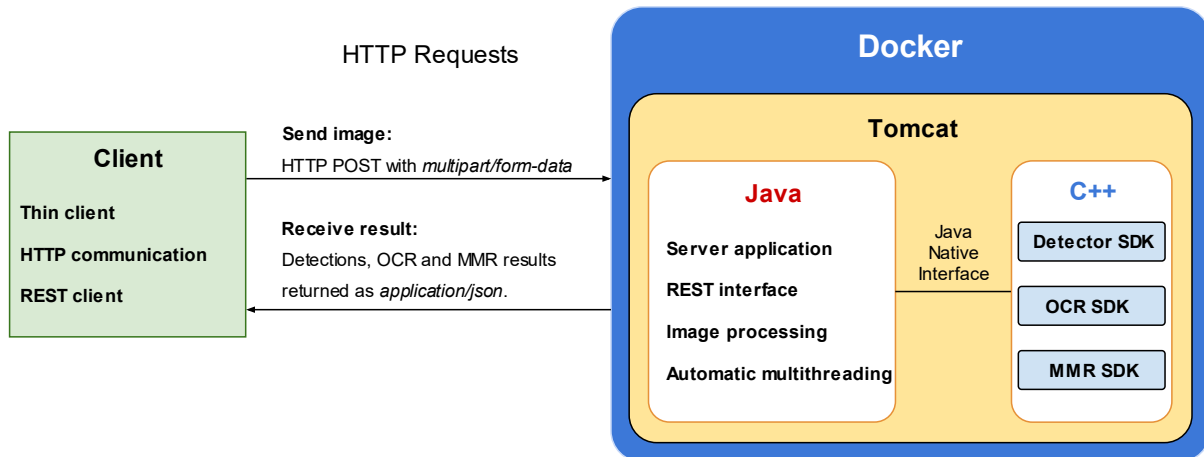


MMR+ANPR REST Server 4.3

Technical Specification



Description

- **Client-Server** architecture with **REST API** written in **Java**.
- The server accepts input images (**JPG, PNG, BMP**) and returns processing results:
 - Detected road users and their components (bounding boxes, plates, windshields)
 - 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 and windshield 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: 8 GB free space
- GPU (optional): NVIDIA Driver version >= 410.48 compatible (e.g., GeForce GTX 1050 Ti)

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, 1040 OCR tasks, 1102 MMR tasks
 - Detections: 1102 road users (1037 boxes, 1040 plates)
- Parallel processing
- Default SDK configuration
 - Fast Detector: LPM module 802 (802-generic.gen-gen-v7.10, all object types)
 - Precise Detector: LPM module 803 (803-generic.gen-none-v7.3, all object types)
 - OCR: LPM module 801 (801-generic.gen-gen-v7.12, worldwide)
 - MMR: precise VCMMGVCT with BOX preference (MMRBOX_VCMMGVCT_PREC_2024Q4.dat / MMR_VCMMGVCT_PREC_2024Q4.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]
Fast Detection	13.1	47
Fast Detection + OCR	13.5	54
Fast Detection + MMR	24.1	944
Fast Detection + OCR + MMR	25.4	947
Precise Detection	31.0	328
Precise Detection + OCR	31.1	332
Precise Detection + MMR	46.4	955
Precise Detection + OCR + MMR	47.7	956
OCR	5.9	37
MMR	19.3	927
OCR + MMR	21.3	928