**MLOPS**

**Project Report**

****

i191669 Zohaib Bashir

i191723 Zaafir Rizwan

i191910 Shazil Ahmed

AI-K

# Introduction

The name of our project is Deepshot, basically what our project does is it takes a video input and then finds major events in them and then extracts those events into clips and finally those clips are merged together to form a highlight video.

# Git

As the project was completed when we started MLOPS, this part we didn’t use as such. So, we didn’t use version control. We utilized other things such as code management and collaboration. We were also able to perform some GitHub actions as well. From what we learnt from Assignment 1 we applied some workflows on the GitHub repo. We also created 3 branches and for each individual member as well as having a central repository where the other 2 members could do a Pull Request and can merge with the central repo.

# Docker

For this portion, we dockerized our customized model. After some experimenting, we were able to successfully dockerize our model. Due to the nature of the project, the size of the image/container was very heavy as we used many technologies that utilize many dependencies so because of that the docker image/container was very heavy. We researched and utilized a docker multi-stage build to lessen the size of the project.

# Jenkins

Now that we have a dockerized version of our code, we just had to link it to Jenkins. For this we made a job on Jenkins that clones the GitHub and then runs it. We utilized the cloud version of Jenkins as it can be accessed from anywhere and any team member can access it. We also had to utilize GitHub Webhooks. But as told above due to our docker image being very large, it imposed some challenges on importing it to Jenkins. After a lot of trial and error we were finally able to upload/push/pull the image and ran a job/built a pipeline for the project

# DVC

For DVC, we used it for saving our model file and for dataset versioning. Using DVC, we can not only look at the different versions of model file outside of MLFLOW but we can also see the changes/versions of the dataset

# MLFLOW

MLFLOW was used in tandem with Airflow. Check below for its explanation

# Airflow

Airflow and MLFLOW were used together for this project. Basically, we created a pipeline where several things occurred. First what will happen is that our remote storage (Google Drive) will be searched to find a particular file, if that file exists it will move onto the next step which is to download the dataset/directory. After that was to clone a repo that our project utilizes by the name of YOLOV5. Then we would simply train and perform tests.