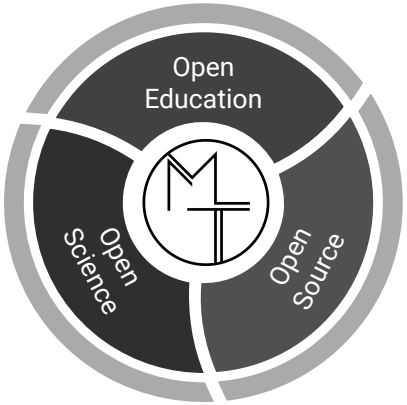
Edge AI Lab #2

MLT & Edge AI Lab

Machine Learning Tokyo (MLT) is an award-winning nonprofit organization 一般社団法人 based in Japan.

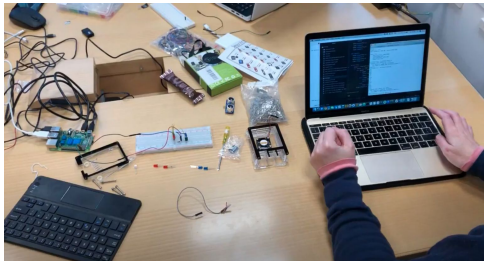
MLT is dedicated to democratizing Machine Learning through open education, open source and open science.

We support an international research- and engineering community of 7,000 members.

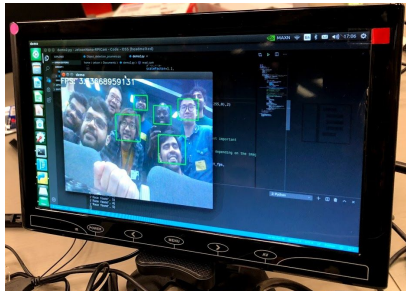


MLT Edge AI Lab

MLT Agritech team visiting
Hacker Farm in Chiba



Edge AI Lab #1



MACHINE
LEARNING
TOKYO

Ref : https://machinelearningtokyo.com

Code of Conduct

MLT promotes an inclusive environment that values integrity, openness and respect.

We are committed to a supportive study and work environment for our community members and the MLT core team, and to provide the opportunity for individuals to reach their fullest potential.

We aim to create a community culture that is collaborative, inclusive and empowering and therefore free of intimidation, bias and discrimination. Thus, we do not tolerate harassment and discrimination of members and participants in any form.

MLT events are for community building and knowledge sharing. We politely ask that company representatives, recruiters and consultants looking to hire or sell their services do not come to MLT events.

This code of conduct applies to all MLT spaces and events, online and offline, including meetups, Twitter, Slack, Discourse, FB, mailing lists. Anyone who violates this code of conduct may be excluded or expelled from these spaces and events at the discretion of the Founding Members.

Ref : https://github.com/Machine-Learning-Tokyo/MLT_starterkit#code-of-conduct

Agenda

- ➔ Introduction to the project
- ➔ EDA
- ➔ Object Detection
- ➔ Object Tracking
- ➔ Testing & Submissions
- ➔ Challenges in deployment on Jetson Nano

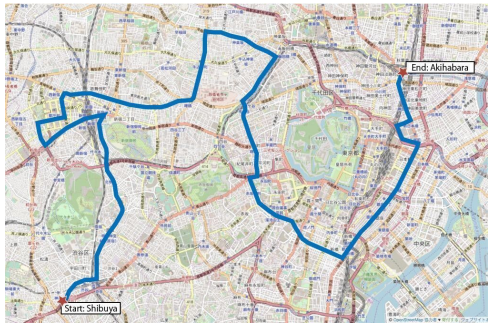
Slack: https://bit.ly/2Yb0uXI
Channel : **#edge_ai_lab**

MACHINE
LEARNING
TOKYO

Problem Statement

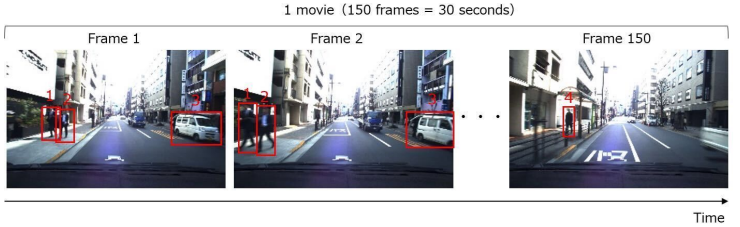
Create an algorithm that detects a rectangular area in which objects are captured from the images taken by a camera forward facing a vehicle and tracks the objects.

Video Data

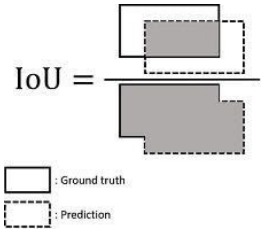


https://signate.jp/competitions/256#abstract

Bbox Detection



Evaluation Stategy



$$\mathrm{MOTA} = 1 - \frac{\sum_t (\mathrm{FN}_t + \mathrm{FP}_t + \mathrm{MME}_t)}{\sum_t \mathrm{GT}_t}$$

FN_t: number of false negatives in frame t
FP_t: number of false positives in frame t
MME_t: number of mismatch error in frame t
GT_t: number of ground truth objects in frame t

MACHINE LEARNING TOKYO

Team Introduction

Alisher
Abdulkhaev

Benjamin
IOLLER

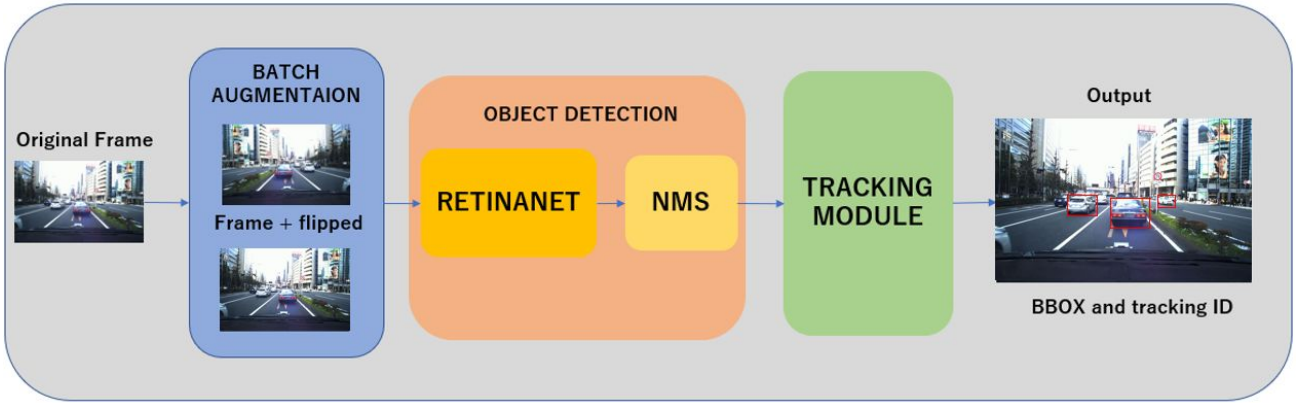
Hajime
Kato

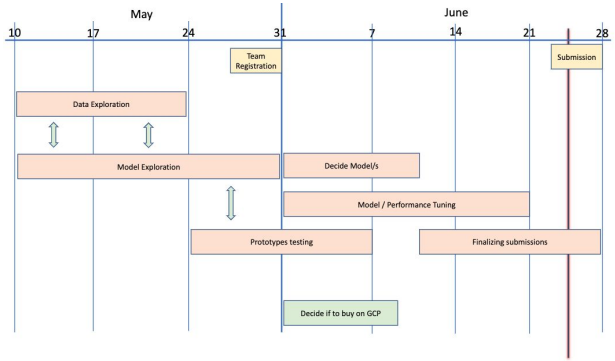
Naveen
Kumar

Yoovraj
Shinde

MACHINE
LEARNING
TOKYO

MLT Solution





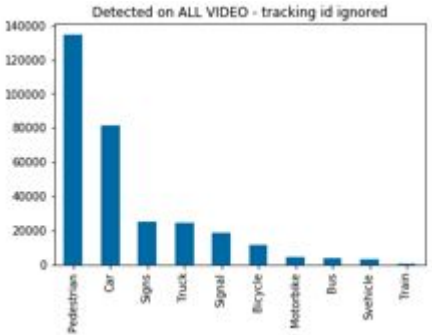
Final Rank	Teamname / Username	Private
1st	RailStar737A	0.62610
2nd	IRAFM-AI	0.61198
3rd	MLT	0.60545

MACHINE LEARNING TOKYO

EDA work on Video Data

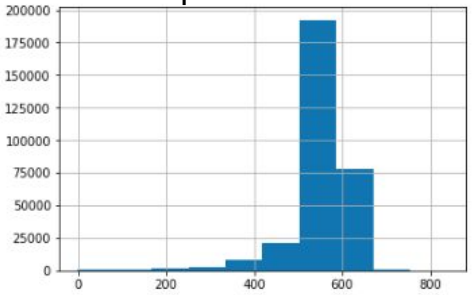
Why? To help selecting the **detector** and tune the **tracker:**

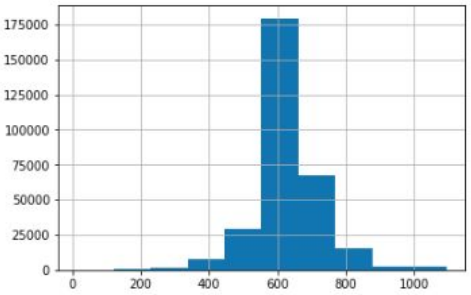
- Object size / scale
- Number of object to track / frame
- Actual images: brightness ...



pandas

BBox position in frame





Also, help for the competition → **Heuristics**

- Position / Ratio / Wrong label in the dataset...

MACHINE LEARNING TOKYO

Object Detection

- **Data**:
  - trained on 5 classes: “Car”, “Pedestrian”, “Truck”, “Svehicle”, “Bus”
  - evaluated on only “Car” and “Pedestrian”
- **Framework**:
  - Keras-RetinaNet
- **CNN Architecture:**
  - ResNet-101 backbone
- **Input data:**
  - 1936x1216x3 (no resizing)

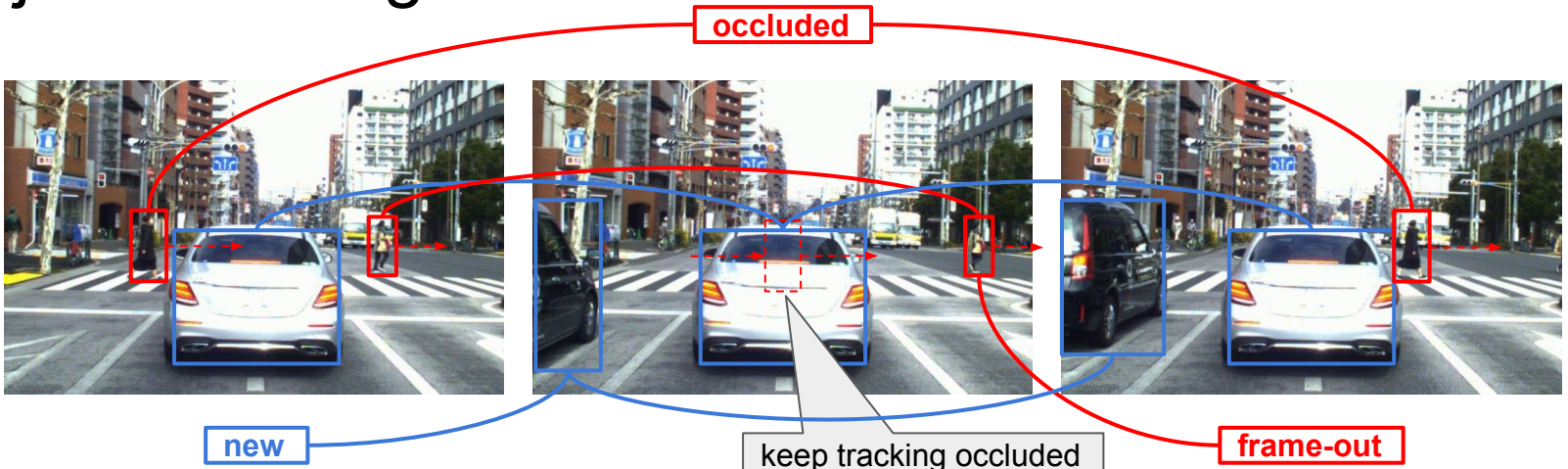
Object Detection

- **Augmentation**:
  - Rotation
  - Translation
  - Shear
  - Scaling
  - Horizontal flip
  - Contrast
- **Score**: class based Average Precision (AP)
  - “Pedestrian”: 0.7713
  - “Car”: 0.9244

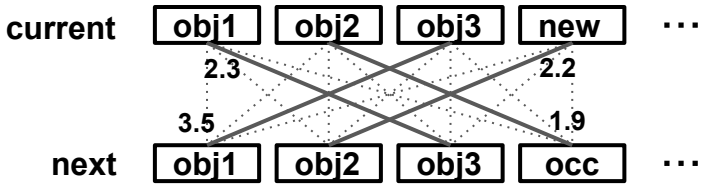
Object Tracking

- **Approach:** maximum weighted matching problem for objects in two adjacent frames - Hungarian Algorithm.
- **Features utilized for matching:** position (coordinate), size, blurred-image similarity (histogram)
- Estimate the position in next frame by linear/quadratic regression.

Object Tracking



Maximum weighted matching



- A score is calculated for each edge, where bbox pairs with close position, similar size and similar image, should have high value.
- It can be solved by Hungarian algorithm.

MACHINE LEARNING TOKYO

Object Tracking

Box distance score

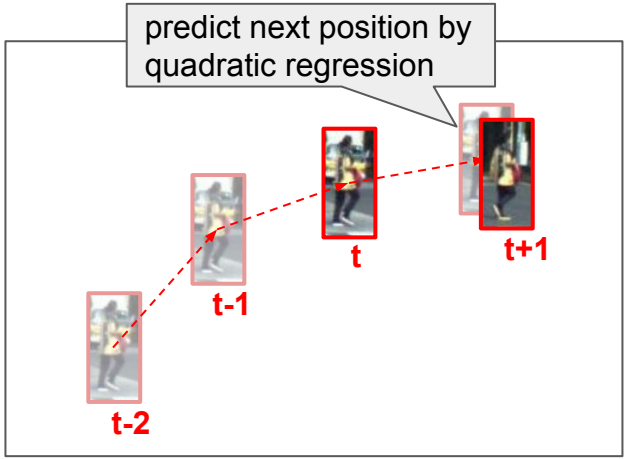
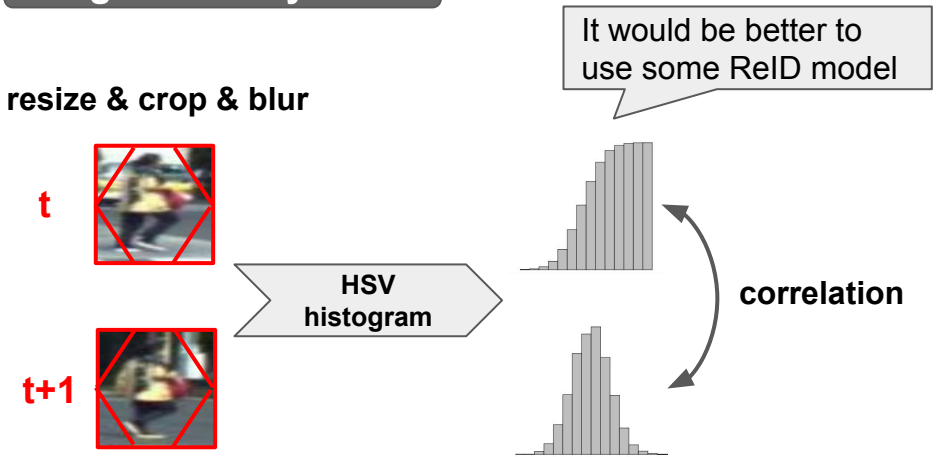


Image similarity score

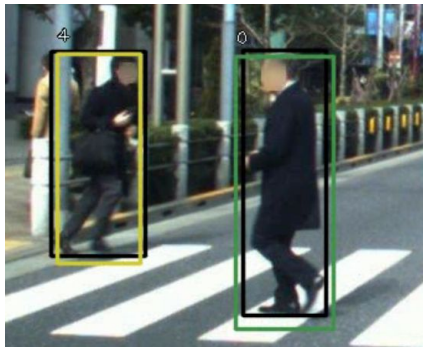


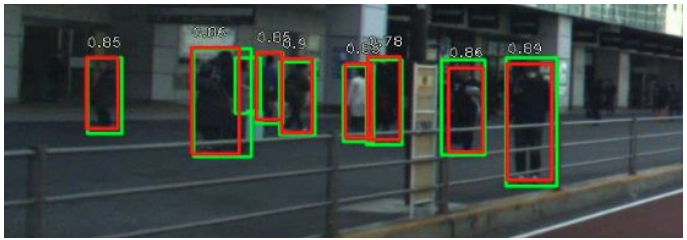
- Parameters are optimized for the ground truth bounding boxes at first, and then fine-tuned for predicted bounding boxes.
- The ID mismatch rate is around 1.5% for the ground truth bounding boxes.

Testing and Submission

No	PIC	original	left + right	flip_lr	dark + bright	pedes nms thr	car nms thr	conf score bias	Result (avg. score)
0	**ALL**	**1**	**1**	**0**	**0**	**0.45**	0.4	**0.2**	**To standardize environment Alisher : 0.6565880517339964 Yuvi : 0.6581984695243673 Ben : 0.6581984695243673 Naveen : 0.6556962259739478 Kato : 0.6581984695243673**
1	Yuvi	1	1	0	0	0.45	0.4	0.2	**0.683137940413561 0.6849171747059786 (fix)**
2	Ben	1	0	1	0	0.45	0.4	0.2	**0.696520928333663 0.6968308980654618 (fix)**
3	Naveen	1	0	0	1	0.45	0.4	0.2	**0.692603211**
4	Ben	1	1	1	0	0.45	0.4	0.2	**0.686553621**
5	Ben	1	0	1	1	0.45	0.4	0.2	**0.694364303**
6	Yuvi	1	1	1	1	0.45	0.4	0.2	**0.682448959052017 0.6842547481594257 (fix)**
7	Ben	1	1	0	0	0.55	0.35	0.1	**0.678503889**
8	Yuvi	1	0	1	0	0.55	0.4	0.2	**0.70120613**
9	Kato	1	1	0	0	0.45	0.4	0.2	**0.695251349**
10	Ben	1	0	1	0	0.4	0.35	0.1	**0.709913565**

40 experiments





- Do not overfit on public test set ($1^{st} \rightarrow 5^{th}$ & $2^{nd} \rightarrow 4^{th}$)
- Test time augmentation
- Adaptive nms: different nms for different classes
- Data exploration is very crucial to catch the mis-annotations and to develop various heuristics (eg., there was no object at y<365 & y>851)
- Someone should keep track of everything: report, next steps, etc.
- Try to find critical improvements: everybody would achieve a particular score, the difference comes from critical improvements - heuristics.
- Do not underrate any idea :)

MACHINE LEARNING TOKYO

Any questions / comments ?

Prerequisites for Deployment

Physical

Camera

Power

Basic Hardware

Optimization

gsstreamer

Fast libraries

NO GUI

Model Deployment

RetinaNet

Challenges

TensorRT

Conversion Process

Product

3d Printing

Real Life Applications

Field Test Video

Prerequisites for Deployment

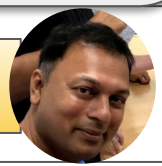
Optimization

GStreamer

OpenCV

NO GUI

Model Deployment



TensorRT

RetinaNet

Challenges

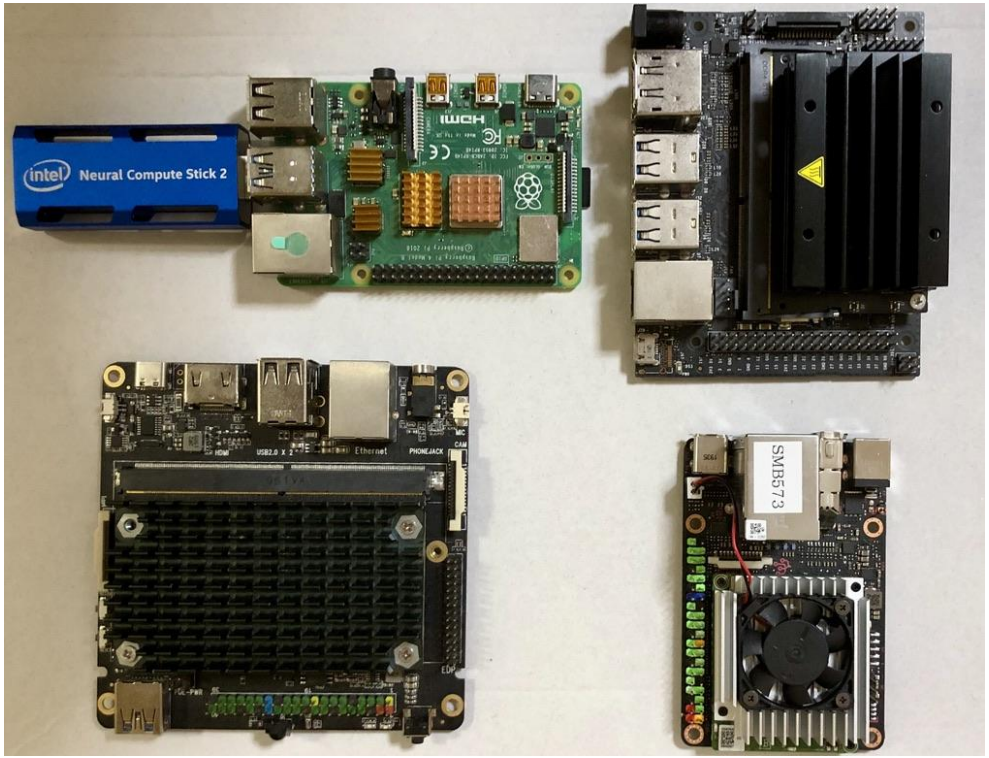
Conversion Process

MACHINE LEARNING TOKYO

From Signate to Edge

	Signate	**Edge**
Image Resolution (Training/Inferencing)	1936 x 1216	640 x 360
Deployment	Cloud, Tesla V100, 5120 cores	Jetson Nano, Maxwell 128 cores
GPU Memory	16 GB (dedicated)	4 GB (shared)
Input Batch Size	2	1
Inferencing Time	2 frames per second	1 frame per second

Model Deployment



MACHINE
LEARNING
TOKYO

Model Deployment

Edge Device	Status	Remarks
Jetson Nano	Success	The model runs without any conversion.
Raspberry Pi 4 with Intel Neural Compute Stick 2	Success	The model needs to converted to the OpenVINO Intermediate Representation. Performance is similar to the Jetson Nano.
ROCK PI N10 (Rockchip RK3399Pro)	No Success	The model conversion failed due to missing OPS for NPU (Neural Network Unit).
Google Coral USB Accelerator / Asus Tinker Edge T	In Progress	There are some issues in the current version of TensorFlow Lite. May be fixed soon.

Optimization

Using Optimized natively compiled libraries

- GStreamer 1.0
- OpenCV 4.4

Runtime memory management

The Jetson Nano is started without GUI to make available more memory for GPU to overcome model loading issue.

Conversion to TensorRT (Work in progress)

We tried to convert the model to **TensorRT** but it failed due to unsupported OPS.

Prerequisites for Deployment

Physical

Camera

Power

Basic Hardware

Product

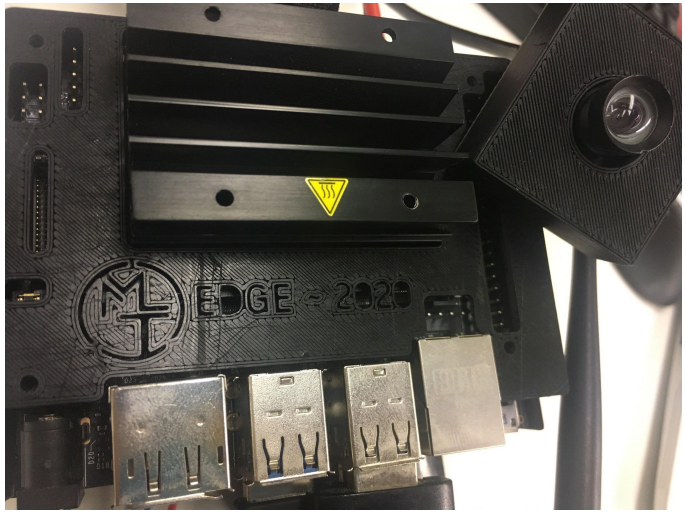
3d Printing

Real Life Applications

MACHINE LEARNING TOKYO

Field Testing

1. Hardware: Jetson Nano - (10W mode)
2. Power: 3S Lipo - BEC - 5v3amp (**2 hours**)
3. Set-up: USB-CSI camera / WiFi AP
4. 3D printing: Bike fixation -



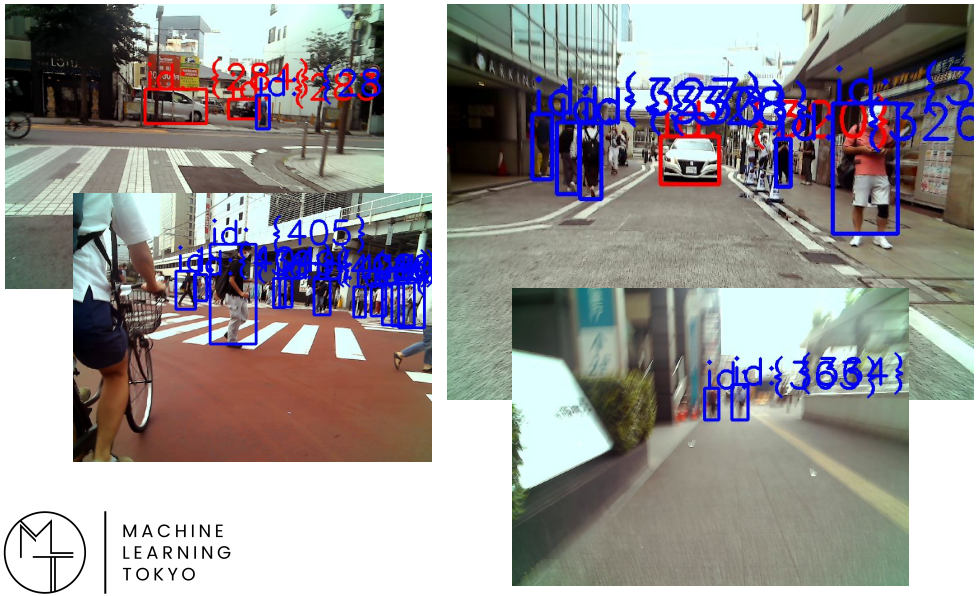
"Not just deploy to deploy, deploy with an application in mind"

Learn about other field - skill up engineering

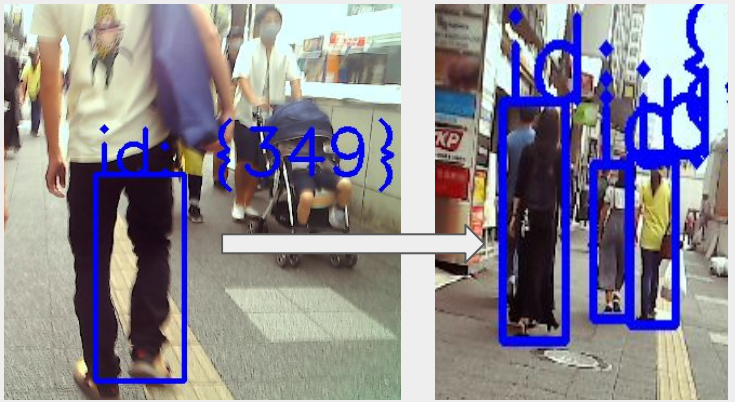
MACHINE
LEARNING
TOKYO

Performances and limitations

- Inferences at **0.95** second and **1.2** second while tracking 3-7 objects.
- Detection even on blurry images (bike vibration).
- Display results without tuning the new resolution model.



Pedestrian as legs...



MACHINE LEARNING TOKYO

Feedback and Suggestions

Future topics to work with Edge AI Lab

- NVIDIA DeepStream SDK
- FPGA (Ultra96 V2 / PYNQ)
- TinyML
- And Other …

GITHUB:
https://github.com/Machine-Learning-Tokyo/EdgeAI Contest3

Blogpost
https://machinelearningtokyo.com/2020/07/21/object-tracking-mlt-solution-for-meti-nedo-edge-ai-competition/

Slack: https://bit.ly/2Yb0uXI
Channel : **#edge_ai_lab**

MACHINE
LEARNING
TOKYO