Recognition of Effort Qualities in Human Movement

CSC 480 Artificial Intelligence
Fall 2020 Team Project

By:

Hunter Harris, Taylor Hayase, Jeremy Lee,
Emily O’Neal, Div Sharma

Abstract

The human form is a complex yet artistic model that can take on many shapes. These positions, or poses, convey information without words; a visual language so to speak. In 1982 a Austro-Hungarian dance artist named Rudolf Laban conceptualized a formal notion for these human poses that would later come to be known as Labanotation. Later on, others expanded upon Laban's work to create Laban Movement Analysis (LMA) which is a method and a language used to describe human movement. The goal of our project is to train Artificial Intelligence to identify and label a series of human poses, or a single movement, using LMA specific keywords dealing with space, weight, speed, and flow [1].

1. Project Overview

The purpose of this project was to develop an AI that is capable of classifying human movements using the keywords from Laban movement analysis. An AI with these capabilities will be able to categorize the intentions behind the movement. This AI focuses more on emotions and feelings behind all types of movement phrases. This means that instead of depending on another person or an instructor, a user would be able to simply dance in front of a camera and then the AI will be able to describe the qualities of the movement that the user just performed. The dancer can take that analysis and use it as a tool for critiquing their own choreography, to see if their intentions match the AI's analysis.

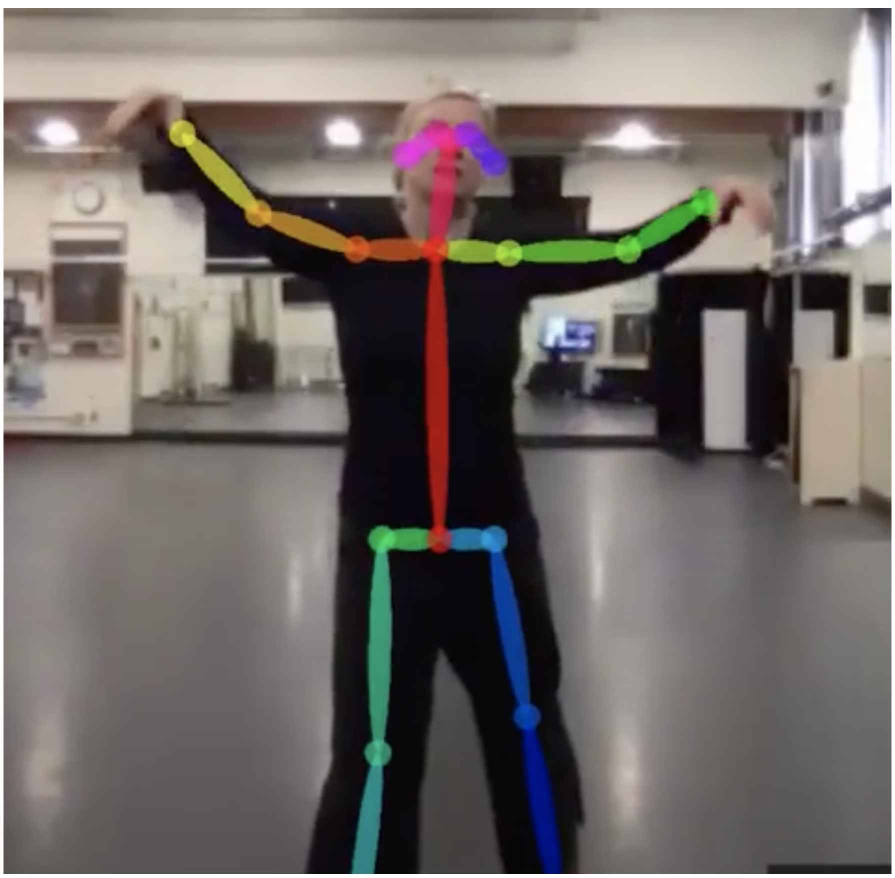
For the scope of this class, the main goal was to create and train the AI to be able to recognize and classify a set group of movements with the eight keywords. This way the AI will have a fairly solid base to start from, and can later on be expanded upon to be able to classify almost all common human movements. Pictured below are the eight keywords and how they are classified in terms of space, speed/time, weight, and flow. We will also explain later on in this report how our project goals changed in response to the challenges we faced.

Effort Category	Fighting Efforts	Indulging Efforts
Direction	direct	indirect
Weight	heavy	light
Speed	quick	sustained
Flow	bound	free

One way people analyze and describe dance movement is through its quality. It has long been debated by academics on how to put movement into words. Movement is so universal that it can seem impossible to put it into words. One solution has been to look at the quality behind the movement, such as the space and time it uses. As a dance student, we often watch each other move and comment and critique each other based on the intentions we saw in each other's dance. We then discuss and the choreographer can take that feedback to see if they were able to portray the qualities intended in the movement. This type of analysis can help dancers and choreographers learn how their movement translates into emotion and what feelings their movements give to their audience.

2. Background, Difficulty, and Relevance

Pose mapping has been a topic of interest for computer vision professionals for upwards of two decades now. In the recent years CMU has broken onto the scene with their open source pose mapping software OpenPose. This program uses deep learning to 2D and 3D map human poses from images, videos, and even live footage such as a webcam, having detail up to 135 keypoints of articulation. What CMU provides with OpenPose is a deep learning model that acts as a jumping off point, allowing developers to tweak and add to its base functionality to best suit their needs.



The technology behind pose mapping is not unique to OpenPose or CMU, and in the past couple of years this technology has made its way out of solely academic purposes and into the commercial market. This can be seen in mobile applications such as HomeCourt, an app which uses pose mapping to monitor the human body as it performs a series of exercises. Pose mapping technology is also leveraged by professionals such physical therapists, personal trainers, and dance instructors. The University of San Diego is pioneering a completely virtual physical therapist, using pose mapping technology to assess patient movements.

With each passing year pose mapping technology is becoming more and more integrated with how we interact with computers. As projects like CMU's OpenPose become more robust, its use cases will increase, and we will continue to see a steady rise in pose mapping technology within our daily lives. However, our project differs from these because we focused on movements, and instead we will just be using these technologies to assist our own project implementations.

The main challenges that we expected of this project were identifying the specific keywords within our data, dealing with inconsistencies in the images, and also applying the results into a useful and comprehensible abstract that will then train our AI. We expected this to be a challenging project since none of us have worked with AI before, so we literally jumped in with no experiences or ideas on how to actually implement what we were aiming to do. However, the dance department was very willing to help supply us with data as well as help categorize our data as needed. We expected data collection to be difficult even with how much the dance department gave us.

3. Features, Requirements, and Evaluation Criteria

Features

Our team's original goal was to create an AI capable of reading in human movements and then identifying the effort qualities within that given movement. However, after working for a few weeks on our project, we adjusted our goals to be a little more feasible for a quarter project. In this quarter we were able to both create a brand new dataset that fits our specific use, and also build a model that is capable of deriving a single effort quality. Originally the beginning of the process needed to have a method to allow users to record a short video of a small movement phrase (from a selection of movements). Then the system would have recognized the four 'fighting efforts' (strong, sudden, direct, bound) and four 'indulging' (light, sustained, indirect, bound) of that movement and give the user the words describing that quality. Then the algorithm would translate the gestures into eight basic effort action drives. After that, it should have been able to implement this AI to help beginners learn new dance moves and see if they are performing them correctly. The final feature is that the algorithm can be used to breakdown larger expert performances to its basic parts as long as all the movement phrases are in the selection.

Requirements

The main requirements of the AI would revolve around correctly obtaining data and being able to process it in OpenPose and TensorFlow to get the required accurate output of EQ's. Our original requirements were the following. The AI should be able to translate video movement to relevant data using OpenPose. In addition, it should be capable of taking multiple poses and look at the changes in those poses over time. Another qualification that we included was that it should learn to understand the different types of quality in human's movement for a specified set of moves (direct vs indirect). In addition, it would be able to take the data and categorize which quality it most falls under. Finally, a stretch goal was to have it differentiate different people's movement qualities for the same type of movement.

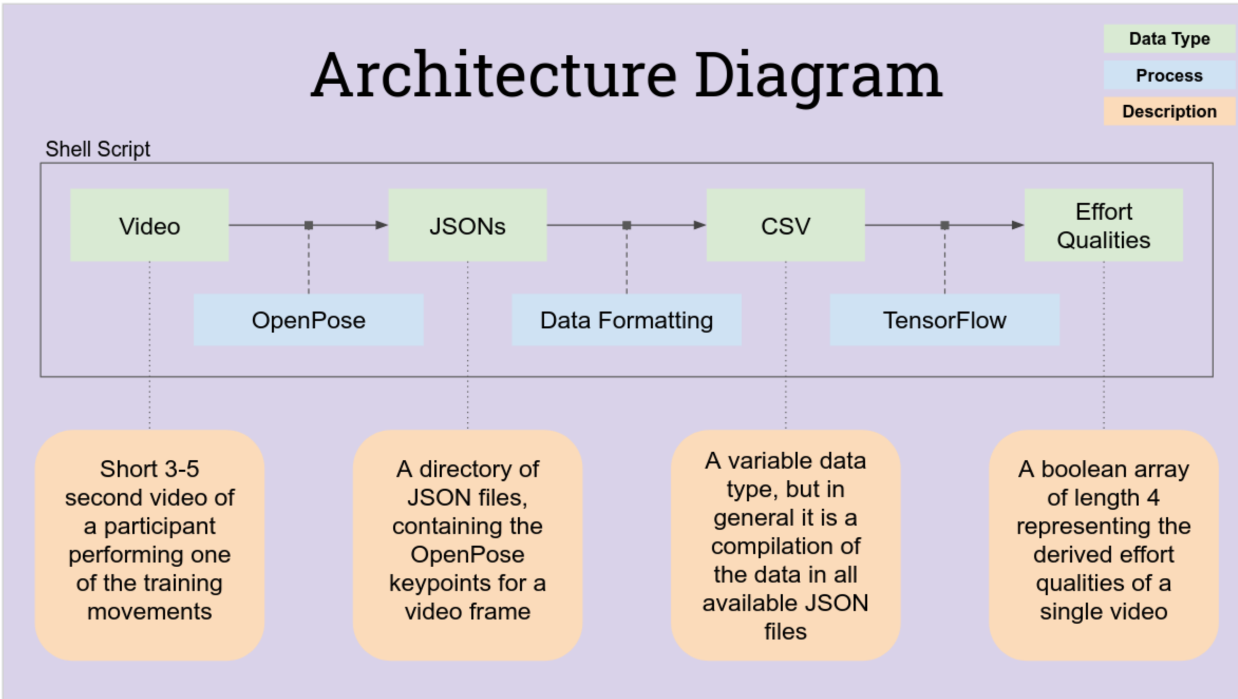
Evaluations

The following are the evaluations that we originally specified for our project. Our created dataset is diverse in its participants, where we have a fairly large range of ages, there is a fair spread of gender and race, as well as a wide variety of users with little to a lot of dance experience. We also had to sort through submissions and filter through videos to find those that most closely followed the effort qualities that we specified. We wanted the model to not be overfitting or underfitting the training data. The group also wants to explore how accurately the model can classify movements and calculate the threshold from those observations. It should be able to have above a 75% accuracy in the end and with more training up to 95%. Also, the code's time complexity based on each pose and how fast we can process it should be in a reasonable range as it needs to do this action multiple times.

4. System Design and Architecture

Functional Components

Our program consists of 2 main pillars, the first of which is OpenPose. We use OpenPose to convert our short movement videos into a directory of JSON files. The second pillar is TensorFlow, where we fit our dense neural network onto a set of formatted data. In between these two processes there is a data formatting bridge that takes our JSON OpenPose data and converts it into a desired format. We chose to use this 3 pronged approach to abstract the data formatting from our TensorFlow model. This allowed us to rapidly change our TensorFlow data types and develop a robust file hierarchy that could be manipulated at will. Separating data formatting from our model kept our code decoupled and made viewing our program from a high level much easier to think about (see diagram below).



The OpenPose section rarely changed, meaning we did not have to touch that aspect of our project after the first few weeks. The other 2 pieces could be updated to fit our needs at that time, making development more straightforward and easier to test. This clear division within our solution benefitted us greatly throughout our work.

Data Structures

While our program did go through many iterations involving multiple data types, some types remained consistent throughout development. Both types relating to OpenPose, the videos we input and the JSON files output, were unchanged once implemented. There was no need to modify our JSON file structure since our data formatting process did that work for us. In fact, its consistency is the reason we were able to develop such a robust formatting script and automate the process as a whole.

5. Implementation

In our approach to create an AI capable of recognizing effort qualities in human movement we had to both create a useful dataset, and train a model to comprehend our dataset. We planned to create a dataset of eight different movements whose effort qualities were known, and then have the AI recognize those qualities. In the following sections we will describe how we created our dataset and used that to train our AI, and we will also go into the issues we had during our implementation.

Technologies, Tools, Languages, Development Environments

The first step in gathering our data required us to determine what exact movements we were going to use for our dataset. Factors such as movement duration, environment space, and replicability were factors that we had to keep in mind when first planning out our data. After determining the set of movements that we wanted participants to replicate our focus shifted to how we could best gather all of the required videos. For this task we employed a VideoAsk, which allowed users to view a short five second video, then record their response mirroring what they had seen. Below are the eight movements that we asked our participants to replicate.

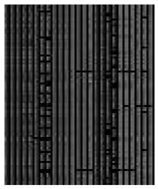
	SPACE/FOCUS	TIME	WEIGHT	FLOW
PUNCH	Direct	Quick	Heavy	Bound
DAB	Direct	Quick	Light	Bound
PRESS	Direct	Sustained	Heavy	Bound
GLIDE	Direct	Sustained	Light	Free
SLASH	Indirect	Quick	Heavy	Free
FLICK	Indirect	Quick	Light	Free
WRING	Indirect	Sustained	Heavy	Bound
FLOAT	Indirect	Sustained	Light	Free

From there, the next hurdle was to take each video and track the position of keypoints on the body. This is where OpenPose, a body tracking software developed by CMU Perceptual Computing Lab, came into play [2]. We use OpenPose to turn our videos into JSON files containing positional data for each of our keypoints. This decision was made because we want our data to focus solely on the body movement rather than the video as a whole.

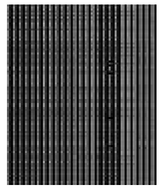
Now that we finally have a useful dataset crafted, the next step was to run it through TensorFlow. We chose TF, a machine learning platform that provides a high level toolset useful for rapid development of your own ML model, over other competitors like PyTorch due to its extensive online documentation [3]. Getting our data into a suitable format required some extra work (more on this in the Obstacles section) but we ended up converting our directories of JSON files into uniformly sized images, then using an image recognition algorithm to iteratively extract 1 effort quality at a time over the course of 4 separate runs. The specific layers we used for our TF model were provided by Keras, the ML base behind TF which acts as a wrapper surrounding the entire technology.

Prototype Functionality

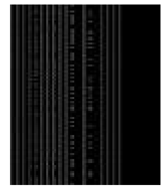
Our initial prototype for TensorFlow was based on Keras' image recognition capabilities. We converted our JSON keypoint files into images for processing, then labelled them with the corresponding motion (see figure below).



slash



glide



float

At this stage in our development we were attempting to derive the motion name instead of Effort Qualities. This was a decision based on the specifics of loading data into TensorFlow, and we were planning on using it as a proof of concept before working further to get the Effort Qualities themselves. Unfortunately, this implementation did not yield meaningful results, as our model struggled to achieve above a 26% accuracy. With no solid pattern being detected, we decided to pivot to our current implementation and scrapped the image recognition approach.

Obstacles and Implementation Issues

Since starting this project, our team has faced several issues and obstacles, some which we were able to overcome and some where we are still trying to find better solutions to.

The first obstacle was deciding which movements we would be focusing on. We needed motions where each effort quality was clearly visible for OpenPose to track, while also being easy enough for non-dancers to be able to replicate as well. During this point in planning, we also had to take into account the environment of the participants since many of them would be recording from a bedroom and thus would not have a lot of space to move around in. With the help of Diana Stanton, the Dance Program Director and a professor in the Department of Theater and Dance at Cal Poly, we were able to narrow down our options into eight easily replicable movements. Following that decision we had to figure out exactly how we would go about gathering our dataset. Since this project was started during quarantine we needed to first think of how we could get data from participants. Fortunately we found a site called VideoAsk which allowed for the participants to record videos of themselves that could then be downloaded and processed through OpenPose. This was paired with a Google form that we sent out to friends and family in hopes of expanding our dataset.

Remote data collection proved to be a multi headed issue over time. As previously discussed, collecting data became increasingly difficult, however the real problem came to light when we started to go through the few submissions that were received. Video qualities varied widely, and while we tried to provide general guidelines such as having your entire body in frame, many videos ended up being entirely unusable. Some submissions that did meet the minimum requirements to be used did not portray the effort qualities well, adding more ambiguity to what was supposed to be a training data set.

Technical issues were also present throughout, mostly stemming from TensorFlow's strict data formatting rules and the unpredictable nature of our video submissions. Our initial approach saw us converting JSON directories into CSV files, but TensorFlow was not willing to assign a single label to a 2D array of varying length. This did however lead to our idea of converting the JSON directories to images instead, each being 75 pixels wide (the number of keypoints we were tracking) and X pixels long (the number of frames in the video submission). Unfortunately we hit a similar wall where all image inputs were required to be the same size. As a short term solution we decided to trim our video images down to the shortest video present, cutting off lots of meaningful data in the process.

To fix this issue, the AI was trained to focus on just the right wrist values. We hoped to remove extraneous data that was making the critical values more difficult to extract. Approaching TF again, this time using a single array of the wrist keypoint values in each frame of the video, we created a new ML model with layers best designed for finding trends in numerical data. We stuck to the binary classification of a single EQ as our test since it is the best representation of our final vision, as well as a highly repeatable process that can be iterated over for each EQ. Our determination and intuition paid off and as of now we are identifying the time EQ with a 75% accuracy. The next step is to implement a model for each EQ, then tie the whole thing together to predict all 4 EQs of a given movement.

While we have overcome many obstacles thus far, there are still quite a few issues we are currently attempting to work through. Data loss/accuracy is our #1 concern right now and we are looking into alternative solutions for this issue. It would also be ideal to find some way to increase our current dataset, as it is not nearly large enough to provide meaningful predictions for untrained data.

6. Validation and Evaluation

As a note, some of our initial requirements became irrelevant after making changes to our project, so we came up with five revised requirements based on our five initial requirements.

Our first revised requirement we set for ourselves was to gather a diverse dataset, with "diverse" pertaining to the number of people who performed the movement. Different people will perform the movement with different speed, technique, and style, so each unique person meant a more diverse dataset. We collected data (with the performers' permission) from dance students at Cal

Poly, and students from the two CSC 480 sections. This resulted in a small but respectable, eighty video dataset. Given that we identified our needs and gathered a dataset starting from nothing, this requirement was a success. We hope to provide the dataset to future projects interested in studying human movement.

The second requirement we set was being able to translate the video movement to relevant data. This was achieved through a successful implementation of OpenPose, as it translated videos to a set of points representing body parts in a JSON file. We did consider a few other pose-mapping libraries including WrnchAI, but we ultimately decided on OpenPose for its flexibility, reliability, as well as clean data formatting. In addition, OpenPose outputted one JSON file per video frame, so we used a script to translate a complete video's JSON files to a single csv file.

The third requirement we set was being able to track the movement of specific keypoints over time. This requirement is similar to the second requirement, but actually marks an important decision we made in our project. After getting low accuracy on the initial approach, we extracted just the point representing the performer's right wrist point to reduce extraneous data in the model. This approach increased our model accuracy as it became able to identify patterns within our data.

The fourth requirement was to be able to derive a single effort quality for any of the set movements (ie. speed: sudden vs sustained). We were initially trying to predict all four effort qualities for any given video. However, we changed our approach to predict a single effort quality as a binary output, using the point of the right wrist given the time. The average accuracy for a training dataset is around 75%. Adding more epochs helped us reach a max of 80%. Of course, we might have different accuracies with different training/test sets, but there we didn't want to overfit our data.

The final requirement was to automate the process of running the program from videos to effort qualities. Some of this endeavor was mentioned already, but a lot of effort was spent writing scripts to automate the running of OpenPose over multiple videos, the compilation of the OpenPose JSON files for a single video, and cleaning the csv data into a readable training set for the model. Overall, we managed to meet most of our requirements and are extremely satisfied with the final result.

7. Conclusions

Though the scope of this project may have shifted over the quarter, we were able to reach a satisfactory conclusion. The program was able to effectively identify and label a series of human poses, or a single movement, using Laban Movement Analysis (LMA) specific keywords dealing with directions, weight, speed, and flow. In order to do this we needed to gather suitable data and create and train our own model. However, our team faced many challenges throughout the quarter which led us to have to adjust our goals for the project. Our many challenges included gathering relevant data, formatting the data for our model, building our model to process our data correctly. The main system of OpenPose and Tensorflow aided in this endeavor and helped us form a cohesive program to identify EQs. Despite the challenges we faced and the adjustments that we had to make we were still able to produce a final product that we are all proud of.

We hoped to be able to use this AI as a tool for dancers, so that the AI would be able to watch the dancer's movements and then tell them what effort qualities their movements emulated. For example, a choreographer may be developing a movement with the intention of showing grief which would need to be direct and bound. The AI could watch the choreographer and signal when a movement strayed from those core efforts. However, the uses for this AI are not limited to just dancers, and our newly created dataset is not only for our use. In addition the base algorithm of the processing of video data from OpenPose to Tensorflow for evaluation can be repeated for multiple other experiments. Therefore, even if our AI is not fully completed, we were still able to contribute new data for future uses.

Being a brand new project we were also left with a lot of future possibilities of further expansion and exploration. During the development of our project we learned many different lessons and we made several plans for how this project can be expanded in the future. We learned a lot of lessons about building up a useful dataset as well as building and training our own model. After the first few submissions outside our group also showed us the importance of clarifying the instructions as much as possible, and also the importance of going through each submission to make sure they contain the correct efforts. When we first started creating our model we tried to give it all of our data and simply asked it to identify all of the effort qualities present in it. However, this proved extremely inaccurate and we had to adjust our model to only focus on a single point and then identify a single effort quality. This proved to be more accurate, and we were able to get a 75% accuracy rate as opposed to the 50% we were getting before.

We also have many ideas and plans for the future of this project. Because we were only able to successfully identify a single effort quality in the time that we were given this quarter, we aim to expand this model to identify all four effort qualities in future development. We are also hoping to expand our dataset in the future to hopefully improve the accuracy of our model. Once we have a nice and established dataset, we want to try and expand the model to be able to identify effort qualities in all different movements, and not just the gestures that we used to train the model. We would also like to create an application that gives feedback in real time, given OpenPose has this capability. All of these expansions and improvements can be taken on as future AI quarter projects, or even a senior project, but we hope to see this project continue to grow even after this class.

8. References

[1]I. Bartenieff and D. Lewis, *Body movement*. New York: Gordon and Breach Science Publishers, 2002, pp. 57-58.

[2]G. Hidalgo, Z. Cao, T. Simon, S. Wei, H. Joo and Y. Sheikh, "CMU-Perceptual-Computing-Lab/openpose", *GitHub*, 2020. [Online]. Available: CMU-Perceptual-Computing-Lab/openpose: OpenPose: Real-time multi-person keypoint detection library for body, face, hands, and foot estimation. [Accessed: 03- Dec- 2020].

[3]Y. Tang, "tensorflow/tensorflow", *GitHub*, 2020. [Online]. Available: tensorflow/tensorflow: An Open Source Machine Learning Framework for Everyone. [Accessed: 03- Dec- 2020].

9. Acknowledgements

We would like to thank:

- Diana Stanton, Professor and Head of the Dance Program at Cal Poly
- Frank Kurfess, Professor in the Computer Science and Software Engineering Department at Cal Poly
- The Dance Appreciation class of Fall 2020
- And all of the participants who submitted videos for our dataset