One vs Rest) .
- The ideal exercise videos are identified for every class i.e. in reference to which every exercise's effectiveness measure for that class will be evaluated.
- 5 ideal gaussian distributions (for each of the top 5 most important features) are modelled for every class with the mean and variance calculated over the pre-identified ideal exercise videos for that particular class.
- For every test video, once we predict the class it belongs to, we model 5 gaussian distributions with the mean and variance calculated over all of the video's frames.
- We then compute the KL divergence (Kullback-Leibler) between the 5 ideal distributions for that class and the 5 modelled distributions for the test video and compute efficiency as a function of the weighted sum of the 5 KL divergences

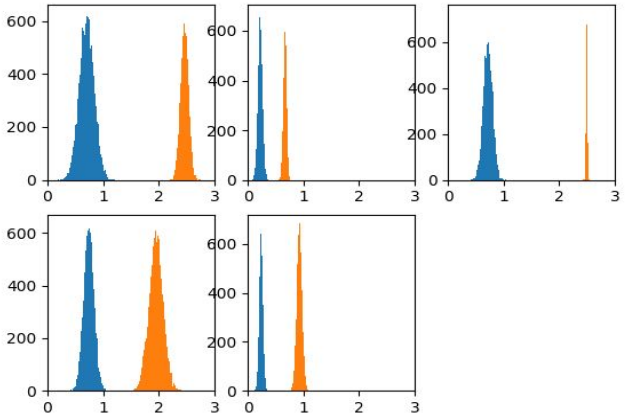
Results

Algorithm	**Hyperparameters**	**Accuracy**
Logistic Regression	Newton-CG and Multinomial	0.6751218573627501
Linear SVM	C=1000	0.7208196708168648
RBF SVM	C=1000	0.9068730809008041
SVM using memory		0.998754328461764
Decision Tree	Max_depth: 27	0.95662100456621
MLP using sklearn	Activation: relu, neurons: 100, layers: 2	0.9606979778212655
Sequential MLP using Keras	Activation: ‘relu’, neurons: 100, layers: 2, epochs: 200	0.9548271361006994

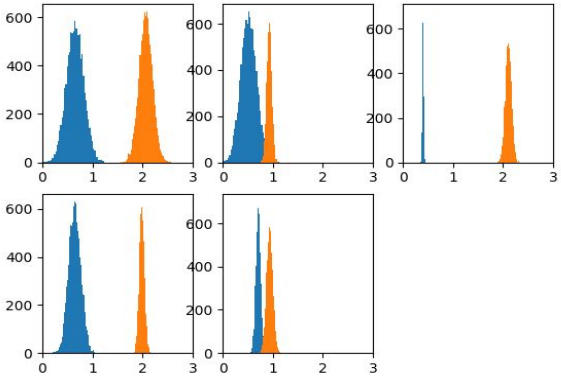
Results



Highly Effective pushup



Ineffective pushup



Jigme’s pushup

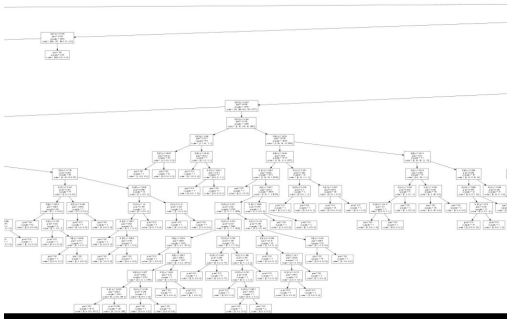
Analysis - Ablative

Plotted graphs comparing results across different models implemented.

Found that all advanced models worked better than baselines (except for RNN).

Analysis

- Confidence Scores - Removal of less confident frames would lead to loss of time based data required for SVM and RNN.
- RNN - Ideally, RNN's would be expected to work in a sequential setting but we didn't get good results. We got a best training accuracy of 0.5. Initially, data was overfitting so we used dropout at 0.2,0.3,0.4. Increased number of layers.
- Effectiveness - Initially, we identified key features for each exercise manually but then we saw that a time based analysis would be required as the position of a particular feature is not fixed. That's how we arrived at a Guassian representation of a temporal model.



Individual Contribution

- Suryatej Reddy: RNN, Decision tree
- Vishaal Udandarao: Effectiveness, MLP sklearn
- Yash Tomar: SVM with memory, OpenPose vectorization
- Jigme Lobsang Lepcha: Sequential MLP using Keras, baseline

Backup Slides

Workflow of RNN

1. Initially, just appended all frames together and trained. Didn’t work at all.
2. Then realized the error was probably frames from different videos getting mixed.
3. So we pre-preprocessed data to sequentialize data from only one video and then appended.

Possible Flaws:

4. Trained on OpenPose rather than video directly.
5. Couldn’t find the right architecture.
6. Differences in each video.