



Federated Learning for Object Detection in Autonomous Vehicles

2021 IEEE Seventh International Conference on Big Data Computing Service and Applications (BigDataService)

Guided by:

Prof. C Krishna Mohan
IIT Hyderabad

Presenter:

Manan Darji
CS22MTECH14004
IIT Hyderabad

Teaching Assistant:

Zarka Bashir
IIT Hyderabad

Contents:

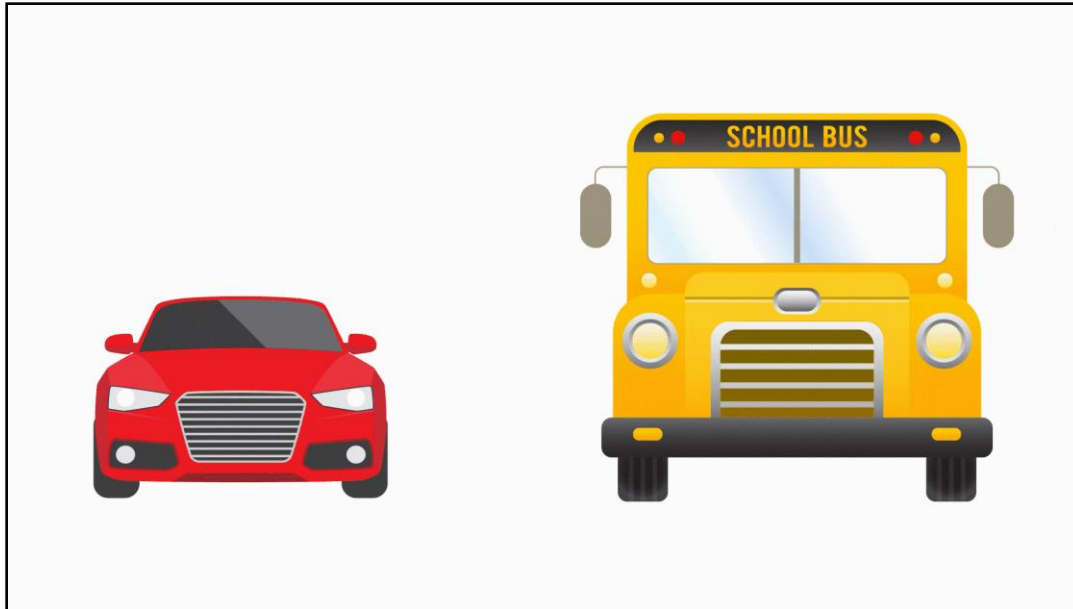
- Introduction
- Literature review
- Limitations (of previous work) and Motivation
- Problem statement
- Proposed method
- Experimentation/results
- Summarization
- Future work



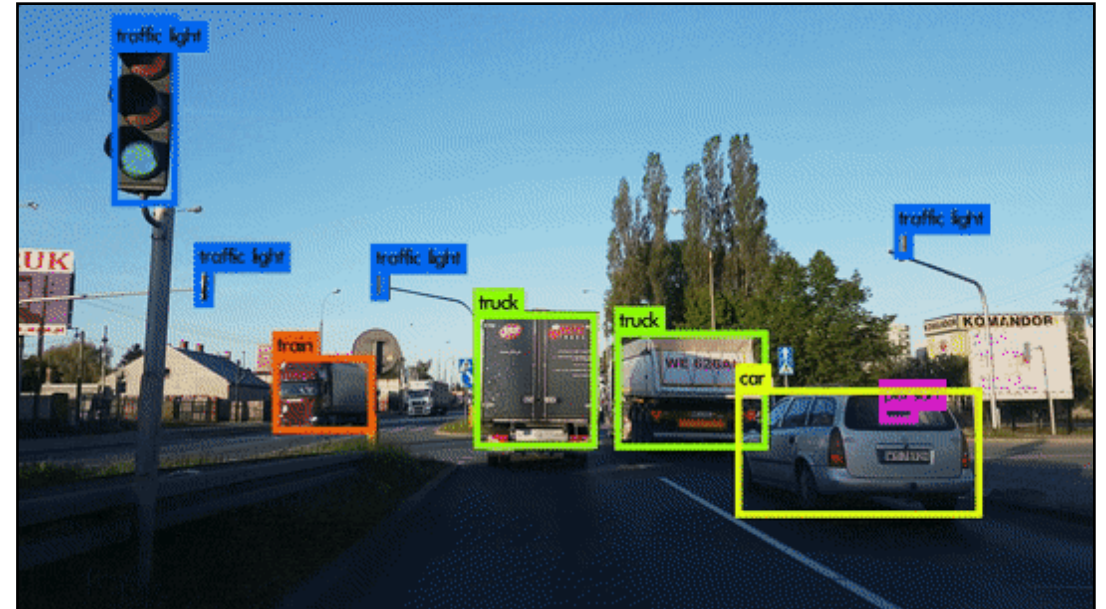
INTRODUCTION

Object Detection in Autonomous Vehicles

- Object Detection is a process of **finding all the possible instances of real-world objects**.
- It is one of the key features of Autonomous Driving Systems.



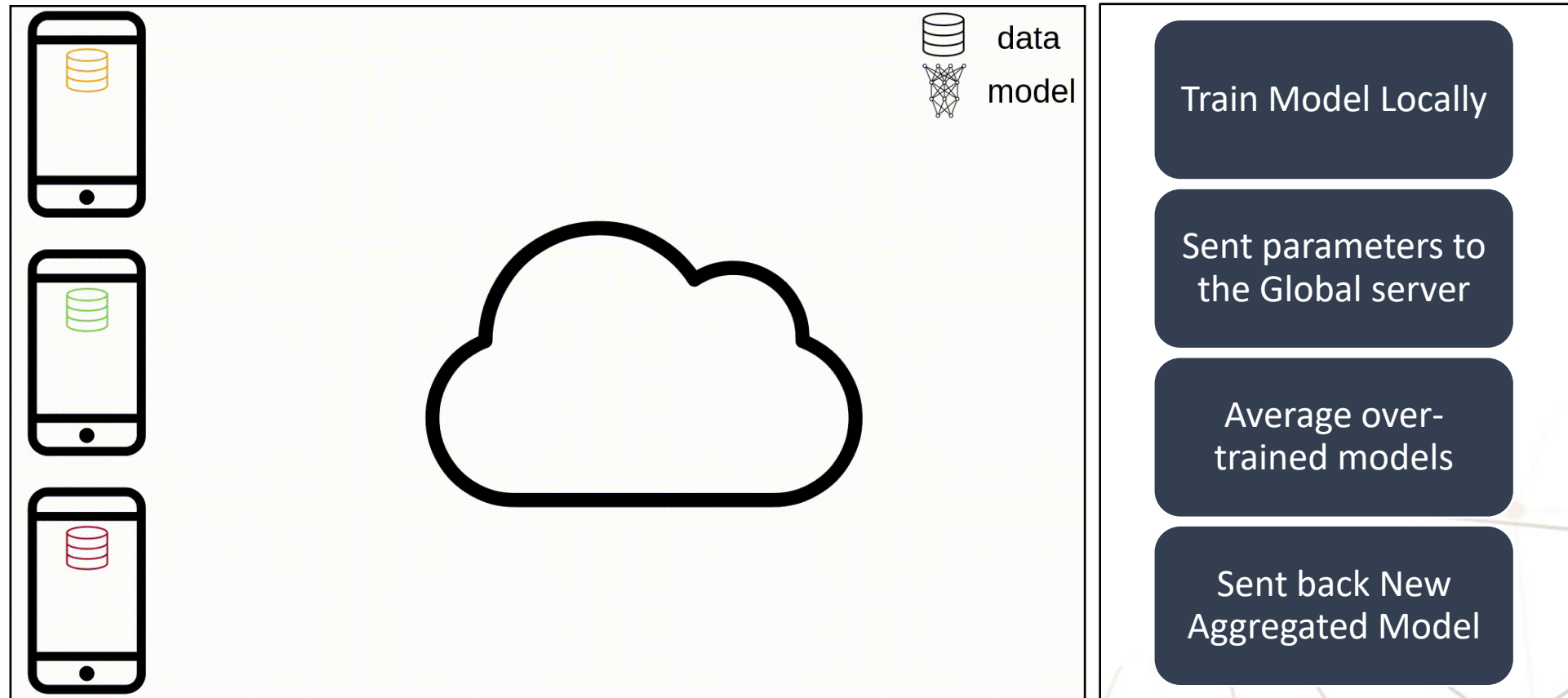
Bounding Box with classes [1]



Object Detection in Autonomous Vehicles [2]

Federated Learning

- Federated learning (also known as collaborative learning) is a machine learning technique that trains an algorithm across multiple decentralized edge devices or servers holding local data samples without exchanging them.



Federated Learning

- In traditional Deep learning, we use mini-batches SGD to train our model.

$$\min_{w \in \mathbb{R}^d} f(w)$$

$$f(w) \stackrel{\text{def}}{=} \frac{1}{n} \sum_{i=1}^n f_i(w)$$

$$f_i(w) = l(x_i, y_i, w)$$

$$w_{t+1} \leftarrow w_t - \eta \nabla f(w_t; x_k, y_k)$$

- Here X_k and Y_k are set of input and labels of size k .
- Basically we are minimizing the Average loss over the selected mini-batch of size k .

-
- So, we can consider every client in federated learning having data of K -size batches only.
 - There are two ways we can combine weights.
 - Each client k submits a gradient(g_k); the central server aggregates the gradients to generate a new model

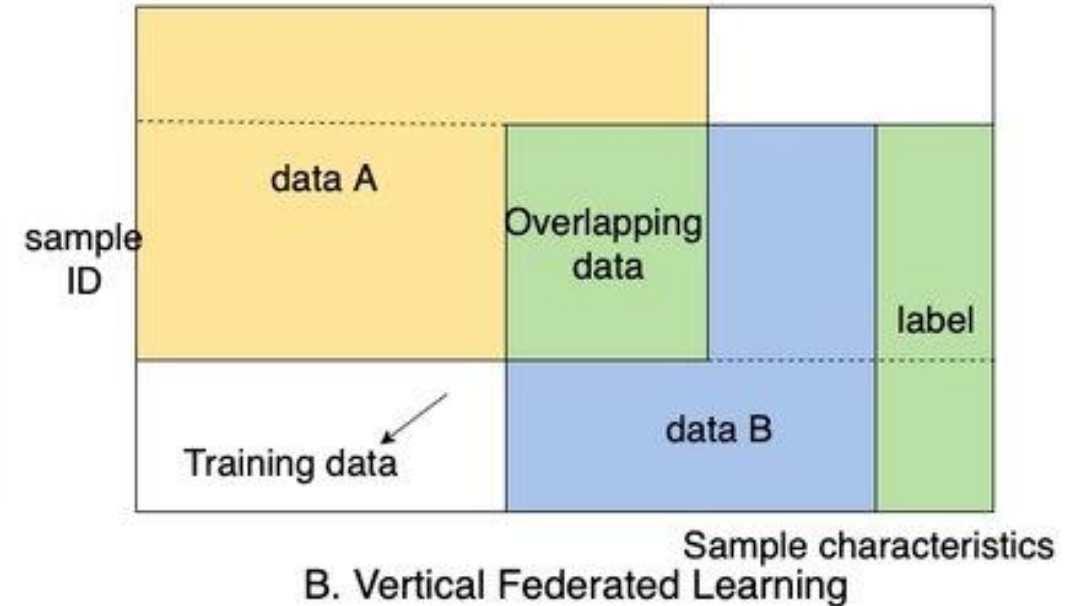
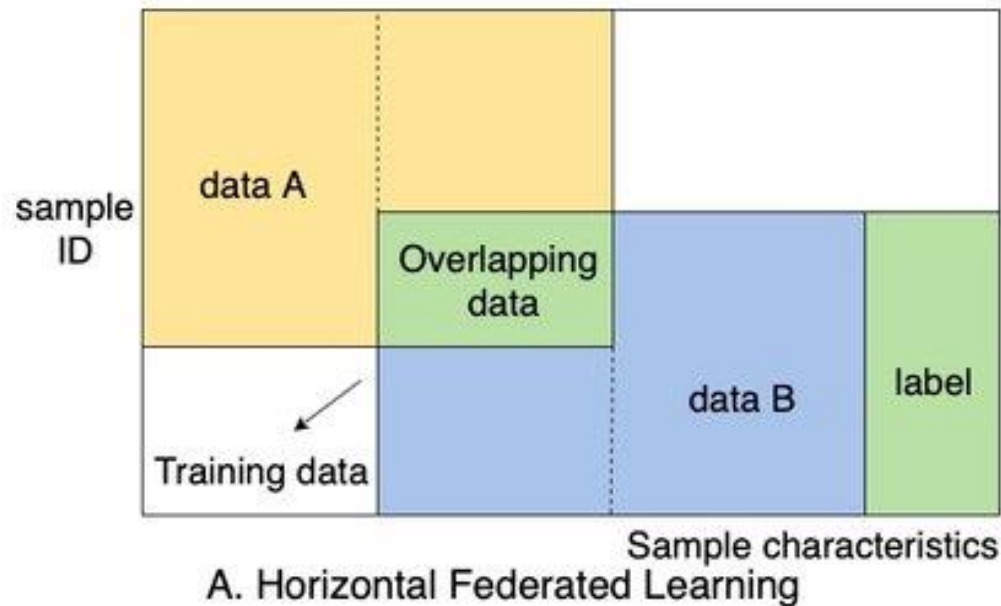
$$w_{t+1} \leftarrow w_t - \eta \nabla f(w_t) = w_t - \eta \sum_{k=1}^K \frac{n_k}{n} g_k.$$

- Each client k computes \mathbf{W} locally; the central server performs aggregation over all \mathbf{W}

$$w_{t+1} \leftarrow \sum_{k=1}^K \frac{n_k}{n} w_{t+1}^k$$

Federated Learning

Horizontal federated learning uses datasets with the same feature space across all devices, this means that Client A and Client B has the same set of features as shown in (A) below. **Vertical federated learning** uses different datasets of different feature space to jointly train a global model as shown in (B).

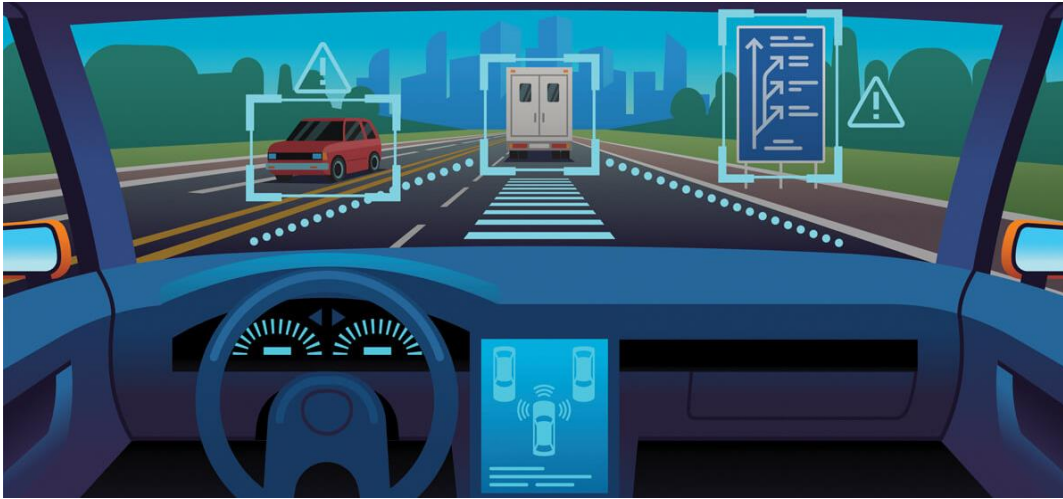


Types of Federated Learning [1]

Literature review

- Why do **Autonomous vehicles** have Privacy issues? It's Not like "Medical Data".
- There is many application for Autonomous vehicles, like **Driving assistance**, **Cruise control**, **Self-parking systems**, or some vision-based toy device that you fix in your car. All this application gets the image/Video data from your edge device/car. That data could be

sensitive, like your current location or places you travel to, photographs of your residence, etc. So, consumers may not consent to share this data.



Literature review

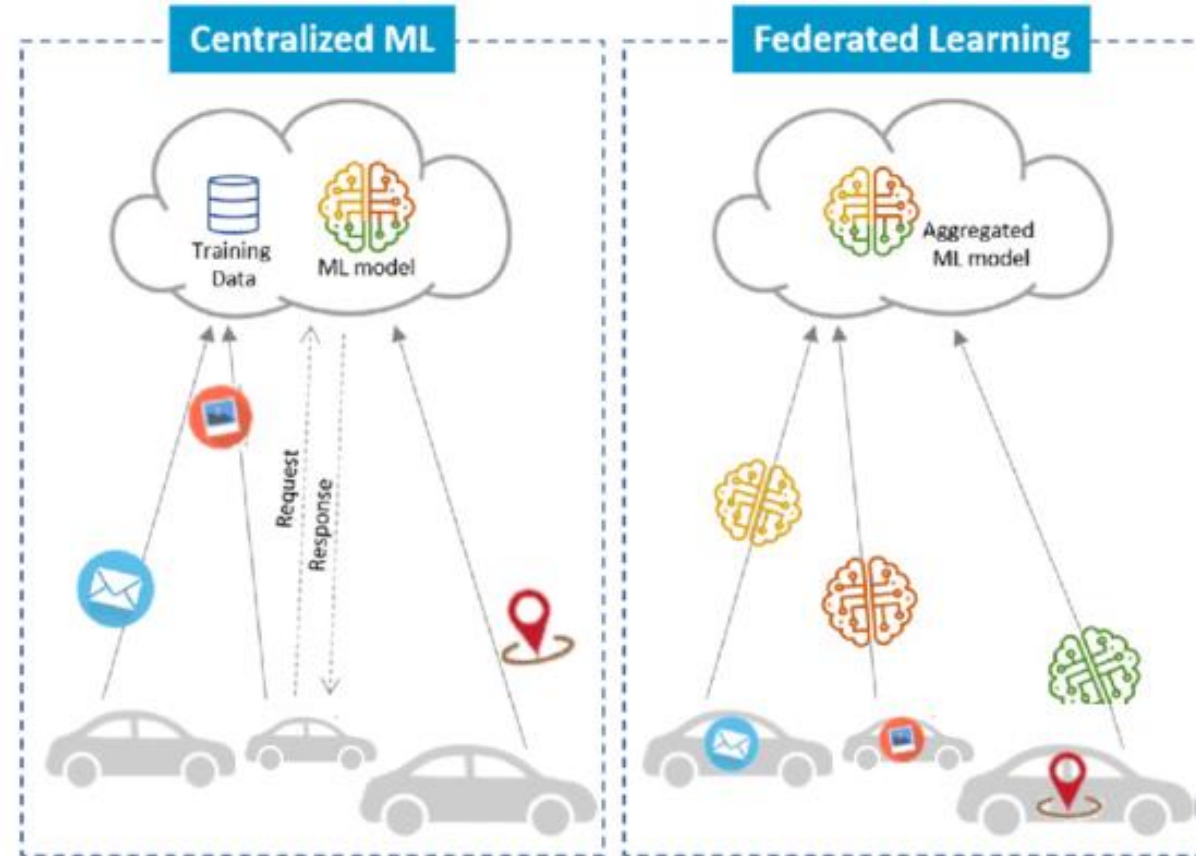
- The number of vehicles with autonomous features like **Driving assistance**, **Cruise control**, and **Self-parking systems** has been **increasing** globally.
- The collaboration of **Autonomous Driving Systems (ADS)** and **Deep Learning (DL)** has improved the performance of various tasks for self-driving vehicles.
- For **data-sensitive** industries such as healthcare and “**Autonomous vehicle**”, it leaves a considerable **privacy risk**. Not only does centralized training has a privacy risk, but it also doesn't adapt to the **dynamic environment** and each device well.
- Companies providing autonomous driving solutions **collect data from private vehicles** that use their products, apart from collecting their own data through demo runs. However, consumers may not consent to share their driving data, for instance, the places they travel to or photographs of their residences. Collecting personal data also **violates privacy regulations and policies**.
- In the scenario of the autonomous vehicle, federated learning empowers adaption to **environment dynamics**, providing feature learning in different **geographical locations**, **weather conditions**, **pedestrians behavior dynamics**.

Limitations (of previous work) and Motivation

- Traditionally, object detection models are usually trained at a centralized location by collecting data from multiple sources. This raises concerns about **Data privacy** among other issues.
- We Know we need a lot of data to train Deep Learning models. If we collect all data from edge devices, it will, of course, increase the **Upstream Data Load**.
- We also have to store all that data, which will also require some **temporary storage** on the server side.
- To train on such large data, we need a **lot of computing**.
- Of course, collecting data from private vehicles violate **privacy regulations and policies** in some cases.

Problem statement

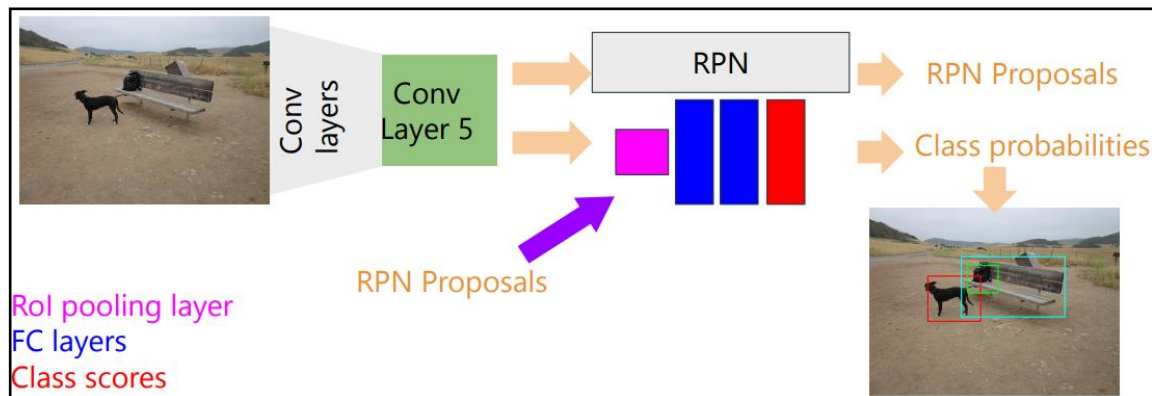
This paper will discuss the implementation of a Federated Learning prototype to train object detection onboard vehicles. Through this prototype, we test the hypothesis that **Federated Learning can yield performance comparable to traditional centralized deep learning** while securing user data locally.



[1] Traditional vs. Federated Learning

Proposed method

- Legacy object detection techniques follow **Discrete pipelines** for extracting features, warping and classifying the images.
- Traditional pipeline uses **RPN (Region Proposal Network)** with **Faster R-CNN**. Approximate joint training, where both RPN and R-CNN are trained together, is used to obtain the results **faster**.
- Although eliminating the selective search speeds up the region proposal process, the R-CNN-based architectural frameworks require enormous region proposals, which are **computationally expensive** and **inefficient**. The solution to this problem is introduced as some **Reinforcement Learning (RL)** model for object detection.
- A different approach to object detection is YOLO. Which is Developed as a Single Convolutional Network. Which localizes the object based on the class probabilities at a pace of **~ 45 fps**. The authors have used **YOLO** for this Prototype.

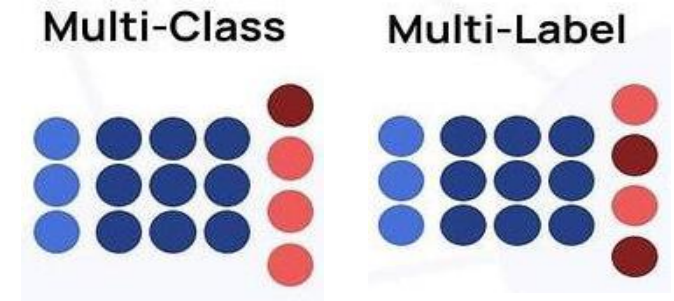


Proposed method

YOLOV3 Model for Object Detection

- YOLOV3 [7] is one of the advanced versions of object detection frameworks.

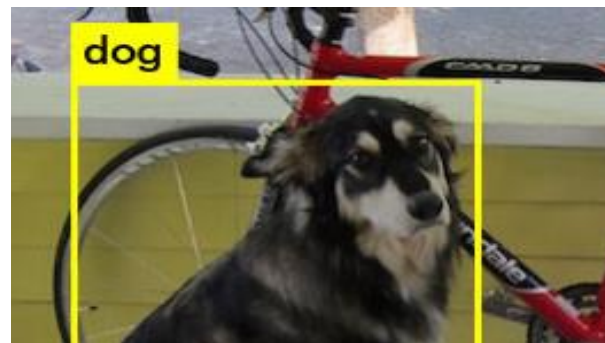
[at that time]



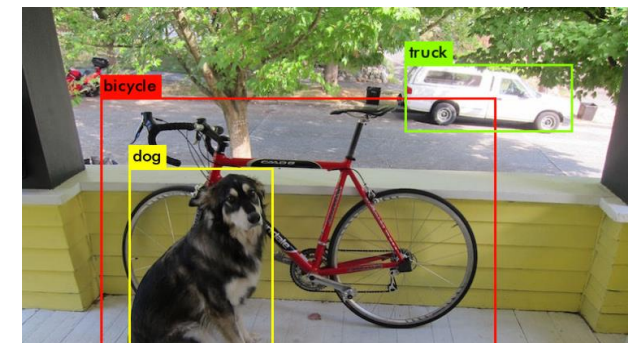
Detect Multiple Objects



Predict Classes for Object

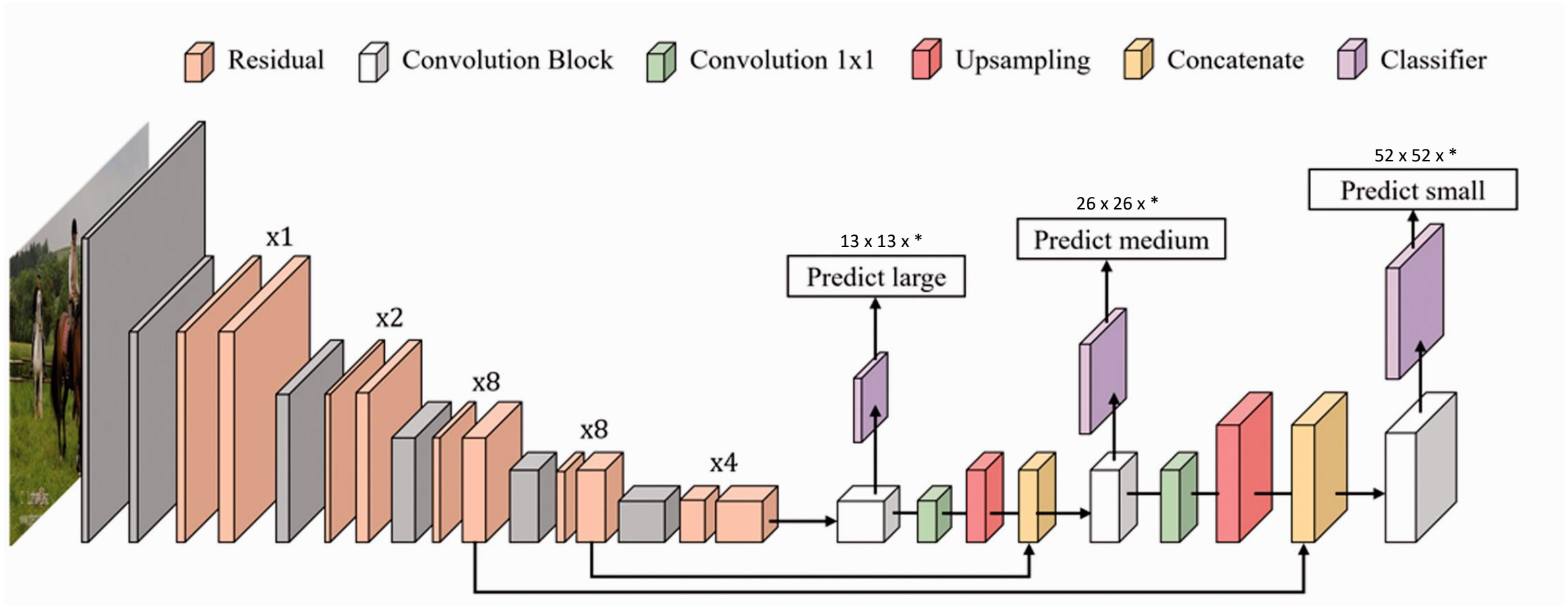


Identify the Location for Object



Proposed method

Yolo Architecture:



- It Contains 106 Layers.
- Prediction at Layers 82, 94, & 106.

Proposed method

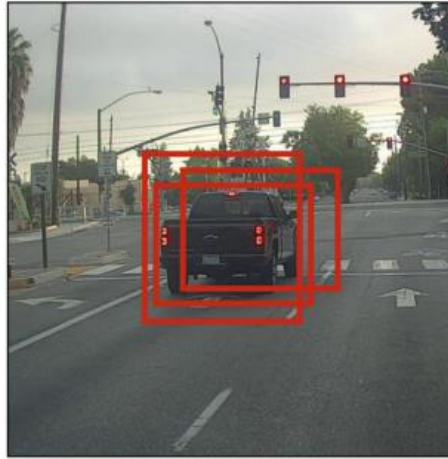
Non-maximum Suppression (NMS)

- **Input:** A list of Proposal boxes B , corresponding confidence scores S and overlap threshold N .
- **Output:** A list of filtered proposals D .

Algorithm:

- Sort all the bounding boxes in the decreasing order of confidence scores.
- Pick the first bounding box, which has the maximum score.
- The next bounding box with the highest confidence is picked.
- Repeat until all the bounding boxes above a threshold value are selected.

Before non-max suppression



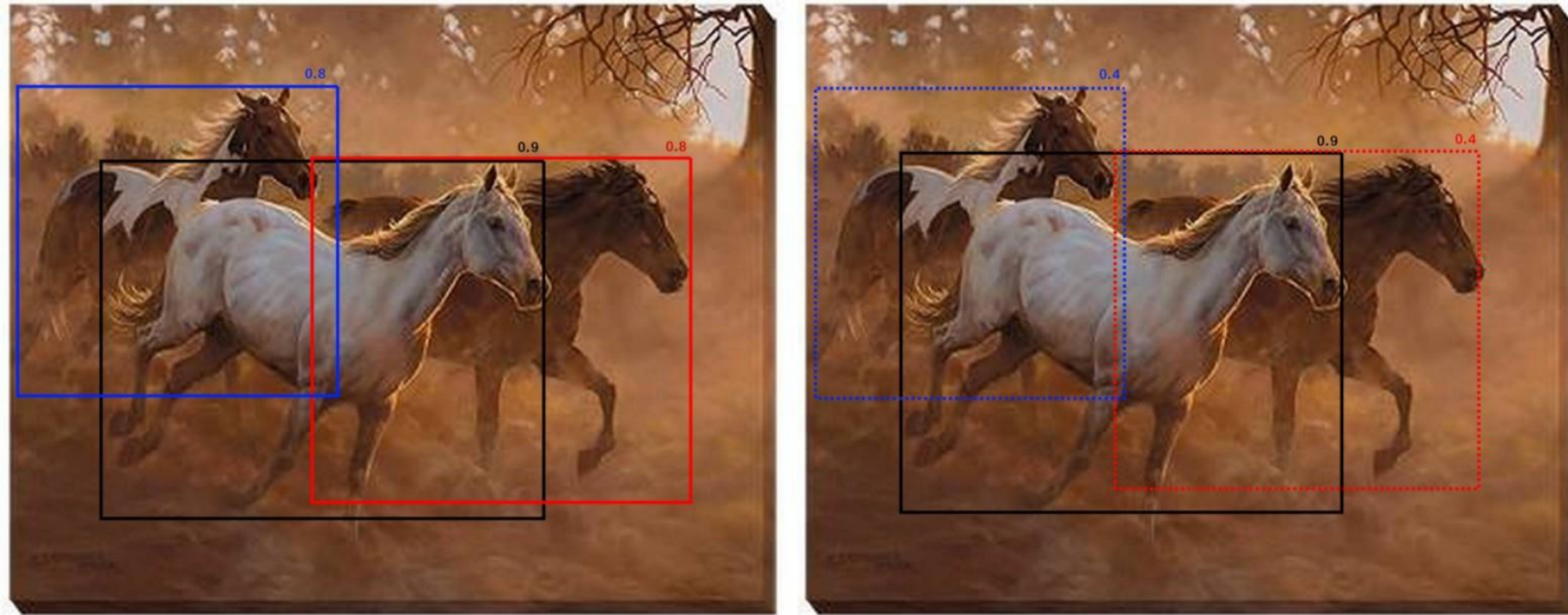
After non-max suppression



Non-Max
Suppression



Proposed method



“instead of completely removing the proposals with high IOU and high confidence, reduce the confidences of the proposals proportional to IOU value”

This is called Soft-NMS.

Proposed method

- Authors Use **horizontal federated learning** with numerous clients $C = \{C1, C2, C3, \dots, Cn\}$ to collaboratively perform object detection on their respective image data samples $D \{D1, D2, D3, \dots, Dn\}$.
- W_i are the training weights obtained from client C_i , trained on
- As defined in equation 2, the FL model is defined as a function of aggregated weights.
- They do this for a given number of communication rounds, R between Server S and Clients C .
- weights aggregation formula is defined in equation 3. For each client in C .

$$\sum_{i=1}^R f(Fl_{agg}) \text{ where } R \in \mathbb{Z} \wedge 1 \leq R \leq n \quad (1)$$

$$f(Fl_{agg}) = Agg\{W_i(D_i, C_i) : i \in \mathbb{Z} \wedge 1 \leq i \leq n\} \quad (2)$$

$$Agg\{W_i(D_i, C_i)\} = \sum_{i=1}^N \sum_{j=1}^L \frac{W_{ij} * D_i}{T} \quad (3)$$

N = Number of clients

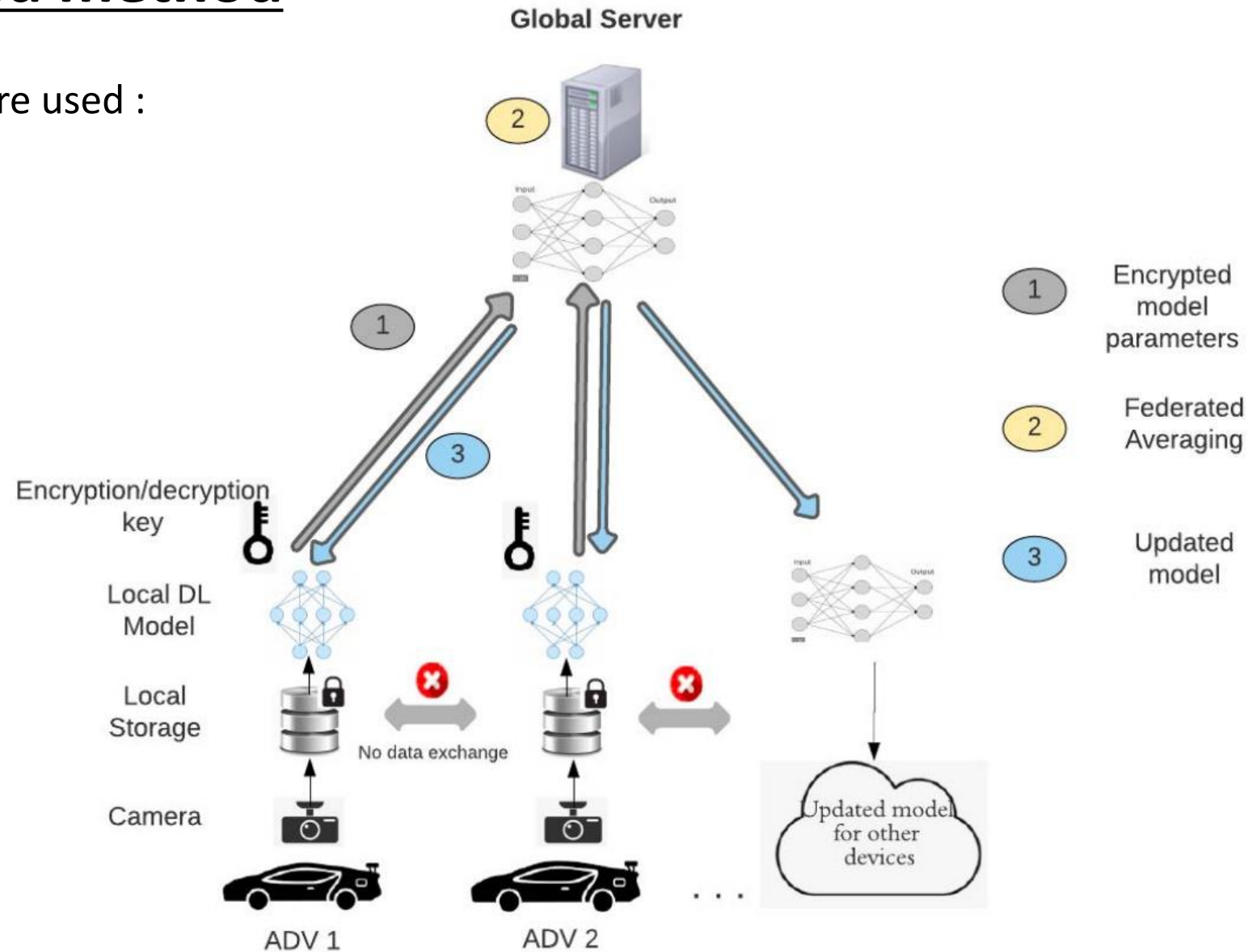
L = Number of layers of yolov3 model

T = Sum of total number of data samples from all clients C .

D_i = number of images used in training on client C_i for e number of epochs

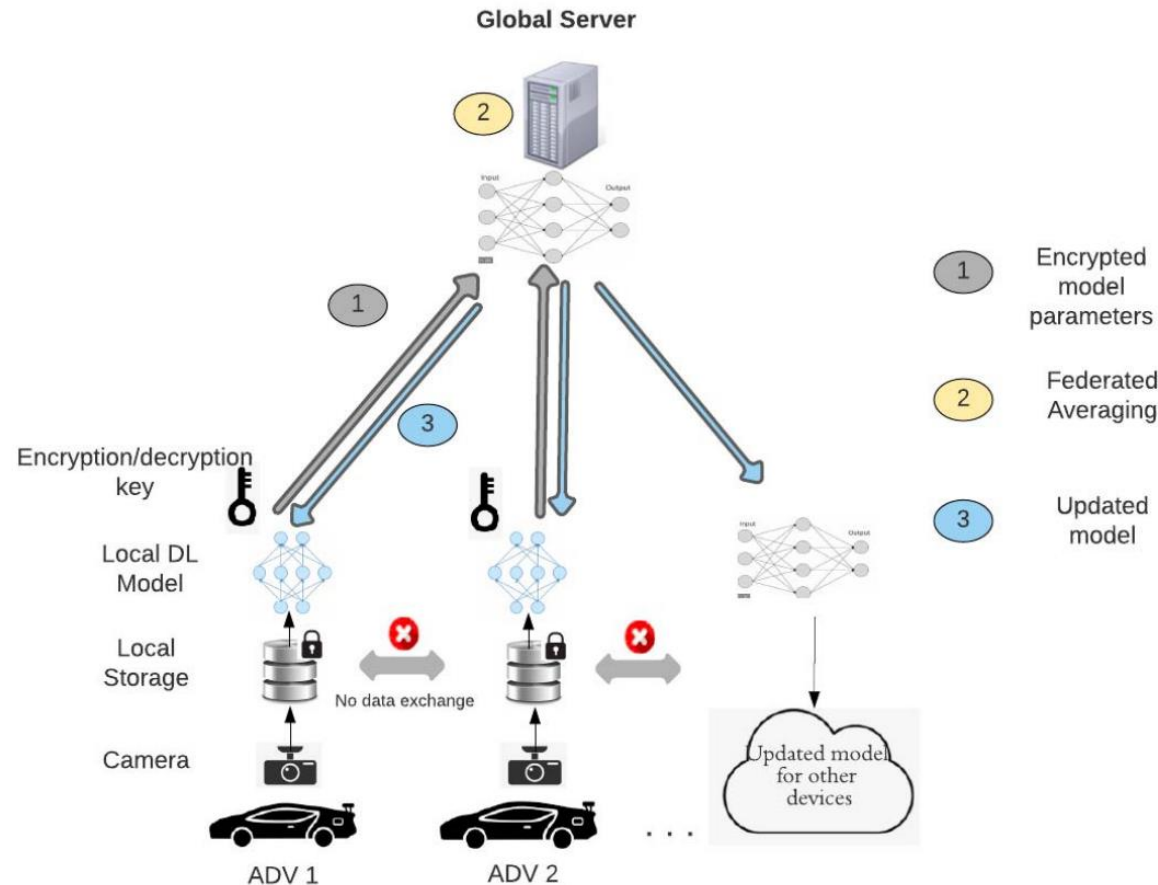
Proposed method

Architecture used :



Proposed method

Federated Learning Algorithm:



Algorithm 1: Federated Averaging Algorithm

Require: : Server S , Clients C_i , Communication Rounds R , Data Sample sizes D

1: At Server

while $Rounds < R$ **do**

Server S initializes S communication to listen to Client;

Receive encrypted weights from all clients C_i ;

Decrypt the weights;

Aggregate weights and save the consolidated weights;

Establish the communication role as sender;

Send encrypted weights.

2: At Clients

while $Rounds < R$ **do**

Bind the connection with Server S ;

Training D number of images for e number of epochs;

Save the weights;

Send encrypted weights to Server S ;

Establish the communication role as listener;

Receive encrypted aggregated weights;

Decrypt the weights;

Save weights and load model to train with aggregated weights;

Dataset Used



- Their tasks of interest are stereo, optical flow, visual odometry, **3D object detection** and 3D tracking.
- For this purpose, They equipped a standard station wagon with two high-resolution color and grayscale video cameras. Accurate ground truth is provided by a Velodyne laser scanner and a GPS localization system.
- Their datasets are captured by driving around the mid-size city of [Karlsruhe](#)[\[Germany\]](#), in rural areas and on highways. Up to 15 cars and 30 pedestrians are visible per image.



Dataset Used

- Data set Used:
 - **KITTI Vision Benchmark Dataset**
- It consists of 8 classes:
 - Car, Van, Truck, Pedestrian, Person sitting, Cyclist, Tram, and Misc.
- Since KITTI dataset is **imbalanced**, in that 'Car' and 'Pedestrian' classes have more occurrences in comparison to other classes.
- **Conversion to YOLO Format:**
 - KITTI labels need to be converted to YOLO format, which is:
 - **<object-class> <x> <y> <width> <height>**
- In our experiment, authors have deployed the YOLO algorithm in Tensorflow with Python. There are several hyperparameters of the model that has been tuned for better detection. Which I explore while implementing the code.

- Data Split Strategy:

Class	Client 1	Client 2	Client 3	Client 4
DontCare	1337	1225	1177	2487
Misc	160	196	245	173
Person Sitting	58	76	43	11
Cyclist	273	336	594	418
Car	5741	7758	6241	6206
Pedestrian	1236	1223	820	488
Truck	155	157	136	432
Van	494	743	682	513
Tram	112	97	144	66

TABLE I: Object Distribution Frequency

Experimentation/results

- **mAP**

- **Precision:**

- is a measure of when *"your model predicts how often does it predicts correctly?"* It indicates how much we can rely on the model's positive predictions.

- **Recall:**

- is a measure of *"has your model predicted every time that it should have predicted?"* It indicates any predictions that it should not have missed if the model is missing.

		True Class	
		Positive	Negative
Predicted Class	Positive	TP	FP
	Negative	FN	TN

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

The best model is the one with higher **Precision** as well as **Recall**.


$$mAP = \frac{1}{|classes|} \sum_{c \in classes} \frac{\#TP(c)}{\#TP(c) + \#FP(c)}$$

mAP is one of the widely used metrics to measure the accuracy of object detection models based on all the classes detected

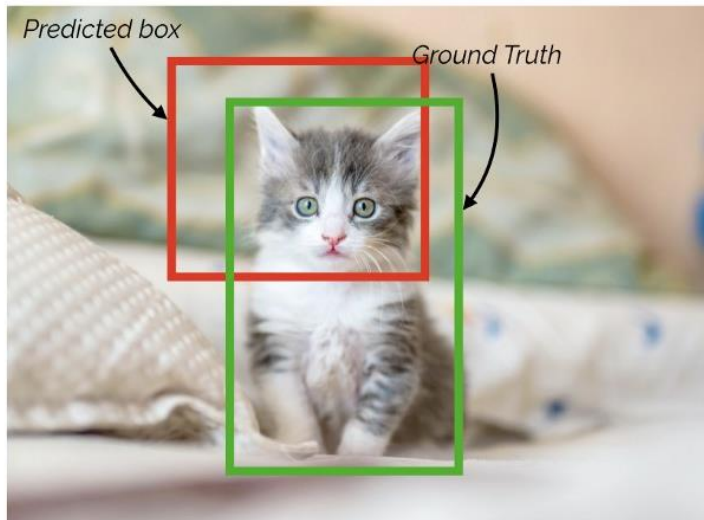
Experimentation/results

- **IoU** (Intersection Over Union)

- It is a measure of finding the area of overlap between two regions over the union of the two regions.
- IoU is used for finding the percentage of overlap between predicting bounding box values and ground truth bounding box values over the area of their union.
- The IoU score value ranges from 0 to 1. The higher the IoU score, the better the prediction.

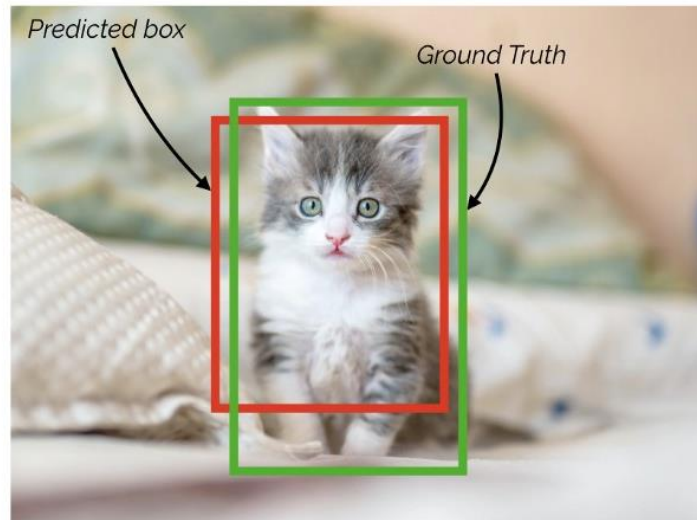

$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$

False Positive (FP)

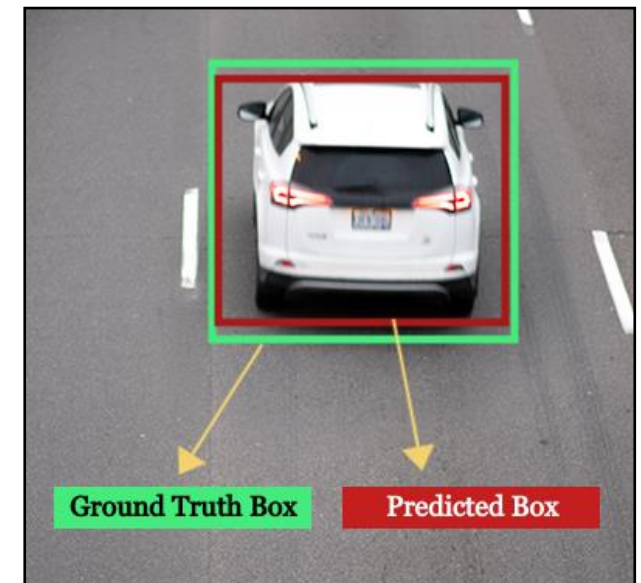


IoU = ~0.3

True Positive (TP)



IoU = ~0.7



Experimentation/results

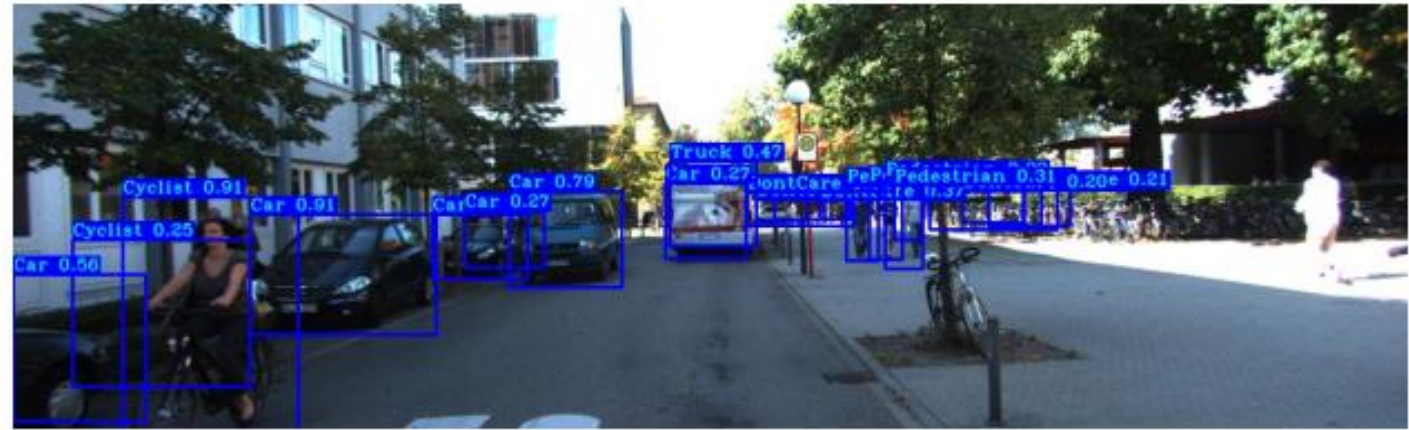
- They ran object Detection using YOLO individually for the four clients for **5 epochs** and repeated the process for **15 communication rounds**.
- They observe that although the model is trained for 5 epochs, the mean **average precision** value is **considerably high**.

Client	Data Size	Rounds	MAP
C1	1615	15	68.5%
C2	1725	15	66.9%
C3	1662	15	64.7%
C4	1632	15	64.4%

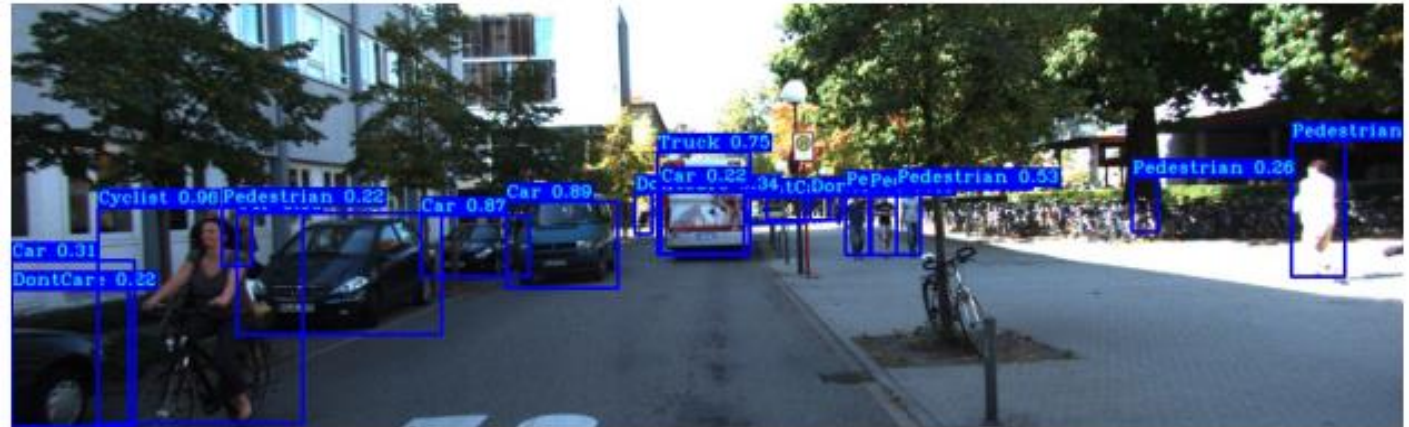
TABLE II: FL Experiment Results

Experimentation/results

The figure shows the detection images for 10 and 15 communication rounds, respectively. There is a considerable improvement after indirectly gaining knowledge from unseen data of other clients, in that objects are classified correctly, and pedestrian towards the right has been detected. These results demonstrate the working of the FL prototype.



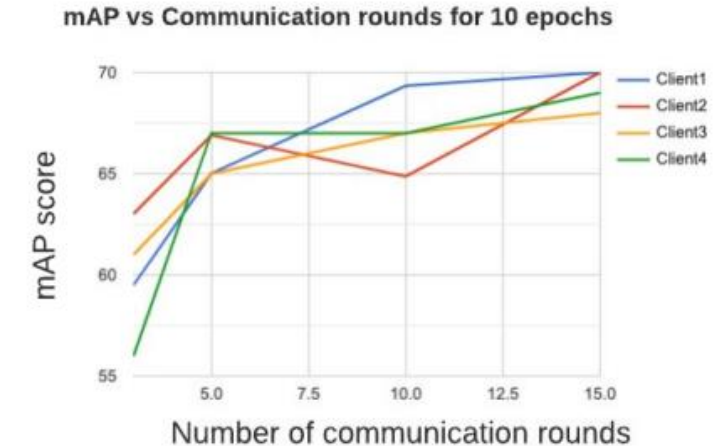
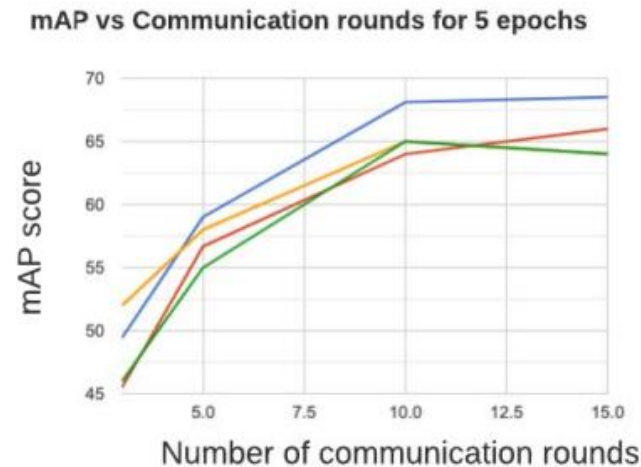
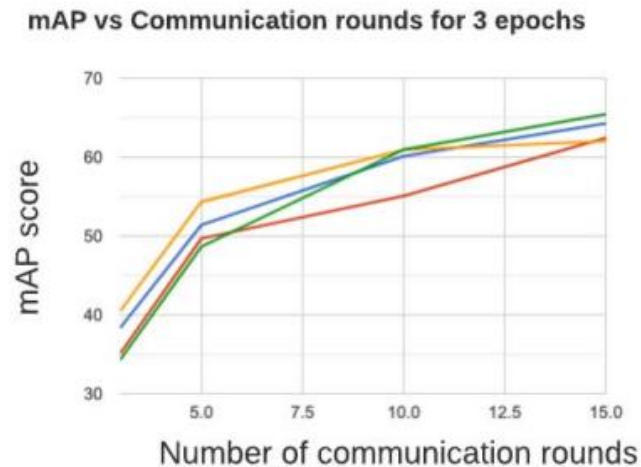
(a) Object Detection with FL for 10 comm rounds



(b) Object Detection with FL for 15 comm rounds

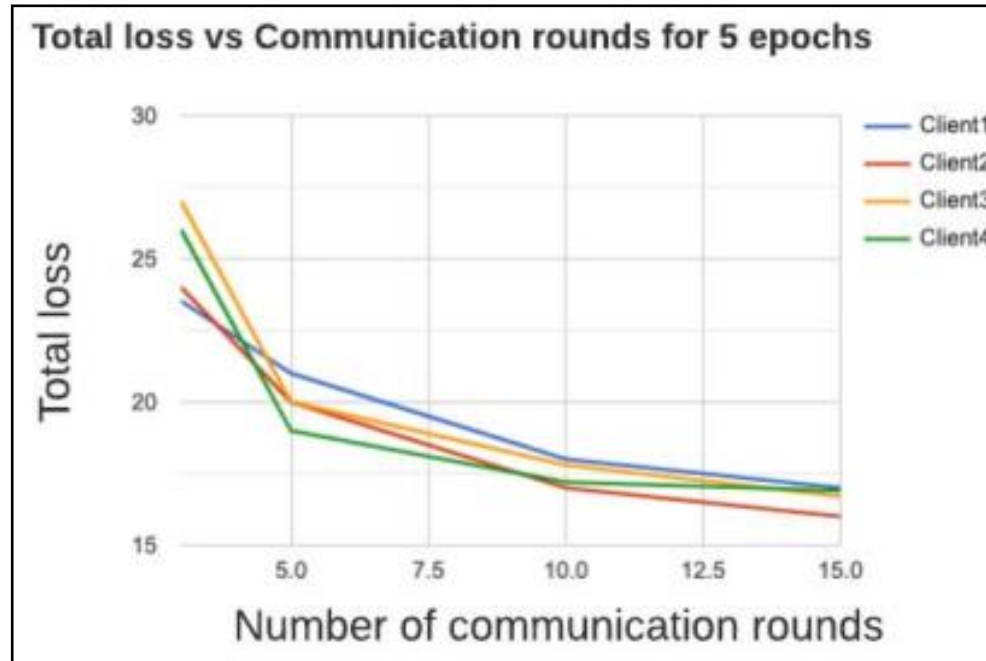
Experimentation/results

- We further experimented with different numbers of epochs and communication rounds for the FL model. From the plots shown in figure [4], we infer that the model converges at mAP value of around 68%.
- Increasing the number of epochs in the range of 3 to 5 to 10 shows a gradual increase in the mAP after each communication ro. However, the improvement in the model performance is negligible after reaching a precision score of 68% for all clients.

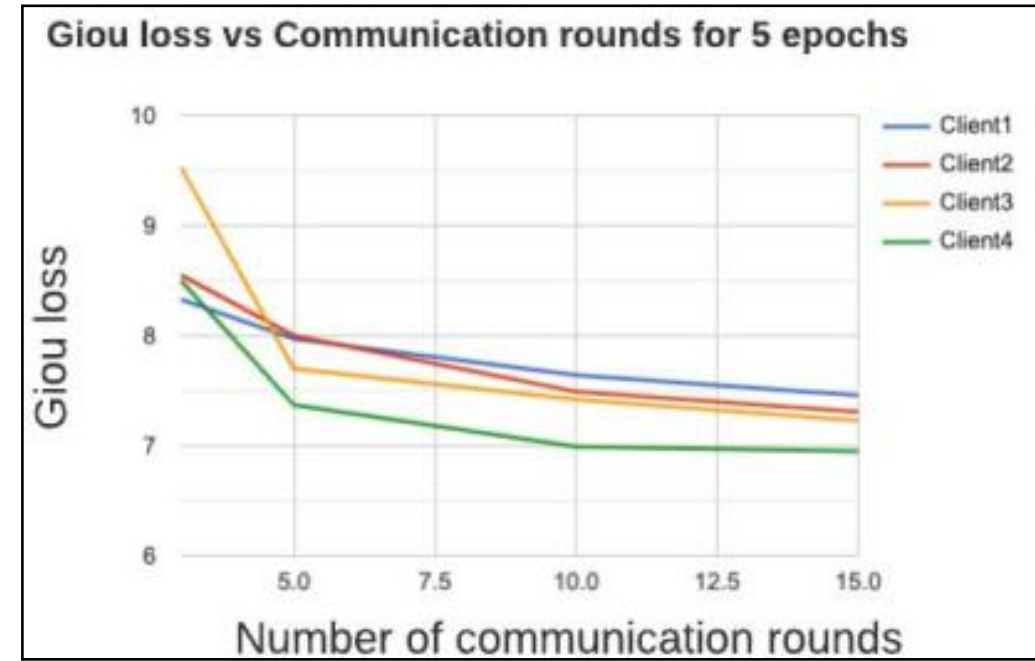


Experimentation/results

- The plots of total loss versus the communication rounds for 5 epochs in figure 5 shows a gradual decrease in the losses until the model converges after 10 communication rounds.



Total Loss



Generalized Intersection over Union Loss

Experimentation/results

- **Comparison of Deep Learning with Federated Learning**

- 6K images are trained using the YOLOV3 model on a **centralized system for 5 and 10 epochs** and the FL model where the same **6K images are distributed among 4 clients** and where each client has YOLOV3 configured to run for 5 and 10 epochs and the weights are tuned using **FL algorithm for 3 rounds**.
- It can clearly be inferred that the performances of FL and DL are comparable, where overall training loss and mAP are reduced and enhanced, respectively, with FL modeling. With the increase in communication rounds for FL, it is possible to achieve similar results as traditional models.
- However, the **interesting part** is the time taken to complete the training. The FL model training takes a **lot less time** than the traditional DL model training, **thus reducing the resource needs**.

Training Type	Data Size	Epochs	MAP	Total Loss
DL	6481	5	44.473	21.43
FL(3 rounds)	6481	5	46.063	26.26
DL	6481	10	65.171	16.30
FL(3 rounds)	6481	10	63.015	8.5

TABLE III: DL vs FL - mAP, Loss, Latency Analysis

Experimentation/results

They have shown that while training the **7K images** of the **KITTI dataset** for one epoch on a **single client** took **27 minutes**, the FL approach with data distributed among **4 clients** took only **10 minutes**. We can therefore deduce that the time taken is directly proportional to the data size.

Training Type	Data Size	Epochs	MAP	Total Loss
DL	6481	5	44.473	21.43
FL(3 rounds)	6481	5	46.063	26.26
DL	6481	10	65.171	16.30
FL(3 rounds)	6481	10	63.015	8.5

TABLE III: DL vs FL - mAP, Loss, Latency Analysis

Summarization

- Authors have constructed **and evaluated a prototype FL system** on the **KITTI Vision Benchmark 2D** image dataset. Object detection models are trained locally on a vehicle's dataset in their prototype. The resultant weights are securely aggregated at the global server using symmetric encryption techniques during data transfer to yield an improved model.
- The FL model converged at **68% mean average precision**. They compared object detection performance using FL to the traditional deep learning approach and noticed a **significant difference between** the two models.
- We further analyzed the latency in an FL and non-FL system. While training the 7K images of the KITTI dataset for one epoch on a **single client took 27 minutes**, the FL approach with data distributed among **four clients took only 10 minutes**. It is important to note that latency in the above context is the time taken during the training phase of object detection. **Network latency is considered to be negligible** for this setup.
- They deduced that the **time taken** is proportional to the **data size**.
- At last, they said that **FL and DL produce comparable results**.

Possible Future Direction

- Currently Trying to reproduce their results. (Code Improvement)
- I might be changing The averaging algorithm used to batter one.
- I might change YOLOv3 to other advance versions of YOLO.
- If time permits, I would like to test this on a different data set (Like one with adverse weather conditions suggested by TA).

References

- D. Jallepalli, N. C. Ravikumar, P. V. Badarinath, S. Uchil and M. A. Suresh, "Federated Learning for Object Detection in Autonomous Vehicles," 2021 IEEE Seventh International Conference on Big Data Computing Service and Applications (BigDataService), Oxford, United Kingdom, 2021, pp. 107-114, doi: 10.1109/BigDataService52369.2021.00018.
- C. Bloom, J. Tan, J. Ramjohn, and L. Bauer, "Self-driving cars and data collection: Privacy perceptions of networked autonomous vehicles," in Thirteenth Symposium on Usable Privacy and Security (SOUPS 2017). USENIX Association, Jul. 2017.
- L. Chen, B. Li, and L. Qi, "Improved yolov3 algorithm for ship target detection," in 2020 39th Chinese Control Conference (CCC), 2020, pp. 7288–7293.
- Communication-Efficient Learning of Deep Networks from Decentralized Data
<https://arxiv.org/pdf/1602.05629.pdf>
- <https://medium.com/@ys2223/a-study-of-federated-learning-in-autonomous-vehicle-system-ca9be70291fc>
- <https://www.bitfount.com/pets-explained/federated-machine-learning>



भारतीय प्रौद्योगिकी संस्थान हैदराबाद
Indian Institute of Technology Hyderabad

Thank you