Controls
Electronics &
Mechatronics

CERN

Framework for Deep Learning Inference

Álvaro García González

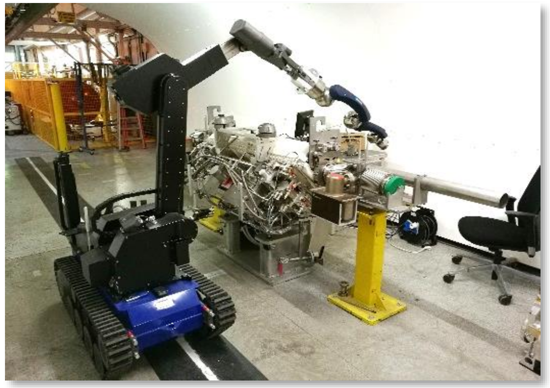
Eloise Matheson

BE-CEM-MRO

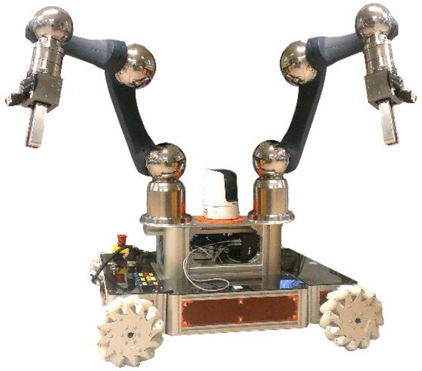
CERN

Robots at CERN

Train Inspection Monorail (CERN made)



EXTRM robot (CERN made)



CERNBot in different configurations (CERN made)

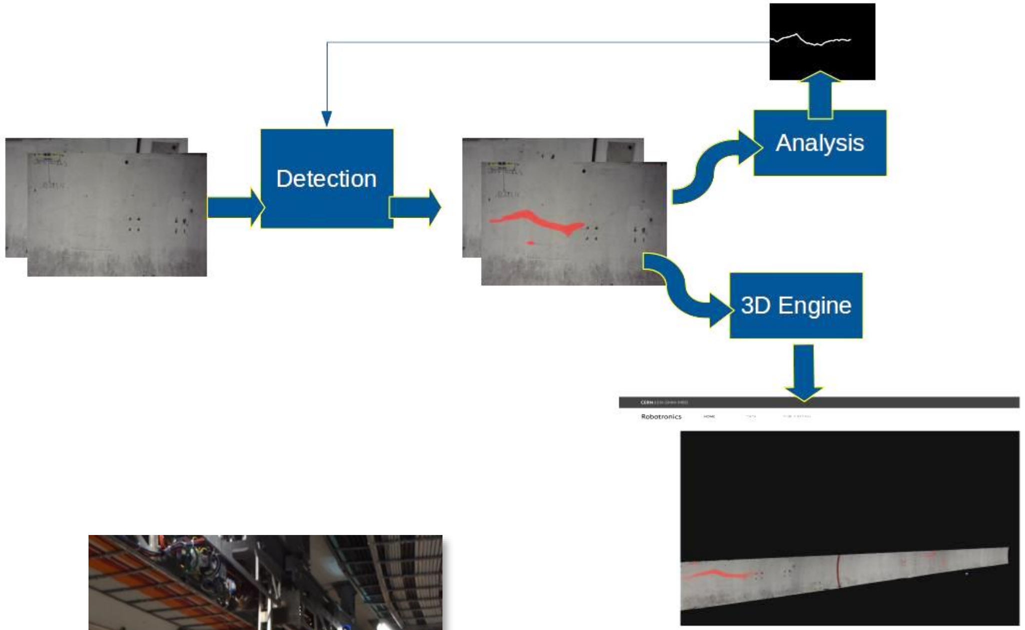
[Mario Di Castro, Alessandro Masi, Luca Rosario Buonocore, Manuel Ferre, Roberto Losito, Simone Gilardoni, and Giacomo Lunghi. Jacow: A dual arms robotic platform control for navigation, inspection and telemanipulation. 2018.]

[Di Castro, Mario, et al. "i-TIM: A Robotic System for Safety, Measurements, Inspection and Maintenance in Harsh Environments." 2018 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR). IEEE, 2018..]

Online Tunnel Structure Monitoring

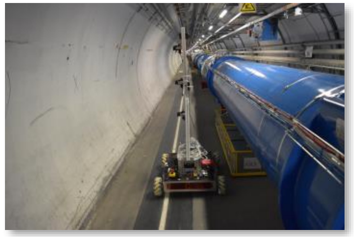
Requested by SMB

- Detects defects (cracks, water leaks, changes) using a Mask-RCNN network.
- High-definition picture collection using TIM and CERNBot.
- 3D reconstruction of wall using structure from motion techniques to compare time evolution of defects (available on web browser or virtual reality headset).
- **HL-LHC condition survey of existing infrastructure carried out with TIM to monitor impact of new civil works.**

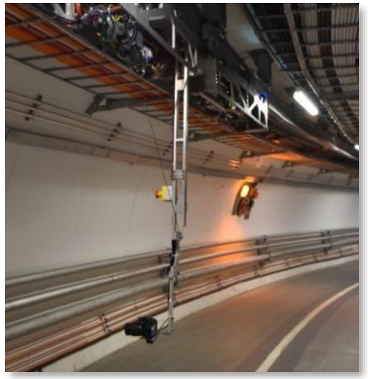


Example of water leak found by TIM2 during TS3 2018

HD camera system for tunnel dome view



System integrated also on other robots



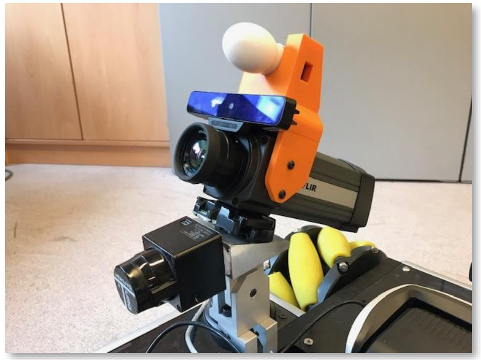
HD cameras mounted on TIM

Example of crack found using vision based machine learning techniques

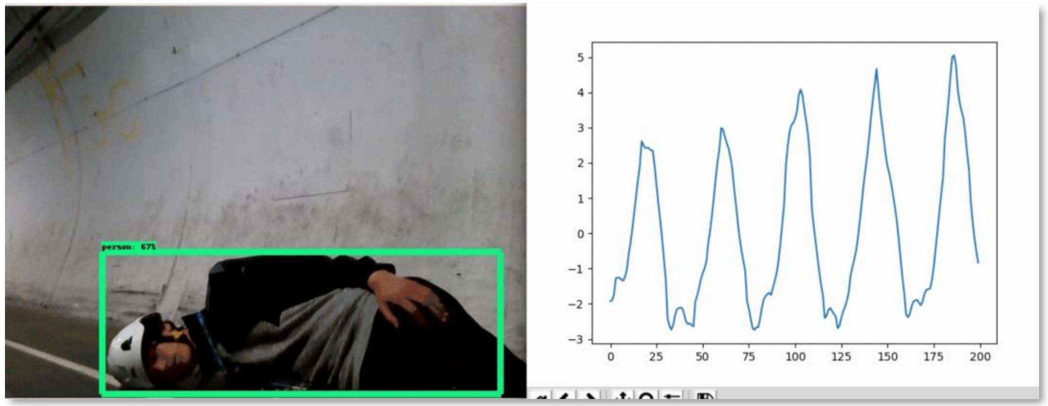
People Recognition and Vital Monitoring

Requested by HSE

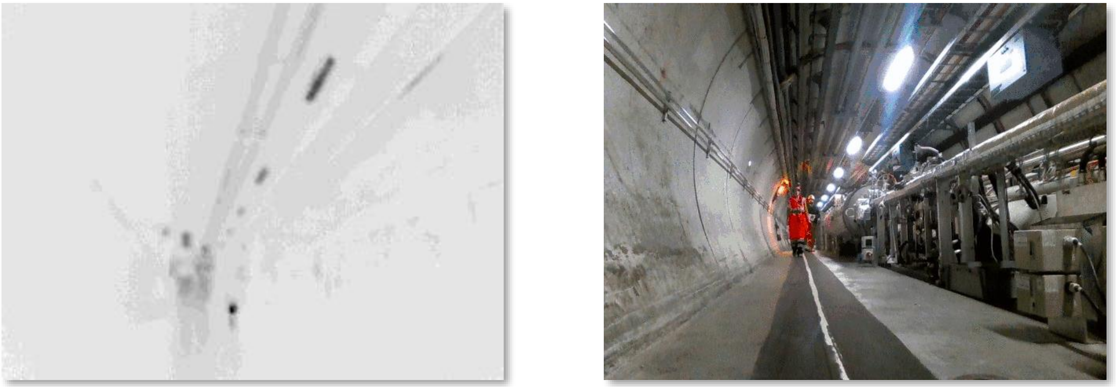
- ➢ Machine learning techniques enhance people detection and vital signals monitoring at distance.
- ➢ People search and rescue is of primary interest in disaster scenarios.
- ➢ People monitoring during rehabilitation.



Vision system (2D Laser, radar, thermal and 2D-3D camera)



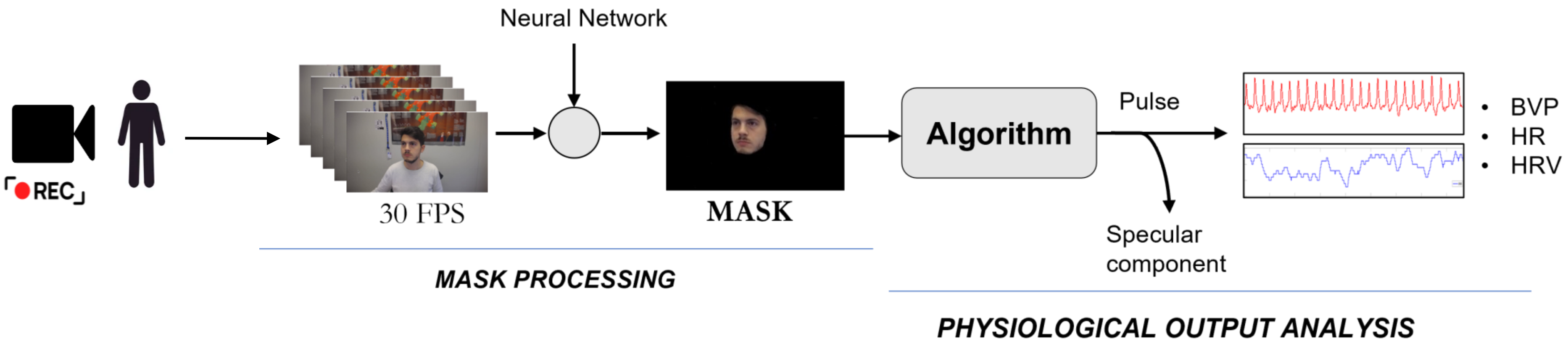
Online respiration monitoring



Online people recognition and tracking

MARCHESE Project

Machine Learning based Human Recognition and Health Monitoring System



Autonomous Navigation

- Autonomous sector door detection, recognition and passage – heavily relies on vision
- Research into optical flow and deep learning to detect and perform pose estimation of the door – CNN-based dense pixel correspondence estimation
- Target Image + Source Image -> Aligned source image

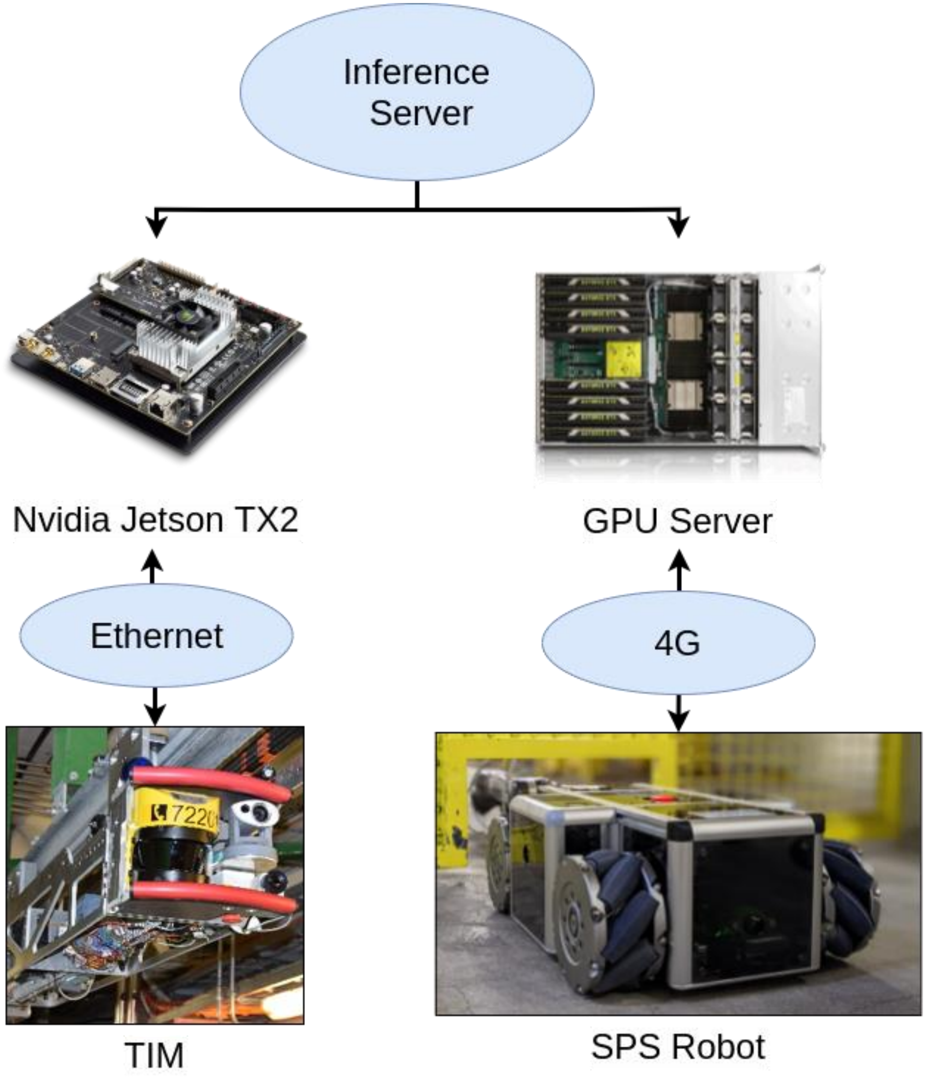
Remote Inference Framework - Requirements

- Easy to use.
- Low-Latency Inference.
- Access from several languages: C++, Python, C#.
- Support for most popular Deep Learning frameworks
- Optimized for reduced bandwidth networks.

Optional requirements:

- Horizontal scaling.
- Dynamic Batching (higher latency).

Example of use



Features

- Developed in C++
- Python, C++ and C# Clients.
- Uses TensorFlow, Pytorch and TensorRT frameworks as back-end.
- Support model formats:
  - Keras SavedModel,
  - TensorFlow SavedModel
  - TorchScript
  - ONNX
- Optional compression of data.
- Multi-model inference
- High performance network protocols.

gRPC

gRPC is an open-source remote procedure call framework developed by Google. RPC allow to call a method on a server as an object on the local system.

- It is implemented over HTTP/2 and uses protocol buffers for message encoding.
- Generates cross-platform client and server bindings for many languages.
- Provides authentication, asynchronous calls, bi-directional streaming and flow control.
- Uses binary payloads, which are efficient to create and parse and hence light-weight.
- Faster than REST API's and optimized for low-latency.

Protocol Buffers

Protocol Buffers is a platform-neutral mechanism used to serialize structured data. It is open source and developed by Google.

- Uses a binary serializer that makes the message files smaller than JSON and XML
- The serializing and deserializing process is also several times faster than JSON and XML.
- It is main advantages are it is simplicity and performance.

Compression

Optional compression for tensor contents.

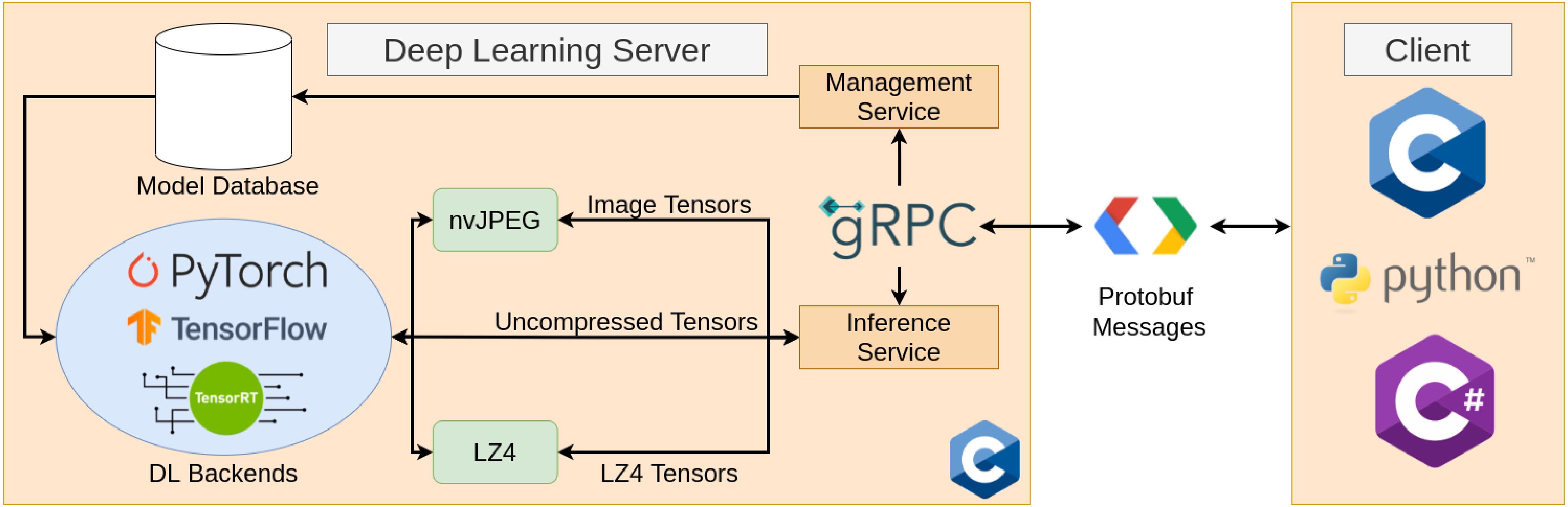
Image Tensors (JPEG):

- Decoded and encoded by a high-performance GPU library (**NvJPEG**).
- NvJPEG is a high-performance GPU library for decoding and encoding JPEG images, developed by Nvidia.

Other Tensors (LZ4):

- Lossless CPU compression algorithm.
- Compression speed > 500 MB/s per core.
- It features an extremely fast decoder, with speed in multiple GB/s per core.

Framework Overview



Server API

The server has two API's: Inference and server management.

The inference protocol is based on the KFServing project.

InferenceService
• **ModelReady**(ModelReadyRequest): ModelReadyResponse • **ModelMetadata**(ModelMetadataRequest): ModelMetadataResponse • **Predict**(InferInputMessage): InferOutputMessage

ManagementService
• **ModelList()** • **LoadModel()** • **UnloadModel()**

Predict Messages

Input:

InferInputMessage
• model: String • tensors: Tensor[] • requested_output_tensors: RequestedOutputTensor • compression: Bool

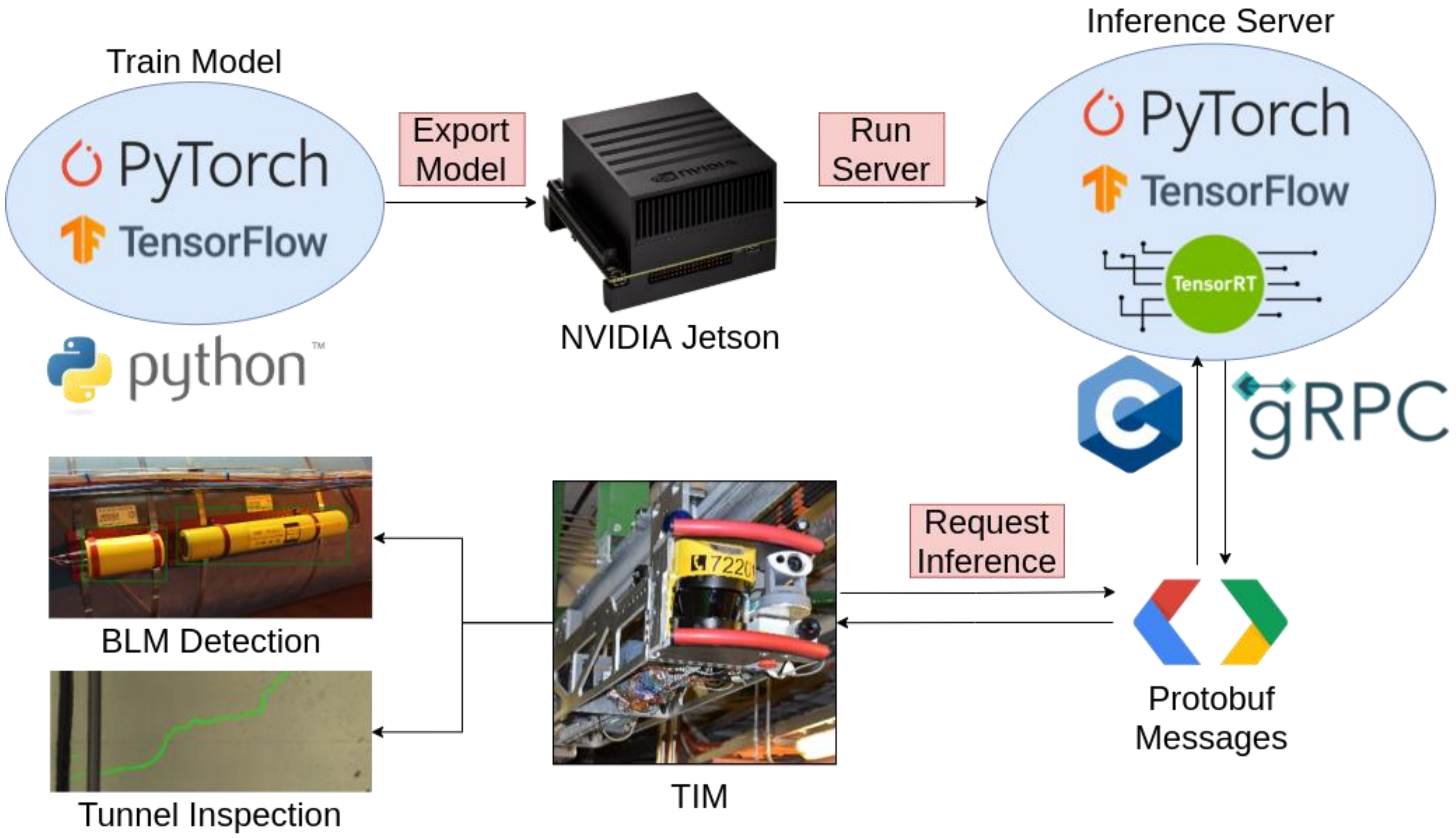
Output:

InferOutputMessage
• tensors: Tensor[] • compression: Bool

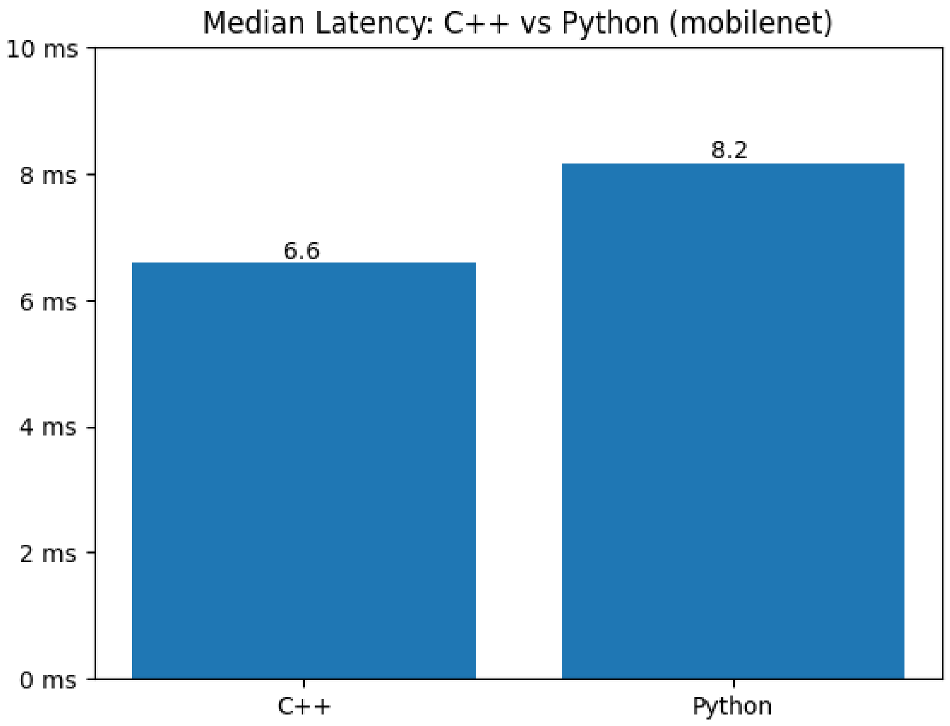
Tensor Message:

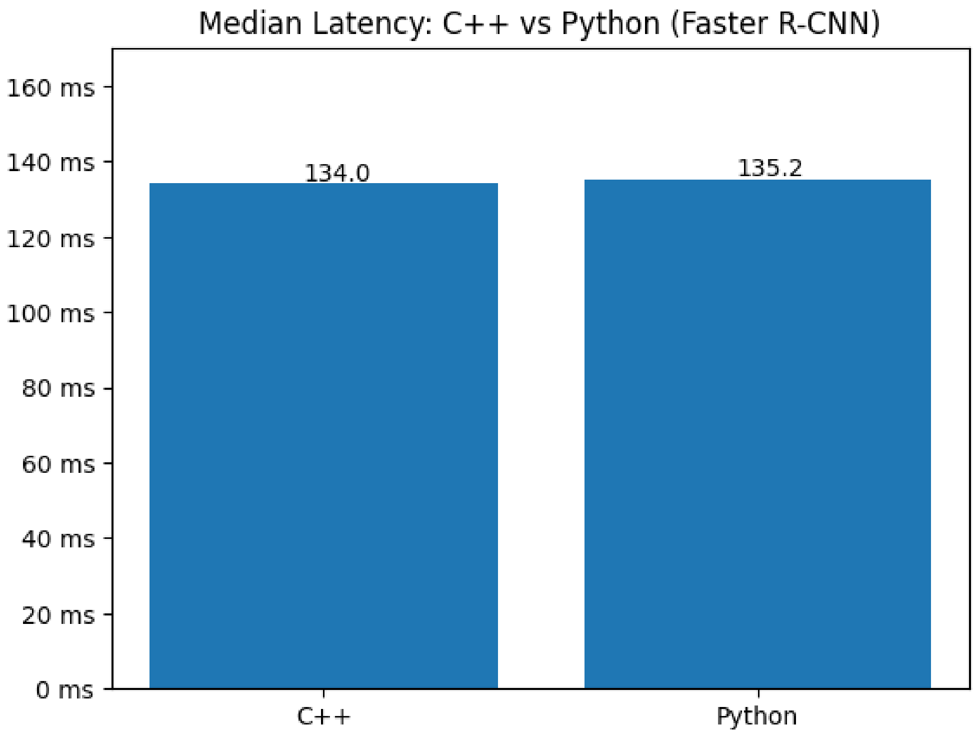
Tensor
• name: String • datatype: Enum • shape: Long[] • image: Bool • data: Byte[]

Workflow Example



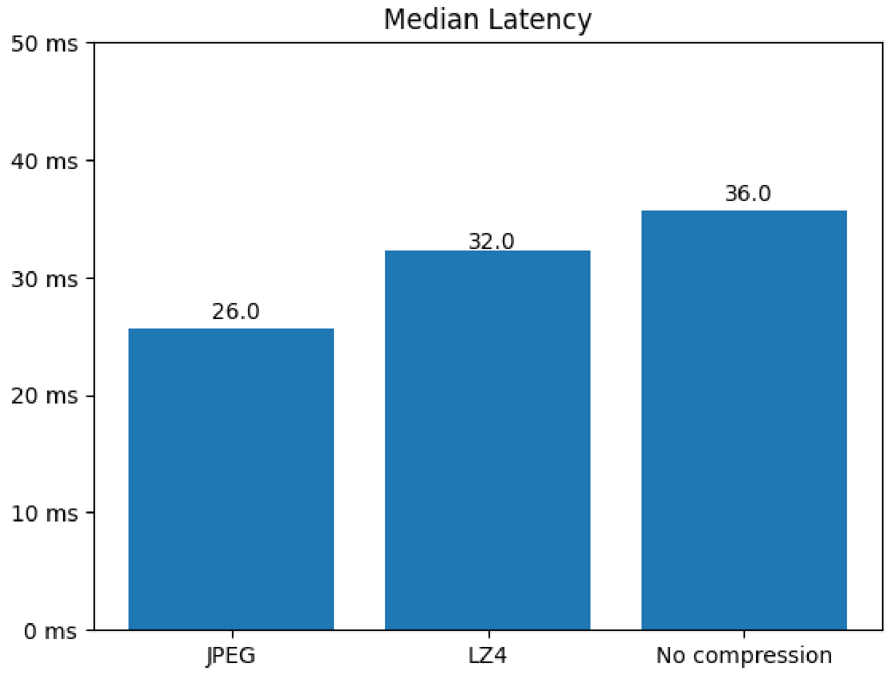
Some results: C++ vs Python TensorFlow





Number of samples: 1,000,000
Image size: RGB 224x224

Some results: Compressed vs Uncompressed



Number of samples: 50,000
MobileNet RGB 224x224

How to improve inference

- **Dynamic Batching:**
  - Higher throughput
- **Reduced precision (FP16, QINT8, QUINT8):**
  - Lower memory usage
  - Faster inference
- **Optimize graph:**
  - Layer fusion
  - Prune model
- **Duplicate models.**

Other alternatives

Nvidia Triton Inference Server

kServe

TorchServe

Flask/DJANGO Server

TensorFlow Serving

Future Work

- Create a model database.
- Test on real environment.
- Test Triton Inference Server.
- Duplicate and ensemble models.