CSCI 1430 Final Project Report:
Just Vision

Team name: Kamya Raman, Max Smith-Stern, Stephanie Lin, Sue An Park.
TA name: Brendan Leahey
Brown University

Abstract

We attempt to implement a simplified version of the popular video game Just Dance. We use live depth image input and sample images from live feed to pass through our pose classifcation pipeline. The pipeline is trained on synthetic data and uses random forest classifiers to predict segmentation maps to make joint predictions, then classify the pose. Our model has reasonably accurate results on synthetic data and maintains somewhat respectable results on our personal dataset tailored to the YMCA dance from Just Dance.

1. Introduction

The goal of our project was to recreate the popular game Just Dance using a three-step pipeline that takes in a depth image, isolates the person, classifies joints, and uses this joint vector to classify a pose. The current limitations of solutions to this problem are striking a balance between the amount of data and computing power we have, adapting to variations and noise in depth cameras, and mitigating the effects of having a long pipeline that is prone to errors.

Mapping 3D image data to 2D joint locations for pose classification is a computer vision technique that can be extended to solving other problems such as predicting pedestrian routes for autonomous vehicles and identifying improper movements for injury prevention. Our attempts to map 3D data to joint locations within the game platform are the beginning of a broader and more widely integrated technique.

2. Related Work

We used the depth map, segmentation map, and joints 2D from the SURREAL Dataset produced by Varol et al. [2]. This is a synthetic dataset that contains 6 million human frames. Limited by our computing power, we downloaded a small subset of the test set (507 images). 90% of the images were used to train the Random Forest Classifiers Random Forest Regressors.

Shotton et al. [1] use depth images due to their invariance to color and texture as well as simplify background subtraction which we also perform. They were most inspirational in our approach. Random decision forests were used for the multi-class classification problem. We use a similar approach using random forest classifiers and random forest regressors.

We used the software to connect to the Kinect camera from Xiang et al. [3]. We used separate software that interfaced the software from Xiang et al. to python. This was from Yamamoto et al. [4]. We adapted this software from Yamamoto et al. to sample live video feed at timestamps we specify.

3. Method

Our goal was to train a model that could predict the joint positions from a human depth image and identify the pose. First, we began by training the model using the SURREAL data set. The training and testing data set was too large to download, so we downloaded the val data set, which was a small subset of the test set. From the val set, 90% of it was used to train the model and 10% was used to test the predictions. From the depth map, we extracted a 5x5 patch for each pixel and calculated the local mean and standard deviation to get the features. In order to speed up the training process, we down-sampled the data set by a stride of four. Next, we used a Random Forest Classifier to classify pixels into a background, upper body, or lower body. Using this information, two additional Random Forest Classifiers were trained to predict either the upper body or the lower body only. This separation helped prevent the upper body being classified as one big body part.

After all the Random Forest Classifiers were trained, we used a Random Forest Regressor for each of the 24 joints. Having one regressor for each joint helped increase the accuracy of the prediction compared to using a single Random Forest Regressor for all the joint positions. Given the predicted joint positions from the Random Forest Regressor, we used the mean shift algorithm to cluster the predicted points and find the most central position of the joint.

After training the model, we integrated our own dataset. Compared to the SURREAL data set that had specifically marked background and human pixels, the Kinect images we captured did not have this distinction. To isolate the dancer from the background we used depth thresholding and CV2's connected component analysis to identify depth clusters. We then used additional area thresholding to ensure only the person was identified and the background was set to 0. We plugged these isolation masks into our joint classifier, and used the joints along with pose labels to train a Random Forest pose classifier. Then, we set up a live depth image feed that would take snapshots of the user's dance poses synced to the YMCA music. Images were converted into the isolation masks, then into a vector of joint locations, and finally a pose classification. We used two scoring metrics, accuracy over all poses and the average probability that the user was hitting a given pose.

4. Results

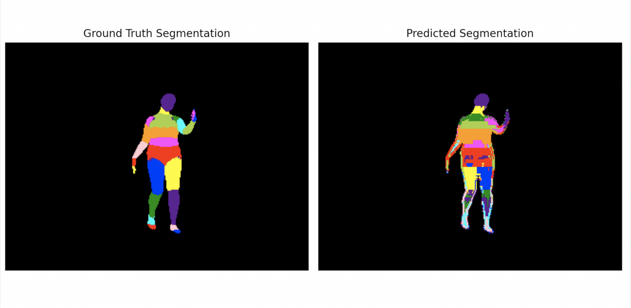

Figure 1. Segmentation Prediction on SURREAL Dataset

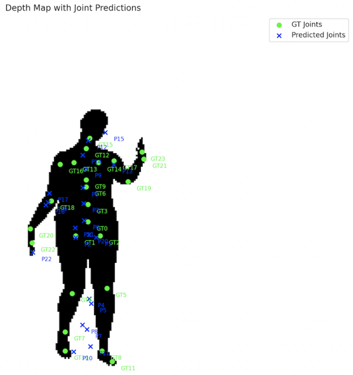

Figure 2. Joint Prediction on SURREAL Dataset

As shown in Figure 1, our model is capable of producing reasonable segmentation of body parts, correctly identifying the main body regions such as the head, neck, torso, shoulders, legs, and feet. However, the segmentation occasionally confuses left and right body parts. For instance, the left and right legs are sometimes mislabeled, with colors swapped relative to the ground truth. This suggests the model lacks a consistent spatial understanding of lateral symmetry.

In Figure 2, the predicted joint positions capture the overall body configuration reasonably well. While key points such as elbows, knees, and ankles are often placed near correct locations, finer details like hands and feet can be imprecise. Despite these inaccuracies, the predicted joints and segments are sufficient to infer the general pose of the subject, demonstrating the model's potential for coarse pose estimation tasks.

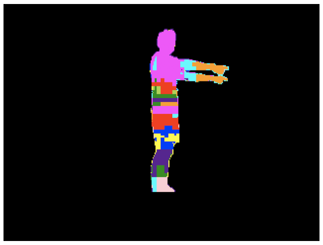

Figure 3. Segmentation Prediction on Our Dataset

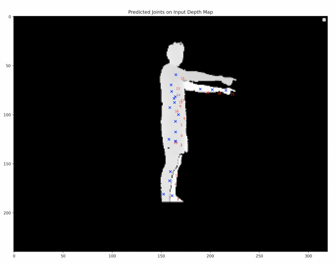

Figure 4. Joint Prediction on Our Dataset

Similar to Figure 1, Figure 3 demonstrates that the model does not effectively distinguish between left and right body parts. For example, both arms are assigned the same color. Additionally, the head, shoulder, and chest regions are grouped into a single large segment, indicating limited granularity in upper body classification. Nevertheless, the overall vertical ordering of colors across the body remains reasonably consistent, suggesting that the model has learned a rough spatial understanding of body part locations.

In Figure 4, the positions of several key joints such as the knees, ankles, elbows, hands, and spine are identified with reasonable accuracy. However, only one of the two arms contains joint predictions. This is due to the segmentation map classifying both arms as a single segment, which directly affects the joint regression stage. Since our approach relies on 24 Random Forest Regressors trained on segmentation labels to predict joint positions, this highlights the critical dependency of joint prediction accuracy on the quality of the segmentation map.

Initially, the joint predictions appeared to be relatively accurate. However, after labeling the predicted joint indices, we discovered that left and right sides were frequently confused. For example, the identified arm contains joints labeled 18, 19, and 21–corresponding to the left elbow, right elbow, and left wrist–all within the same arm, indicating a failure to distinguish between sides.

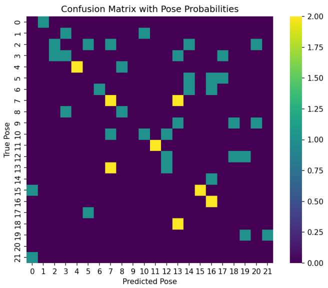

Figure 5. Pose Classification Confusion Matrix

For pose classification, the variation in our joint estimations and segmentation created unstable results. When using the joint classifications in a random forest for pose classification, our model tended to mix up similar poses such as ”thumb right”, ”shoot right”, and ”clap right”, which rely on accurate joint positions for smaller joints to make a prediction. Along with the fact that our segmentation mask could not distinguish between sides, side-based moves were classified poorly. Figure 5 shows a sample confusion matrix for pose classification using joint classifications. However, when we used just the isolation masks with a random forest classifier, our accuracy was much higher, highlighting that the issues within our joint classifier compounded the misclassifications.

4.1. Technical Discussion

The fluctuating results of our joint classification model demonstrate difficulties in creating high-quality labeled data, the limitations of machine learning compared to deep learning, and the importance of interpretability. During the process we spent a long time searching for a viable dataset to work with for our project. Specifically finding a dataset that had labeled per-pixel segmentation masks proved a challenge, and we originally tried to make our project work with only ground truth joint labels with limited success. One solution that we attempted to purse to make our segmentation and joint predictions more robust, especially since SURREAL data is synthetically generated, was to augment our dataset with more natural variations. However, we were unable to find additional data, so we had to work with different data augmentation tricks to make sure joint predictions on SURREAL data could generalize to our own dataset.

During the whole process, we were adamant about not using a neural network. We faced many setbacks, trying several different versions of our segmentation and joint predictors with different parameters or methodologies. While a convolutional neural network (CNN) is now the gold standard for pose and joint classification, we wanted to challenge ourselves to rely on machine learning algorithms. However, our end results highlight the limitations of these methods. This raises interesting questions about the comparative efficacy of machine learning and deep learning techniques, and what that means for future development in human pose recognition if machine learning techniques continue to take a backseat.

Another important takeaway was the importance of interpretability when debugging and improving our model. As we had many steps in our pipeline, each step we had to work with graphs to see what part of the pipeline may have been failing. Using these intermediate outputs, we noticed that when displaying the segmentation predictions on our data set, the body parts were classified as horizontal bands instead of detailed pixels initially. We weren’t able to figure out where the lines of classifications were coming from and tried multiple approaches to make the classfication more detailed. For instance, we added local mean and standard deviation in extracting the features and changed from using one Random Forest Regressor to one regressor per joint to potentially increase the level of detail in each joint. We were able to partially avoid the large strips that covered most of the upper body, but we couldn’t understand where the issue was coming from. This illustrates both the importance of interpretability in improving machine learning models and the importance of learning both lower-level and high-level features for classification tasks to improve accuracy.

Finally, the higher accuracy on our isolation masks when compared to our joint classifications demonstrates the importance of striking a balance between higher- and lower-level feature learning. When improving accuracy, we noticed that the segmentation map would predict upper body parts as lower body parts, which would significantly shift the joint predictions. We decided to create separate classifiers for the upper and lower body, which improved our results. By informing our classifier about the higher-level organization of segments, we were able to improve the resulting segmentation masks. The lack of higher-level feature learning could also explain why training the random forest pose classifier on our isolation masks was more successful when compared to our joint classifications.

4.2. Socially Responsible Computing

When thinking about human pose recognition it is important to ensure a diverse and representative dataset for unbiased results. Upon research of our SURREAL dataset, the pose meshes used to create depth images accounted for a diverse set of human bodies. As for our dataset, we took depth images of ourselves, so while we are a small subset of the human population, given more time, the dataset could include more varied pose data.

Since Just Vision is utilizing a Kinect camera it is also important to consider respecting the privacy of our users. We account for this by displaying the visual input to the user, so they are aware of what is being captured, as well as deleting all the snapshots taken immediately after their score has been calculated. The snapshots that are taken are only used to determine pose classification and are not stored for future use.

Finally, if we had a chance to continue to improve this project, we would consider adding more accessible features so that users with diverse needs could still use our platform effectively. Some features that could be easy to implement is text-to-speech instructions, so those who are visually impaired could also use our software. Another accesible feature that would be a good addition to our project is ignoring lower body joint classifications by using only the upper body random forest classifier for a more accessible seated dance routine. Although we did not get a chance to implement this ourselves, future work would involve making modifications to address these concerns.

5. Conclusion

We created a Just Dance duplicate for one song to allow users to dance for entertainment. Users are able to see their isolated movement on the screen while simultaneously following along with a video tutorial. Using captured depth image movements, we isolate the person's body from the background and create a segmentation map from a model trained with a Random Forest Classifier and the SURREAL dataset. With this segmentation map, we estimate each joint using a Random Forest Regressor. Using these joint vectors we determine the accuracy of the user's pose and calculate their game score.

We wanted to create a platform for people to have fun dancing with friends, encouraging healthy movement and cognitive activity. We see this project expanding to more songs from diverse genre to more broadly align with user preferences, uniting people over the excitement of dancing together.

Looking forward, our project gives good insights for the benefits and limitations of machine learning techniques on the task of human pose recognition. While Random Forest classifiers and regressors tend to generalize well on limited data, our results demonstrate limited efficacy for noise invariance and higher-level feature learning.

References

[1] Jamie Shotton, Andrew Fitzgibbon, Mat Cook, Toby Sharp, Mark Finocchio, Richard Moore, Alex Kipman, and Andrew Blake. Real-time human pose recognition in parts from single depth images. In *Microsoft Research*, 2011. 1

[2] Gül Varol, Javier Romero, Xavier Martin, Naureen Mahmood, Michael J Black, Ivan Laptev, and Cordelia Schmid. Learning from synthetic humans, 2017. 1

[3] Lingzhu Xiang, Florian Echtler, Christian Kerl, Thiemo Wiedemeyer, Lars, hanyazou, Ryan Gordon, Francisco Facioni, laborer2008, Rich Wareham, Matthias Goldhoorn, alberth, gaborpapp, Steffen Fuchs, jmtatsch, Joshua Blake, Federico, Henning Jungkurth, Yuan Mingze, vinouz, Dave Coleman, Brendan Burns, Rahul Rawat, Serguei Mokhov, Paul Reynolds, P.E. Viau, Matthieu Fraissinet-Tachet, Ludique, James Billingham, and Alistair. libfreenect2: Release 0.2, April 2016. 1

[4] Ryuichi Yamamoto, maderafunk, MikoyChinese, and Lars Andersson. r9y9/pylibfreenect2: v0.1.4 release, May 2020. 1

Appendix

Team contributions

Kamya Raman At the beginning of the project I helped outline our program and detail what all the moving parts would look like. I helped create our first joint classifier attempt by implementing the Microsoft paper, which ended up having poor results. I also worked on a function that loaded all the SURREAL data into a usable format, the pose random forest classifier, the isolation mask code, and helped experiment with the joint classifier to improve results on both datasets.

Max Smith-Stern I spent a lot of time trying to get the Kinect camera trying to work with our pipeline. I also got the live demo to work playing synced music and video while sampling from the live video feed so we could run our pipeline on those images. I also did my best to assist my teammates to do whatever the project needed. That often involved debugging or tuning classifiers to work with our data. I was able to tune the seg maps so they only predicted on the person when they were also predicting on the background. I also worked with Stephanie to create our dataset for the YMCA dance.

Stephanie Lin As the beginning of the project, I downloaded a program to run the Kinect camera we used for depth image input and created a file that would capture float32 images based on key press. This code allowed up to create you dataset of dance poses. After image collection, I cleaned and deleted poses that were not captured well. Later on the project, I attempted to test a version of our joint estimation model on the dataset we made, normalizing the images and increasing the accuracy. Towards the end of the project, I created a pipeline that would run all the components together into one cohesive game. Throughout the entire project, I continuously ran and tested models, seeking to increase accuracy.

Sue An Park I focused on joint classification and refining its accuracy. Building on the foundation and ideas initially explored by Kamya, I enhanced the performance of joint classification through a two-step process: first, using a Random Forest Classifier to predict body part segmentation; then, applying a Random Forest Regressor to estimate joint positions. Upon observing that the upper body was often classified as a single large region, I further attempted to improve the prediction accuracy by dividing the body into upper and lower parts for separate classification. I also experimented with multiple parameters for the classifiers and regressor to find the most optimal conditions.