SnowScore

A Pro Skier Classification Using OpenPose

Ying Guo: adaguo@stanford.edu
Liyu Tan: tlyhenry@stanford.edu

Introduction

Watching people practice skiing skills on slope, it's quite straightforward to recognize the expertise level with their movements, tracks and body positions. Currently, some researchers are using ski-pros' motion and positional data, which are collected from the wearable IoT sensors at 8 body key positions, to classify skier's expertise level. In other word, skier's expertise level are able to classified by positional and motion data at key body position.
The challenge we want to take in this project is to eliminate the strong dependency on IoT sensor for motion and positional data collection. After we did some research on existing computer vision DL framework, openpose came into our sight, which is a well built framework to extract key body point for pose estimation, and the data we want to leverage is the existed online video data.
More concretely, we take ski-pros' and amateurs' video recordings as input to run with openpose body-25 model to extract motion and positional data at 25 body key positions, and trained with proper neural network to get score to represent expertise level.

Current Progress

The first challenge we have is how to collect the data, we do understand that the data is really important for deep learning. And in our case, there are no existing ski skeleton data in internet. We have tried several open pose estimation, and find the openpose we currently use is what we want: we give it a video, and it will generate a sequence of json file, each file contains the people_pose skeleton, that contains 25 points for it.

Data Collection

- **Video Download**
  we use the youtube-dl to down the sequence of urls which are getting from a python crawler.
- **Video Split**
  we are using the video-splitter to split the video into 5 seconds.
- **Video data cleaning**
  we found that in a video, there are many places that only have scene, or the skier is too small, or too many skiers, those will make our data in low quality. So we manually see the video, and removing theose low quality video slots.
- **Extract key point using openpose**

We spend a couple of days to setup the environment for running the open pose in our computer. Then, we tried to use a video as an input, however, it took a really long time to get the results. And for a 5 second video, it took 20 - 30 minutes for us to get around 80 json outputs in CPU mode

Now, we have the sample data. The next step is prepraring data for training.

Data Pre-processing

- **Extract dataset from json file**
  As explained in data collection, we have 100+ json files extracted from 5 seconds video. For each json file, we can get the pose skeleton data for one frame, which we have 25 data in the form (x, y, accuracy). To have well format data, we currently use a cut off value for the number of frame and eliminate accuracy.
- **normalize and reshape**
  We will get 75 * 25 sets of (X, Y) for now. And we do a normalization for the x and y by using the formula (xi - x_min) / (x_max - x_min) for each video. Then, the input become a numpy array with shape (1, 2 * 25 * 75).

Baseline Model

Due to we spend a lot of time to collect, clean and process training data, our baseline model binary classifier is pretty simple, which is 5 layers Neural Network with (256, 128, 16, 1) hidden units and activation function is relu and sigmoid, and loss function is binary cross-entropy.
Currently, because our data set is very limited and lack of diversity, our model is overfit. We will fix it later.

Remaining Work

Data

[the most risk part, due to computational power and lack of data diversity]

- **Collecting More data**
  We have to increase diversity of data, especially for amateurs.
- **Continue cleaning and pre-processing the collected data**
  Running pre-processing open-pose on data is very time-consuming. For cpu-only settings, 5s data usually requires more than 2hr to run body-25 model. We are setting the GPU cloud machine to speed this process up.
  Based on the output of currently preprocessing data, we need to find a better way to handle those output have invalid data.
  We also need to find a better way to normalize our data.
  Based on our previous data processing experience, the accuracy of open pose key point extraction strongly related to the size of person in the scene. We have to do data cleaning up for those un-qualified data before running open-pose to save limited computational power.
- **Setting up GPU cloud server to speed up open-pose processing**

Training

- **Model**
  After simplify data with open-pose body-25 model, the training data dimension has reduced significantly from **640(width)*360(height)*75(frames)*m(# of examples)** to **25(key body position)*2(x, y)*75*m**. Due to the size of training data, we save a lot time for training on iterations and experiment on different model. We have 2 training model on our list to experiment, a CNN vgg 19 model and a sequential model.

- **Loss Function**
  Ideally, we want to have an exact score to represent skier's expertise level. However, due to the huge barrier that we encountered in data preprocessing, currently we have to simplify the problem to train a binary classifier to classify pro-skiers and amateurs. The loss function we are using is binary cross-entropy.
  If we could overcome the data barrier, we will spend time to review the loss function, ideally, we want to give a score to represent skier's expertise level.
- **Evaluation**
  The metrics we will use are precision and accuracy.

Citation:

- Open pose framework from CMU
  https://github.com/CMU-Perceptual-Computing-Lab/openpose/blob/master/doc/prerequisites.md
- GolfSwing classifier using open pose
  https://github.com/personableduck/GolfSwing
- IoT sensor based skiing expertise classifier
  https://www.microsoft.com/developerblog/2017/06/29/iot-sports-sensor-machine-learning-helps-amateurs-up-their-game/
- Video split for data pre-processing
  https://github.com/c0decracker/video-splitter
- Youtube downloader
  https://github.com/ytdl-org/youtube-dl