Identify plays based upon video footage

Karam Shbeb
Innopolis Uinversity
k.shbeb@innopolis.university

Mohammad Shahin
Innopolis Uinversity
m.shahin@innopolis.university

Mahmood Darwish
Innopolis Uinversity
m.darwish@innopolis.university

***Abstract*—This document is the Project Deliverable D1.2 for the Practical Machine Learning and Deep Learning course. Our project aims to detect football (soccer) passes, including throw-ins and crosses. Github repository.**

I. Creating the training set

During our work we first needed to create the training set. First, instead of looking at events as if they happen in an exact timestamp, we created a small range around the event timestamp. We took all the frames in that range and added them to a folder with the same name. At the end, we would have 4 folders each folder will be filled with the images from frames where an event was happening. It is true that we have 3 events to catch (passes, throw-ins, and challenges) but we also added a 4-th kind of event to be caught. That event is "background". Basically, every frame that doesn't belong to an event will be added to the event "background". Because we need to train our model on the frames where no event is happening too. That way we transform the problem into a 4 class classification problem. We will input for the model a time series of photos (a video) and for each photo it will predict which one of the 4 classes it is.

II. Base-line model

For the model, we decided to leverage transfer learning and go with EfficientNet pre-trained model. That model has shown great results with image classification problems as shown here. Right now we're using the "tf_efficientnet_b5_ap" implementation from the "timm" library as is. We're even using the training script from "timm". We're training the model on the data but cutting up the videos into pictures and training the model on them is taking much more time than expected. We weren't done with the training before writing the report. What we did is make an option in our script to only cut up to 10 frames from each video and train the model on them with a small epoch number. This allowed us to confirm that at least the script is working. The notebook can be accessed by going to the git repo and going to the folder notebooks. The notebook is titled "model_creation".

III. Work to come

Currently, our plan is train the model fully. Then write the code to test the model. We assume the performance is going to be abysmal, this is because we are only cutting up the videos and feeding them to the model without any modifications and expecting a result. We are gonna use this as our base-line model and try to improve on it. We will try different parameters, maybe different pre-trained models and try to get the accuracy to an acceptable level. We might also try different data pre-processing steps to help the model (like gray scaling the input for example).

IV. Work distribution

In this section of the work, Karam worked on cutting up the videos and preparing the training set. Mohammad and Mahmood both worked on model selection by doing some research on image classification models. Mahmood also wrote the report.