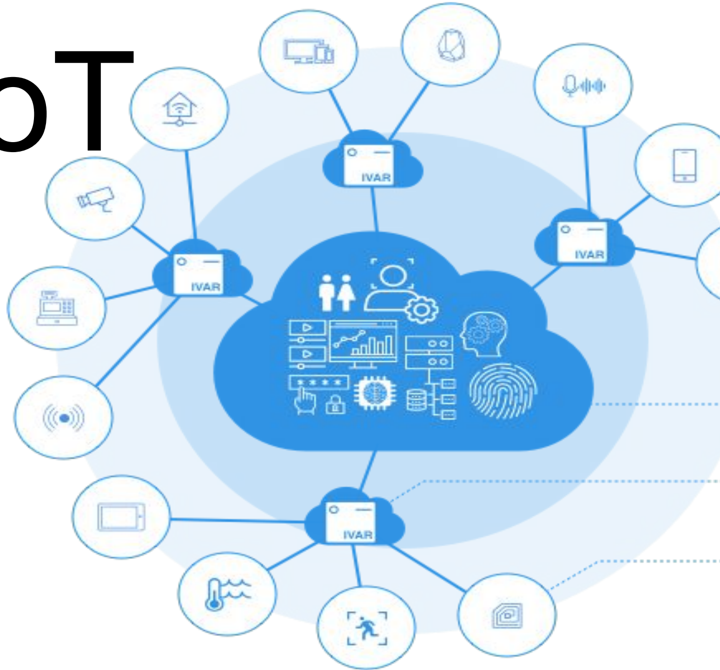
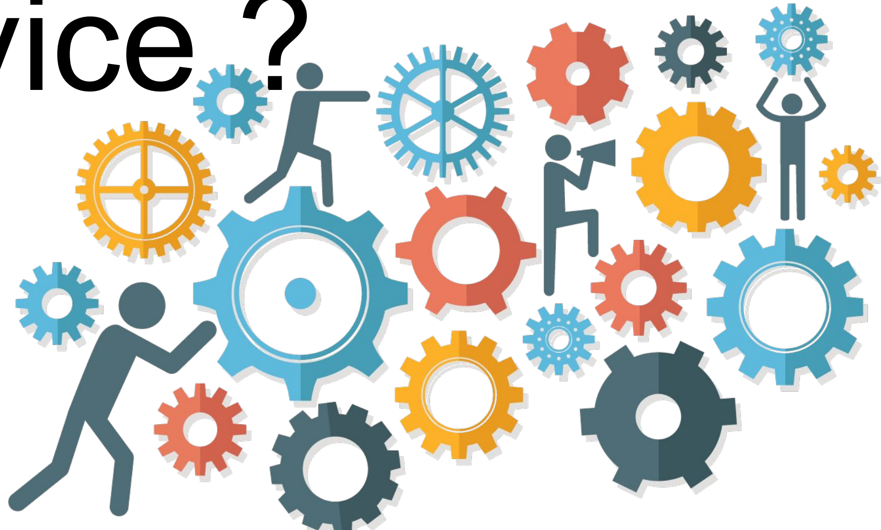
Real-Time Data processing and AI for Distributed IoT



Group 15

Rajitha Opanayaka (E/15/246)
Rashmi Thilakarathne (E/15/363)
Amila Weerasinghe (E/15/385)

Why it's so hard to do real time AI processing at the end device ?



Problem definition

- Resource Limited IoT devices
- Latency and network congestion
- Inefficient use of cloud resources
- When data is offloaded to the cloud, it causes some privacy issues

What are current solutions ?

Related work

Related work	Proposed work
Use Fog computing for real time process	Use ROOF and Fog for real time process
Map reduce to distribute computation	Map reduce based approach focus on limited resource devices
Top down approach to train model in cloud and deploy to edge	Train the model at the edge and IoT devices
Train the global model at the edge or cloud(Federated Learning)	Peer to Peer /Cluster (Static /Dynamic)/Master slave architecture
Use policy technology to deploy model trained in top to deploy in bottom edge devices	Use bottom up approach, having defined threshold limits and priorities to decide when to offload.

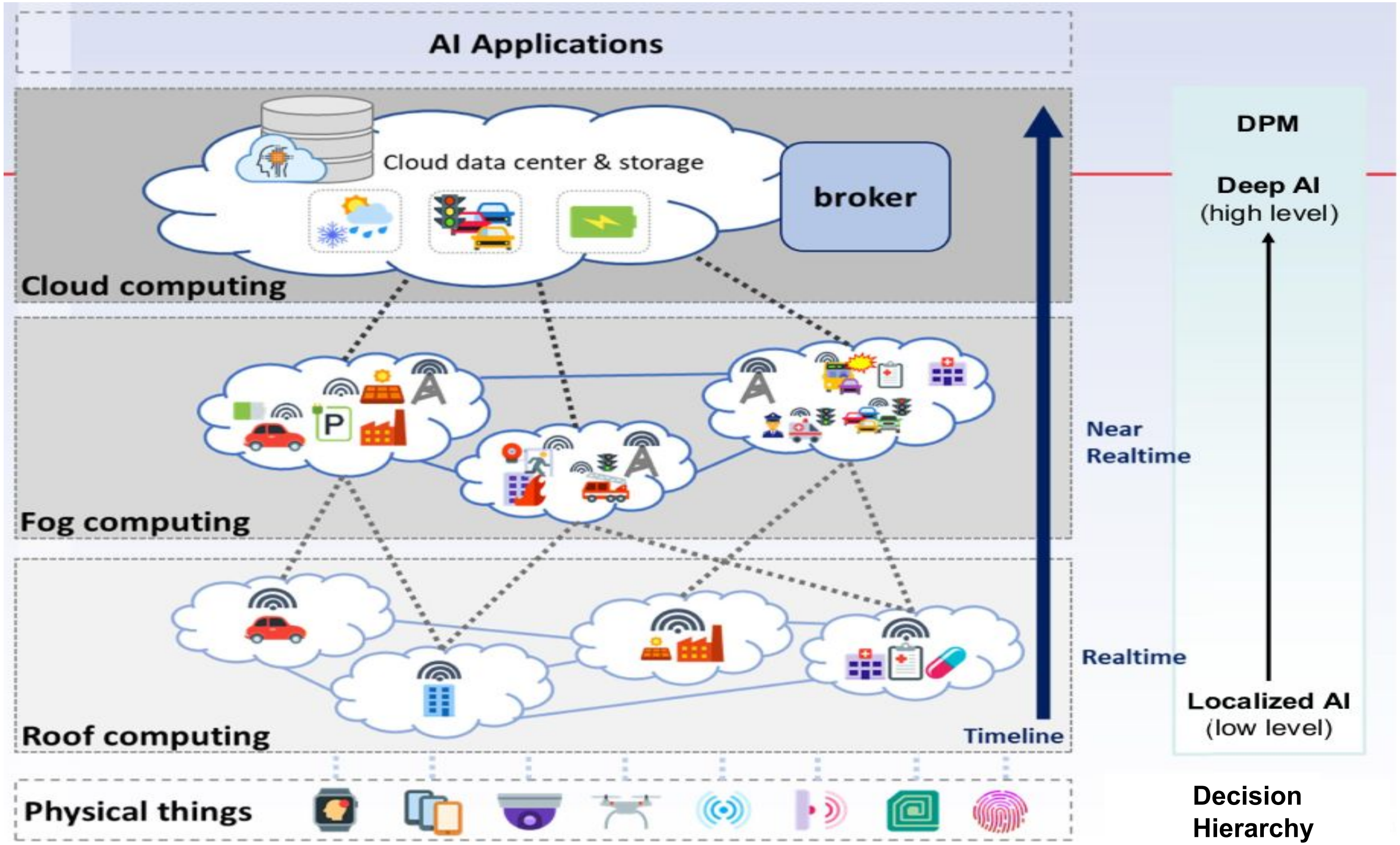
Proposed methodology:

- Instead of developing deep AI in cloud center, **novel localized AI algorithms** to be implemented in low level edge using **ROOF and Fog** computing.

- **Policy technology**-As decisions are need to made in real time. There should be a policies to when to offload data to cloud as data communication is costly.

- **Federated Learning** to fine tune the learning parameters

- Distribute Computation using **MapReduce** based approach

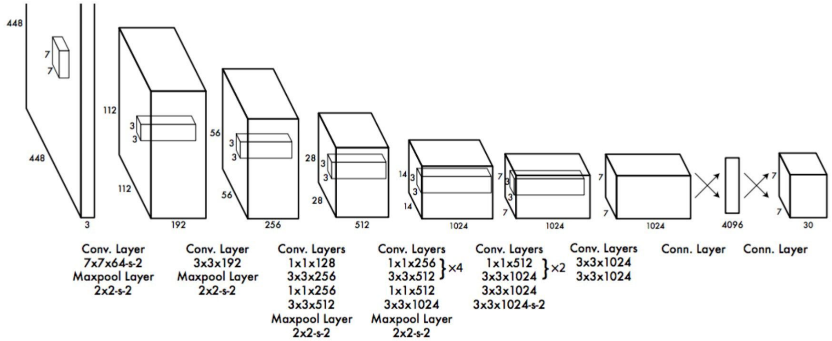


AI Applications
Cloud data center & storage
broker
Cloud computing
Fog computing
Roof computing
Timeline
Physical things
Near Realtime
Realtime
DPM
Deep AI
(high level)
Localized AI
(low level)
Decision Hierarchy

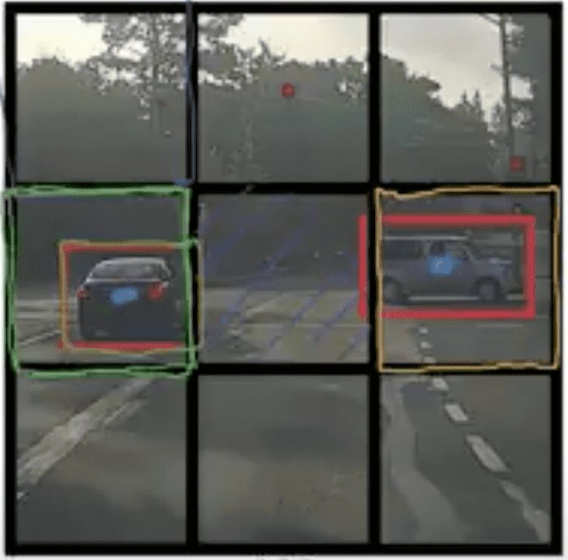
So our plan is to run a object detection algorithms on IoT devices in real time

Why Yolo ?

- Fast
- High accuracy



YOLO algorithm



Labels for training

For each grid cell:

$$y: \begin{bmatrix} p_c \\ b_x \\ b_y \\ b_h \\ b_w \\ c_1 \\ c_2 \\ c_3 \end{bmatrix} \quad \begin{bmatrix} 0 \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \end{bmatrix} \quad \begin{bmatrix} 1 \\ b_x \\ b_y \\ b_h \\ b_w \\ 0 \\ 1 \\ 0 \end{bmatrix} \quad \begin{bmatrix} 1 \\ b_x \\ b_y \\ b_h \\ b_w \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

Target output: $3 \times 3 \times 8$

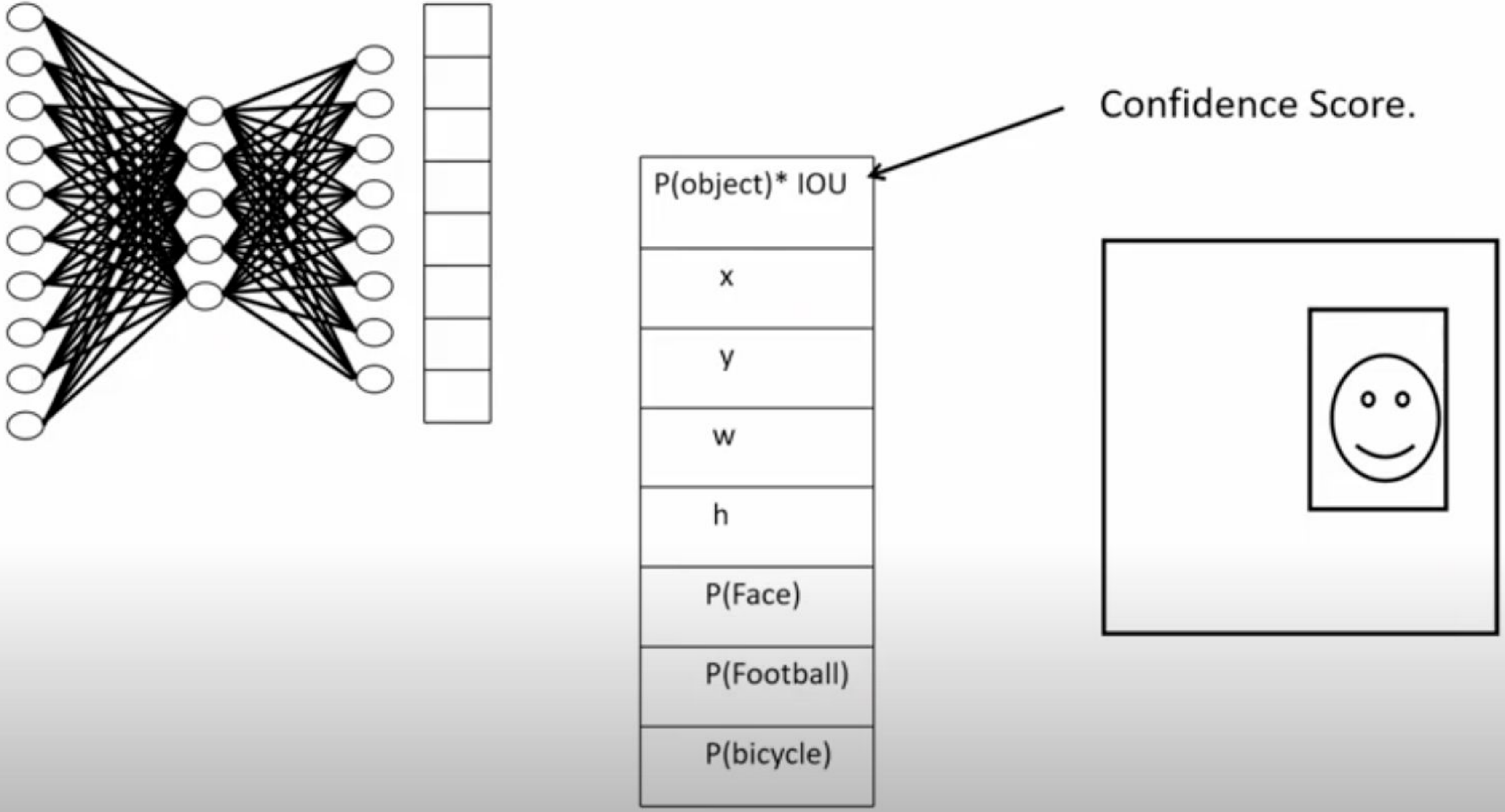
X (100×100×3) → CONV → MAXPOOL → .... → $3 \times 3 \times 8$ y

→ $19 \times 19 \times 8$

361

[Redmon et al., 2015, You Only Look Once: Unified real-time object detection]

Andrew Ng


Confidence Score.
P(object)* IOU
x
y
w
h
P(Face)
P(Football)
P(bicycle)

Our changes to Yolo

MapReduce based approach

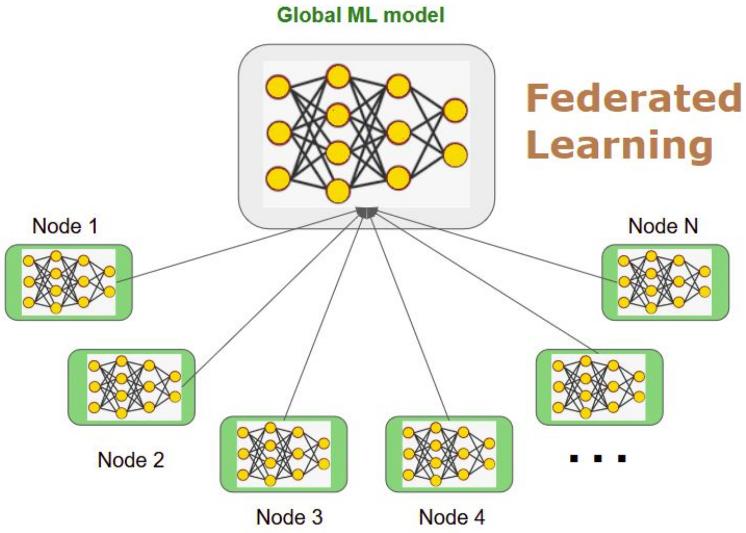
- Distribute computation over multiple nodes

Bottom up approach

- Less network delays
- Computation in closer proximity to the user

Federated learning

- Data privacy
- Learn a shared model for prediction



Evaluation

Optimized YOLO CNN Performance Results in different environment

Environment	Execution time (ms)
Google Colab cloud CPU	3060.6624
Google Colab cloud GPU	93.4970
Single Raspberry Pi	65531
Three Raspberry Pi	32744

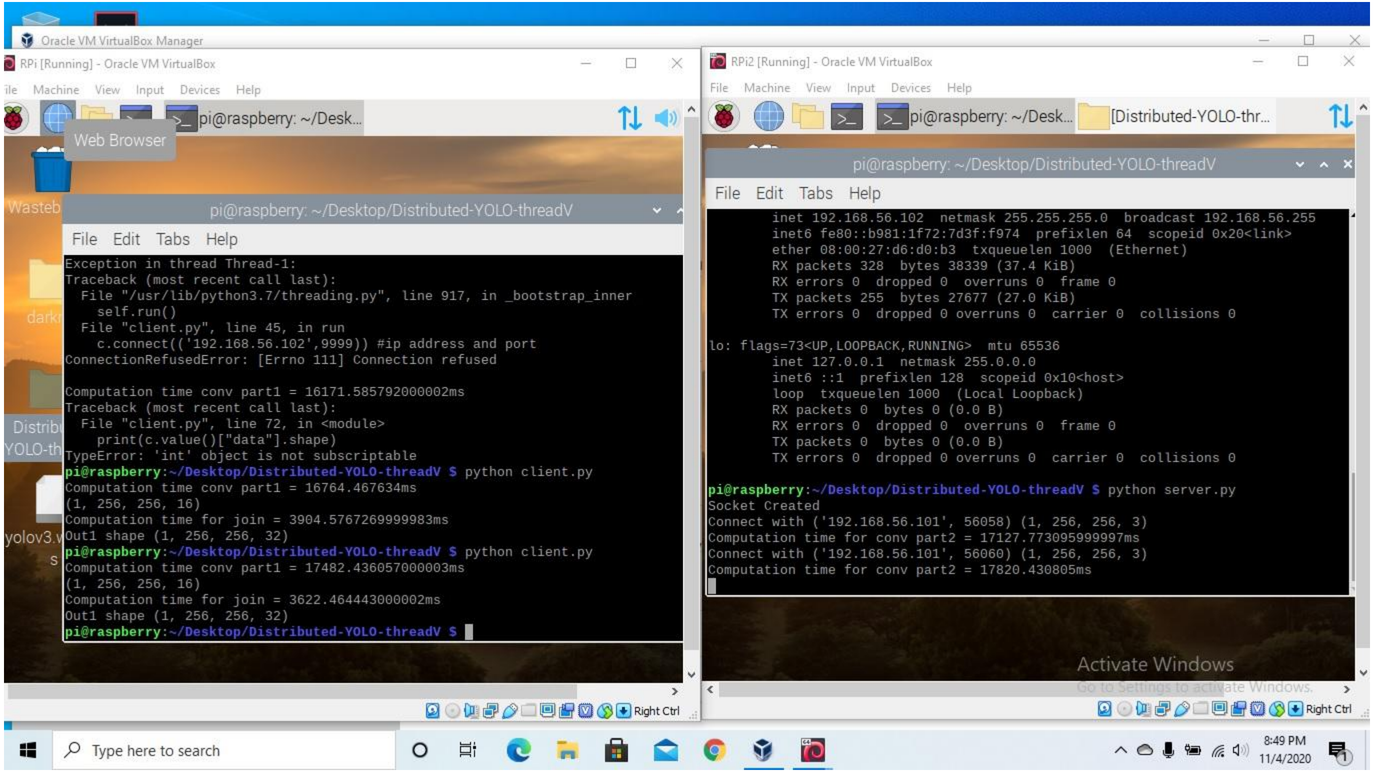
Compare and Analysis

$$\frac{CPU\ Execution\ time\ (s)}{GPU\ Execution\ time(s)} = \frac{3.0606}{0.0934} = 32.76 \quad (1)$$

$$\frac{Resource\ Availability\ in\ Colab\ Cloud - GPU(RAM)}{Resource\ availability\ of\ Single\ Raspberry\ Pi\ (RAM)} = \frac{12\ GB}{512\ MB} = 23.43 \quad (2)$$

$$\frac{Parallel\ implementation\ Execution\ time(ms)}{Single\ Raspberry\ Pi\ Execution\ time\ (ms)} = \frac{32744}{65531} = 0.499 \quad (3)$$

DEMO



Oracle VM VirtualBox Manager
RPi [Running] - Oracle VM VirtualBox
Machine View Input Devices Help
pi@raspberry: ~/Desk...
Web Browser
pi@raspberry: ~/Desktop/Distributed-YOLO-threadV
File Edit Tabs Help
Exception in thread Thread-1:
Traceback (most recent call last):
File "/usr/lib/python3.7/threading.py", line 917, in _bootstrap_inner
self.run()
File "client.py", line 45, in run
c.connect(('192.168.56.102',9999)) #ip address and port
ConnectionRefusedError: [Errno 111] Connection refused
Computation time conv part1 = 16171.585792000002ms
Traceback (most recent call last):
File "client.py", line 72, in <module>
print(c.value()["data"].shape)
TypeError: 'int' object is not subscriptable
pi@raspberry:~/Desktop/Distributed-YOLO-threadV $ python client.py
Computation time conv part1 = 16764.467634ms
(1, 256, 256, 16)
Computation time for join = 3904.5767269999983ms
Out1 shape (1, 256, 256, 32)
pi@raspberry:~/Desktop/Distributed-YOLO-threadV $ python client.py
Computation time conv part1 = 17482.436057000003ms
(1, 256, 256, 16)
Computation time for join = 3622.464443000002ms
Out1 shape (1, 256, 256, 32)
pi@raspberry:~/Desktop/Distributed-YOLO-threadV $
RPi2 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
pi@raspberry: ~/Desk...
[Distributed-YOLO-thr...
pi@raspberry: ~/Desktop/Distributed-YOLO-threadV
File Edit Tabs Help
inet 192.168.56.102 netmask 255.255.255.0 broadcast 192.168.56.255
inet6 fe80::b981:1f72:7d3f:f974 prefixlen 64 scopeid 0x20<link>
ether 08:00:27:d6:d0:b3 txqueuelen 1000 (Ethernet)
RX packets 328 bytes 38339 (37.4 KiB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 255 bytes 27677 (27.0 KiB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
inet 127.0.0.1 netmask 255.0.0.0
inet6 ::1 prefixlen 128 scopeid 0x10<host>
loop txqueuelen 1000 (Local Loopback)
RX packets 0 bytes 0 (0.0 B)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 0 bytes 0 (0.0 B)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
pi@raspberry:~/Desktop/Distributed-YOLO-threadV $ python server.py
Socket Created
Connect with ('192.168.56.101', 56058) (1, 256, 256, 3)
Computation time for conv part2 = 17127.773095999997ms
Connect with ('192.168.56.101', 56060) (1, 256, 256, 3)
Computation time for conv part2 = 17820.430805ms
Activate Windows
Go to Settings to activate Windows.
Right Ctrl
Type here to search
8:49 PM
11/4/2020

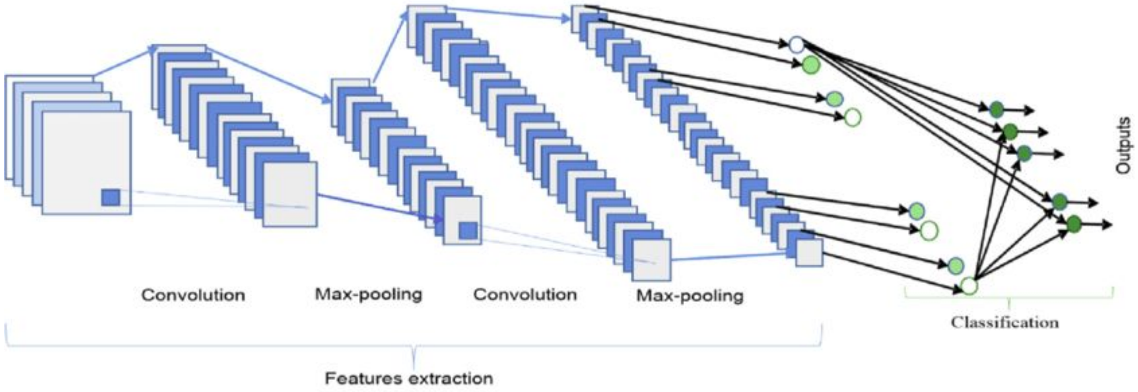
Conclusion and future direction

Recap

- Real time decision using FL ,map reduce base techniques with limited resource
- Computation distribution

Next steps

- Apply federated learning
- Offloading techniques
- Improve CNN



Project Plan

Title	T/M	Start	End	
Start of Semester 7	M	06/07/2020	06/07/2020	-
Research	T	13/07/2020	07/08/2020	20 days
Research	T	05/10/2020	23/10/2020	15 days
Implement CNN	T	05/10/2020	19/10/2020	11 days
Distribute CNN	T	11/10/2020	23/10/2020	10 days
Start of Semester 8	M	09/11/2020	09/11/2020	-
Improve CNN and accuracy	T	09/11/2020	30/01/2021	60 days
Apply Federated Learning	T	09/11/2020	15/02/2021	71 days
Apply offloading techniques	T	09/11/2020	15/02/2021	71 days
Finalize Project	T	01/02/2021	26/03/2021	40 days
End	M	26/03/2021	26/03/2021	-

Our Proposed Work

Milestones

7 semester

- ML model for limited resources IoT devices

8 semester

- Designing a federated learning algorithm.
- Designing offloading based on the threshold
- Designing the Policy framework based on the priorities

Our Accomplishments

- CNN for YOLO
- Approach to distribute computation

THANK YOU

Q&A