LABELLE: A DEEP LEARNING APP THAT HELPS YOU LEARN BALLET

Sarah Fan[1], Kevin Guo[2] and Yu Sun[3]

[1]Sage Hill School, 20402 Newport Coast Dr, Newport Beach, CA 92657
[2]University of Southern California, Los Angeles, CA 90007, Irvine, CA 92606
[3]California State Polytechnic University, Pomona,
CA, 91768, Irvine, CA 92620

ABSTRACT

Human Pose Estimation has proven versatility in improving real-world applications in healthcare, sports, etc. [1]. Proper stance, form and movement is instrumental to succeeding in these activities. This paper will explain the research process behind the deep learning mobile ballet app, LaBelle [2]. LaBelle takes in two short videos: one of a teacher, and one of a student. Utilizing MediaPipe Pose to identify, analyze, and store data about the poses and movements of both dancers, the app calculates the angles created between different joints and major body parts. The app's AI Model uses a K-means clustering algorithm to create a group of clusters for both the student dataset and the teacher dataset [3]. Using the two sets of clusters, LaBelle identifies the key frames in the student-video and searches the teacher cluster set for a matching set of properties and frames. It evaluates the differences between the paired frames and produces a final score as well as feedback on the poses that need improving. We propose an unsupervised guided-learning approach with improved efficiency in video comparison, which is usually both time and resource consuming. This efficient model can be used not just in dance, but athletics and medicine (physical therapy like activities) as well, where stance, form, and movements are often hard to track with the naked eye.

KEYWORDS

Artificial Intelligence, Machine Learning, Deep Learning, K-means Clustering, Computer Vision, Ballet.

1. INTRODUCTION

Deep learning methods, especially in computer vision, are beginning to gain more and more popularity as technology advances to produce accurate models. Human Pose Estimation (HPE) addresses this area of deep learning research, computer vision [4]. It has versatile real-world applications in healthcare, sports, etc. [5]. We will be discussing dance, specifically ballet, which is the foundation for most forms of dance. Ballet serves as a steppingstone to most other forms of dance, so it is universally appreciated as one of the best ways to improve one's dance skills.

There are many ways to learn ballet [6]. With today's technology, you can watch a professional or teacher recording and learn through imitation. Alternatively, you can record yourself dancing, and compare it to the teacher's dance, critiquing your own movements to improve. Manually comparing two videos, however, is tedious, inefficient, and resource-consuming. We want to develop a deep learning-based App, *LaBelle*, that helps users learn ballet. To achieve this, we first needed to develop an accurate deep learning model that can classify, detect, track, and segment videos for useful data and information. For these tasks, deep convolutional neural

David C. Wyld et al. (Eds): SIP, CRBL, WiMNeT, BIOM, EDUPT, NLDM, SOFTFM, COMIT - 2022
pp. 135-146, 2022. CS & IT - CSCP 2022 DOI: 10.5121/csit.2022.121714

networks (CNN) are the most efficient, given their ability to extract patterns from target video clips (images) with high precision and accuracy [7]. Next, we need a good algorithm to efficiently compare two videos. This algorithm must process and store data for hundreds to thousands of video frames both accurately and quickly, to ensure a smooth and consistent user experience. Lastly, we need to design a clean UI layout in a practical app to display our score and feedback.

The rest of the paper is organized as follows: Section 2 describes our research background, direction, and crucial research elements in detail; Section 3 presents our approach to resolving any challenges we faced and relevant details about the experiments we carried out; Section 4 presents the results and analysis; Section 5 gives a brief summary of other works that tackle a similar problem; Section 6 gives closing remarks and discusses the future of this project.

2. CHALLENGES

Figure 1 shows the problem our research aims to solve. The process can be summarized as the following: (1) user inputs teacher's dance video (maximum length two minutes) and student's dance video (maximum length 40 seconds); (2) app processes the videos and applies our deep learning algorithm (using HPE) to the frames to analyze the data; (3) user receives a standardized score and is shown video frames with several poses to improve on.

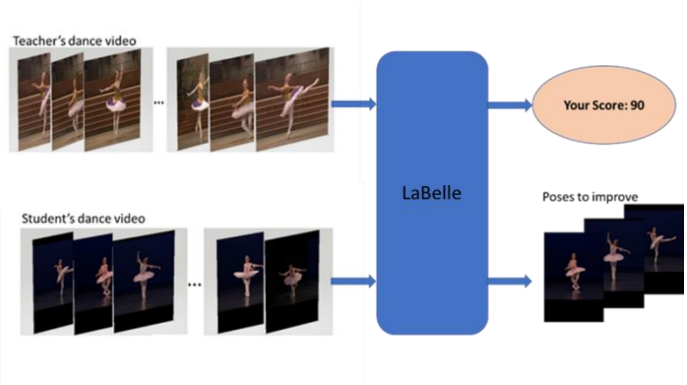

Figure 1. Problem definition

2.1. Human Pose Estimation

Human Pose Estimation (HPE) is a computer vision task that includes detecting, associating, and tracking semantic key points. On each video frame of target videos, the key points on a body need to be identified and tracked to describe the dancer's pose, referred to as landmarks. The connection drawn between key points and their movement is used to model the dancer's movement. There are three models we follow to process the human body: skeleton-based model, contour-based model, and volume-base model (shown in Figure 2).

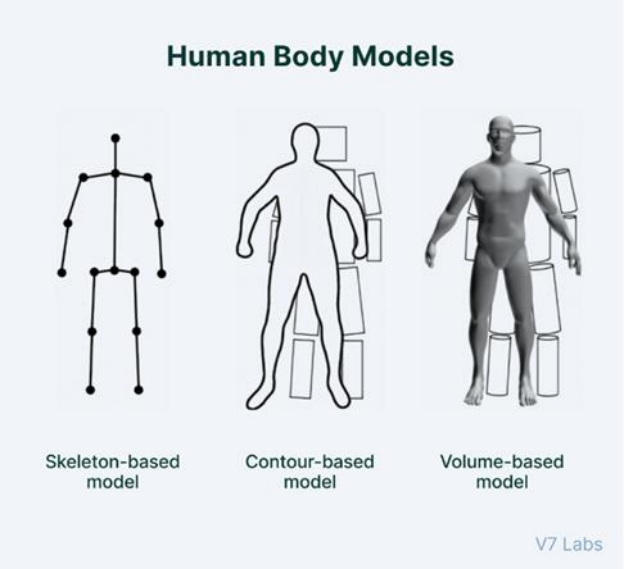

Figure 2. Human body models

The performance of semantic key-point tracking in live video requires high computational resources. The teacher's video clip may be up to 3 minutes in length, with each second consisting of 24, or 60 frames. That's 4000-10,000 frames in total. For the student's video, it can vary from 30 seconds to one minute. That's up to 1800 frames. This is an extremely large set of data to be collecting and processing.

2.2. Key Frame Detection

Within a dance video clip, the key frame is the frame/shot defining the starting or ending point of a transition (by the dancer). When we compare two video clips, one of the teacher and one of the student, both will be dancing to the same piece of music, but the video lengths will differ. Not only that, the starting and ending points of the video clips will be different. Additionally, the performers' height and weight are rarely the same, with arm, leg, and torso/body proportions making contour and volume vary greatly despite performers completing the same dance. Finally, the angle at which the videos are taken may differ.

2.3. Clustering Algorithms

It's almost impossible to compare two videos frame by frame in real time, even if they were the same length. Even if the student video is short, it could still consist of up to 900 frames. To compare the two videos efficiently, we need to group the frames together in a process called clustering, where frames with similar properties are group around a centroid set of desired properties.

2.4. Datasets

Strong datasets are a fundamental factor in a deep learning system. An extensive and diverse dataset is a crucial requirement for the successful training of a deep neural network. For our research purposes, we decided to create our own dataset, deriving data from sample videos of professionals, paired with several videos of students performing to the same piece. These videos are cut into multiple clips: 10 professionally performed classical dances are downloaded, and correspondingly, 10X6 student video clips are generated.

3. SOLUTION -- LABELLE

Our solution to this problem is deep learning-based App: LaBelle. LaBelle is structured into seven layers of processing. The first layer (L1) consists of two MediaPipe Post models (M1 and M2) to process the input video clips, one for the teacher video clip and one for the student video clip. The second layer (L2) consists of two blocks (B1 and B2) that calculate the major angles for important joints in the skeleton-based dancer model. The third layer(L3) consists of two blocks (C1 and C2) that group the angle-based properties into clusters for both teacher and student. The fourth layer (L4) consists of one block (D11) to detect the key frames for student video clip. The fifth layer (L5) consists of one block (D12) to find the centroids' feature property for each cluster of student video clip based on the starting and ending frame. The sixth layer (L6) has one block (F2) to find the matching property in the teacher video clip for each centroid's feature property generated from the student video. The last layer (L7) has one block (S12) to compare matching properties of teacher and student, generating a score to send to the user along with the frames corresponding to the centroids bearing lowest score to improve on.

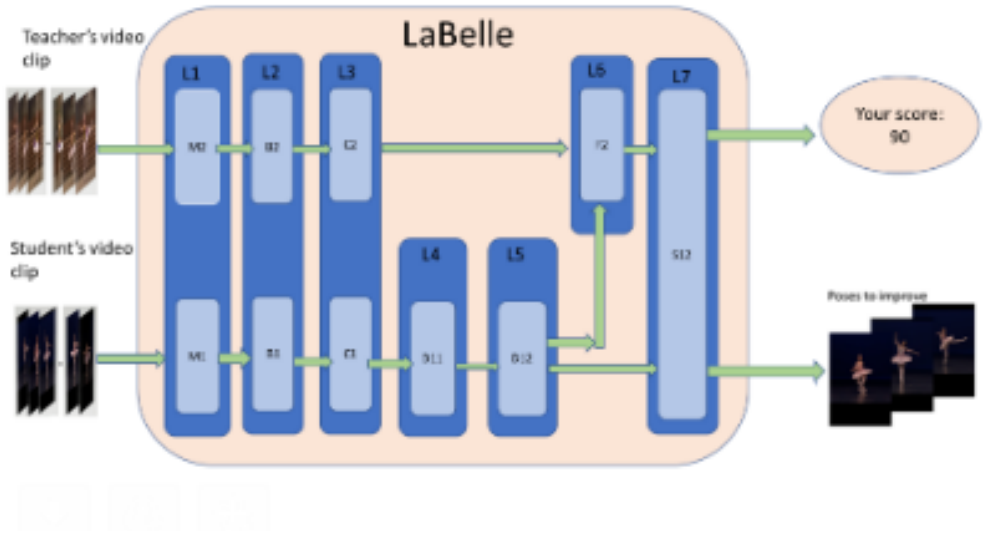

Figure 3. Our solution

3.1. MediaPipe Pose (L1 Layer: M1 and M2)

MediaPipe is a cross-platform library developed by Google. It provides high quality ready-to-use Machine Learning solutions for computer vision tasks. MediaPipe Pose is one of the solutions we use for high-accuracy body pose detection and tracking [9]. The Pose Landmark Model (BlazePose GHUM 3D) allows us to predict the location of 33 pose landmarks.

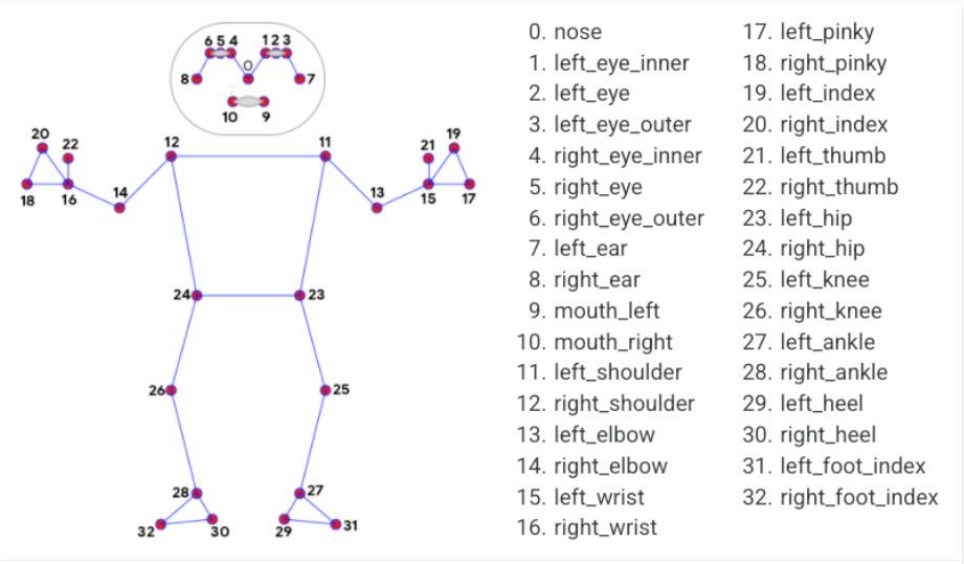

Figure 4. MediaPipe Pose landmarks [5]

3.2. Property Extraction: Angle Calculation (L2 Layer: B2 and B1)

MediaPipe Pose detects/tracks 33 landmarks, outputs 33 landmark connection. However, to improve the performance, we define a set of key angles between two landmark connections. For example, we use (∠16, 14 ,12) for the right elbow. We use (∠14, 12, 24) for the right shoulder. We use (∠24, 26, 28) for the right knee. This is done for the left side of the body as well. As shown in Figure 5 and Figure 6, we use MediaPipe's built in libraries to plot the human's pose on a 3D graph. The coordinates of 3 adjacent joints are called into the plot angle function of our algorithm, which calculates and returns the angle between the three landmarks. This process is repeated and carried out for each video frame until all desired joint angles are returned, resulting in a final tuple array consisting of all major body angles for individual video frames.

The pre-defined key angles can be altered. As dancers' skill level progresses, more angles in the hand and feet can be applied to provide an even more rigorous level of tracking and critique.

```
angle, image = plot_angle(mp_pose.PoseLandmark.LEFT_SHOULDER.value,
                          mp_pose.PoseLandmark.LEFT_ELBOW.value,
                          mp_pose.PoseLandmark.LEFT_WRIST.value, landmarks, image, h, w + val)
```

Figure 5. MediaPipe code implementing key joints into angle calculation

```
def plot_angle(p1, p2, p3, landmarks, image, h, w):
    # Get coordinates
    a = [landmarks[p1].x,
         landmarks[p1].y]
    b = [landmarks[p2].x, landmarks[p2].y]
    c = [landmarks[p3].x, landmarks[p3].y]

    # Calculate angle
    angle = calculate_angle(a, b, c)
    # print(angle)
    draw_angle(tuple(np.multiply(b, [w, h]).astype(int)), image, round(angle))
    return angle, image
```

Figure 6. MediaPipe code to generate angles given key joints

Table 1. Example of key properties (angles) for random frames

	∠16, 14 ,12	∠14, 12, 24	∠24, 26, 28	∠11,13,15	∠13,11,23	∠23, 25, 27
Frame 0	50	80	180	50	80	120
Frame 1	180	120	100	45	180	180
Frame 2	70	45	180	180	45	70
Frame 3	90	35	130	120	90	180
Frame 4	45	75	120	180	180	120
Frame 5	120	140	180	110	55	60
Frame n	30	160	150	180	45	180

3.3. K-means Clustering (L3 Layer: C2 and C1)

We limit our teacher's video length to about 3 minutes. For each second, we can generalize about 30 frames. That's approximately 5400 frames in total. For the students' video, typically 30 seconds long, there's approximately 900 frames. We want to cluster our large number of frames into similar groups based on their properties, using our key angles as properties. The videos uploaded by the user will change each time, so it's impossible to preset cluster/centroid properties. To solve this problem, we need an unsupervised and autonomous clustering algorithm.

K-means (Macqueen, 1967) is widely used in unsupervised learning algorithms that tackle clustering tasks. It uses "centroids," different randomly initiated points in the data, and assigns every data point to the nearest centroid (distance is dependent on similarity between data point and centroid). Once every data point has been assigned, the centroid is adjusted to the average of all the points assigned to it. Sklearn.cluster. KMeans works with KMeans using Lloyd's or Elkan's algorithm.

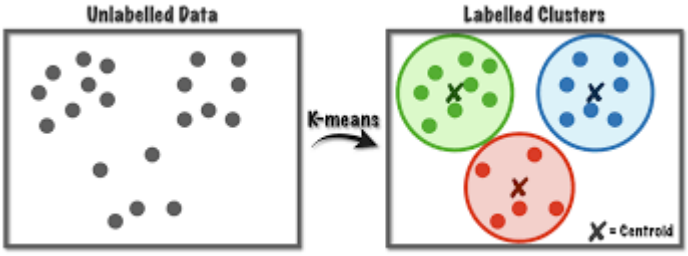

Figure 7. K-means clustering: unsupervised clustering algorithm

3.4. K-mean Hyper Parameter Tuning using Silhouette Score

Both video clips need to be processed using clustering. To determine the optimal number of clusters for both video clips, we use silhouette scoring to tune the hyper parameters. One way we can do this is by assigning a random number of clusters ranging from 3 to 30 with random initial centroid values. Alternatively, we can assign meaningful initial centroid values: by studying Basic Ballet Positions and Movements shown in Figure 8, we can summarize ballet positions into 14 initial centroid frames [15]. We summarize the properties for those unique positions shown in Table 2. With this guide, we use random selection, random range (14, 30), to determine the number of clusters, and initial centroid properties will be used to initialize the sklean.cluster. KMeans algorithm.

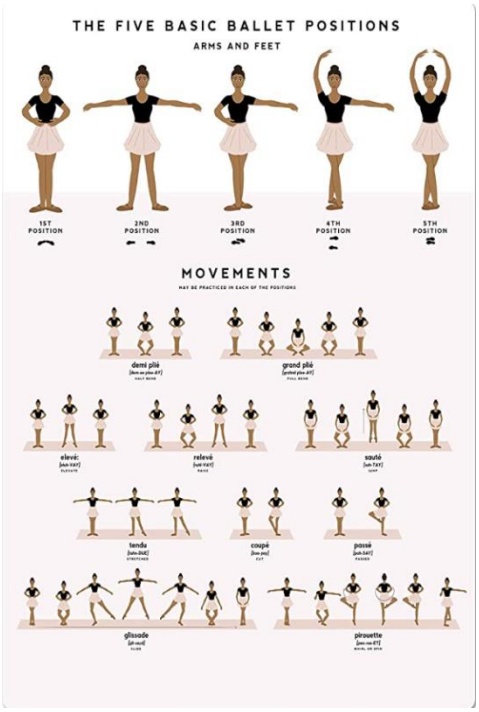

Figure 8. Basic ballet positions and movements

Table 2. Initial centroids based on basic ballet positions and movements

	∠16, 14 ,12	∠14,12,24	∠24, 26, 28	∠11,13,15	∠13,11,23	∠23, 25, 27	∠12, 24, 26	∠11, 23, 25
1st Position	120	30	180	120	30	180	180	180
2nd Position	180	90	180	180	90	180	160	160
3rd Position	120	30	180	180	90	180	180	180
4th Position	180	160	180	180	90	180	180	180
5th Position	160	160	180	160	160	180	160	160
Demi Plie	120	30	120	120	30	120	160	160
grand Plie	120	15	60	120	15	60	120	120
Coupe	120	30	90	120	30	180	120	180
passe	120	30	60	120	30	180	90	180
eleve	120	30	180	120	30	180	160	160
tendu	180	90	180	180	90	180	180	160
Glissade1	180	60	160	180	60	180	160	135
Glissade2	180	90	180	180	90	180	135	135
Glissade3	180	60	180	180	60	160	135	160

3.5. Key Frame Identification (L4 and L5 Layer: D11 & D12)

Based on the clusters generated from the student's video clip, each cluster will determine its own key frame, then the middle frame is identified and used as key frame.

3.6. Compare and Generate Final Score (L6 and L7 Layer: F2 & C12)

To compare the two videos, we go through labels of all the centroids in the student video clip and find the modified labels by fitting the centroids within the teacher's clusters [10]. Then we find the most similar property from the teacher's video clip, producing the most similar poses for both student and teacher, allowing us to calculate the score. Show in Figure 9 is part of the algorithm, given a predetermined video frame. We use matrix norms from linear algebra to help us find the smallest distance (least difference between frames). We conclude the final score and output the pose that generates the lowest score, which means least matching, for user to improve in the future practice.

```
def get_nearest_neighbor(image, indexes, frames):
    a = np.array(image)
    min_dist = sys.maxsize
    nearest = indexes[0]
    for idx in indexes:
        b = np.array(frames[idx])
        dist = np.linalg.norm(a - b)
        if min_dist > dist:
            nearest = idx
            min_dist = dist
            # print(min_dist, nearest)
    return nearest
```

Figure 9. Portion of algorithm used to find the two most similar frames

4. EXPERIMENT

In this section we give some examples with real data sets to demonstrate the performance of the proposed k-means algorithm. We show these unsupervised learning behaviors to get the best number of clusters for our algorithm to match the video between teacher and student. The dataset we are using is from the data we collected through the training. To find the best K to using in the production level we try to train the dataset for both 40 k-means random states and 80 random states, run several k-means, increment k with each iteration, and record the SSE.

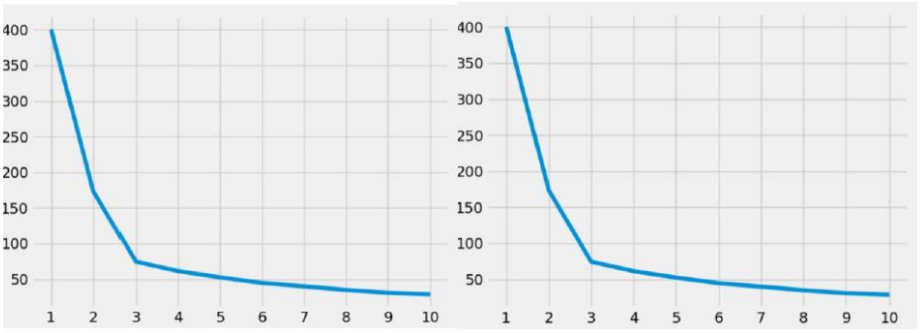

Figure 10. SSE curve with N cluster changes for tested with 40 and 80 random states

The diagram shows K bigger than 3 has the acceptable SSE control.

The silhouette coefficient is a measure of cluster cohesion and separation. It quantifies how well a data point fits into its assigned cluster. instead of computing SSE, compute the silhouette coefficient, to determine the best k for our algorithm:

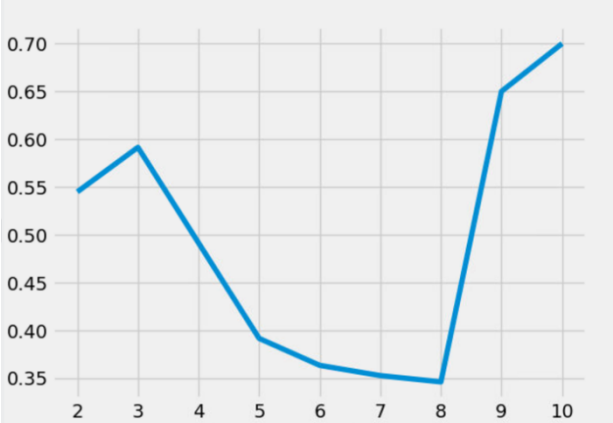

Figure 11. Silhouette scores for k shows that the best k is 10 since it has the maximum score

Ground truth labels categorize data points into group. These types of metrics do their best to suggest the correct number of clusters. To determine the best number of clusters, we fitted our dataset and get the plot.

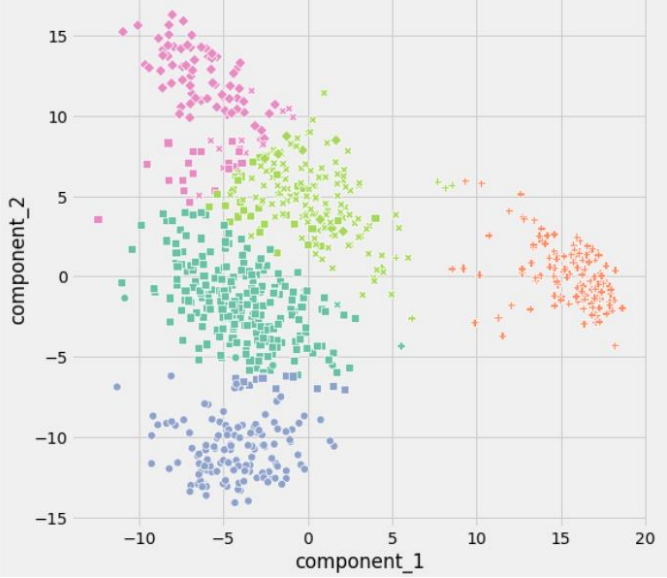

Figure 12. Ground truth labels for clusters 2, 4, 6, 8, 10

Ground truth labels also shows that cluster with 10 (orange points) has the best error value.
The comparison shows that clusters with 8 has the best result to matching the student video and coach's video.

To test the effectiveness of the application, we also create a survey in Google Forms [13]. 100 participants were gathered to take the survey, which is a large enough sample size to mitigate the effects of any variability.

Scores	Numbers
Under 5	2
6	7
7	11
8	10
9	28
10	42

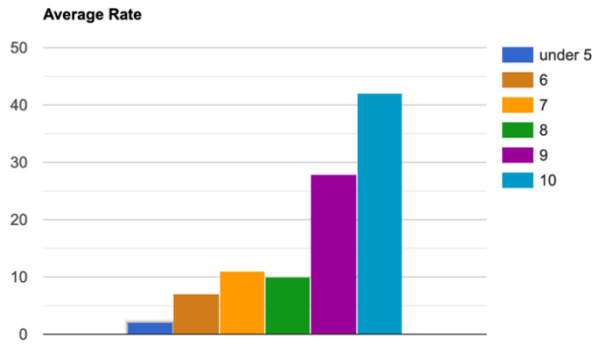

Figure 13. Survey scores for using the app

According to the results of the first experiment, we choose 8 as the best performance K values for matching the video between student and coach. The second experiment shows the app can be a great help to dancers. The participants are regarded as a small sample of the public, which means that the general public can make use of the application.

5. RELATED WORK

Several researchers have worked on the K-means algorithm. In the paper “Unsupervised K-means clustering algorithm.", Sinaga, Kristina P., and Miin-Shen Yang construct an unsupervised learning schema for the k-means algorithm so that it is free of initialization without parameter selection [11]. They also simultaneously find an optimal number of clusters. In the paper, the k-means Algorithm: A Comprehensive Survey and Performance Evaluation, Syed Islam explains the advance of k-means algorithm and discuss the using of comparison among different k-means clustering algorithms.

In the paper "Real-time comparison of movement of Bulgarian folk dances.", Uzunova, Zlatka. explain a method recording the dancing with two Kinetic sensors and analyze the data with machine learning algorithm. To understand how to apply and pick the data, we also check the research paper related to the machine learning filed, In the paper " Machine learning algorithms-a review” Mahesh, Batta published, he explained how different machine learning algorithm applying in data analyzing.

6. CONCLUSION

In this paper, we explain a method for analyzing student ballet movement by Comparing coach's movements and the student movements. We cut the coach's 3 minutes video length into approximately 5400 frames and student’s 30 second video into approximately 900 frames to compare with a K-means algorithm [14]. Our results show that to make the k-means algorithm work best, we need a good dataset and use different methods to find the best K number that fits the algorithm dataset. Both SSE curve and silhouette scores show that the algorithm is stable.

Further improvement: In the future we could collect more data from the user to improve the model performance and processing.

REFERENCES

[1] Andriluka, Mykhaylo, et al. "2d human pose estimation: New benchmark and state of the art analysis." Proceedings of the IEEE Conference on computer Vision and Pattern Recognition. 2014.
[2] Joorabchi, Mona Erfani, Ali Mesbah, and Philippe Kruchten. "Real challenges in mobile app development." 2013 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement. IEEE, 2013.
[3] Likas, Aristidis, Nikos Vlassis, and Jakob J. Verbeek. "The global k-means clustering algorithm." Pattern recognition 36.2 (2003): 451-461.
[4] Deng, Li, and Dong Yu. "Deep learning: methods and applications." Foundations and trends® in signal processing 7.3–4 (2014): 197-387.
[5] Sarker, Iqbal H. "Machine learning: Algorithms, real-world applications and research directions." SN Computer Science 2.3 (2021): 1-21.
[6] Novack, Cynthia J. "Ballet, gender and cultural power." Dance, gender and culture. Palgrave Macmillan, London, 1993. 34-48.
[7] O'Shea, Keiron, and Ryan Nash. "An introduction to convolutional neural networks." arXiv preprint arXiv: 1511.08458 (2015).
[8] I. H. Witten, "Data Mining: Practical Machine Learning Tools and Techniques, Third Edition," Morgan Kaufmann Series in Data Management Systems, 2011.
[9] Zhang, Fan, et al. "Mediapipe hands: On-device real-time hand tracking." arXiv preprint arXiv: 2006.10214 (2020).
[10] Yuan, Fang, et al. "A new algorithm to get the initial centroids." Proceedings of 2004 International Conference on Machine Learning and Cybernetics (IEEE Cat. No. 04EX826). Vol. 2. IEEE, 2004.
[11] Sinaga, Kristina P., and Miin-Shen Yang. "Unsupervised K-means clustering algorithm." IEEE access 8 (2020): 80716-80727.
[12] Uzunova, Zlatka. "Real-time comparision of movement of Bulgarian folk dances." Computer Science and Education in Computer Science 14.1 (2018): 93-108.
[13] Vasantha Raju, N., and N. S. Harinarayana. "Online survey tools: A case study of Google Forms." National Conference on Scientific, Computational & Information Research Trends in Engineering, GSSS-IETW, Mysore. 2016.
[14] Likas, Aristidis, Nikos Vlassis, and Jakob J. Verbeek. "The global k-means clustering algorithm." Pattern recognition 36.2 (2003): 451-461.
[15] Yuan, Fang, et al. "A new algorithm to get the initial centroids." Proceedings of 2004 International Conference on Machine Learning and Cybernetics (IEEE Cat. No. 04EX826). Vol. 2. IEEE, 2004.