Lauretta.io Assessment
Question 2: Video-based Action Recognition

Sports Action Classification:

I have implemented a video action classification model specifically for some sports actions as through literature and project survey, I couldn't find many projects which specifically deal with video-based sports action classification. Therefore, I was interested in making a relatively smaller scale model for this task.

The first idea of development included using sports labels from the Kinetics400 dataset but due to constraints on size (400GB) and timeline for the assessment, a smaller UCF101 dataset was used for this project. UCF101 dataset contains around 13 diverse labels for sports-based actions, which can be found in the following link.

CRCV | Center for Research in Computer Vision at the University of Central Florida (ucf.edu)

After selecting the model, the next step was to choose an action recognition. Through work experience in this topic, I found that the MMaction2 library which is based on Pytorch is a great choice for fast implementation of state-of-the-art video models. I choose a pre-trained model called I3d based on the following paper

CVPR 2017 Open Access Repository (thecvf.com)

I3d model comes pre-trained on the Kinectics400 dataset, and the implementation is provided by the MMaction2 library. This model was the state-of-the-art video-based action recognizer model. After preparing the dataset (following section). The model was trained and tested on the UCF sports action dataset, after initial few hyperparameter optimizations, the model produced around 87.88 percent accuracy and 100 percent Top5 accuracy, making this initial attempt at fast small-scale sports action recognition model successful.

Model prediction example:

Data Preparation Guide:
Code- data_preparation.ipynb

I am using a model trained on Kinetics400 therefore the UCF101 dataset needs to be in a certain format. This code processes the dataset such that it can be used with the I3d model. The final dataset can be downloaded from the following google drive link, this can be directly used with the MMaction2 library, it is not required to run the instructions again.

https://drive.google.com/drive/folders/16xW-zZwH9DIJCnJi0Z0Iw0P0NrgmED4G?usp=sharing

- To prepare the dataset, first, we download using the CRCV link from the previous section and find the “ucf action” folder extract it, and store it as “ucf_action”.
- Since the video format is in “.avi” we need to convert it to “.mp4” to be compatible with the MMaction2 library. We achieve this in the last cell of this notebook.

Model training and testing guide:
Code: training_and_testing.ipynb

- Follow the 1-3 steps to install the MMaction2 and its dependencies.
- Place the “ucf_action” folder from the google drive link in the “mmaction2” folder created by the previous step.
- To train, we need I3d kinetics400 model, which can be found in the following link

Action Recognition Models — MMAction2 0.24.1 documentation
Find and download i3d_r50_video_32x2x1_100e_kinetics400_rgb
and store it in the "checkpoints" folder.
The model can be trained following steps from the notebook and every 5 epochs the model will be saved in “tutorial_exps” folder in mmaction2

After training the model, execute the next step to get the results on the test dataset. Following is the cell output.

[>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>] 33/33, 0.6 task/s, elapsed: 57s, ETA:
0s
Evaluating top_k_accuracy ...

top1_acc 0.8788
top5_acc 1.0000

Evaluating mean_class_accuracy ...

mean_acc 0.8692
top1_acc: 0.8788
top5_acc: 1.0000
mean_class_accuracy: 0.8692

To test a saved model, go the testing from saved model section and provide its path in the cfg.load_from parameter and run the cell.

This model can be executed in real-time, where the video stream after certain frames can be saved and given as input to the model which can provide the predicted labels.