2023 Synopsys ARC AIoT Design Contest

Project Proposal

Project Topic-WOLO

Presenter- Live without livers

Agenda

- **Abstracts**
- **Challenge and Innovation**
- **Design and Reliability**
- **Project Progress**
- **Test Result**
- **Overall Summary**

Agenda

- **Abstracts**
- **Challenge and Innovation**
- **Design and Reliability**
- **Project Progress**
- **Test Result**
- **Overall Summary**

According to the Ministry of Transportation and Communications, the casualties of motorcycle riders are **twenty times more** than that of car drivers and others in 2022 in Taiwan.

18,001

391,223

WOLO *A smart monitor for scooter and bicycle riders*

WOLO - *We Only Live Once* - is aiming to **analyze the road condition** by deep learning. It can detect the vehicles and their distance in the camera to **alarm the users in real time**.

Agenda

- Abstracts
- **Challenge and Innovation**
- Design and Reliability
- Project Progress
- Test Result
- Overall Summary

Innovation

Current similar system like TESLA are...

Expensive

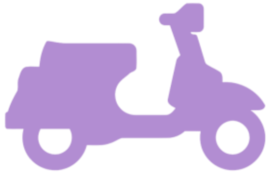
TESLA FSD cost more than 10,000 US dollars.

Require compound sensors

Waymo uses lidar ,radar, and cameras on their self-driving cars.

Designed for cars only

Driver aiding system like Tesla and Waymo are designed for cars only.



Innovation

The innovation of WOLO

Expensive	→	Cheap for scooters
Require compound sensors	→	Need only cameras
Designed for cars only	→	Embedded system

Challenges

Memory Size

- Available memory size < 1MB
- Computation and Parameters both need memory space

Image Quality

- Transfering images from external camera is too slow
- Only 640x480 grayscale images by internal camera are available

Model Complexity

- YOLOv7-tiny has 3M parameters
- The model has too many OP to be placed on ARC EM9D

Result transmission

- The elapsed time of Wi-Fi module is 1 second

Agenda

- Abstracts
- Challenge and Innovation
- **Design and Reliability**
- Project Progress
- Test Result
- Overall Summary

Hardware Architecture

ARC EM9D AIoT DK

Object detection /
Running Dist-YOLOv7

Camera 640x480x1

RW8300

HTTP

Smartphone

WOLO app/
display result

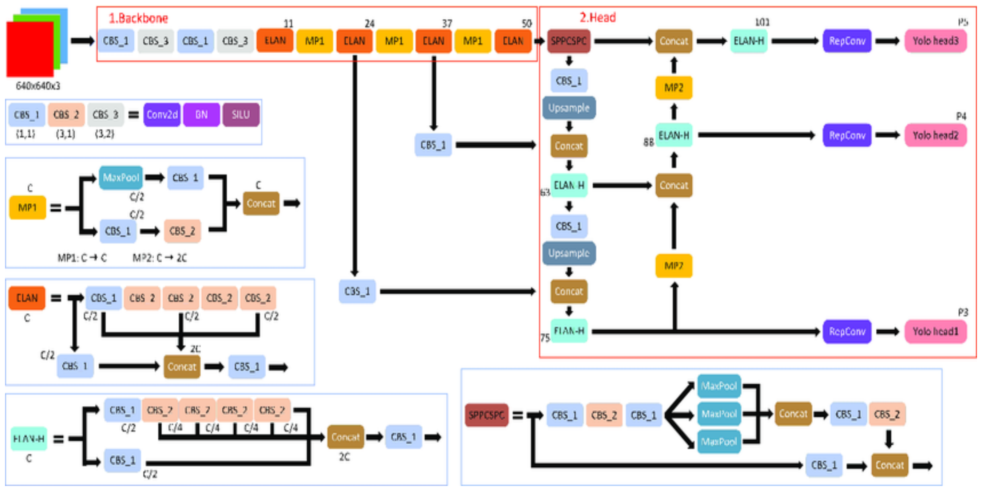
Model Prototype

YOLOv7

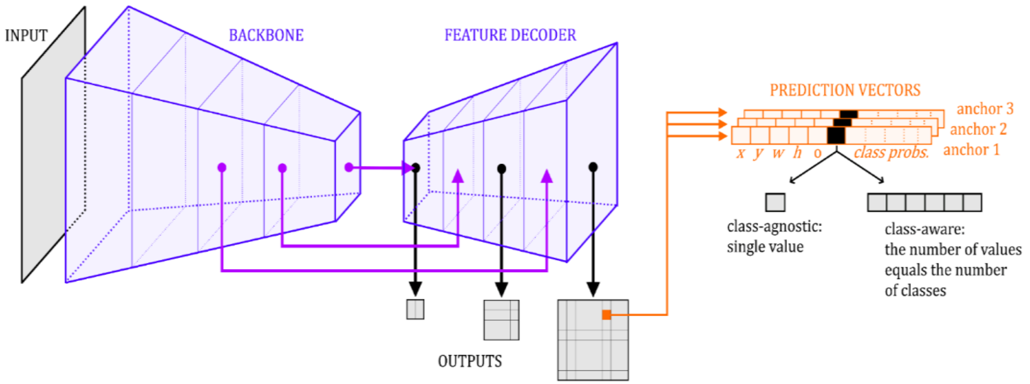
- A convolutional based model for object detection
- An model with simplicity and high accuracy
- Provide a tinier architecture for real time jobs

Dist-yolo

- Adapted from YOLOv3
- Predict distance by adding a distance channel at last layer



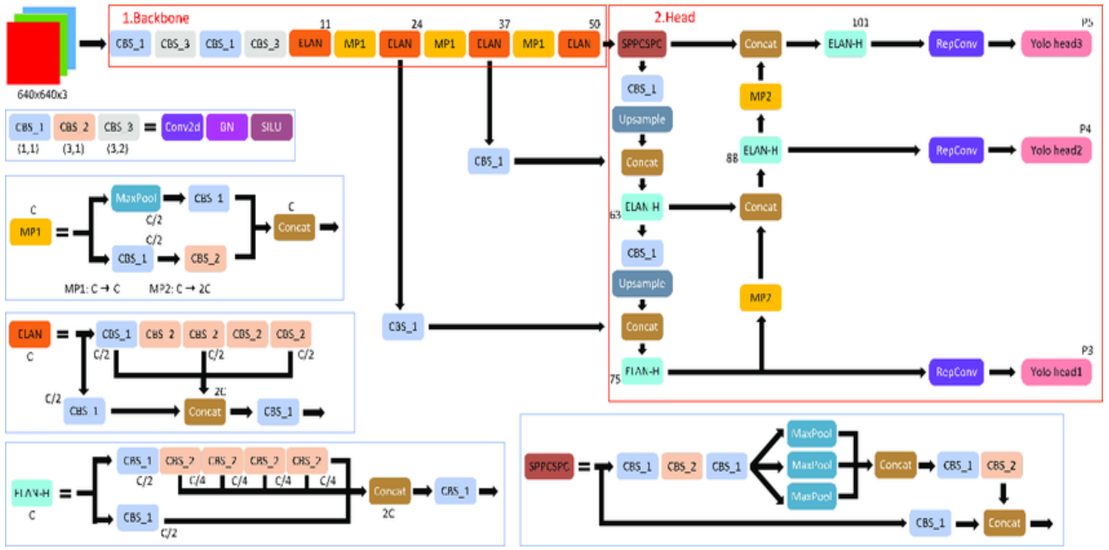
Network of YOLOv7 link



Network of Dist-yolo(based on YOLOv3) link

Model Modification

- Adding distance prediction
- Modified loss for distance



Yolo head ≡ x, y, w, h, conf. + class conf. → Yolo head ≡ x, y, w, h, conf. + class conf. + dist

Add extra channel for distance estimation

$$l_5(i,j) = \omega(\hat{d}_{i,j} - d_{i,j})^2 \quad \rightarrow \quad l_5(i,j) = \omega\frac{|\hat{d}_{i,j} - d_{i,j}| + \sigma}{d_{i,j} + \sigma}$$

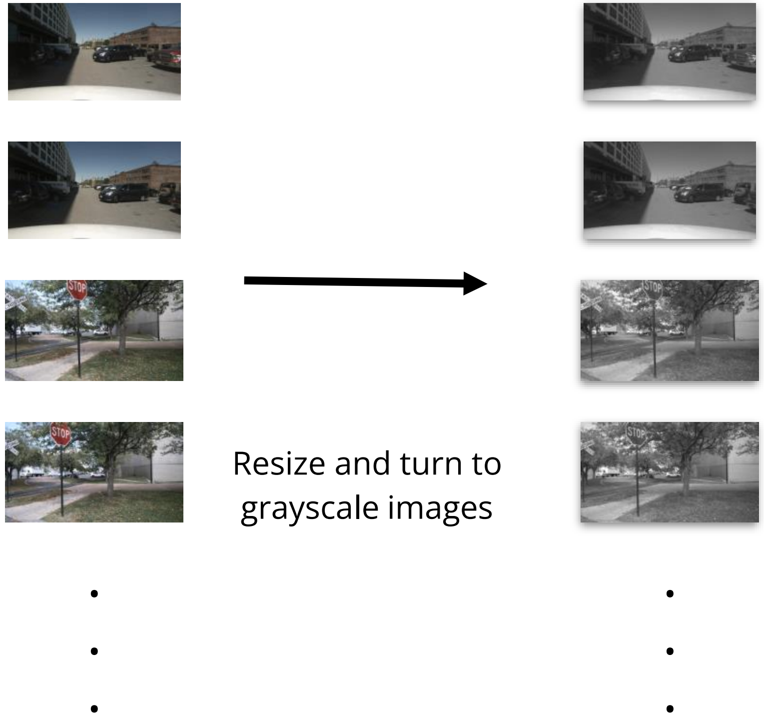
Change L2 loss to L1 loss, and scale with ground truth

Data Process

NuScenes Dataset

- Extract key frames
- Resize to 640x640 and turn to grayscale
- merge and reduce the number of classes

For each 10 second clip, extract 3 detailedly labeled key frames from 6 different cameras



Label form: (x,y,w,h,class,distance)

Review: Challenges

Memory Size

Image Quality

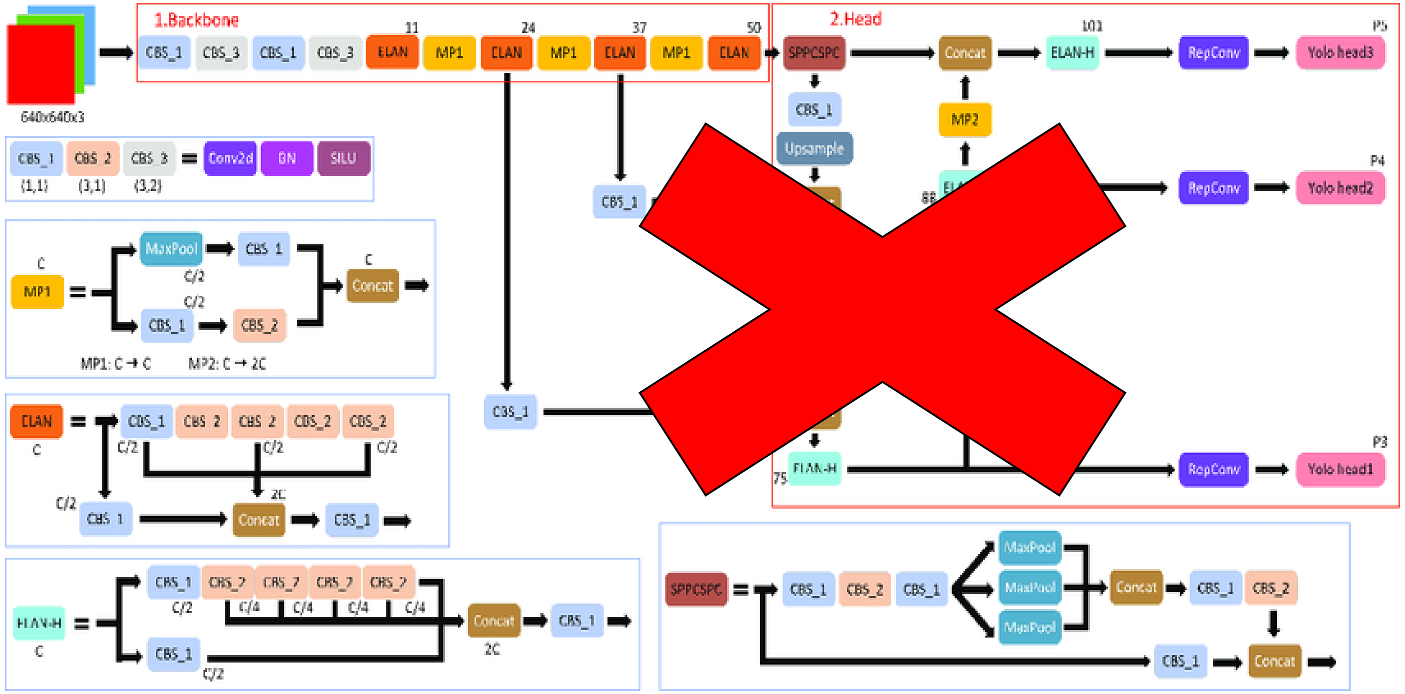
Model Complexity

Result transmission

Solutions I : Memory Size and Model Complexity

- Truncate the feedback loop of Dist-YOLOv7

- Scale down channel numbers of each layer by 0.25



Solutions II : Memory Size and Model Complexity

- Convert operation to C code, compute on cpu.

```
static tflite::MicroMutableOpResolver<20> micro_op_resolver;
micro_op_resolver.AddConv2D();
micro_op_resolver.AddMaxPool2D();
micro_op_resolver.AddFullyConnected();
micro_op_resolver.AddReshape();
micro_op_resolver.AddSoftmax();
micro_op_resolver.AddConcatenation();
micro_op_resolver.AddLogistic();
micro_op_resolver.AddAdd();
micro_op_resolver.AddArgMax();
micro_op_resolver.AddStridedSlice();
micro_op_resolver.AddDequantize();
micro_op_resolver.AddReshape();
micro_op_resolver.AddMul();
micro_op_resolver.AddPadV2();
micro_op_resolver.AddReduceMax();
micro_op_resolver.AddQuantize();
micro_op_resolver.AddSub();
micro_op_resolver.AddPad();
micro_op_resolver.AddResizeNearestNeighbor();
micro_op_resolver.AddLess();

//lack of
//LeakyRelu
//Transpose
//BroadcastTo
//ScatterNd
//gather
//splitV
//Pow
//Cast
//NonMaxSuppressiionV4
```

```
static tflite::MicroMutableOpResolver<8> micro_op_resolver;
micro_op_resolver.AddConv2D();
micro_op_resolver.AddRelu6();
micro_op_resolver.AddMaxPool2D();
micro_op_resolver.AddPad();
micro_op_resolver.AddConcatenation();
micro_op_resolver.AddResizeNearestNeighbor();
micro_op_resolver.AddDequantize();
micro_op_resolver.AddQuantize();
```

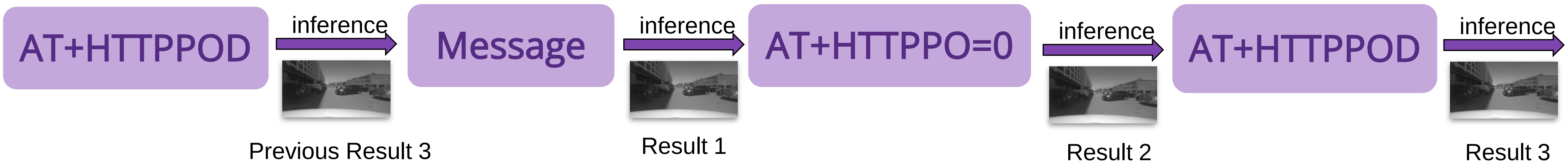
- Reduce from 20 to 8.
- Remove not supported operation

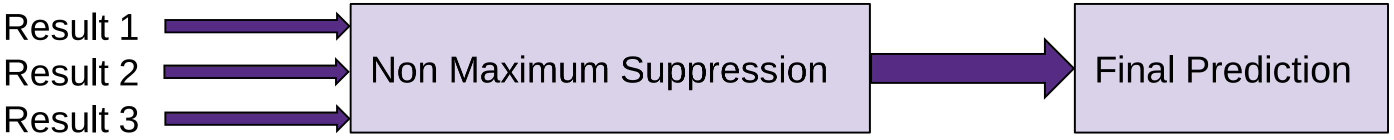
Problem : Result Transmission

- Send by AT command.
- 3 command for sending single message.
- Every command need 300ms delay.

Solution : Result Transmission

- Do inference between commands.
- Use NMS to remove overlapped bounding boxes.





Agenda

- Abstracts
- Challenge and Innovation
- Design and Reliability
- Project Progress
- **Test Result**
- Overall Summary

ARC Model Result

- INT8 Quantized, reduced model size.
- High memory usage (96.81%), make good use of ARC resource.

Memory region	Used Size	Region Size	%age Used
ICCM0:	0 GB	64 KB	0.00%
ICCM1:	268624 B	320 KB	81.98%
SYSTEM0:	926632 B	957168 B	96.81%
DCCM:	104 KB	256 KB	40.62%
XCCM:	32 KB	32 KB	100.00%
YCCM:	32 KB	32 KB	100.00%

```
unsigned int my_model_tflite_len = 359760;
```

```
Total image size= 794 KB( 0xc69a0 )
```

	Original Dist-YOLOv7	Dist-YOLOv7 (Modified)	Quantized (INT8)	Quantized (INT8)
Processing Platform	PC	PC	PC	ARC EM9D
Number of Parameters	6,025,236	304,436	304,436	304,436
Recall (%)	52.71%	47.07%	40.33%	51.82%
Precision (%)	91.73%	85.71%	90.81%	96.24%
Distance MSE(m)	2.35m	4.77m	3.59m	3.34m
Inference time	7.2ms	4.0ms	202.3ms	337ms

ARC inference result using camera

Image from ARC EM9D

Inference using TFLM on ARC EM9D

Agenda

- Abstracts
- Challenge and Innovation
- Design and Reliability
- Project Progress
- Test Result
- **Overall Summary**

Summary

- **Combine YOLOv7 and dist-yolo**
  - Modified YOLOv7 architecture, make it can estimate distance of objects.
- **Simplify the model**
  - Truncated and scale the channels size of model to reduce model size and increase inference speed.
- **Quantize model to INT8 and deploy on ARC EM9D AIoT DK**
- **Use React Native to create User APP**

Appendix : Detection Ability

- Transfer learning YOLOv7 on NuScenes dataset.
- Compare result with Dist-YOLOv7

	Number of Parameters	Recall (%)	Precision (%)	Inference time
YOLOv7	6,025,236	54.11%	95.70%	6.9ms
Dist-YOLOv7	6,052,872	52.71%	91.73%	7.0ms

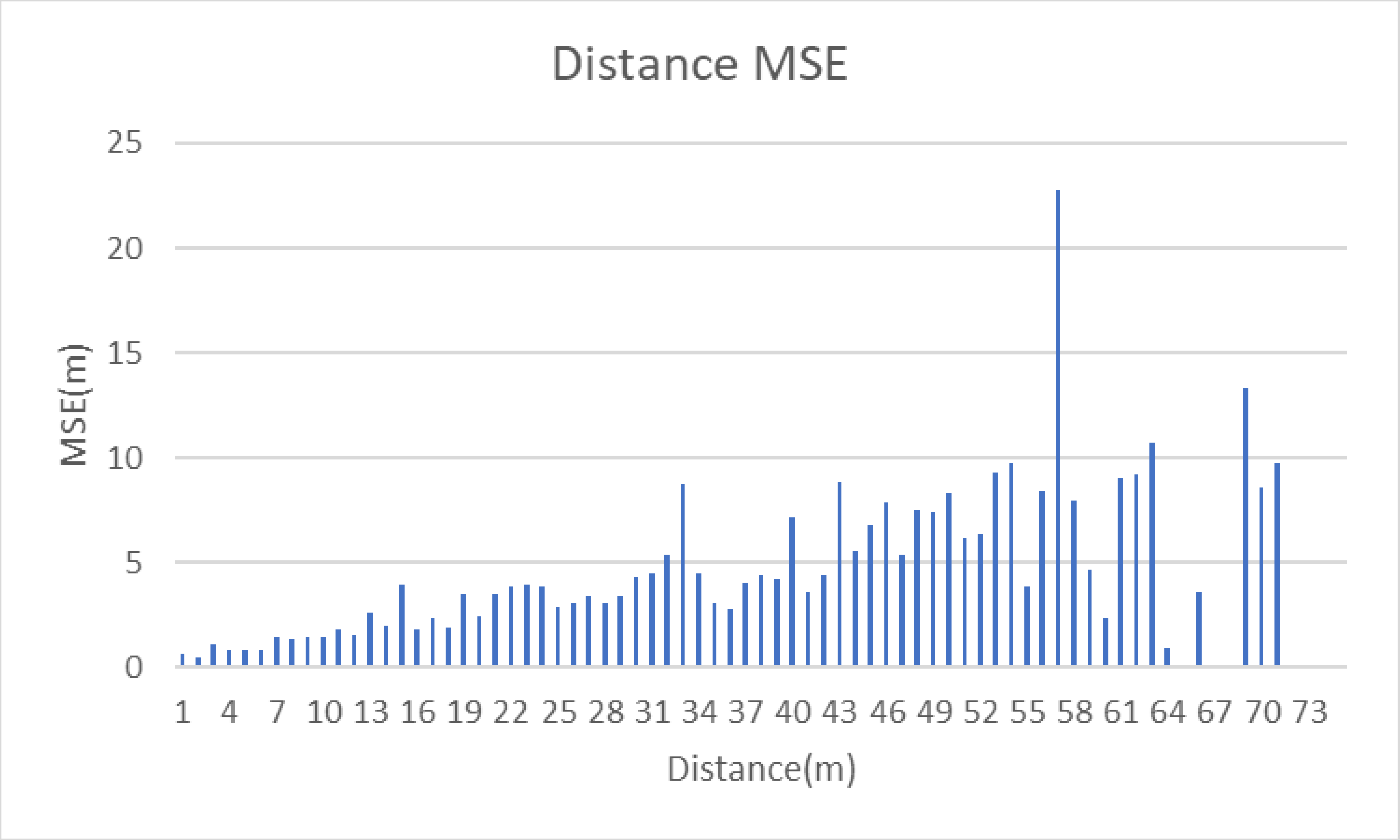
Appendix : Fisheye Effect

- Simulate fisheye effect on NuScenes validation set.

	Number of Parameters	Recall (%)	Precision (%)	Inference time	Distance MSE(m)
Dist-YOLOv7 (Modified)	**304,436**	**47.07%**	**85.71%**	**4.0ms**	**4.77m**
Dist-YOLOv7 (Modified) Fisheye	**304,436**	**46.16%**	**83.66%**	**4.0ms**	**5.08m**

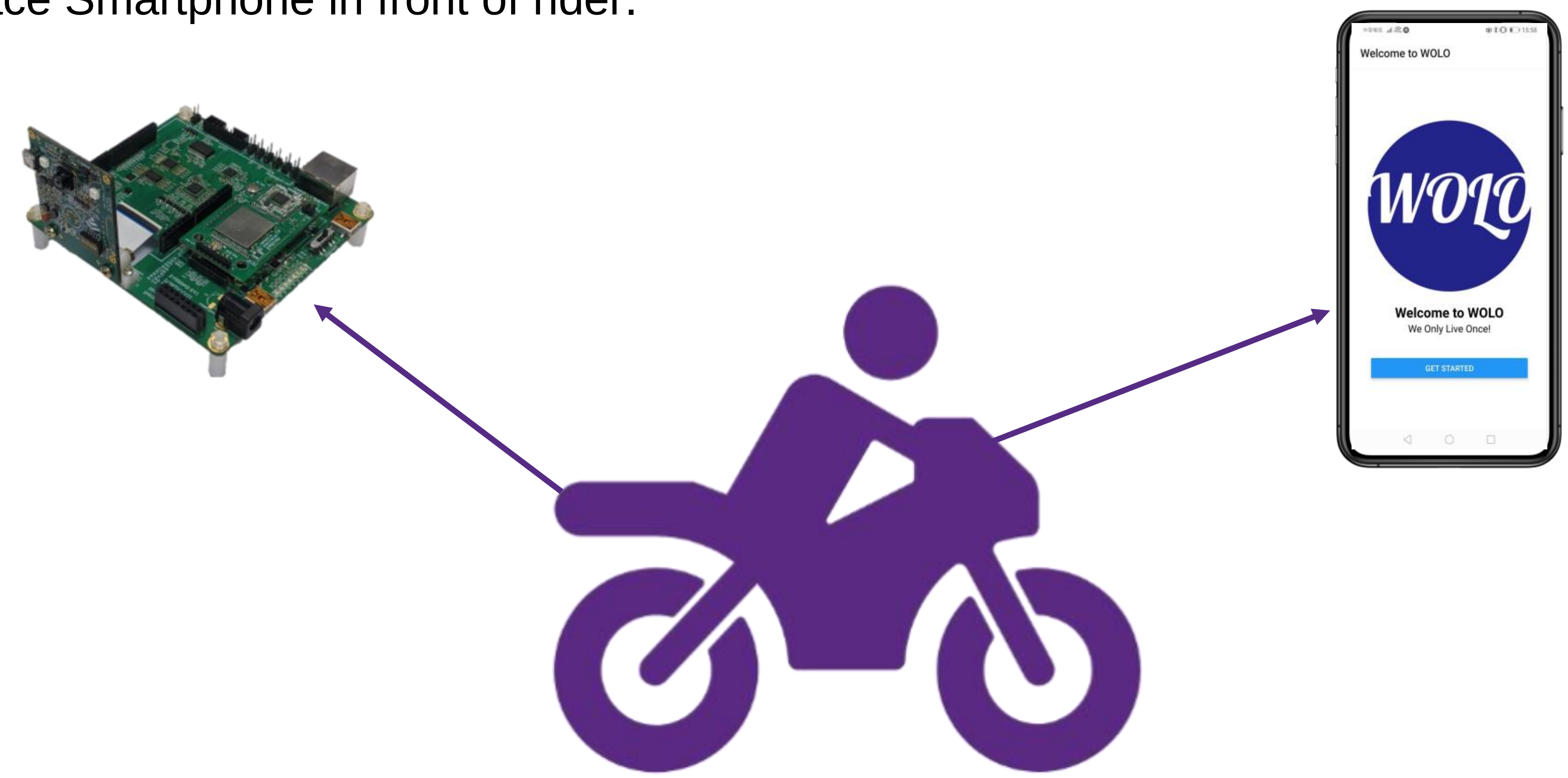
Appendix : Distance MSE

- Distance MSE for different distance objects



Appendix : Placement

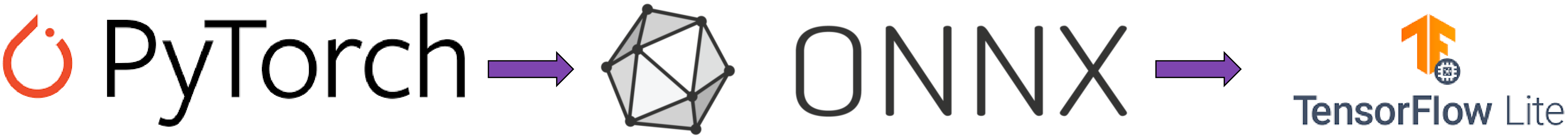
- Place ARC EM9D AIoT DK in the rear of motorcycle.
- Place Smartphone in front of rider.



Appendix : Post Training Quantization

- Post Training Quantization, convert model to INT8.

- Quantize step
  1. Convert model to onnx (PyTorch -> ONNX)
  2. Convert model to tflite and Quantize to INT8 (onnx -> tflite) (https://github.com/PINTO0309/onnx2tf)
  3. Convert model to C header file (tflite -> C)



Appendix : ARC inference result on validation set

Inference using pytorch on PC (fp32)

Inference using TFLM on ARC EM9D (INT8)

Appendix : Source Code

- React Native app code : https://github.com/coherent17/WOLO-app
- ARC EM9D AIoT DK firmware : https://github.com/c119cheng/WOLO-ARC-EM9D
- Model Training : https://github.com/c119cheng/distyolo_v7

Appendix : Demo Video explanation

- Demo Video :https://youtu.be/egK0S4pPOws

- How it is done
  - The video of road is taken using smartphone, because is hard to record a video using ARC EM9D. The inference of video is using the same model that is on the ARC EM9D (model quantized in INT8), we first read a single frame of video, convert it to gray level and resize to 320x256, as the image take by ARC EM9D's camera, then use the model to get the result, finally plot the result back to the RGB image, and make these images as a video.

ARC DESIGN CONTEST 2023

SYNOPSYS®