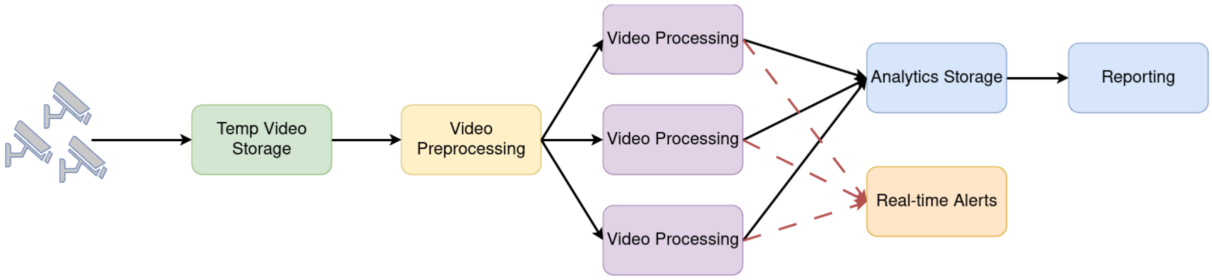
Athens University of Economics and Business
Department of Informatics
Msc. COMPUTER SCIENCE
Fall semester 2024-2025
Prof: V. Kalogeraki
Project Description

Traffic management cameras are one of the most modern tools for improving safety and efficiency on the road network. Installed at critical points in cities and highways, these cameras allow for the immediate monitoring of traffic flow, the identification of events of interest and the recording of violations. Through them, the authorities can provide real-time updates, adjust signaling according to traffic needs, and intervene immediately in cases of emergency. At the same time, traffic cameras contribute to the fight against delinquency, enhancing safety and ensuring the correct application of the rules of the Road Traffic Act.

In order to support this network of cameras and to be able to record and monitor traffic quickly and efficiently, modern cloud computing systems are increasingly used. The large volume of data generated in these systems makes it necessary to use architectures for processing large volumes of data streams (Big Data). Systems such as Apache Kafka, Apache Spark and OpenCV are often used in such applications.

A typical architecture for processing such data streams is the following:



The flow consists of two sub-flows. The first is a flow that generates vehicle movement data and traffic statistics over a period of several days or weeks, while the second is a near-real-time flow that is responsible for immediately notifying the authorities in cases where a high-risk illegality is observed, e.g. excessive speed of a vehicle.

As part of your assignment, you are asked to design and implement such an architecture that supports both flows.

You will be given a video from traffic management cameras in England (as shown in the image above) which is approximately 32 minutes long and has a frame rate of 25fps.

As part of the work, you are required to implement the following steps:

1. In the first step, you are asked to implement a Video Preprocessing function, which will break the video into smaller 2-minute clips. (The last may result in less than 2 minutes.)
2. In the next step, you are asked to implement a Video Processing function that will use a Computer Vision algorithm using a computer vision library to detect and track vehicles within the clip. This step should be implemented in such a way that horizontal scaling can be done to speed up the process, if necessary.
   Every three dashed lines between the lanes are 20m apart. So as shown in the image below, the distance between the two green lines can be considered to be 20m. For the purposes of this work, you do not need to take perspective into account.

3. After the Video Processing step, the flow is broken into two sub-flows. The results from the Video Processing function are stored in a storage and forwarded for further processing to the Reporting function. At the same time, the urgent events that arise must be forwarded immediately to the Real-time alerts function in order to be updated in real time.

   Questions:

   With the above data in mind, you are asked to implement the following questions:

   Q1: The speed at which each vehicle is moving.
   Q2: The number of vehicles per stream.
   Q3: The number of vehicles that have exceeded the speed limit, in total for the 32 minutes of the entire video. Consider the limit for cars to be 90 km/h and for trucks 80 km/h.
   Q4: If a vehicle is moving at more than 130 km/h, a real-time alert should be generated. It can simply be printed in the log of the Real-time alerts function.
   Q5: The number of vehicles per stream and every 5 minutes.
   Q6: The average speed of the traffic per stream and every 5 minutes. That is, it will be printed in a log file (inbound, 1st 5min, 60kmh.) … (outbound, 5th 5min, 60kmh.)

   Delivery:
   Phase 1 (Deadline 19/12): implementation of the basic architecture of the data flow processing system
   Phase 2 (Deadline 23/1): completion of the work, answering questions