BalletNetTrainer: An Automatic Correctional Feedback Instructor for Ballet via Feature Angle Extraction and Machine Learning Techniques

Jiayao (Emily) Li
ARQuest SSERN International
Irvine, CA, USA
emilyjiayaoli@gmail.com

Haridhar Pulivarthy
ARQuest SSERN International
Kirkland, WA, USA
hpuliv@gmail.com

Abstract

Ballet is a complex art form that consists of precise movements atypical to the human body and requires meticulous technique. Students receive only a limited portion of their instructor's attention in a classroom setting and private coaching is often financially inaccessible, which results in overlooked mistakes. As a result, these momentary errors become habitual and culminate in serious injuries that would be prevented with proper body alignment. This research focused on enhancing the current technology in instructional programs for ballet by training a teacher model with open-source video data of professional dancers. We used a novel Feature Angle Extraction system as a medium to standardize variances among different dancer videos. Our pipeline consisted of the pose estimation system OpenPose to identify dancers' joint positions, Dynamic Time Warping to align video frames, Random Forests Classifier to feature select characteristics of correct movements, and Decision Tree Classifiers to generate feedback. The outcome of this study was a highly accurate application that computes a percent depiction of movement quality and returns user-friendly feedback upon the input of a single-view video.

Keywords: Motion analysis, pose estimation, dynamic time warping, random forest classifier, ballet

1. Introduction

Ballet is a complex art form that consists of meticulous movements atypical to the human body and requires precise technique and body alignment. Even though human ballet teachers can instruct dancers, their attention is often divided between several students. Personalized feedback is rare during regular classes and achieving correct body alignment while turning for certain poses (e.g. pirouettes) may be difficult for inexperienced dancers. The repetition of subtle mistakes in movements such as these may result in serious injuries. We seek to prevent these in our work by developing an accessible, automatic ballet instructor that can supplement in-person guidance.

This software provides personalized feedback using a single-view video of the user, which can be recorded on most mobile devices. It uses a previously developed, now optimized, 3-D pose estimation system to recognize the dancer's joints over a series of video frames. These joints are processed using a novel angle feature extraction system, which derives a pair of angles (front view and side view) from each joint's x, y, and z coordinates to serve as a comparative medium between the student video and the teacher model. The joint angles are standardized with Dynamic Time Warping and then analyzed using a newly developed vector-geometry algorithm. An instructor model is trained and tested with videos from beginner, pre-professional, and professional (ground truth) ballet dancers. Finally, the program's feedback includes the correctness of the user video and possible improvements in English sentences.

1.1 Objectives

This research aims to enhance the current instructional programs for ballet. By examining the applications of pose estimation in generating user-friendly feedback, BalletNetTrainer seeks to expand the role of technology in preventing injury and improving technique in common physical activities like dance. We intend to improve the quality of instructional feedback by providing an alternative to the classification-based CNN systems of the past. BalletNetTrainer will instead implement a novel Feature Angle Extraction system, which can encode the position and orientation of the user, as a medium to standardize variances between videos. This system will allow for targeted feedback that can be easily interpreted by the user to improve their technique. Furthermore, our data pipeline should be applicable to other physical movements with minimal alterations.

2. Literature Review

2.1 Previous Feedback Applications:

We identify several similar instructional programs that provide feedback to the dancers and athletes. Chen et al proposed PoseTrainer for several similar programs focused solely on ballet pose recognition through classification, which does not take into account the movement of the dancer over a time series [Chen et al., 2018]. Thus, these programs are limited in their usefulness because users would rather receive technical corrections than a mere classification of their movement. Zell et al. used pose estimation based on forces and torques to analyze physical movements [Zell et al., 2017]. Chen and Richard simplified this process to an angle and distance analysis, which they used to provide feedback on the accuracy of various exercises. Kishore et al. included a similar approach in their analysis of accuracy on an Indian classical dance dataset [Kishore et al., 2018].

2.2 Previous Pose Estimation Systems:

Our application must use an accurate 3-D pose estimation system to best collect data from the user. There are several available options, varying in the conditions they require from the user: specific sensor configurations, frame count, lack of background interference, and more. Toshev et al. were the original proponents of deep neural networks to improve the accuracy of pose detection [Toshev et al., 2014]. They found the location of key points using regression on CNN features. Newell et al. further improved this CNN architecture by using repeated bottom-up and top-down processing [Newell et al., 2016]. Wei et al. created an alternative to Newell et al.'s stacked hourglass architecture by refining joint estimates through sequential passes [Wei et al., 2016]. Shotten et al. were early innovators of 3D pose estimation; they used single depth maps (as opposed to RGB camera images) from the Microsoft Kinect to predict 3D joint positions [Shotton et al., 2011]. Then, Cao et al. used Part Affinity Fields to estimate poses in real-time, expediting the process used by Shotten et al. and making it practical for user consumption in the OpenPose application [Cao et al., 2017]. In 2020, Mehta et al. created a new 3D pose estimation program called XNect, which had accuracy similar to OpenPose but could function in real-time [Mehta et al., 2020]. Due to the relative recency of pose estimation as a practical and versatile tool, there is less research into its applications.

3. Data Collection

The dataset used to train the instructor model was obtained from Youtube. It is composed of 20 videos of professional ballerinas, of varying body types, belonging to prestigious dance companies. They each perform two rotations of the pirouette, which the model uses as the ground truth of a correct pirouette. This ground truth was tested against 25 'student' videos (less experienced dancers performing the same pose). To verify the quality of the instructional model, nine award-winning ballet teachers from around the nation were asked to evaluate the student videos and provide feedback. The BalletNetTrainer's feedback was later compared to this feedback to test the program's precision.

4. Methods

The technical steps of the BalletNetTrainer application will now be described using a pipeline model (see Figure 1). The process begins when the user records a video of a ballet pose and concludes when the BalletNetTrainer application provides feedback on the accuracy and areas of improvement to the user. Since many factors impact the quality of a dancer's pirouette, simply classifying a student's video to either be incorrect or correct using algorithms such as K-Means Clustering would not be useful. To provide robust and practical feedback for dancers, we create a teacher model that can be compared to the student video to identify mistakes, compute accuracy in a percent depiction, and finally return user-friendly feedback.

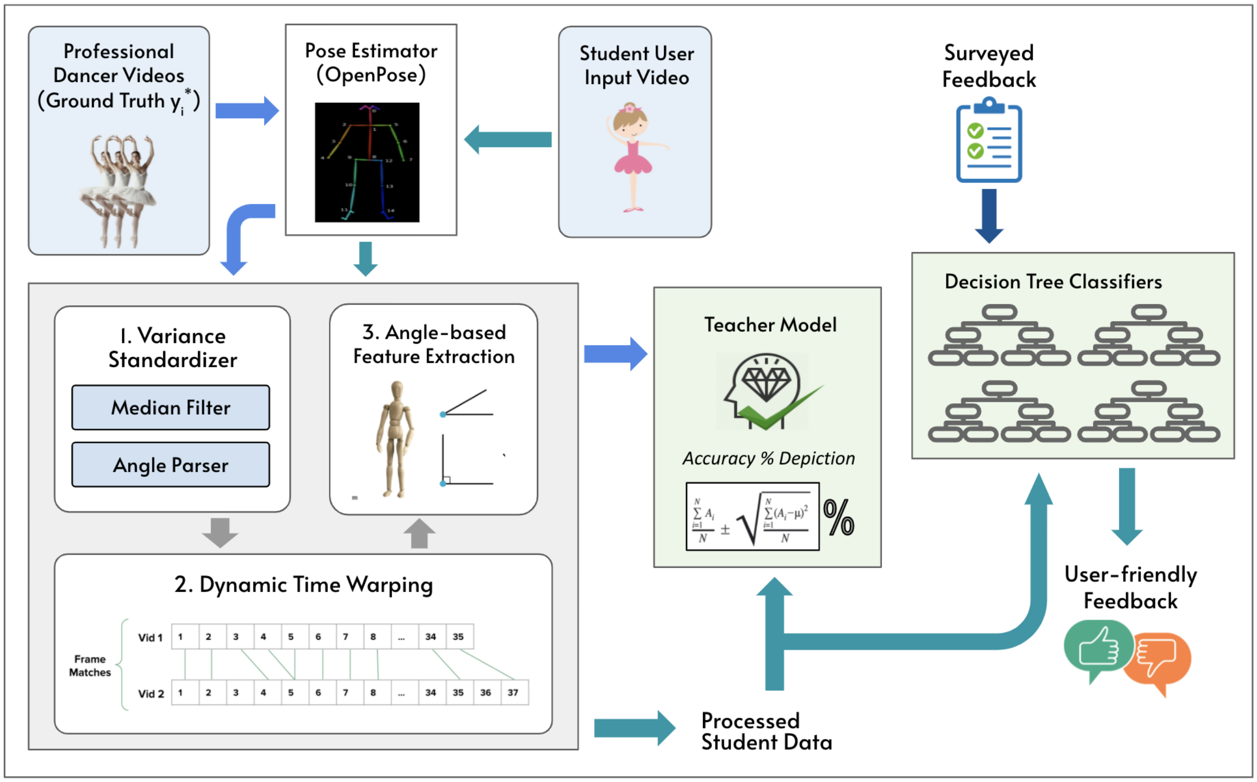


Figure 1: BalletNetTrainer Pipeline for both professional and student video data.

4.1 Preprocessing / Video Recording

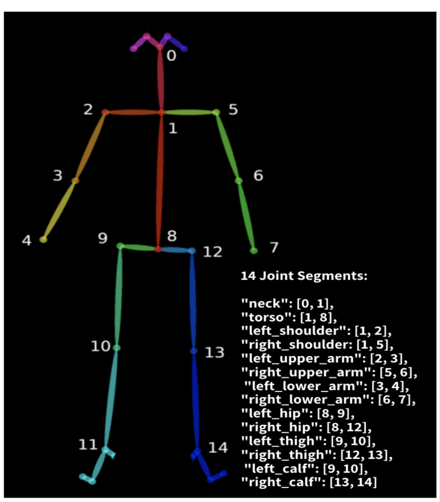


Figure 2: Pose skeleton generated from x, y, z key points of 14 joints (Source: Openpose)

In the first stage, the user records a video of themselves performing a ballet pose.

4.2 Pose Estimation

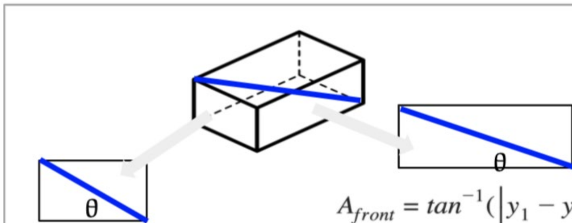
To estimate the user's pose, BalletNetTrainer labels key joints on RGB images using OpenPose. OpenPose is a pre-trained model that has two branches in the same sequential process, both of which use multi-stage CNN:
1. The first finds the confidence mappings of the key points on the image, which can be used to determine the relevance of the data.
2. The second learns the part affinity fields (PAF), which are vectors that can be used to compare the position and orientation of limbs [Cao et al., 2017].

OpenPose is the first pose estimation system to use PAF independent of body part location estimation. Removing this parallel refinement process dramatically decreases processing time with only a minimal sacrifice in accuracy, optimizing it for applications such as BalletNetTrainer.

When cropped videos of students performing pirouettes are inputted into OpenPose, the program outputs x, y, and z coordinates for the 14 joint keypoints for every frame in the video (see Figure 2).

4.3 Angle Parser

Scaling variances between the student video and teacher model such as distance to the camera and limb size do not necessarily detract from the accuracy of the student's movement. BalletNetTrainer accounts for these variances by deriving two sets of joint angles from OpenPose's x, y, and z coordinates using vector trigonometry, one for the front-view and one for the side-view (see Figure 3). This novel approach is effective because a measure of angular displacement can compare student and teacher data while ignoring scaling variances while a measure of linear displacement (e.g. limb vector length) cannot. These angles would be used in the following stages to determine pose accuracy and feedback options.



$$A_{front} = tan^{-1}(\left|y_1 - y_2\right| / \left|x_1 - x_2\right|)$$

$$A_{side} = tan^{-1}(\left|z_1 - z_2\right| / \left|y_1 - y_2\right|)$$

Figure 3: Angle calculations using trigonometry using X, Y, Z key points.

After the user video is processed by OpenPose, it is run through the angle parser for the keypoint output. First, the list is segmented according to confidence and the x, y, and z key points. Each grouping of data represents a limb such as an arm or a leg. We keep track of the limbs that are partially or fully obscured (i.e. the limbs that have low confidence) and then combine all of them to construct a skeletal approximation of the student's pose in each frame.

Although the overall accuracy of the application decreases when we ignore joint lengths, this is necessary to account for the aforementioned scaling variances. Alternatives such as motion analysis using data from multiple consecutive frames are highly susceptible to noise and fail when movements are executed extremely quickly (e.g. consecutive pirouettes).

Additionally, user videos are reoriented if their poses are mirrored or rotated. This is identified by comparing torso angles, which should generally be identical regardless of scaling. If a reorientation is needed, the key points are adjusted according to a multiplier determined by the difference between the torso angles. This automatic process prevents factors such as crooked cameras from impacting the angles and, eventually, the feedback generated.

4.4 Median Filter

In an earlier iteration, we found that the DTW process we use next performs poorly with noisy data. To minimize this flaw, we run the angle sequences through a median filter of size 3 [Dinc et al. 2015].

4.5 Dynamic Time Warping (DTW)

BalletNetTrainer generates feedback by comparing the data from the input video to a premade teacher model. However, these datasets can be of varying lengths and movements can be executed at different speeds without necessarily being wrong. Thus, simply comparing corresponding frames would be inaccurate.

To correctly compare these videos, we use Dynamic Time Warping (DTW) [Berndt et al., 1994, Portilla et al., 2020], which measures the non-linear similarity of the pose between two time-series (video frames, in this instance). More accurately, we determine the optimal warping path between the two videos, which is the alignment of frames, represented by index-pairs, where the videos are most similar. But first, the similarity between frames must be defined; typically, it is the Euclidean distance between two points on a pose skeleton, which is defined by the key points that

OpenPose identifies. However, this strategy fails to account for differences in body and limb size. Instead, BalletNetTrainer uses the difference between angles derived by the aforementioned angle parser to determine similarity. During DTW, we dynamically associate the angle in the second sequence most similar to a given angle in the first. With two angle sequences $Q \in R^n$ and $C \in R^n$, we construct an accumulated cost matrix $D \in R^{nxn}$, where $D_{i,j}$ is the local cost measure, or the difference between angles q_i and c_j. We iterate through every value in the matrix and dynamically find the optimal match of frames in each sequence. These matches are represented in the final warping path, which can then be used to compare the two angles across the time series.

Since each warping path represents only one angle per video, a pair of videos will produce 28 warping paths. These will be condensed to a single path in the post-DTW processing phase. After this is complete and a correct model has been created to act as a digital 'teacher,' the user video must be processed through a median filter and DTW (wherein the second time series is the correct model) as well. Again, this process will result in 28 warping paths, one for each pair of angles. These will also be condensed to a single path in the post-DTW processing phase, ultimately resulting in a sequence of index pairs that can be used to accurately generate feedback for the user. The index pairs are visually represented in Figure 4 by connected video frames.

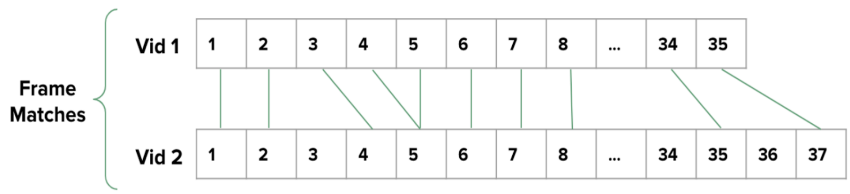


Figure 4: Every frame from Video 1 must be associated with a frame from Video 2.
However, the opposite is not necessarily true.

4.6 DTW Result Analysis

After processing the user data through DTW, the optimal warp paths provide one set of indexed pairs each. However, the purpose of this angle processing is to temporally align the entire pose, not only the individual key points.

To reduce this vast amount of data to a single set of matches, we first separate the data by frame number rather than the associated joint. Once every frame of a specific number has been gathered into one set, that set is analyzed to determine a method of simplification.

Although mean, median, and mode all reduce the data to a single number, each optimally suits a particular type of data set. To determine when each method should be used, we tested them extensively on user videos with varying conditions regarding duplicates and data distribution. In doing so, we determined the distribution and duplicate thresholds for the median and mode methods, allowing the default method to be the mean if the other two are not suited.

Once these conditions are set, the application can automatically condense the products of DTW into a warp path that allows for the accurate comparison of entire video frames rather than single key points.

4.7 Construction of Teacher Model

Since there are many components of a dancer's pirouette, simply classifying a student's video as either correct or incorrect using algorithms such as K-Means Clustering would not be useful; their feedback is not constructively implementable for the user. To provide applicable feedback to dancers, we create a teacher model that first identifies a student's mistake, then computes a percentage depicting its accuracy, and finally returns user-friendly feedback.

4.7.1 Feature Angle Extraction (FAE)

A major challenge in accurately evaluating the quality of the student dancer's movement is identifying the important features of a correct pirouette. Such a process is necessary to prevent false positives by determining whether angles are within the acceptable threshold due to proper form rather than characteristics belonging to all dancers.

The FAE system developed consists of a Random Forest Classifier [Breiman et al., 2001] that ranks the feature importance of the 14 joint angles based on teacher videos in the training dataset. These results are verified by another survey-based approach, wherein qualified human ballet teachers from across the country identify the important features of a pirouette.

These features are computed from the raw output of OpenPose using vector trigonometry. All of the processed data from teacher videos is run through the Random Forest Classifier. 60% of the data was used for training and 40% for validation.

4.7.2 Creation of Teacher model

To create the teacher model, we use a data-driven method based on 45 pirouette videos from various professional dancers. Since these videos serve as the ground truth for our teacher model, credible videos of professional dancers with varyings physiques were gathered to reduce sampling bias.

The teacher videos are processed through the aforementioned steps. Then, for each identified feature angle, a threshold of acceptable angle values is computed for all of the matching frames. With these thresholds, we account for a slight margin of error to create a realistic model for comparison. The mathematical model below ensures that the threshold is large enough to ignore trivial deviations but small enough to recognize false positives:

$$T = \left[\frac{\sum_{n=1}^{N} A_{n,j}}{N} - \sqrt{\frac{\sum_{n=1}^{N} (A_{n,j} - \mu)^2}{N}}, \frac{\sum_{n=1}^{N} A_{n,j}}{N} + \sqrt{\frac{\sum_{n=1}^{N} (A_{n,j} - \mu)^2}{N}} \right] \forall \ j \in J$$

$A_{i,j}$ = feature angle for video i at joint j
N = teacher dataset population
μ = population mean
J = set of feature angles

4.7.3 Accuracy Percent Depiction

Then, every angle of each feature in each of the frames is tested against the threshold; only feature angles that are below the threshold are labeled as correct. Incorrect frame numbers are stored to generate corrective feedback. Overall correctness is calculated using the following metrics:

$$\%\ accuracy = \frac{amount\ of\ frames\ that\ exist\ on\ interval\ T}{total\ amount\ of\ frames} \ \forall \ j \in J$$

4.7.4 User-friendly Feedback

To ensure the practicality of BalletNetTrainer, the program returns comprehensive and useful feedback in English. This is accomplished through a decision tree classifier that was designed to use Gradient Boosting, a “deviance” loss function, an estimator count of 1000, and a learning rate of 0.1. All the other hyperparameters were initially the default values set by the Scikit-learn library. The model was chosen for its interpretability, which is appropriate for exploratory knowledge discovery in ballet feedback. The model was evaluated using the Precision, Recall, and F-1 Scores [Hossin et al. 2015]. To mitigate overfitting during training, the hyperparameters max_depth (the maximum depth of the tree) and min_samples_leaf (the minimum number of samples needed for a split) were set to 6 and 3 respectively.

Each model classifies a video’s movement as requiring or not requiring corrections. In the former case, the model classifies the specific pre-set correction and at what specific instance of the movement *(e.g. drop shoulders and engage core slightly if the front-view feature angles for the shoulder joint is larger than the threshold at early frames)*. Prewritten correctional feedback was given by surveyed human ballet instructors. The attributes involved in training the models include the evaluated percentage, stored angles and frames at which the student is labeled to be executing an incorrect movement, and computed deviation of the angle from the desired angle.

5. Results and Discussion

This section will discuss the observed results from the Random Forest experimentation in the Feature Angle Extraction system. We present quantitative and qualitative results of BalletNetTrainer for four randomly chosen sample students at different levels ranging from beginner to early advanced out of the 45 videos trained. For each video, the feedback consists of a percentage rating for each feature, the overall rating, and the user-friendly English feedback.

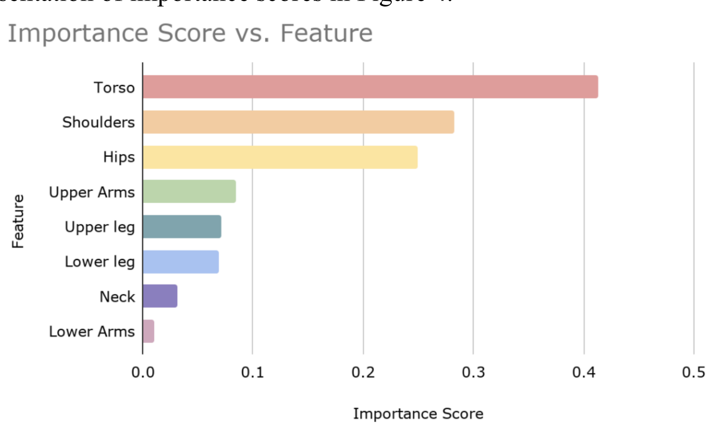
5.1 Feature Angle Extraction

Figure 5 presents the most significant angle features in giving rise to proper pirouette movement. It presents this information as a table of the calculated importance scores for all the joints. Figure 6 illustrates the table in a bar graph for visualization, ranging from the highest to the lowest scores.

Figure 5: Similar Joints are identified as important Feature Angle importance ranked and weighed.

Feature	Importance Score
Torso	0.413
Shoulders	0.283
Hips	0.250
Upper Leg	0.085
Upper Arms	0.071
Lower leg	0.069
Neck	0.032
Lower Arms	0.011

Figure 6: Graphical representation of importance scores in Figure 4.



5.2 Numerical Results for Percent Depiction

In Table 1, teacher video data from the validations set resulted in overall correctness ratings ranging from 0.98 to 1.00 on a scale of 0-1, with the most ideal value being 1, and the least is 0.

Table 1: Results for 4 sample teacher videos with updated threshold model (validation dataset).

Teacher Vid #	F1: Torso	F2: Shoulders	F3: Hips	F4: Arms	Overall%
T2	1.00	1.00	0.97	1.00	0.99
T4	1.00	1.00	1.00	1.00	1.00
T5	0.97	1.00	1.00	0.93	0.98
T6	1.00	1.00	1.00	1.00	1.00

Since having an overall rating close to 1.00 signifies that at every frame of the video, almost all of the feature angles are within the allowed thresholds, the high ratings of accuracy that BalletNetTrainer outputs for unfamiliar teacher videos in the validation set agree with hypothesized high scores. This indicates that the proposed method of using the feature angles as a medium for accuracy evaluation can effectively standardize the different variances among professional dancers' videos to prevent false-negative results.

In Table 2, BalletNetTrainer outputted overall ratings ranging from 0.58 to 0.78 for the 25 student videos tested, which are significantly lower ratings in comparison to the teacher test videos. This is as expected of a dancer in training.

Table 2: Results for 4 sample student videos with updated threshold model (test dataset).

Student Vid #	F1: Torso	F2: Shoulders	F3: Hips	F4: Arms	Overall%
S1	0.91	0.86	0.83	0.52	0.78
S2	0.85	0.73	0.76	0.45	0.70
S5	0.82	0.75	0.69	0.80	0.77
S6	0.68	0.63	0.6	0.43	0.58

The outcome that teacher test videos received overall high scores while student test videos did not demonstrates that the pirouette features identified as areas of study effectively capture characteristics of movements such as body alignment in the professional teacher videos correct and student dancers wrong since the angles of similar two joints remain the same regardless of differing joint lengths.

5.3 Generated User Feedback from Decision Tree Classifier

Table 3 presents the observed performance of our decision tree classifier model in terms of its ability to correctly classify each input video data with the correct feedback for the four identified feature angles for the validation dataset. We present results for before and after optimizing the hyperparameters of the model (see section 4.7.4).

Table 3: Confusion matrix for DTC Feedback Generation model on the validation dataset

	Features Studied	Precision	Recall	F-1 Score
Before Optimization	Torso	0.98	0.88	0.92

	Shoulders	0.73	0.76	0.74
	Hips	0.67	0.72	0.69
	Arms	0.43	0.51	0.47
After Optimization	Torso	1.00	0.95	0.97
	Shoulders	0.98	0.95	0.96
	Hips	0.71	1.00	0.83
	Arms	0.78	0.86	0.82

5.4 Quantitative Results

Figure 7 illustrates examples of the correct shoulder position (by a professional ballerina) versus the incorrect position (by a student dancer) as well as the feature angle statistics that we use when creating a percent depiction for the shoulder feature during the pirouette. Red data points are torso angles of the teacher model and the blue data points of a sample student.

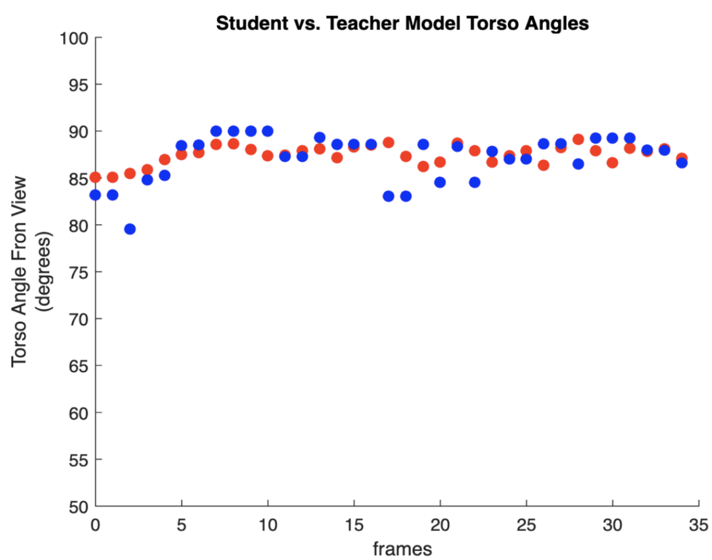


Figure 7: Correct torso data from a professional compared to amateur data.

5.5 Proposed Improvements

There are two external limitations to our program, first in the OpenPose program and second in the user's recording device.

Since the usefulness of the BalletTrainer program is directly related to the accuracy of the pose estimator, poor data presents a constant and unpredictable obstacle to effective feedback. Although we can compensate for this through techniques such as the median filter, these cannot replace a truly accurate dataset.

Also, some mobile devices (which we expect to be the primary tools for recording user input) record at only 30 frames per second. Although newer models can reach 60 frames per second, this is not to be expected. Thus, there is some ambiguity between frames and the angle-difference approach we take can only infer linear movement.

Additionally, the BalletNetTrainer program is currently only able to analyze pirouettes. However, the general pipeline can be expanded to accommodate other types of physical activity (other dances, sports, exercise, etc..). Through this versatility, the BalletNetTrainer approach seeks to transform available video resources of professionals into digital instructors that can help people better perform physical activities. Although this program cannot yet fully replicate the experience of personalized, human coaching, it can supplement that coaching or provide effective instruction when it is not available.

6. Conclusion

The BalletNetTrainer program is an accessible, automatic ballet instructor that can supplement traditional guidance with specific and easily applicable feedback. There are several notable takeaways from this study. Our novel angle-based approach to pose correction involves a Feature Angle Extraction (FAE) system and the generation of a teacher model. The FAE system developed for BalletNetTrainer selects the groups of angles most relevant to the evaluation of a dancer's technique through a Random Forests Classifier. It is observed that the torso, shoulders, hip, and arm angles were the quantitative evaluation of the importance of the predictor variables. Those features have been assessed in various professional videos to identify a suitable teacher model. Comparison of the user's data to this model allows BalletNetTrainer to provide potential improvements targeting particular features of the user's pose.

Unlike programs that rely on CNN and feature maps, BalletNetTrainer stores angle data to encode the position and orientation of the user. This approach provides higher-quality feedback, which has the potential to reduce injury and improve technique in dancers of all skill levels. Furthermore, our comparison-based pipeline can be expanded to incorporate physical movements relating to other sports given the needed expert video data.

References

[Chen et al., 2018] Steven Chen and Richard Yang. Pose trainer: Correcting exercise posture using pose estimation. ResearchGate, 2018.
[Zell et al., 2017] P. Zell, B. Wandt, and B. Rosenhahn. Joint 3d human motion capture and physical analysis from monocular videos. In CVPR Workshops, 2017.
[Kishore et al., 2018] P. V. V. Kishore, K. V. V. Kumar, E. Kiran Kumar, A. S. C. S. Sastry, M. Teja Kiran, D. Anil Kumar, M. V. D. Prasad. Indian Classical Dance Action Identification and Classification with Convolutional Neural Networks. In Advances in Multimedia, 2018.
[Toshev et al., 2014] A. Toshev and C. Szegedy. Deeppose: Human pose estimation via deep neural networks. In CVPR, 2014.
[Newell et al., 2016] A. Newell, K. Yang, and J. Deng. Stacked hourglass networks for human pose estimation. In ECCV, 2016.
[Wei et al., 2016] Wei, S.E., Ramakrishna, V., Kanade, T., Sheikh, Y.: Convolutional pose machines. Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference on (2016).
[Shotton et al., 2011] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake. Real-time human pose recognition in parts from single depth images. In CVPR, 2011.
[Cao et al., 2017] Z. Cao, T. Simon, S. Wei, and Y. Sheikh. Real Time multi person 2d pose estimation using part affinity fields. In CVPR, 2017.
[Mehta et al., 2020] Dushyant Mehta, Oleksandr Sotnychenko, Franziska Mueller, Weipeng Xu, Mohammad Elgharib, Hans-Peter Seidel, Helge Rhodin, Gerard Pons-Moll, and Christian Theobalt. XNect: Real-time Multi-Person 3D Motion Capture with a Single RGB Camera. In ACM Transactions on Graphics (TOG), 2020.
[Dinc et al. 2015] Dinc, Imren & Dinc, Semih & Sigdel, Madhav & Sigdel, Madhu & Aygun, Ramazan & Pusey, Marc. (2015). Chapter 12 - DT-Binarize: A decision tree based binarization for protein crystal images.
[Berndt et al., 1994] D. Berndt and J. Clifford. Using dynamic time warping to find patterns in time series. In AAAIWS, 1994.
[Portilla et al., 2020] Portilla, Ricardo, et al. "Understanding Dynamic Time Warping." *Databricks*, 8 Sept. 2020, databricks.com/blog/2019/04/30/understanding-dynamic-time-warping.html.
[Breiman et al., 2001] Breiman, L. Random Forests. Machine Learning 45, 5–32 (2001). https://doi.org/10.1023/A:1010933404324
[Hossin et al. 2015] Hossin, M., & Sulaiman, M. N. (2015). A review on evaluation metrics for data classification evaluations. International Journal of Data Mining & Knowledge Management Process, 5(2), 1.

Biography

Jiayao (Emily) Li is a classically trained ballet dancer and a junior at Northwood High School from Orange County, California. In the past, she has completed research with professors at UC Irvine's Donald Bren School of Information & Computer Science. She is the founder of eZeTrack, a startup that developed and maintains an IOS app that helps its users keep track of food consumption in hopes of reducing food waste through a smart digital fridge. Her research interests are computer vision, robotics, and biotechnology.

Haridhar Pulivarthy is a senior at Skyline High School in Sammamish, Washington. He has previously completed research as part of Project Ignite at Carnegie Mellon University. His research interests include artificial intelligence, robotics, and mechatronics.