Categorizing Poses and Estimating them by Foot Pressure Heat Maps

Yashasvi Asthana

May 05, 2020

Contents

1 Abstract

Posture (Pose) analysis is very important to understand how a person moves and what is their centre of mass. There are many ways for estimating the pose of a person, like by using Motion Capture or by using Computer Vision techniques to analyze a pose in a picture. In this project, I propose a method to estimate the pose based on the foot pressure map of that person at that time. The objective is to divide Poses into certain classes. Then use those classes as labels for the foot pressure data and train a neural network. The resulting neural network will then be able to classify poses based on the foot pressure heat maps. The network can be used in medical fields to understand the common daily postures of a patient just by monitoring their foot pressures. There are many other applications of this research in the realm of health and sports.

2 Introduction

There are two parts of this project. The first part is to categorize similar human Poses using clustering algorithms and then labeling them based on the MoCap data of the Taiji performances. K-means is used for to find the average significant poses in the Mocap data-set. The best possible number of clusters is calculated using Calinski Harabasz Evaluation. The best K-value is 25, but K=24 is used to cluster the poses because 24 labelled key poses were provided later. The clustered poses are then used to label the Foot pressure data. SVM is used as the baseline model to classify Poses based on the Foot pressure data. The maximum baseline test accuracy is 38%. A convolutional neural network with 7 layers performs much better with an average test accuracy of 71%. This project only uses one of the 10 Subjects' data due to time and computational constraints.

3 Related Work

3.1 Related Work on the dataset area

The Taiji (Tai Chi) dataset that will be used for this project was generated in the Motion Capture lab at the Pennsylvania State University[2]. This dataset has been used for analyzing Pose stability. A convolutional neural network with residual architecture, named PressNET, was trained to regress the foot pressure heatmaps of different poses.[1]

3.2 Related work on the Pattern Recognition Approach

There are many state of the art clustering algorithms like Density-Based Spatial Clustering of Applications with Noise (DBSCAN)[3], Agglomerative Hierarchical Clustering, Expectation Maximization and many more.[5] Convolutional Neural Networks have been used for a long time since their inception. These are the best tools for image based classification as they help reduce the number of parameters by a lot. The recent advances in these networks is explained in detail by Jiuxiang Gu et al.[4]

4 Data

4.1 Motion Captured

The primary data collected is long (5min +) choreographed Taiji (Tai Chi) sequences of multiple subjects with synchronized motion capture, foot pressure, and video data.[2] The subjects wear capture markers and perform a choreographed 24-form Taiji Sequence. The markers record movements of the subjects at 100 frames per second. I will be using the MoCap data as this is much more accurate than 2D poses generated from the video frames. Moreover, we won't ever be needing this MoCap data once the model is trained as the input for our model will be the foot pressure heat maps. There are 31,784 frames of the subject 7 performing the Taiji sequence and every frame has corresponding 12 joint markers in 3D space and a joint confidence. .[1]

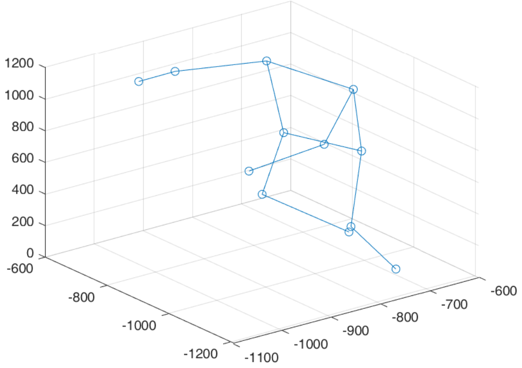


Figure 1: Example of a Mocap frame

4.2 Foot Pressure

Foot Pressure heatmaps of subjects performing the 24-Form Taiji Sequence are collected at 100 fps using a insole pressure measuring system. "The foot pressure heatmaps generated are 2-channel images of size 60x21." The MOCap and foot pressure data are synchronised based on the time in the performance, hence there is 31,784 frames worth of data for both (for subject 7).

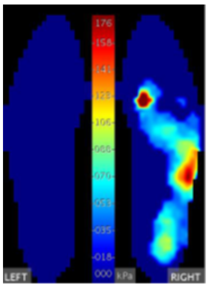


Figure 2: Example of a Foot Pressure Heat Map

5 Methods

5.1 Clustering of Poses

The first part of this project was to categorize the continuous Taiji sequence into different Postures. The subjects perform the 24-Form Taiji sequence, which means that there should be around 24 different postures that everyone was going through. Two very different approaches of clustering were tried - K-means and DBSCAN.
First task was to create the feature space for the dataset. The Mocap data contains 12 joint markers in 3D space and a joint confidence for every marker which is 0 if the joint is not detected and 1 if the joint is detected. So, the number of features become 12*4 = 48. Based on the visualization of the data, I added 4 new features -

- Euclidean Distance between left wrist and shoulder.
- Euclidean Distance between right wrist and shoulder.
- Euclidean Distance between left hip and ankle.
- Euclidean Distance between right hip and ankle.

Therefore, the total number of features in the new feature space is 52. These new features will help capturing the essence of some poses and will also help the clustering algorithm generate clusters with more separable means.
To find the optimal K for K-means, Calinski Harabasz criterion score and visual separation among the final average poses is used while iterating over different K values.
DBSCAN was a not a good choice for this clustering problem as the Taiji dataset consists of continuous motion sequence of the subjects. This type of continuous data where one point leads to the other naturally is not suitable for DBSCAN. To find the most suitable value for Epsilon (step size), I calculated Euclidean distances between all the combinations of the high dimensional points. Then, I used the average of the distances between 3 nearest neighbours of all the points as the Epsilon value for DBSCAN. Based on the results of K-means, the minimum number of points in the cluster were around 200, so the value of minPts was set to 200. After running the DBSCAN with these parameters, there were 2 final clusters and many outliers. This proves that DBSCAN is not suitable for such continuous datasets.

5.2 Classification of Poses using Foot Pressure Map

The Foot Pressure data matrix is created by applying the given foot mask to the given pressure activations of each frame of subject 7. The resultant matrix is a 4D matrix of size 60x42x1x31784 (used in the CNN). The Pose labels generated from the previous task are used as labels here. To calculate the baseline result of Support Vector machine, this data matrix is transformed to 31784x2520 ($60 * 42 = 2520$). The SVM classifier is trained over 80% of the total frames. The remaining 20% is used as test data.
For the convolutional neural network, these layers were created:

- Layer 1: Input Layer of size 60x42x1

- Layer 2: Convolution Layer with 60, 5x5 window size
- Layer 3: Batch Normalization Layer
- Layer 4: Fully Connected layer with 24 nodes
- Layer 5: 25% Dropout Layer
- Layer 7: Output Layer

Cross-entropy is used as the Loss function. The resulting network is trained over 29,784 randomly selected frames, with 1000 frames for validation and the remaining 5000 frames for testing. The training parameters are as follows:

- Solver: Stochastic Gradient Descent with Momentum (SGDM)
- Initial Learning rate: 0.001
- Max Epochs: 10
- Validation Frequency: 30
- Shuffling data every epoch

After training, T-distributed Stochastic Neighbor Embedding (t-SNE) is used to visualize the distribution of the outputs from the first convolution layer and the softmax layer.

6 Hypothesis

There are 24 forms in the Taiji sequence so there must be around 24 plus number of Poses at least. We have seen that the Foot pressure heat maps can be regressed based on the Pose in [1], so we know that there is a relation that we can exploit. So, the Hypothesis is that we can use that relation to classify Poses based on the Foot pressure heat maps. I created this as a classification, and not a regression problem because of the one to many mapping of the Foot pressure to Pose data.

7 Results

7.1 Clustering of Poses

In the Figure 3, the CH Score is highest at K = 25. Moreover, when analysing the means of the postures, at K = 25 there were no distorted average poses. We want the average poses to be as physically accurate as possible because these will act as labels in the next classification task. Our final system will return one of these average poses based on the foot pressure map and we do not want any pose to be anatomically impossible.
Later, I was provided with 24 labelled key poses which I used as the starting points of the K-means with K = 24. This process gives us very clear 24 poses which are highly separable and closely related to the key poses, hence 24 Pose labels are used in the following classification task.

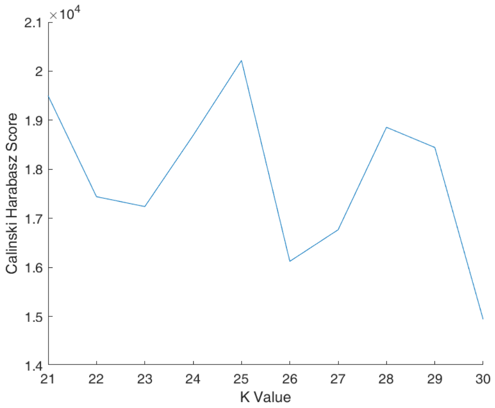
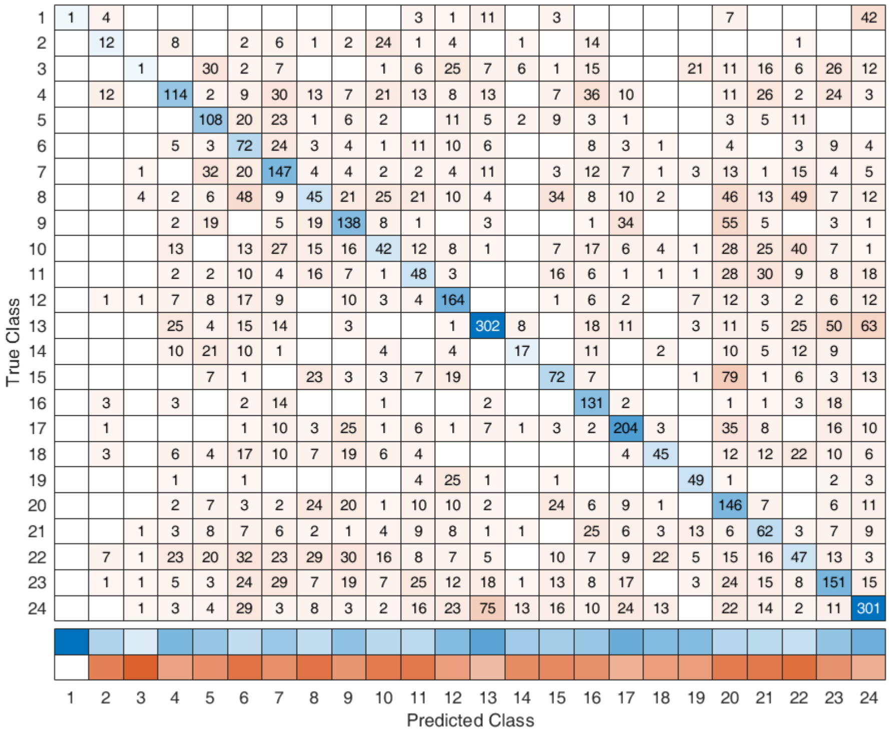


Figure 3: Calinski Harabasz Evaluation

7.2 Classification of Poses using Foot Pressure Map

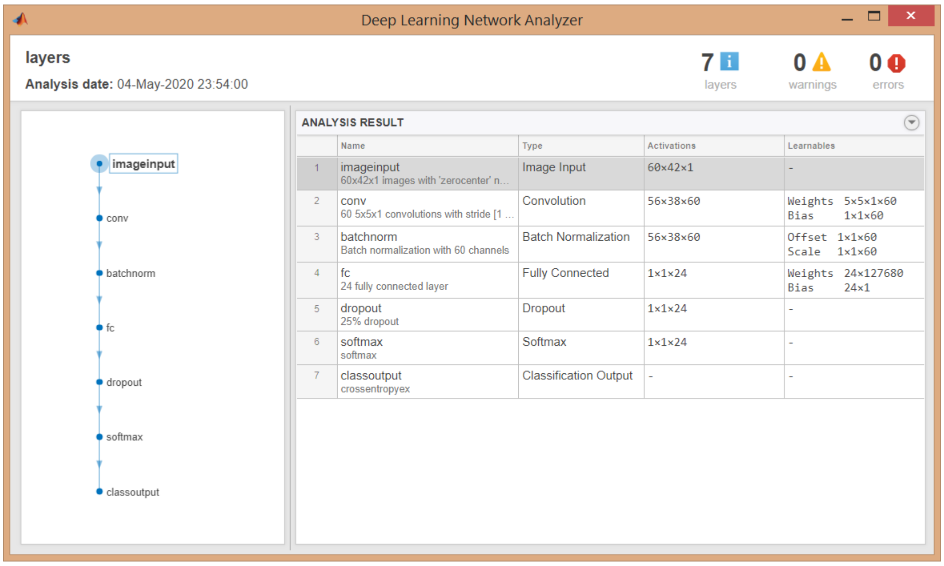
First, let us see the performance of the baseline (SVM). The Figure 4, shows the confusion chart for all the test data. The final test accuracy of SVM ranges between 32% to 38% based on different train-test splits. This is much better than a random guess, which will only lead to an accuracy of around 5% (100/24 to be exact).



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
1	1	4									3	1	11		3					7				42
2		12		8		2	6	1	2	24	1	4		1		14						1		
3			1		30	2	7			1	6	25	7	6	1	15			21	11	16	6	26	12
4		12		114	2	9	30	13	7	21	13	8	13		7	36	10			11	26	2	24	3
5					108	20	23	1	6	2		11	5	2	9	3	1			3	5	11		
6				5	3	72	24	3	4	1	11	10	6			8	3	1		4		3	9	4
7			1		32	20	147	4	4	2	2	4	11		3	12	7	1	3	13	1	15	4	5
8			4	2	6	48	9	45	21	25	21	10	4		34	8	10	2		46	13	49	7	12
9				2	19		5	19	138	8	1		3			1	34			55	5		3	1
10				13		13	27	15	16	42	12	8	1		7	17	6	4	1	28	25	40	7	1
11				2	2	10	4	16	7	1	48	3			16	6	1	1	1	28	30	9	8	18
12		1	1	7	8	17	9		10	3	4	164			1	6	2		7	12	3	2	6	12
13				25	4	15	14		3			1	302	8		18	11		3	11	5	25	50	63
14				10	21	10	1			4		4		17		11		2		10	5	12	9	
15					7	1		23	3	3	7	19			72	7			1	79	1	6	3	13
16		3		3		2	14			1			2			131	2			1	1	3	18	
17		1				1	10	3	25	1	6	1	7	1	3	2	204	3		35	8		16	10
18		3		6	4	17	10	7	19	6	4						4	45		12	12	22	10	6
19				1		1					4	25	1		1				49	1			2	3
20				2	7	3	2	24	20	1	10	10	2		24	6	9	1		146	7		6	11
21			1	3	8	7	6	2	1	4	9	8	1	1		25	6	3	13	6	62	3	7	9
22		7	1	23	20	32	23	29	30	16	8	7	5		10	7	9	22	5	15	16	47	13	3
23		1	1	5	3	24	29	7	19	7	25	12	18	1	13	8	17		3	24	15	8	151	15
24			1	3	4	29	3	8	3	2	16	23	75	13	16	10	24	13		22	14	2	11	301

Figure 4: SVM Test Confusion Chart

Figure 5, shows all the layers of the convolutional neural network that gives a good result. Because of using only one subject's frames, I decided to create a shallow network to avoid over-fitting of the training data. The cross-entropy loss and the training accuracy of this CNN is shown in the Figure 6. The test accuracy of this network ranges from 69% to 75% with 71% average test accuracy. One of the results with test accuracy of 71.40% is depicted using the confusion chart in the Figure 7, and the ROC curve in the Figure 8.



layers

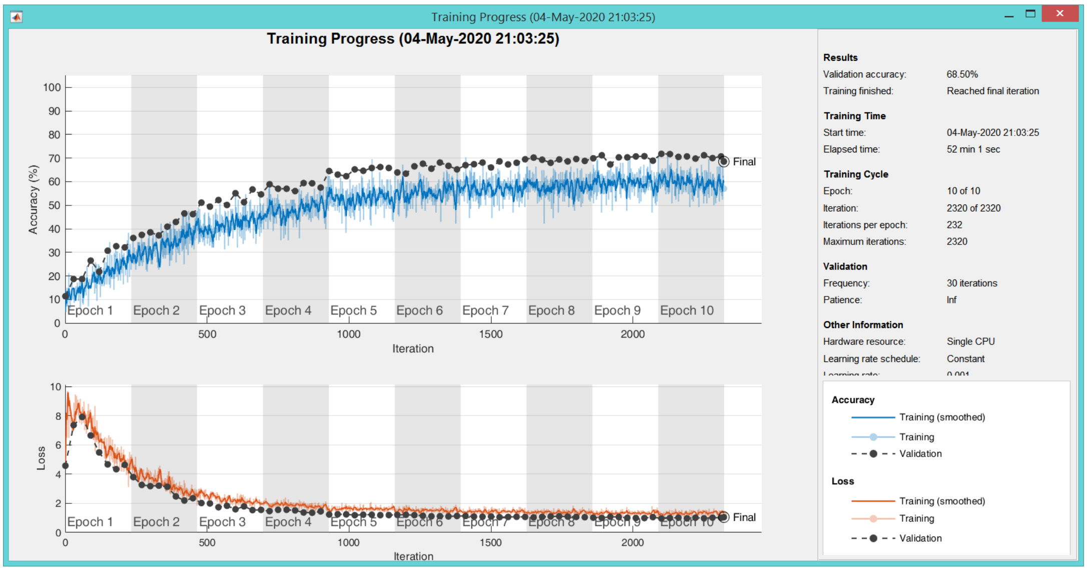
Analysis date: 04-May-2020 23:54:00

7 layers | 0 warnings | 0 errors

ANALYSIS RESULT

	Name	Type	Activations	Learnables
1	imageinput 60x42x1 images with 'zerocenter' n...	Image Input	60×42×1	-
2	conv 60 5x5x1 convolutions with stride [1 ...	Convolution	56×38×60	Weights 5×5×1×60 Bias 1×1×60
3	batchnorm Batch normalization with 60 channels	Batch Normalization	56×38×60	Offset 1×1×60 Scale 1×1×60
4	fc 24 fully connected layer	Fully Connected	1×1×24	Weights 24×127680 Bias 24×1
5	dropout 25% dropout	Dropout	1×1×24	-
6	softmax softmax	Softmax	1×1×24	-
7	classoutput crossentropyex	Classification Output	-	-

Figure 5: CNN Layers



Training Progress (04-May-2020 21:03:25)

Results

Validation accuracy:	68.50%
Training finished:	Reached final iteration

Training Time

Start time:	04-May-2020 21:03:25
Elapsed time:	52 min 1 sec

Training Cycle

Epoch:	10 of 10
Iteration:	2320 of 2320
Iterations per epoch:	232
Maximum iterations:	2320

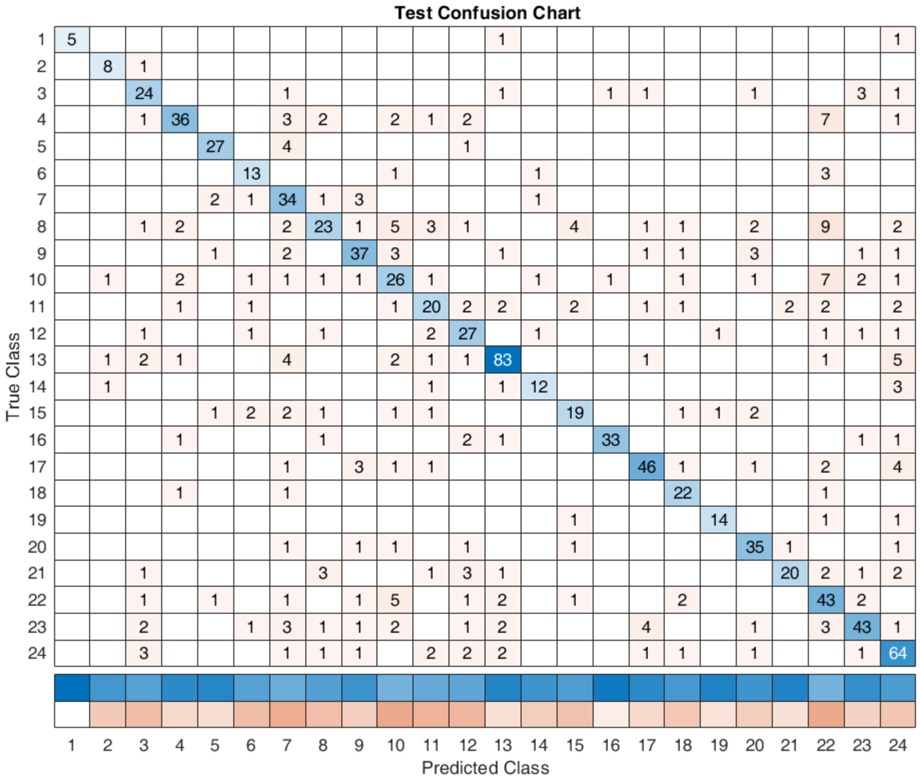
Validation

Frequency:	30 iterations
Patience:	Inf

Other Information

Hardware resource:	Single CPU
Learning rate schedule:	Constant
Learning rate:	0.001

Figure 6: CNN Training



True Class \ Predicted Class	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
1	5												1											1
2		8	1																					
3			24				1						1			1	1			1			3	1
4			1	36			3	2		2	1	2										7		1
5					27		4					1												
6						13				1				1								3		
7					2	1	34	1	3					1										
8			1	2			2	23	1	5	3	1			4		1	1		2		9		2
9					1		2		37	3			1				1	1		3			1	1
10		1		2		1	1	1	1	26	1			1		1		1		1		7	2	1
11				1		1				1	20	2	2		2		1	1			2	2		2
12			1			1		1			2	27		1					1			1	1	1
13		1	2	1			4			2	1	1	83				1					1		5
14		1									1		1	12										3
15					1	2	2	1		1	1				19			1	1	2				
16				1				1				2	1			33							1	1
17							1		3	1	1						46	1		1		2		4
18				1			1											22				1		
19															1				14			1		1
20							1		1	1		1			1					35	1			1
21			1					3			1	3	1								20	2	1	2
22			1		1		1		1	5		1	2		1			2				43	2	
23			2			1	3	1	1	2		1	2				4			1		3	43	1
24			3				1	1	1		2	2	2				1	1		1			1	64

Figure 7: CNN Test Confusion Chart

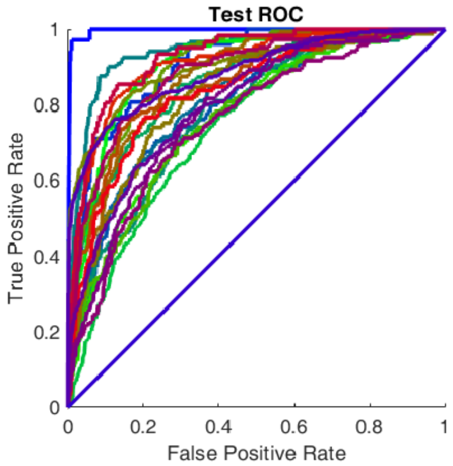


Figure 8: CNN Test ROC

After achieving a decent accuracy, we can visualize the behavior of our neural network using t-SNE plots of output activations from intermediate layers.

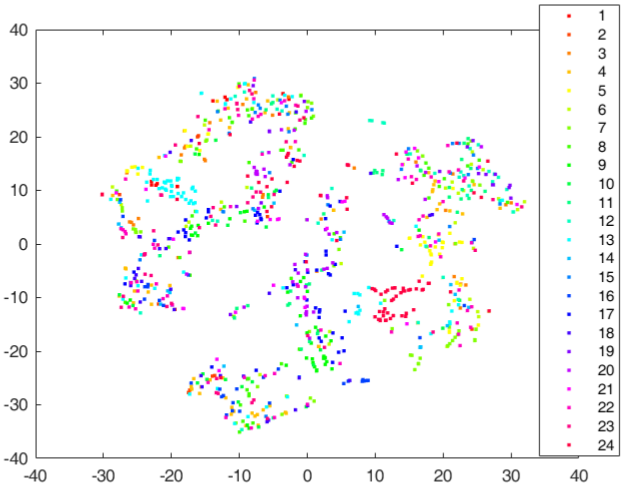


Figure 9: Convolution Layer Validation tSNE

The Figure 9 shows the 2D distribution of the Validation data after passing through the Layer 2 (Convolution Layer). We can see that there is no visible separation in the data points based on the given 24 class labels. The Figure 10 shows the 2D distribution of the Validation data after passing through the Layer 6 (Softmax Layer). We can see that the classes are becoming separable after passing through this layer, which is evident while classification.

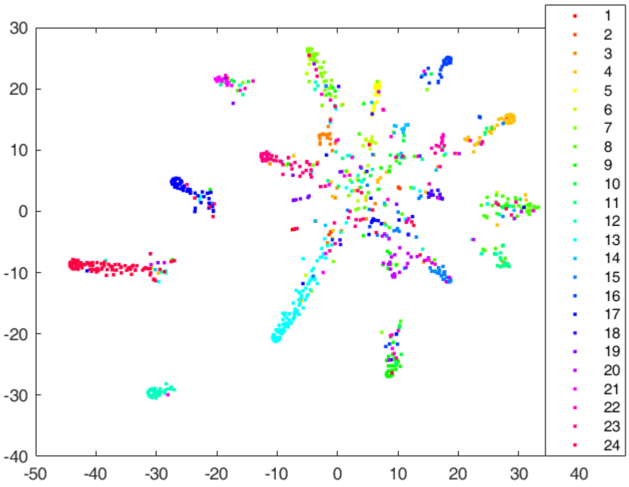


Figure 10: Softmax Layer Validation tSNE

8 Conclusion and Future Work

Through this project, the hypothesis that Poses can be classified using Foot Pressure Maps has been proved. The given convolutional neural network performs much better than the SVM baseline, with average accuracy of 71%. Only one subject's (Subject 7)

Mocap and Foot Pressure data is used throughout this project. In the future, if we use all the 10 subjects, we can build a deeper neural network with a higher test accuracy.

References

[1] Christopher Funk, Savinay Nagendra, Jesse Scott, Bharadwaj Ravichandran, John H. Challis, Robert T. Collins and Yanxi Liu
Learning Dynamics from Kinematics: Estimating 2D Foot Pressure Maps from Video Frames
https://arxiv.org/abs/1811.12607v4 2, 3, 5

[2] *Taiji (Tai Chi) Data* http://vision.cse.psu.edu/research/taijiDataset/index.shtml 2, 3

[3] Ester, Martin and Kriegel, Hans-Peter and Sander, Jörg and Xu, Xiaowei
A Density-Based Algorithm for Discovering Clusters a Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise
https://www.aaai.org/Papers/KDD/1996/KDD96-037.pdf 2

[4] Jiuxiang Gu, Zhenhua Wang, Jason Kuen, Lianyang Ma, Amir Shahroudy, Bing Shuai, Ting Liu, Xingxing Wang, Li Wang, Gang Wang, Jianfei Cai and Tsuhan Chen
Recent Advances in Convolutional Neural Networks https://arxiv.org/abs/1512.07108v6 2

[5] *Types of clustering algorithms* https://towardsdatascience.com/the-5-clustering-algorithms-data-scientists-need-to-know-a36d136ef68 2