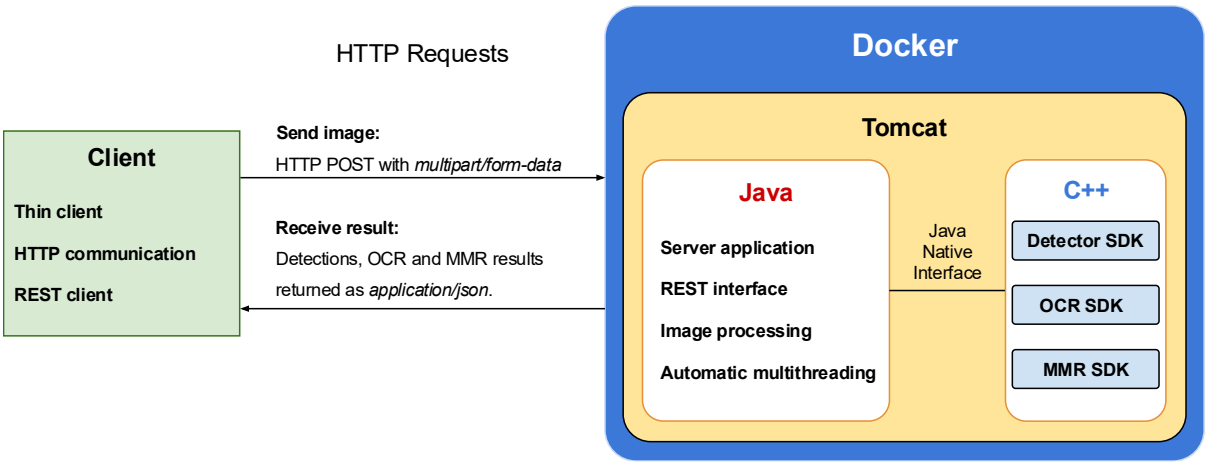
MMR+ANPR REST Server 4.5

Technical Specification



Description

- **Client-Server** architecture with **REST API** written in **Java**.
- The server accepts input images (**JPG, JPEG**, **PNG**, **BMP, TIFF, WEBP, AVIF**) and returns processing results:
  - Detected road users and their components (vehicles, plates, windshields, wheels)
  - Type and text of detected plates (OCR)
  - Vehicle classifications (MMR)
- **Input** image is supplied to the server using **HTTP POST** request:
  - The content type is ***multipart/form-data***
    and the image file data is contained as a parameter.
  - Optionally, the request may contain positions of road users (manual detections)
    and the specification of the requested analyses.
- **Output** road user detections, OCR texts and vehicle classifications:
  - Returned as ***application/json*** media type.

Technologies

- System running as **Docker** image allows scalability.
- Application running on **Tomcat** web server.
- Application itself is written in **Java** and uses **JNI** (Java Native Interface)
  for communication with image processing SDKs.
- Image processing SDKs are written in **C/C++**.

Contained SDKs

- **LPM SDK**
  - Box (vehicle), plate, windshield and wheel detector running on input images.
  - Plate OCR running on detected plates.
- **MMR SDK**
  - Vehicle classification (view, category, make, model, generation, variation, color, and tags recognition) running on detected license plates or vehicle boxes.
- All current SDKs support GPU computation.

Supported Operating Systems

- **Linux**
  - Ubuntu 18.04 and higher – x86_64 platform

Minimal Hardware Requirements

- Processor: 2 GHz, 2 cores (e.g., Intel Core i5)
- RAM: 4 GB
- Hard disk: 16 GB free space
- GPU (optional): NVIDIA Driver version >= 410.48 compatible (e.g., GeForce RTX 30-series)

Performance

For the performance test, the following configuration was used:

- GPU – graphical card NVIDIA® GeForce® GTX 1660, 6 GB GDDR5
- CPU – processor Intel® Core™ i5-9400F @ 2.90 GHz, 32 GB RAM
- 1000 images, Full HD resolution (1920 x 1080 pixels)
  - 1000 detection tasks, 1043 OCR tasks, 1120 MMR tasks
  - Detections: 1120 road users (1085 boxes, 1043 plates)
- Parallel processing
- Default SDK configuration
  - Detector: LPM module 802 (802-generic.gen-gen-v7.14, all object types; segmentations explicitly enabled only when segmentation task requested)
  - OCR: LPM module 801 (801-generic.gen-gen-v7.16, worldwide)
  - MMR: precise VCMMGVCT with BOX preference (MMRBOX_VCMMGVCT_PREC_2025Q4.dat / MMR_VCMMGVCT_PREC_2025Q4.dat)
  - 1 processing thread per SDK

The following table summarizes the average processing time of the input file depending on the requested tasks and the processing unit.

Tasks	GPU [ms]	CPU [ms]
Detection	11.1	44
Detection + OCR	11.7	51
Detection + MMR	33.9	804
Detection + OCR + MMR	35.3	805
Detection + Segmentation	15.2	60
Detection + Segmentation + OCR	16.0	66
Detection + Segmentation + MMR	35.6	805
Detection + Segmentation + OCR + MMR	37.5	806
OCR	6.4	37
MMR	27.3	785
OCR + MMR	29.3	787