cyclomedia

Coding Assignment

Versie 0.2

2021-08-09

Introduction

This includes the coding and design assignments for ***Senior Cloud Software Engineer and Devops*** Position.

- The report should be done in PDF or another common standard
- All coding should be done by you; Avoid copying large segments from the internet.
- Include build and run instructions with your coding
- Send your results to the following the persons
    - Jeroen Guelen (jguelen@cyclomedia.com)
    - Julien Vijverberg (jvijverberg@cyclomedia.com)
- If you feel, you're spending too much time on this assignment, wrap it up by adding what you would have done to make it run or to improve the code.

Assignment 1: System Architecture

Design a data processing pipeline which runs the following steps:

- Object detection: For each image, finds the bounding boxes of all objects of interest (e.g. traffic signs). This typically runs on a GPU.
    - Input: Image or URL to image
    - Output: List of detections (rectangles)

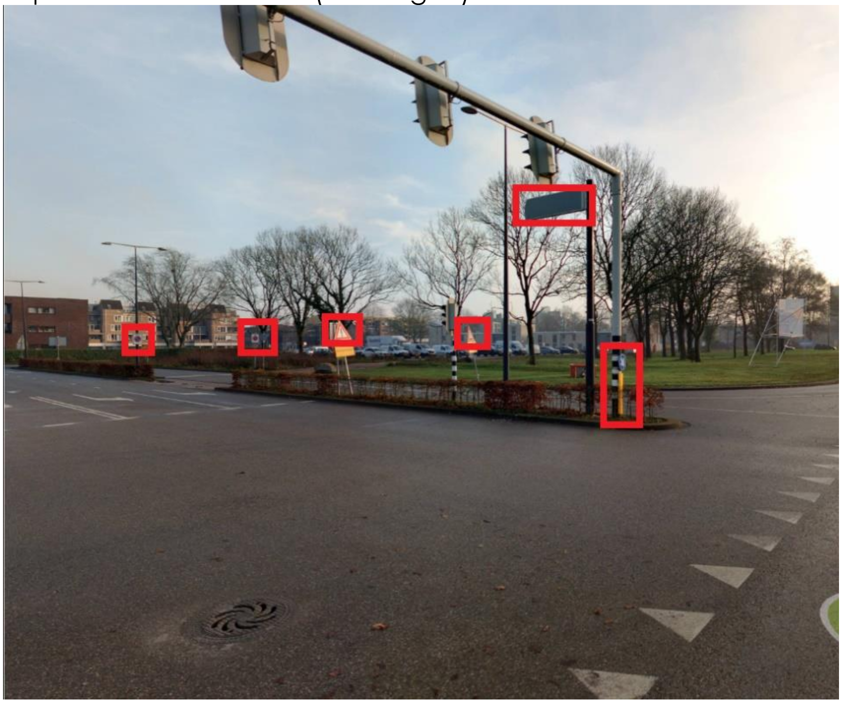

Figure 1: Example output of object detection

- 3D positioning via triangulation: For all images recorded in a (sub-)region and the detection, determines which detection on an image belongs to a specific, real-world object of interest. This typically runs on a CPU.
    - Input:
        - 3D recording locations,

        - known parameters per recording to project a point into 3D,
        - Detections per image.
    - Output:
        - List of Object3d of which each has
            - position (x,y,z)
            - its associated detections per image.
- Object recognition: Given all the detections corresponding to a single real-word object of interest and the source images, determine the category of the object. For example, determine whether it is a stop sign. Depending on the actual algorithm
    - Input
        - List of Object3d of which each has
            - position (x,y,z)
            - its associated detections per image.
    - Output
        - List of Object3d of which each has
            - position (x,y,z)
            - its associated detections per image.
            - A label describing its category

You can assume that

- All processing steps are available in docker containers, so you don't have to design the algorithms. You can change the docker container contents.
- 3D positioning can be subdivided in pre-defined regions depending on the locations and we can perfectly handle the overlap between those pre-defined regions.
- Data is coming in only as batches. For example, a full municipality should be processed in a single run.

Also address the following…

- Be as specific as possible about the technologies you would be using and why.
- Define what is communicated how and where.
- How can we trade-off latency vs through-put vs processing costs in your design ?

Assignment 2: Code

The coding assignment is to design and code a HTTP service which provides flexible image processing functionality.

The functionality which needs to be supported include

- Input must be able to read from JPG images defined by the caller, although this can be a path to the image.
- The following image-processing operations....
  - Resizing the image to any reasonable size smaller than the origin image.
  - Split the image into left-top/right-top/left-bottom/right-bottom sub-images and process/store them separately
  - Blur the full image, for example by applying a Gaussian filter or by downsampling followed by upsampling.
  - Any combination of the above operations
- Images should be saved to JPG and PNG images defined by the callers

So the caller of the service defines a list of images, a list of operations which should be executed on all these images, the output for each.

Divide the project up into 2 parts:

- The service which accepts user requests and kicks off the processing.
- The processing code which runs the image processing in parallel.

For the service part, you can use C#, C++ or python. You're free to define the HTTP API yourself.

For the processing part, you should use python and you must use a parallel processing framework, Hence, the process running the service should not run as the main process.

Tips

- All images can be assumed to be present as local files on the same machine where the service runs or any location which makes your life easier. Similarly, images can be saved to any location.
- You can assume that there will be only 1 request at the same time although more requests may come in while the processing is still running.
- You do not have to take into account the issue of dependencies between requests.
- You can assume the height and width of all input images are 512x512 + 1024x1024 or any size limitation which is convenient for you, expect that the code should be able to support at least two different sizes.
- You could use flask as a web framework for the service
- You could use opencv to process images
- You're free to use any open-source distributed processing framework including one as simple "*make*"

Add the design of the HTTP API to the report.
Add the setup, build and run instructions to the report or the a README file in the root of your code tree.
s