

Projects Portfolio

This section better explain some projects and their technical details

Multimodal Violence Detection System (Master Thesis research Project)

This project it's been developed during the 6 month of internship at the Fondazione Bruno Kessler(FBK), a highly qualified research center based in Trento. To address the specific task of identify the violent moment in a video, we have used a Multimodal Deep neural network which accept in input two different modality, audio and video. The Network is composed by two parallel branches(ResNet50 was used for both modality), one for each modality, which are cleverly structured to optimize computations in a dynamic way. The project was implemented using Python with libraries such as PyTorch, OpenCV, Torchvision, ffmpeg and including the most common libraries(dumpy,matplotlib, etc..). For the project management and updates we've used Docker and Git.

Pipe counting Algorithm

This project it's been developed for the exam of signals images and videos. It was a group project(two people per group) and we decided to use classics computer vision techniques to recognise and count different shapes of pipe(circles, squared). our approach was simple and effective, we first implemented a preprocessing pipeline for enhancing some specific characteristics of the images, and then we defined some dynamic filters for the different shapes which will be used for reference to recognise the shape in the image using a threshold for match ranking. The main library used for this project were numpy and openCV, the programming language used was python.

Camera Motion Tracking

This project was designed and implemented for the exam of Computer Vision. The task was to extract the motion path of a moving camera by it's recorded video, the input is the video itself and from it I extracted the motion path by using a combination of two algorithms, SIFT(scale-invariant-features-transform) to identify the best features to track and knnMatch used to match the features from different frames. The algorithm was implemented using OpenCV and numpy in Python language.

Deep Reconsruction-Classification Network

This group project(two person) was made for the Deep Learning course. It consist of implementing an Unsupervised Domain Adaptation strategy using the dataset provided in the course, which in our case was Adatiope, the dataset was composed by two different domain of the same class(real world object and synthetic object). we used the DRCN(Deep Reconstruction classification network) domain adaptation technique which consisted of jointly training 2 networks, a classification network and a image reconstruction network, this is done to ensure a "good" domain-invariant feature representation. We mainly used pytorch to build and train the network.

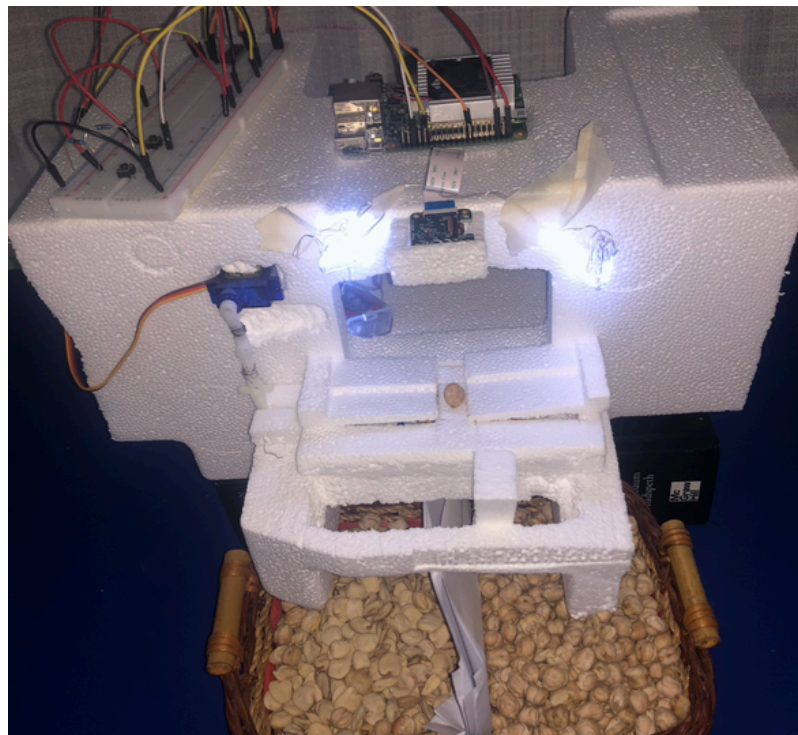
Sentiment Classifier in NLP

This project it's been developed for the exam of Natural language Understanding. The task of this project was to train a neural network using a dataset of sentences from different situations to classify the sentiment of sentences. It's important to notice that one of the most important part of this project was preprocessing the data in the right way (removing punctuation, tokenisation...). After the processing of the dataset I've used BERT encoder from the Hugging face Library to extract a representation vector to train the classifier reaching an accuracy on the testing set of approximately 91%.

OpticSorter: Programmable Optical Sorter Based On Deep Learning and Computer Vision (Bachelor Thesis Project)

Implementation and creation of an actual prototype for classifying agricultural products. the prototype was capable of distinguishing a rotten fruit from an high quality fruit of the same type. the prototype was build using recycled materials as polystyrene. I used a single board computer (SBC) developed by google called CORAL, specifically designed for AI application deployed on the spot. I created a dataset of 2500 (1250 rotten bean and 1250 high quality bean) images to train a convolutional neural network (built in Tensorflow and Tensorflow lite).

here I show an image of the actual prototype trained to distinguish chickpeas from grass peas:



Parallelising the k-means algorithm for faster performances on server cluster

group project (two person) implemented for the course of High Performance Computing. we created different synthetic dataset (using python) for testing with different dimensionality and different sizes, we implemented the K-means algorithm, which is a fundamental algorithm for unsupervised clustering and grouping, using C. Then we parallelised the k-means by using two different approaches and libraries (OpenMP and MPI). With MPI, each process has its own memory space and executes independently from the other processes. With OpenMP, threads share the same resources and access shared memory. In the end we implemented an hybrid strategy to use both libraries and better optimise the parallelisation.