ARCHITECTURES AND INFRASTRUCTURES FOR ARTIFICIAL INTELLIGENCE
PRACTICE 1: CLOUD COMPUTING SERVICES

Objectives

- Learn how to outsource AI processing to the cloud to see the advantages of cloud computing

Introduction:

The practice consists of making an application for Smart City to control a **Smart Lighting System** (SLS) in a certain area of the city. A smart lighting system is responsible for controlling the on/off of the luminaires based on the presence of sensitive elements (people or vehicles) in a defined control area.

An increasingly common use is to use digital cameras as presence detection sensors.

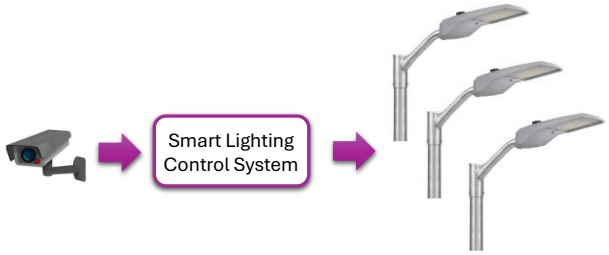


Fig. 1: System overview

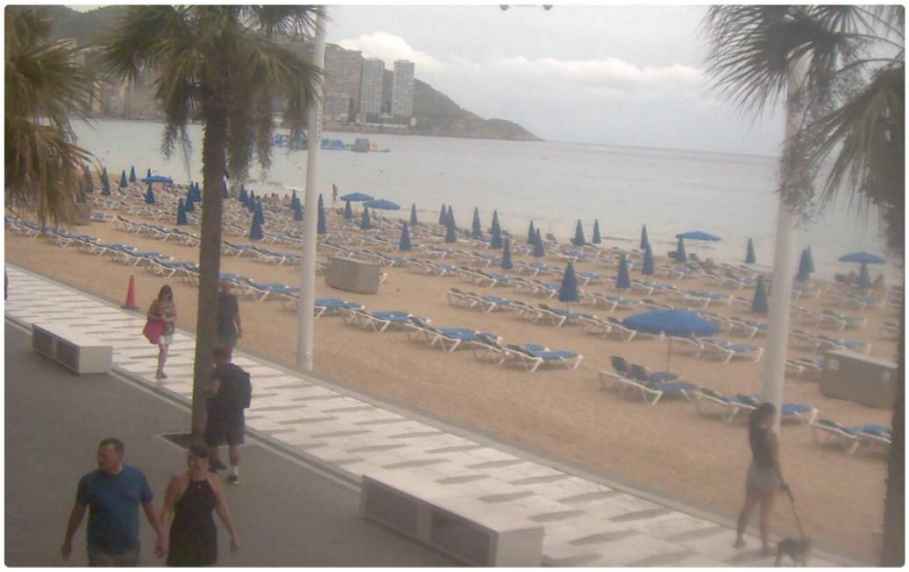
In this practice, public webcams will be used to capture the images of the control areas, and a local machine will be used as a processor for the digital camera. As you know, there are a large number of publicly accessible webcams available on the internet that broadcast a continuous stream of video. For example, the following:

http://www.insecam.org/
https://www.earthcam.com/
https://www.skylinewebcams.com/
https://www.webcamtaxi.com/en/

This video stream will be processed to identify the elements that determine the operation of the controller on the lighting system.

PART 1: LOCAL WORK

Task 1.1. Identify a public webcam that shows a street where people or vehicles are circulating. The camera should be fixed, for example, like the one shown in the following figure:



Online camera overlooking the sea and the beach. The beach is located in Palma, Spain

Fig. 2: Webcam example.

Task 1.2. On a local machine (laptop or lab machine), perform a process that captures a video frame from the previous webcam at a frequency of 5 seconds. The frame resolution must be sufficient to identify the elements.

Note 1: Install the following commands for prepare the environment correctly:

- sudo apt-get update && apt-get install -y
- sudo apt-get update && apt-get install ffmpeg libsm6 libxext6 -y
- sudo apt-get install python3
    - Verify the version: python3 –version
- pip install -r requirements.txt

Note 2: Use the example process provided by the Teaching Team.

Task 1.3. On a local machine (laptop or lab machine), perform a process to count the people in each frame.

It uses the Ultralytics YOLOv8 (https://ultralytics.com/yolov8) AI model capable of performing a large number of actions of object tracking, instance segmentation, image classification and pose estimation automatically.

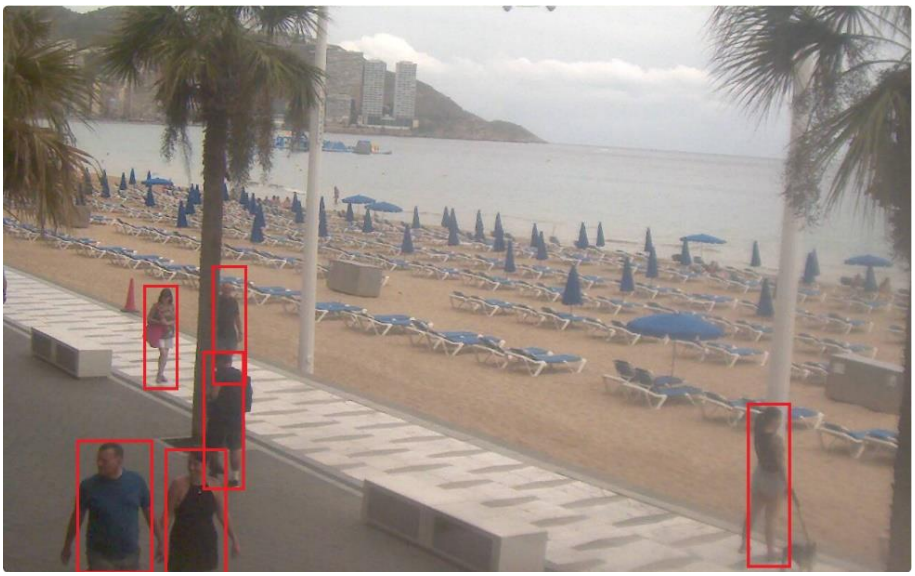


Fig. 3: Person detection

Save the number of people next to the date in a CSV document and draw a graph illustrating the number of people per frame over time:

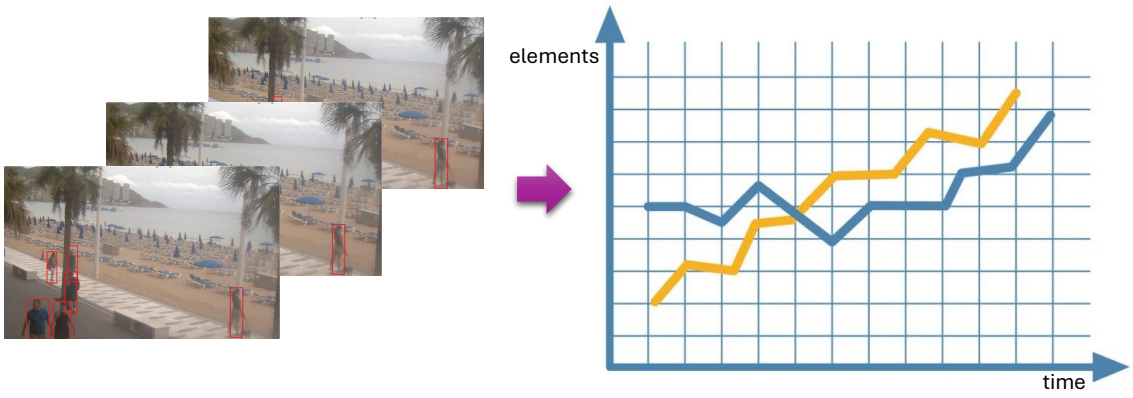


Fig. 4: Elements detection evolution

PART 2: OUTSOURCING PROCESSING TO VIRTUAL MACHINES IN THE CLOUD

In this second part, the processing of element detection will be outsourced to the cloud.

Fig. 5: Outsourcing overview

Task 2.1. Create a virtual machine (VM) in the MS Azure cloud (the "Standard B1s" size is recommended). Install the library packages needed to implement the services in this practice: python3-opencv, python3-pip.

Note 1: You can rely on the example documentation provided by the Teaching Team.

Note 2: If you see a "killed" error when installing any package, it is because the RAM has been exhausted and the system kills the process. Add the "–no-cache-dir" option in that case.
Note 3: If you see a “error: externally-managed -environment”, use the following command before pip3 install: sudo rm -rf /usr/lib/python3.12/EXTERNALLY-MANAGED

Task 2.2. On your local machine, capture a frame of the webcam using the process developed in task 1.2, and send it to the cloud VM for processing.
Note: Use the example process provided by the Teaching Team to submit the data.

Task 2.3. In the VM created in the cloud, it performs a process to count the people in each frame. Use the Ultralytics YOLOv8 **AI model** used in task 1.3. It stores the results in a CSV file.

Task 2.4. Get performance data from running your VM in the cloud: detection efficiency, memory used, processor usage, bandwidth consumed, monetary cost, response time.

PART 3: MULTI-CAMERA MANAGEMENT

In an SLS, the control system manages several connected cameras and a collection of luminaires that completely cover the area to be controlled

Fig. 3: Wide control area

Local processing performance is highly conditioned by the available machine resources, whether in the IoT sensor devices themselves (digital cameras) or in low-cost devices at the Edge (Arduino, Raspberries).

Task 3.1. It identifies a growing number of webcams from which you can send frames for analysis. These frames can be sent from the same local machine or from several with a frequency of 1 frame every 5/sec. per camera.

Task 3.2. Send the frames of new cameras to the VM created in part 2 and get the performance data to identify the processing limit, i.e. how many simultaneous cameras it can handle. Get performance data from running your VM in the cloud: memory used, processor usage, bandwidth consumed, monetary cost, response time.

Task 3.3. Send the frames of new cameras to the Cloud application and get the performance data to identify the processing limit, i.e. how many simultaneous cameras it can handle. Get

performance data from running cloud applications: detection efficiency, monetary cost, response time.

PART 4: OUTSOURCING PROCESSING TO AI SERVICES IN THE CLOUD

In this part, an application will be created to run each frame on MS Azure AI services.

Fig. 7: Outsourcing overview with Azure Machine Learning

Certain types of services related to high-performance computing are available in MS Azure. These services provide high-performance computing and GPU infrastructure with resource-rich clusters.

They also provide AI processing services such as **Azure Machine Learning.** This service allows you to allocate the necessary computing resources and make a payment for their use. It allows you to have from simple CPU and GPU resources, to compute clusters with a large number of nodes and memory:

https://azure.microsoft.com/es-es/products/machine-learning#tabx69c4fee3c8844375955db36c4f74da17

The AI services available in Azure consist of APIs that allow you to use these services to integrate them within applications.

https://learn.microsoft.com/es-es/azure/ai-services/

In this part, we are going to replace the YOLOv8 artificial intelligence model used in the previous parts with an Azure AI service. This is going to be achieved by using the following Computer Vision AI service for image labeling:

https://learn.microsoft.com/es-es/azure/ai-services/computer-vision/concept-tag-images-40

This service allows you to tag the objects that appear in the image.

Task 4.1. Create an AI Service that runs the required processing on MS Azure.
Note 1: You can rely on the example documentation provided by the Teaching Team.

Task 4.2. Deploy the application that invokes Azure AI services to the VM created in Part 2.

Task 4.3. Get performance data from running your VM in the cloud: detection efficiency, memory used, processor usage, bandwidth consumed, monetary cost, response time.

PART 5: OUTSOURCING SERVERLESS PROCESSING IN THE CLOUD (OPTIONAL)

A Serverless application will be created to run the processing of each frame in MS Azure.

Fig. 6: Outsourcing overview with Azure Functions

Task 5.1. Create a serverless application to run on MS Azure.
Note1: You can rely on the example documentation provided by the Teaching Team.

Task 5.2. Get performance data from running cloud applications: detection efficiency, monetary cost, response time.

Task 5.3. Send the frames of new cameras to the Cloud application (created in Part 3) and get the performance data to identify the processing limit, i.e. how many simultaneous cameras it can handle. Get performance data from running cloud applications: detection efficiency, monetary cost, response time.

For all parties:

Write an internship report in which you describe the steps followed in each part, the problems encountered and the design decisions for the resolution of the work to be done. Add a section of conclusions of the work carried out. Include any screenshots you see necessary to prove that you have followed the steps indicated in the statement.

Advanced (+2 points):

1) On top of the developed system, he creates an intelligent system that counts the people who pass through each camera in a period of time. The transit of the same person through the image in several frames must be counted as a single person.
2) On top of the developed system, it creates an intelligent system that counts the people and vehicles that pass through each chamber in a period of time.
3) Design and develop improvements that allow the operation of the architectures of each of the developed parts to be optimized. For example:
   -Establish minimum image resolution that maintains efficiency in detection.
   -Compress image to reduce bandwidth.
   -Implement motion detection techniques in local machines.
   -etc.

Delivery rules:

- The practice can be done in groups of 2 or 3.
- The paper must follow the format defined for Springer's *Lecture Notes in Computer Science publications*
  https://www.springer.com/gp/computer-science/lncs/conference-proceedings-guidelines
- The submission will be made through the virtual campus work submission tool.

- Valid document formats are *MS Word* (.doc, .docx), *OpenDocument* (.odt), or *Portable Document Format* (.pdf).