CYSF 2024 Journal
Karate Scorer

January 5

My current idea looks like the following:

Pre-trained CNN → Landmarks → Process to LSTM → Classification if it's a hit or not

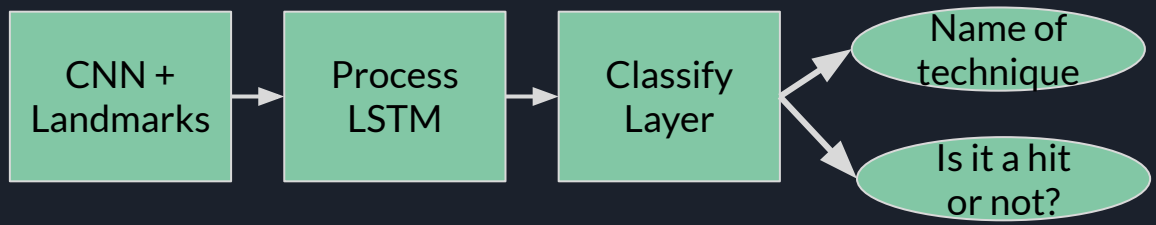
Could be an interesting idea, but need the following:

- What data do I need? How many clips of people fighting or doing punches?
- Will the final classification layer have two outputs → the name of the technique and if it's a hit?

Possible list of techniques:

- Blocking/Idle
- Punch → how many variations of the punch?
- Kick → side, round, hook

Last layer should have an output dimension of 5 maybe?



Existing CNN pose estimator?

I don't think I should re-create my own CNN structure. This tutorial says media pipe is good:https://bleedaiacademy.com/introduction-to-pose-detection-and-basic-pose-classification/

Looking at Google's website, it's not too bad. It has a high frame rate, multiple poses and outputs the coordinates of the landmark points. How can I extract these key points?

Anyways, I managed to extract key points, but should I plot them in 3D? Some work I've seen does it, maybe I can assemble a video review with multiple angles?

Jan 15: Related work with Karate and Classification?

https://www.researchgate.net/publication/363541990_Applying_Deep_Learning_and_Computer_Vision_Techniques_for_an_e-Sport_and_Smart_Coaching_System_Using_a_Multiview_Dataset_Case_of_Shotokan_Karate

This paper trained an OpenPose model to extract keypoints. They then utilized an LSTM to classify them. Other approach they used were Spatial-Temporal Graph Convolutional Neural Networks. Additionally, it was only images, not necessarily fighting or sequences of images

- Further reading tells me it's a method to classify input data that are graphs. Since the skeleton is a graph, they used this method.
  - Idk if I want to do this as it can be a complicated.
  - Could be an interesting approach

Jan 15: Related work with Karate and Classification?

https://www.researchgate.net/publication/340636808_iKarate_Improving_Karate_Kata

Used microsoft's "Kinect" sensor for the pose estimation and classification tasks. I could use the kinect sensor, however the documentation seems a bit old. Additionally, I don't feel like learning another coding language…

- I feel more inclined to use Google's Media Pipe

An interesting idea came to mind: For Kata or punching, I could maybe consider the different angles related to the arms, legs and shoulders for classification?

Another thing important → They normalized datapoints!

Jan 15: Related work with Karate and Classification?

https://www.mdpi.com/1424-8220/21/24/8378 Kinda a similar approach to what I've previously read. They utilized OpenPose for keypoint extraction then utilized a decision tree or MLP from Weka to classify the points.

Something interesting I saw was that they considered the different heights of the person. Like the previous example, should I normalize the data points so people with different heights aren't interfered?

- Actually, maybe not – just for kata. It's important to keep track of height in sparring.

Jan 15: Related work with Karate and Classification?

https://www.youtube.com/watch?v=nTUs6F4CbTc&ab_channel=AutomataLearningLab

Talks about how they applied computer vision to jujitsu . He outlines this technique called Dynamic time warping:

Jan 15: Related work with Karate and Classification?

Other interesting papers:

https://papers.academic-conferences.org/index.php/ecair/article/view/930

Other ideas:

Create a game tree with the types of punches and kicks thrown and the opponent response.

- Note change in position

I can also measure how aggressive someone is by recording where they are in the ring. I need to find a way to normalize the values so that the measurement of the ring is somewhat reliable.

- How do I measure the quality of the punches or techniques?

- Got things working
- I need to improve the latency → is this the OpenCV library fault or the Mediapipe?
- Did some research, turns out the media pip runs on 30ish FPS while most sport related videos should be at around 60FPS

Challenges (Feb 20)

I tried the mediapipe implementation, however I am running into issues since fighters overlap and cause tracking to be fuzzy. The following methods did not work:

- Average colour tracking
- Comparing distances
- Finding the direction that each athlete was heading to keep track of them.

I decided to switch to a yolov 8 pre-trained model that has better inference, but less key points unfortunately.

Model testing

- I needed data augmentation techniques