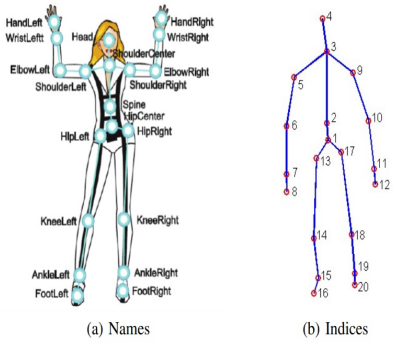
Robot Understanding of Human Behaviors Using Skeleton Based Representations

Anthony Olvera[1]

***Abstract*— In this work we examine how a machine or a robot can understand human behaviors using skeleton based representations and methods in artificial intelligence. We parse a data-set labeled by certain human actions collected by a Xbox kinect sensor. The data set gives pose estimates of 20 human joints for each frame in a video of a human preforming a certain action. We then extract features using three separate approaches. Relative angles and distances of a star skeleton representation, a histogram of joint position differences method and a histogram of oriented displacements method. The feature vectors for each method are written to files and split into a train and test set. We then use a support vector machine and the lib-svm frame work in an attempt to predict human behaviors. We use a grid search method to find the optimal parameters C and α to use with the models. We also investigate how the bin size of the histograms in our feature vectors affect model prediction accuracy. We find that the histogram of oriented gradients method outperforms both other methods reaching an accuracy as high as 75% with optimal hyper parameter values. The data-set and code for the implementation can be found at `https://github.com/arolvera/HumanSkeletonEstimation`**

I. INTRODUCTION

As technology advances and robots become more prevalent in society they will need to interact with humans more frequently, in closer proximity's and in more collaborative manners. Hence it is essential that that machines have a way of understanding human behavior and human social norms to better interact and function in certain situations. In this work we investigate multiple approaches in human skeletal representations for feature extraction to be used with machine learning algorithms which can classify human behaviors. For this investigation we utilize the MSR-action 3d data-set [1]. The data set consists of the Cartesian position estimation of 20 human joints across the frames of a video stream of a human performing 16 different types of actions such as drink, eat read book etc. The joints and representation can be seen below in figure 1. The data was collected using an Xbox kinect sensor [2] which uses stereoscopic vision and structured light technologies to estimate the position and orientation (pose) of a human body. We implement the relative distances and angles representation of a human star skeleton representation seen in 2. We also implement a histogram of joint position differences representation [3]. Finally we implement a histogram of oriented displacements representation [4]. Using the feature vectors obtained from these three methods we split the data into a testing and training set for the implementation of a support vector machine or SVM [5]. To implement the SVM we utilize an open source framework called libsvm [6] which is compatible with the python programming language which is what we used for our implementation. We use libsvm to perform a grid search method [7] to find the optimal hyper-parameters for out model which achieve the highest classification accuracy. we the train our model with these parameters across a range of bin sizes for the histogram and plot the accuracy for each bin size to analyse how the size of the bin affects the models accuracy. We find that there is an optimal number of bins for each representation which achieves highest accuracy. We also find that the histogram of oriented displacements model achieves the highest accuracy and fewest classifications out of the three human skeleton representations. We then write an integrated program which takes as its input the representation and desired number of bins and outputs the classification accuracy and confusion matrix [**?**] for the given representation and bin number. We cover the representations, experiments and results in greater detail in the following sections.



(a) Names (b) Indices

Fig. 1. Skeleton joint names and indices from Kinect SDK.

II. APPROACH

A. Relative Distances and Angles Representation

The first method used for human skeletal representation and feature extraction is called the relative distances and angles or RAD representation. The human is represented as a star skeleton as shown in figure 2

where we are interested in each distance relative to the center hip joint and each angle between each two adjacent body extremities. For each frame in a given action we calculate these five distances and angles for each xyz

[1]Anthony Olvera is a student in the Computer Science Department, Colorado School of Mines, 1500 Illinois St, Golden CO, 80401, USA aolvera@mines.edu

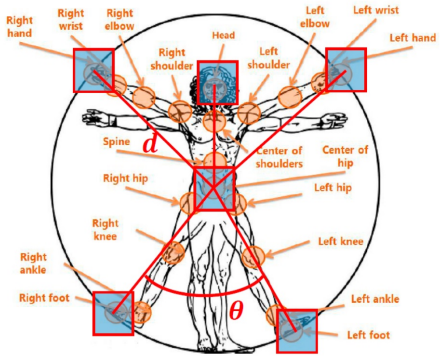


Fig. 2. Illustration of human representation based on relative distance and angles of star skeleton

coordinate and compute a histogram of N bins for the distances and M bins for the angles. N and M were selected as 10 by default but were adjusted later in the process which we discuss in the experimental results section. When a frame is complete we normalize the histogram by dividing each count by the number of frames for that particular action. This gives consistent results across certain actions with a varying number of frames. Next we conduct feature scaling on the data using min-max normalization as show in equation 1

$$x' = \frac{x - min(x)}{max(x) - min(x)} \tag{1}$$

This normalizes the counts to be between 0 and 1 which will help with classification accuracy when we use an SVM to predict the human action. Lastly we concatenate the histograms end to end for each x, y an z and this is our feature vector. We then append the data to our test or train files rad and rad.t respectively. The process is illustrated in more detail in algorithm 1.

B. Histogram of Joint Position Differences Representation

The histogram of joint position differences representation is similar to the relative distances and angles representation except that it takes into account all 20 human joints. In addition this representation ignores pair wise angles. Given a 3D location of a joint (x, y, z) and a reference joint (x_c, y_c, z_c) we calculate the joint displacement using 2

$$(\Delta x, \Delta y, \Delta z) = (x, y, z) - (x_c, y_c, z_c) \tag{2}$$

For this work the center hip joint was selected as the reference joint. Then for each temporal sequence of human skeletons a histogram is computed for each displacement $\Delta x, \Delta y, \Delta z$ using N bins. Note there is no M as the HJPD method ignores the angles between relative joints. N was selected as 10 by default as before but will be fine tuned in the future. These histograms are then concatenated together as a feature vector after feature scaling using the same method as the relative distances and angles representation. After all histograms for each temporal sequence are constructed they are written to the test and

Algorithm 1: RAD representation using star skeletons

Input : Training set `Train` or testing set `Test`
Output : `rad_d1` or `rad_d1.t`

for *each instance in* `Train` *or* `Test` **do**
  for *frame t = 1, ..., T* **do**
    Select joints that form a star skeleton (Figure 3);
    Compute and store distances between body extremities to body center $(d_1^t, ..., d_5^t)$;
    Compute and store angles between two adjacent body extremities $(\theta_1^t, ..., \theta_5^t)$;
  end
  Compute a histogram of N bins for each $\boldsymbol{d}_i = \{d_i^t\}_{t=1}^T$, $i = 1, ..., 5$;
  Compute a histogram of M bins for each $\boldsymbol{\theta}_i = \{\theta_i^t\}_{t=1}^T$, $i = 1, ..., 5$;
  Normalize the histograms by dividing T to compensate for different number of frames in a data instance;
  Concatenate all normalized histograms into a one-dimensional vector of length $5(M + N)$;
  Convert the feature vector as a single line in the `rad_d1` or `rad_d1.t` file.
end
return *`rad_d1` or `rad_d1.t`*

train files as a feature vector hjdp and hjdp.t respectively. Figure 3 illustrates the histogram of joint position differences representation.

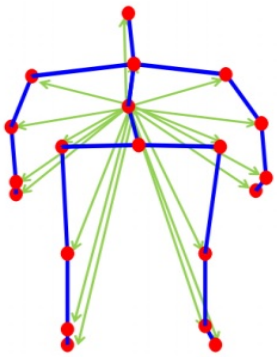


Fig. 3. Illustration of histogram of joint position differences

C. Histogram of Oriented Displacements Representation

The last human skeleton representation we implemented is called the histogram of oriented displacements representation. For each temporal step, that is frame to frame, we obtain the 3D trajectory of each joint and calculate the three corresponding 2D trajectories by projection onto the xy, xz and yz planes in Cartesian space. Then histograms are represented where the counts are the lengths of the 2D trajectories and the bin is selected as the range the angle relative to positive x axis falls between 0 and 2π radians. We calculate the angle using the following equation.

$$slope = \frac{P_{t+1}y - P_t y}{P_{t+1}x - P_t x} \tag{3}$$

Where P_t is the 2D position of a given joint at time t. An illustration of how the bin number is selected is depicted below in figure 7. Again the bin number is a hyper parameter selected as 10 by default but tune-able for optimal performance.

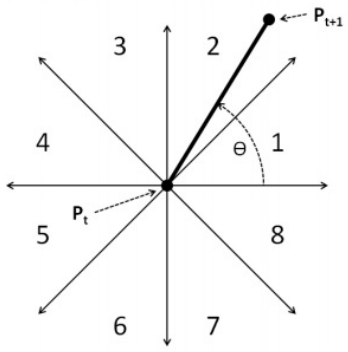


Fig. 4. Illustration of bin selection using HOD representation

Because dealing with the trajectory as a whole misses temporal information we also implement a temporal pyramid. That is the histograms are decomposed in three tier pyramid subdividing each histogram by two at each level this captures the temporal encoding of a certain action across a sequence of time-steps. Each histogram is the concatenated and normalized by frame numbers and scaled using the same method as the previous two representations. Then the histogram is the feature vector written as a line in our hod and hod.t files respectively.

D. Machine Learning Approach for Human Behavior Understanding

Now that we have created three separate data based representations of a human behavior or action, we wish to utilize a machine learning approach to classify the human behavior. The specific area of machine learning we are dealing with is referred to as supervised learning. We refer to it as supervised learning because we train our model with a labeled data-set with known ground truth. There are many methods for supervised learning approaches for classification such as decision trees and ensemble methods. However we choose to utilize something called a support vector machine. A support vector machine is a tool which attempts to find and optimal hyper-plane which maximizes the margin between multiple groupings of data. An illustration can be seen below in figure 5

It is a non-probabilistic binary classifier. Support vector machines are more traditional than some of the newer machine learning approaches such as deep learning, however, they are still very popular amongst researchers due to their simplicity and ability to produce significant results. In addition the are not as computationally expensive as some

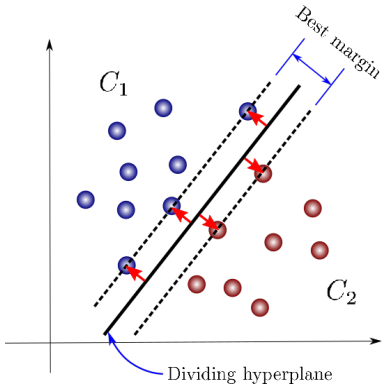


Fig. 5. Support Vector Machine

other methods and can be more suitable for a weaker or embedded computer which can be common in robotics. For this work we utilize a tool referred to as libsvm. Libsvm is an open source framework compatible with many programming languages providing tools for implementation of a support vector machine. For a better understanding how to use libsvm please refer to [8]. This guide explains how the libsvm API's were integrated into our source code. First we must convert out train and test files to the appropriate format. The files are converted to the following form. $< label >< index1 >:< value1 >< index2 >:< value2 >$ ... Where label is the action label. In this work we are focused on the following six actions; CheerUp, TossPaper, LieOnSofa, Walk, StandUp, and SitDown, which correspond to the respective indices; a08, a10, a12, a13, a14, and a16. Libsvm provides a grid searching method for selecting optimal hyper-parameters which examines an exponentially growing sequence of SVM hyper-parameters C and γ to find the optimal hyper-parameter values. For each representation we've applied libsvm to learn a C-SVM model with an RBF kernel and our training data. Our program trains and tests a the model for a given representation then outputs the resulting accuracy and and confusion matrix when applying the model on the test file. We then analyse how accuracy varies according to bin number. The results can be seen in the next section.

III. RESULTS AND DISCUSSION

A. Results

The plots for the grid search hyper-parameter selection for each representation can be seen below. The optimal C and γ values for each representation can be seen above each representations plot. The RAD method has an optimal $C = 8$ and $\gamma = 2$, where the HJPD method has optimal $C = 2$ and $\gamma = 0.125$. Lastly the HOD method has optimal $C = 128$ and $\gamma \approx 0$. Using these hyper-parameters we then trained our SVM model for each representation using the training files. We then apply these models on the testing file and compare with ground truth labels to calculate the models accuracy. In addition we can see model accuracy by behavior label using a confusion matrix.

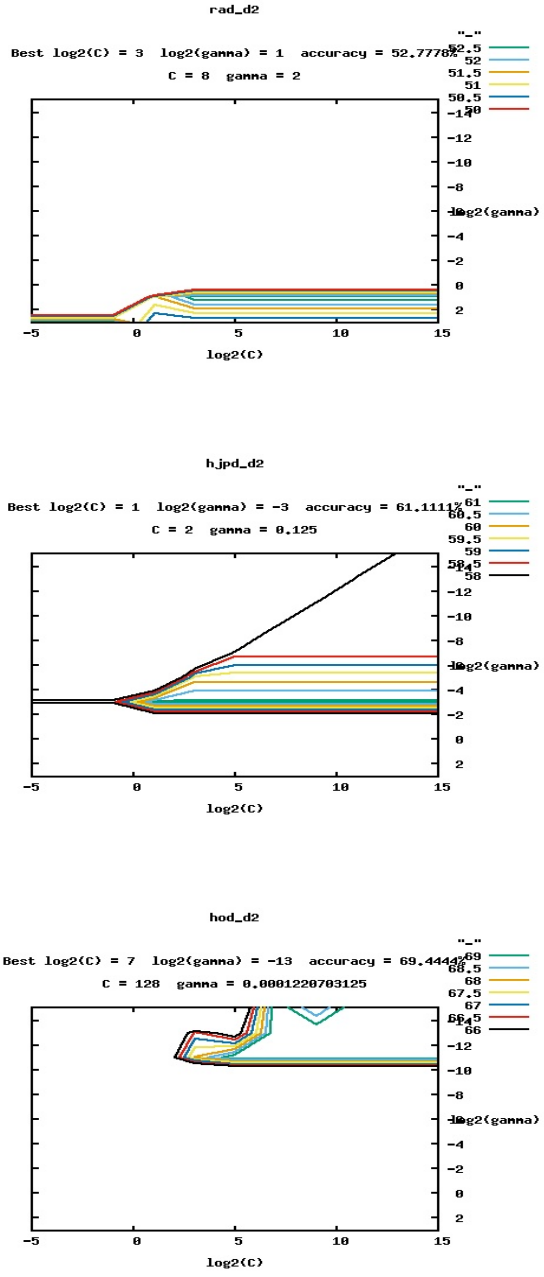


Fig. 6. Grid Search Plots For Each Representation

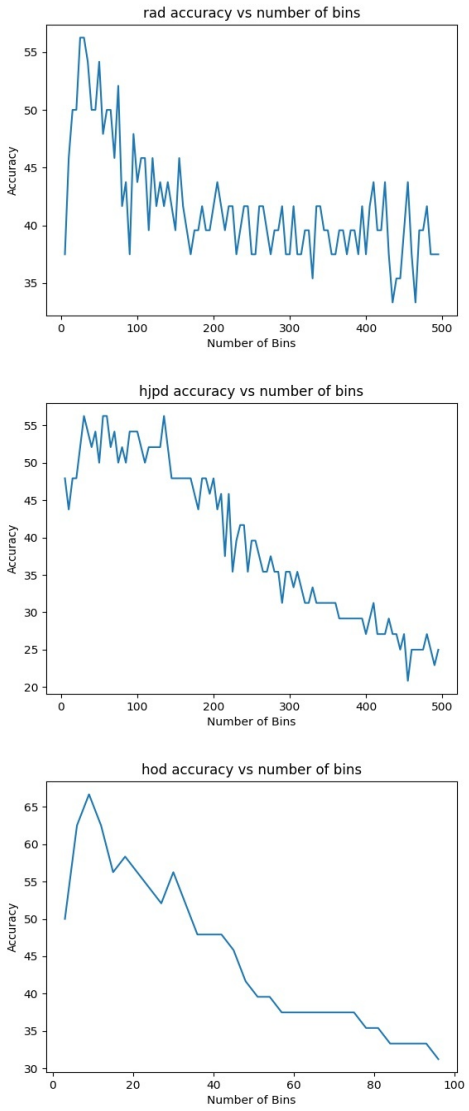


Fig. 7. Bin Number vs Accuracy For Each Representation

	Maximum Accuracy %	Bin Number
RAD	56.125	29 & 30
HJPD	60.4167	26
HOD	75	16-24

TABLE I

MAXIMUM ACCURACY AND CORRESPONDING BIN NUMBER

We wrote a program that trains and tests the model for a sequence of bin numbers starting at 5 and incrementally increases by 5 bins until it reaches an upper limit. The upper limit was selected as 500 for both the HOD and HJPD representations. Due to the longer run-time requirement for the HOD representation the upper limit was selected as 100 for this representation. The results can be seen in figure 7 all three representations follow a similar pattern. Rising to some maximum accuracy given at a certain bin number the dropping in accuracy as the bin number is increased.

Next, given we know our optimal number of bins for each representation we can feed that number to our program and generate the confusion matrix for each representation to illustrate our classification accuracy across the different human behaviors. Below are the matrices for the three separate representations.

B. Discussion

It can be seen that the RAD representation is the poorest representation in terms of classification accuracy. This may be due to the fact that it only takes into account the 5 joints form a star skeleton. Perhaps more

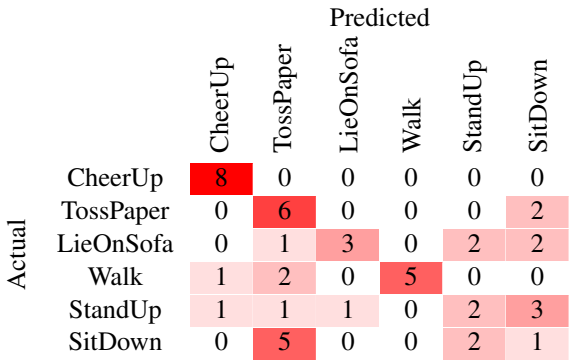


Fig. 8. RAD Confusion Matrix With Optimal Bin Number

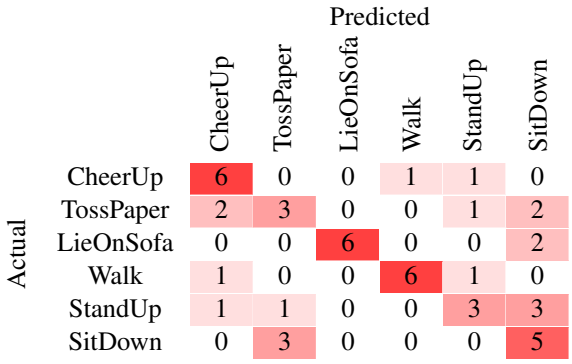


Fig. 9. HJPD Confusion Matrix With Optimal Bin Number

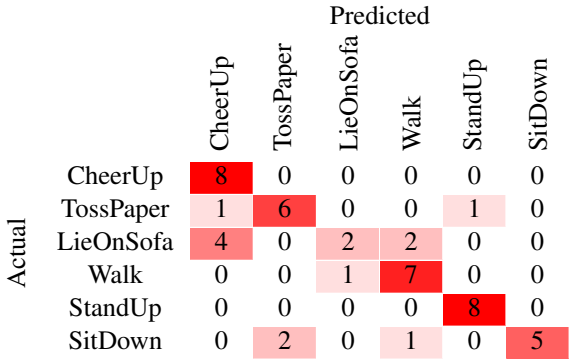


Fig. 10. HOD Confusion Matrix With Optimal Bin Number

detail regarding the remaining 15 human joints can lead to better robot understanding of human behaviors. The HJPD model seems to do better than the RAD representation but struggles with the TossPaper action and also the StandUp action. The last representation, HOD outperformed both other representations, with exception to the LieOnSofa action. Both other models more frequently and correctly classified this action. This seems to make sense as the HOD representation relies on a temporal sequence of joint trajectories. Lying on the sofa is a sedentary activity, hence it makes sense that this representation frequently mis-classifies this action. Thus it may be reasonable to use a different representation for understanding sedentary human behaviors. Overall a classification accuracy of 75% is not entirely bad and is sufficiently larger than a random guess, which with six human actions would be around 16.7% and all models were able to significantly beat this accuracy. However 75% is still not accurate enough to be deployed in certain scenarios, such as those where a mis-classification could have costly consequences or become a safety issue.

IV. CONCLUSION

In this work we investigated how a robot or a machine can better understand human behaviors using human skeleton representations and a machine learning approach. As technology advances humans will need to interact with machines more often and in closer proximity's thus it is essential that a robot can have a better understanding of common human actions. We utilized the MSR-action 3D data-set to create a relative angles and distances, histogram of joint position differences and histogram of oriented gradients skeleton representations. We trained a support vector machine for each of these representations using optimal hyper parameters that we've obtained through a grid search method. We analysed how accuracy varies with changing histogram bin number. We discovered a classification accuracy as high as 75% using the HOD representation, with the only drawback being low classification accuracy for the LieOnSofa action. The HJPD method was the runner up with an accuracy as high as 60.4%. The RAD method was the poorest in terms of accuracy with a maximum of 56.1%. We conclude that these methods may be an effective means of robot behavioral understanding in certain scenarios however the accuracy is not high enough to be appropriate for other applications where the cost of a mis-classification could be much higher.

REFERENCES

[1] https://wangjiangb.github.io/my_data.html

[2] https://en.wikipedia.org/wiki/Kinect

[3] H. Rahmani, A. Mahmood, D. Q. Huynh and A. Mian, "Real time action recognition using histograms of depth gradients and random decision forests," IEEE Winter Conference on Applications of Computer Vision, 2014, pp. 626-633, doi: 10.1109/WACV.2014.6836044.

[4] Mohammad A. Gowayyed, Marwan Torki, Mohamed E. Hussein, and Motaz El-Saban. 2013. Histogram of oriented displacements (HOD): describing trajectories of human joints for action recognition. In Proceedings of the Twenty-Third international joint conference on Artificial Intelligence (IJCAI '13). AAAI Press, 1351â1357.

[5] Noble, W. What is a support vector machine?. Nat Biotechnol 24, 1565â1567 (2006). https://doi.org/10.1038/nbt1206-1565

[6] https://www.csie.ntu.edu.tw/ cjlin/libsvm/

[7] Lameski P., Zdravevski E., Mingov R., Kulakov A. (2015) SVM Parameter Tuning with Grid Search and Its Impact on Reduction of Model Over-fitting. In: Yao Y., Hu Q., Yu H., Grzymala-Busse J. (eds) Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing. Lecture Notes in Computer Science, vol 9437. Springer, Cham. https://doi.org/10.1007/978-3-319-25783-9_41

[8] https://www.csie.ntu.edu.tw/ cjlin/papers/guide/guide.pdf