Object Detection For Autonomous Vehicles

Team Members :

Abhijit rao.S SE21UARI003 N.Gagan SE21UARI094

Varshini.V SE21UARI183 Sri Ashritha SE21UARI153

Srinivas Gurram SE21UARI190

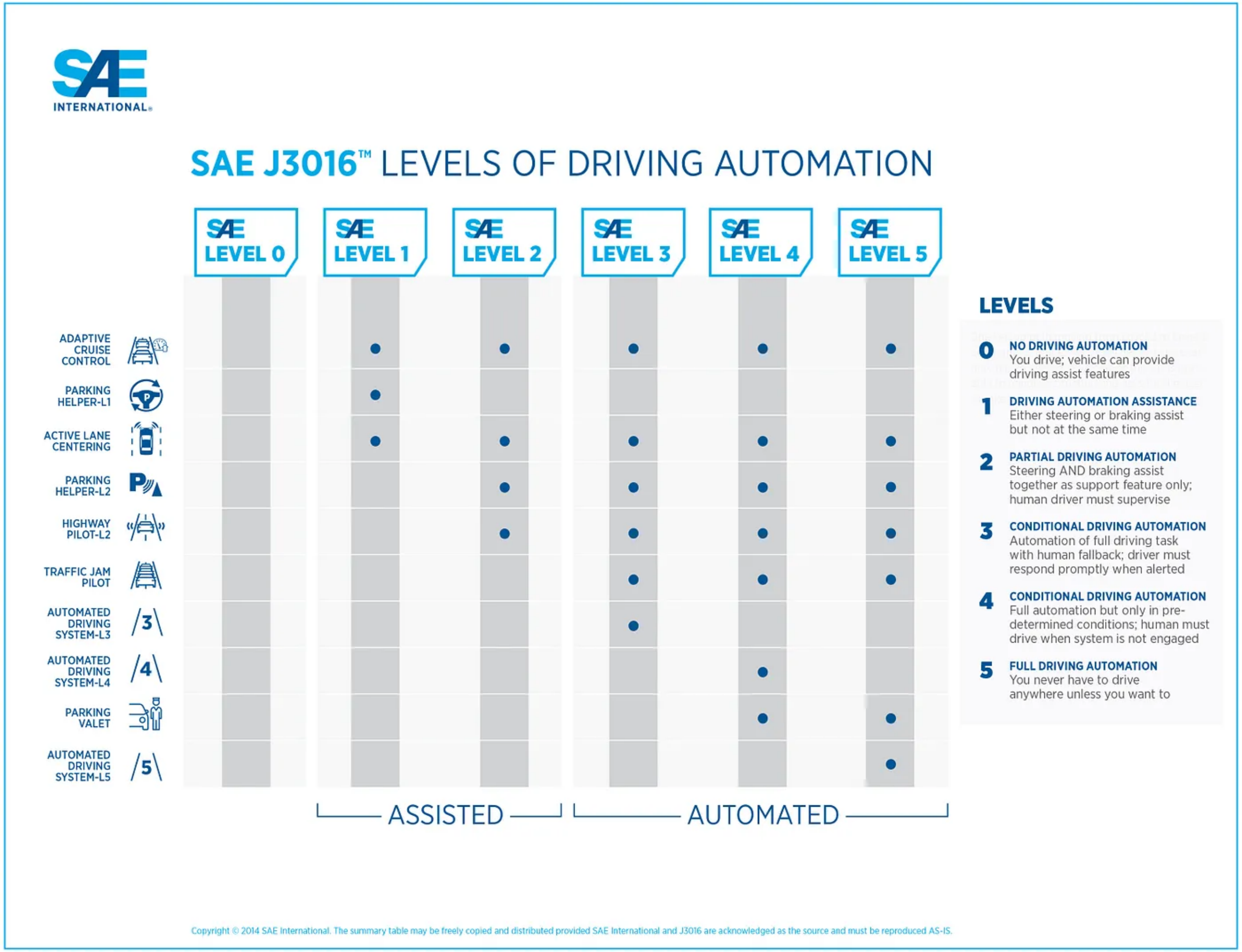
Project Description

Detecting small objects like traffic signs and lights is more difficult due to their complex design. Also, issues like uneven backgrounds and visual distortion from poor weather and lighting conditions make it difficult to identify small objects accurately. The project aims to develop an efficient algorithm for object detection in autonomous vehicles using the YOLO and CARLA.

SAE J3016 levels of automation

SAE Level	Name	Driving Environment Monitor
0	No Automation	Human Driver
1	Driver Assistance	
2	Partial Driving Automation	
3	Conditional Driving Automation	ADAS System
4	High Driving Automation	
5	Full Driving Automation	

Autonomous vehicles are categorized into six levels



SAE INTERNATIONAL

SAE J3016™ LEVELS OF DRIVING AUTOMATION

	SAE LEVEL 0	SAE LEVEL 1	SAE LEVEL 2	SAE LEVEL 3	SAE LEVEL 4	SAE LEVEL 5
ADAPTIVE CRUISE CONTROL		●	●	●	●	●
PARKING HELPER-L1		●				
ACTIVE LANE CENTERING		●	●	●	●	●
PARKING HELPER-L2			●	●	●	●
HIGHWAY PILOT-L2			●	●	●	●
TRAFFIC JAM PILOT				●	●	●
AUTOMATED DRIVING SYSTEM-L3				●		
AUTOMATED DRIVING SYSTEM-L4					●	
PARKING VALET					●	●
AUTOMATED DRIVING SYSTEM-L5						●

ASSISTED (Levels 1–2) · AUTOMATED (Levels 3–5)

LEVELS

0 NO DRIVING AUTOMATION
You drive; vehicle can provide driving assist features

1 DRIVING AUTOMATION ASSISTANCE
Either steering or braking assist but not at the same time

2 PARTIAL DRIVING AUTOMATION
Steering AND braking assist together as support feature only; human driver must supervise

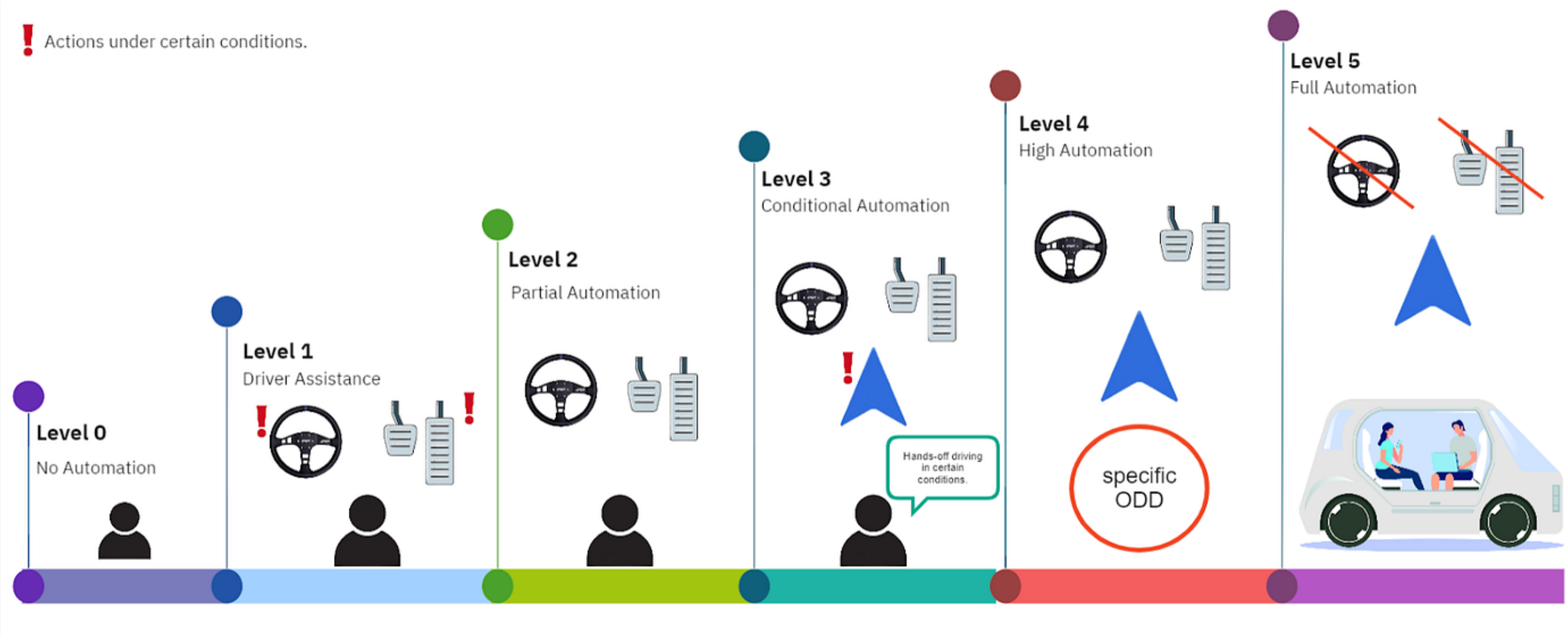
3 CONDITIONAL DRIVING AUTOMATION
Automation of full driving task with human fallback; driver must respond promptly when alerted

4 CONDITIONAL DRIVING AUTOMATION
Full automation but only in pre-determined conditions; human must drive when system is not engaged

5 FULL DRIVING AUTOMATION
You never have to drive anywhere unless you want to

https://medium.com/@overthevehicle/autonomous-vehicle-a-little-deep-dive-about-level-3-73a58393c43c

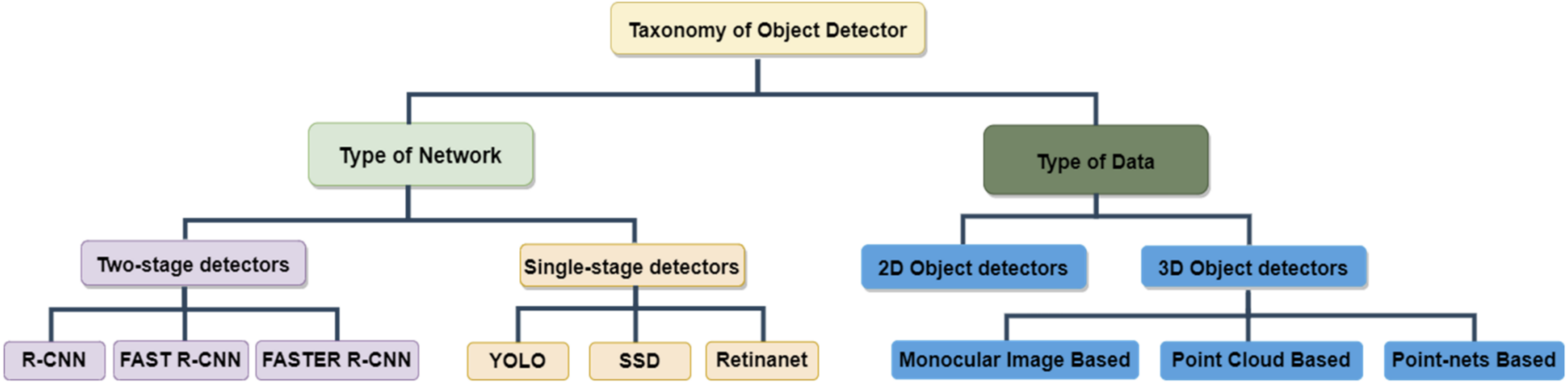
Levels Of Automation



What is object detection?

Object detection is a computer vision technique that uses neural networks to identify and locate objects in images, videos, and live footage.

Classification of Object Detectors



Object detection consists of two tasks

Localization

Classification

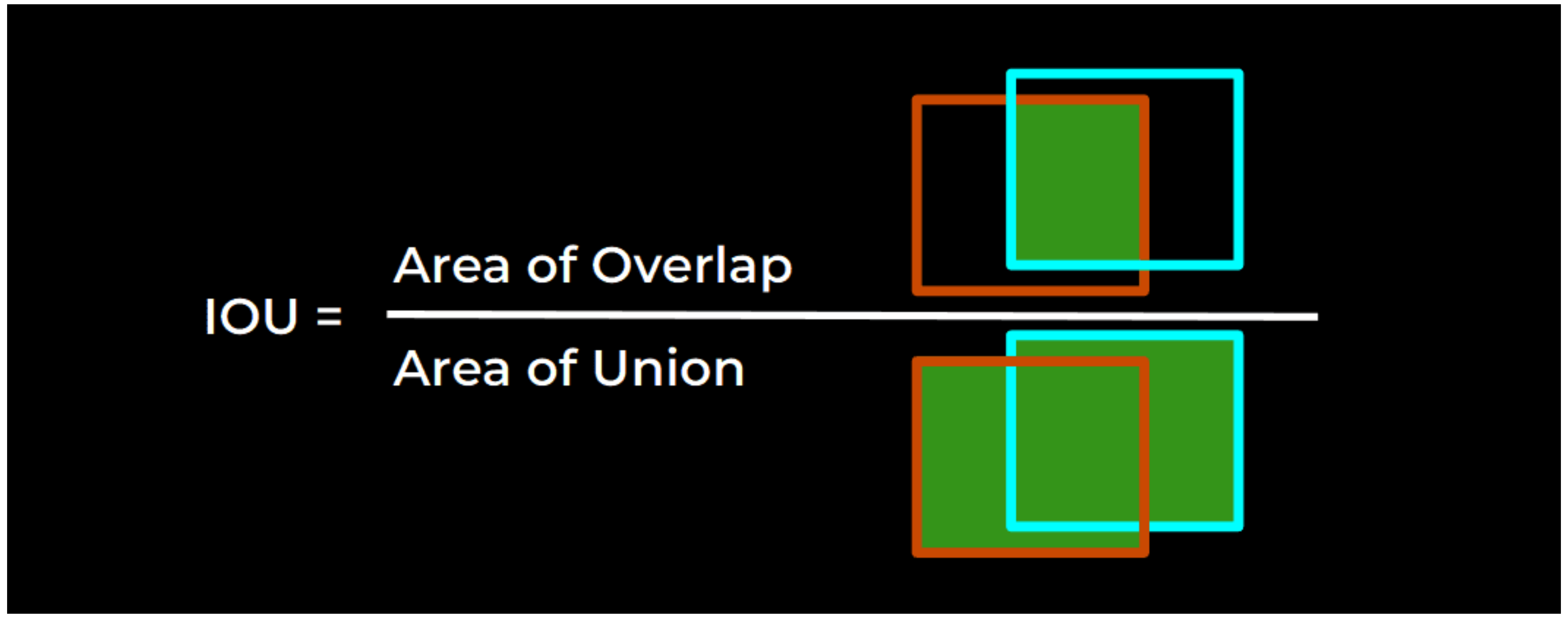
Two stage object detectors consist of

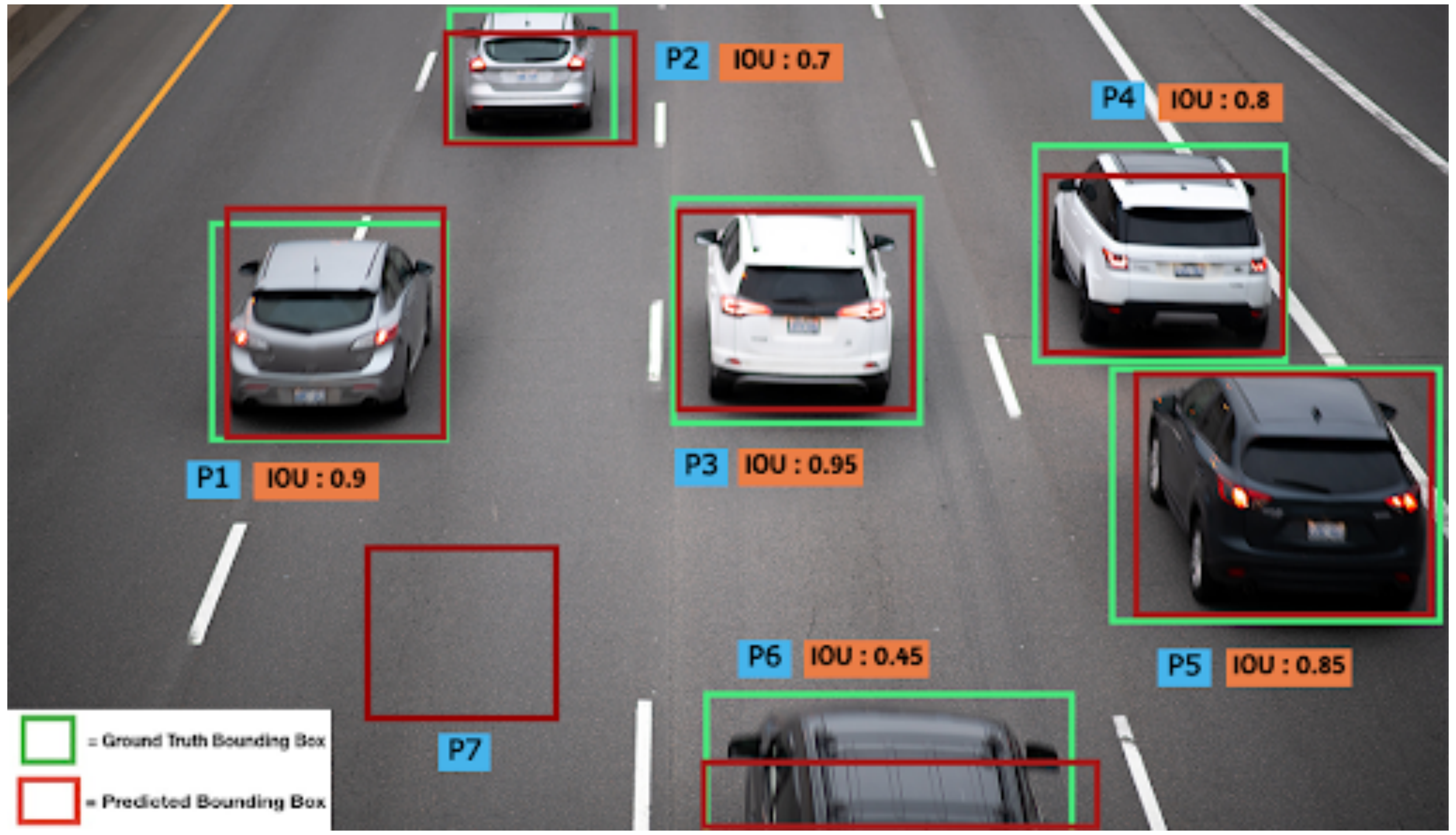
Proposal of ROI

Classification of ROI

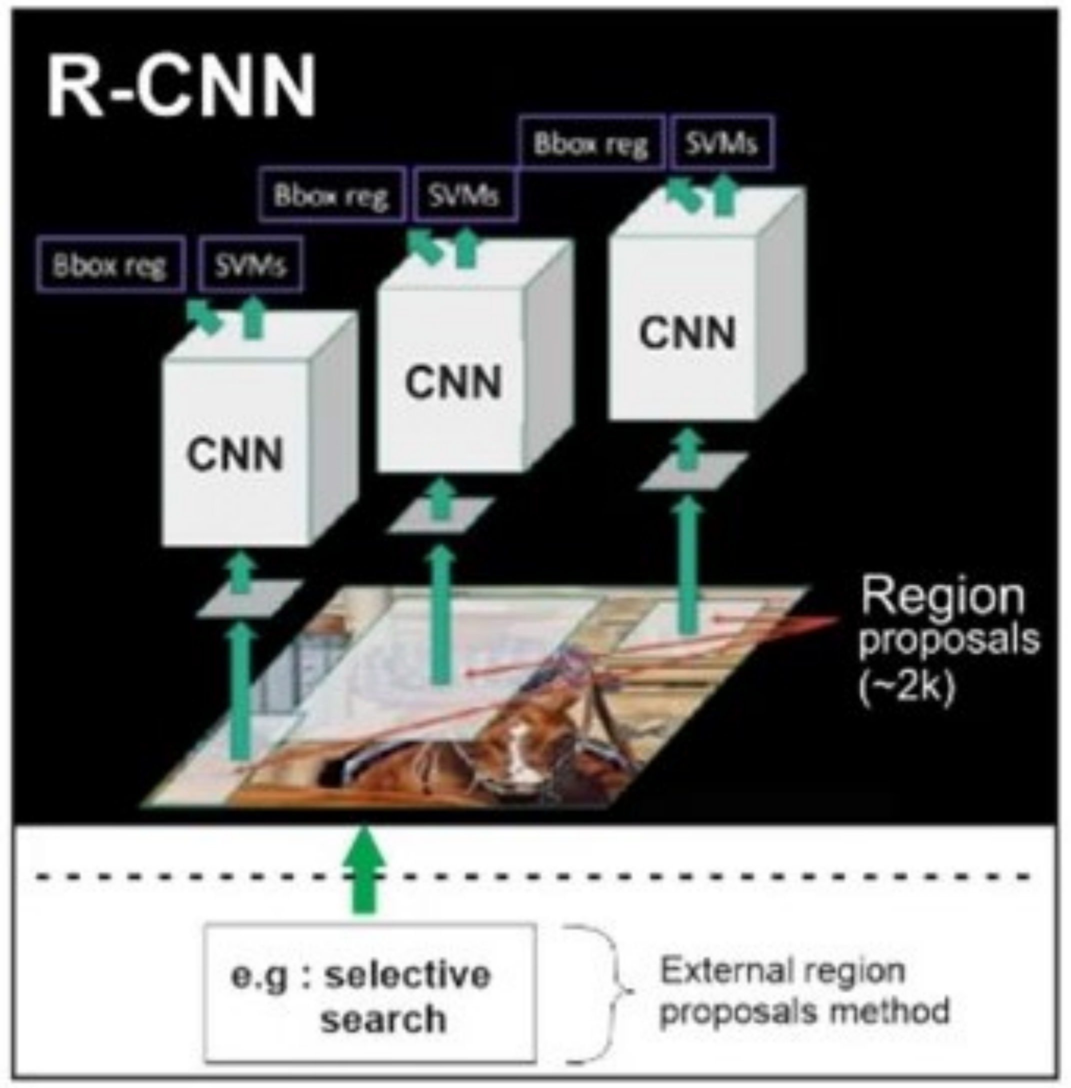
Single stage object detectors use a single feed forward neural network that creates bounding boxes and classification in the same stage.

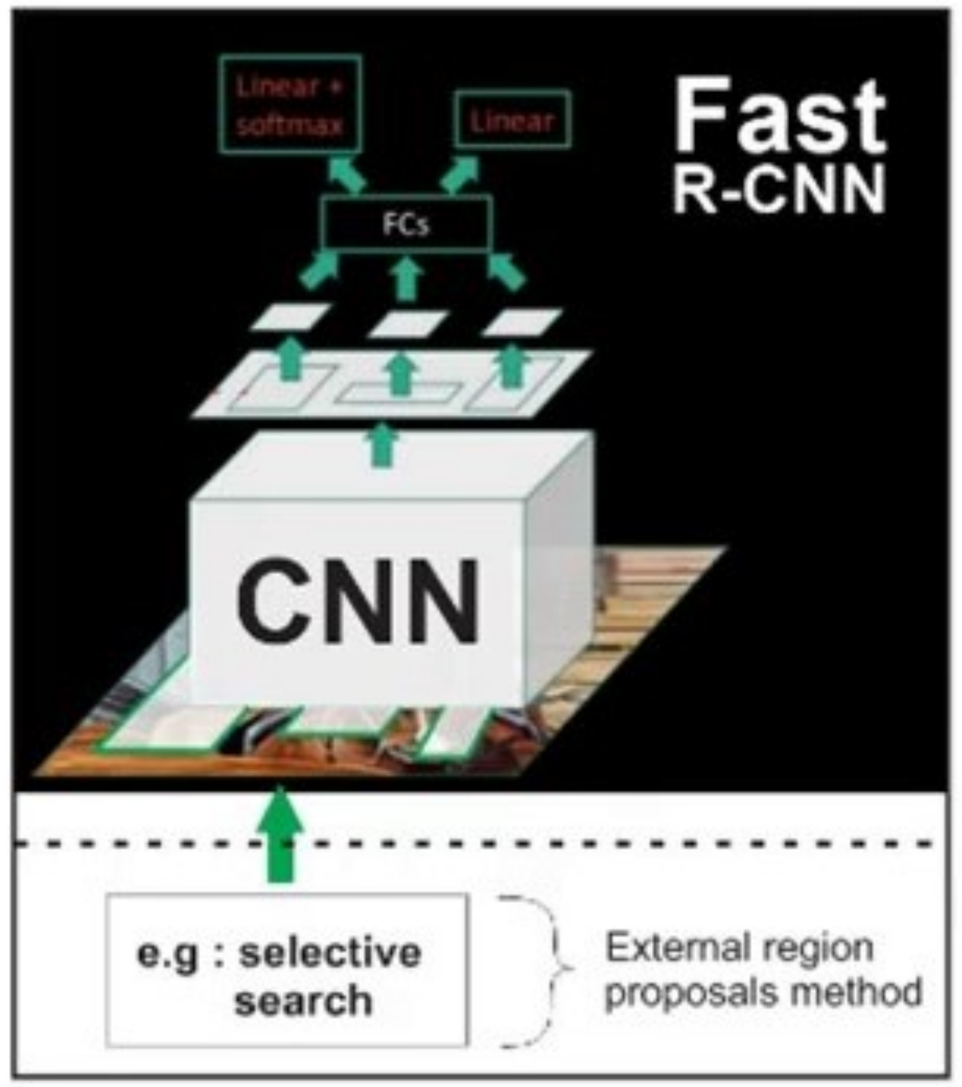
We use mAP(Mean Average Precision) and IoU(Intersection Over Union) to determine the performance of an Object Detector

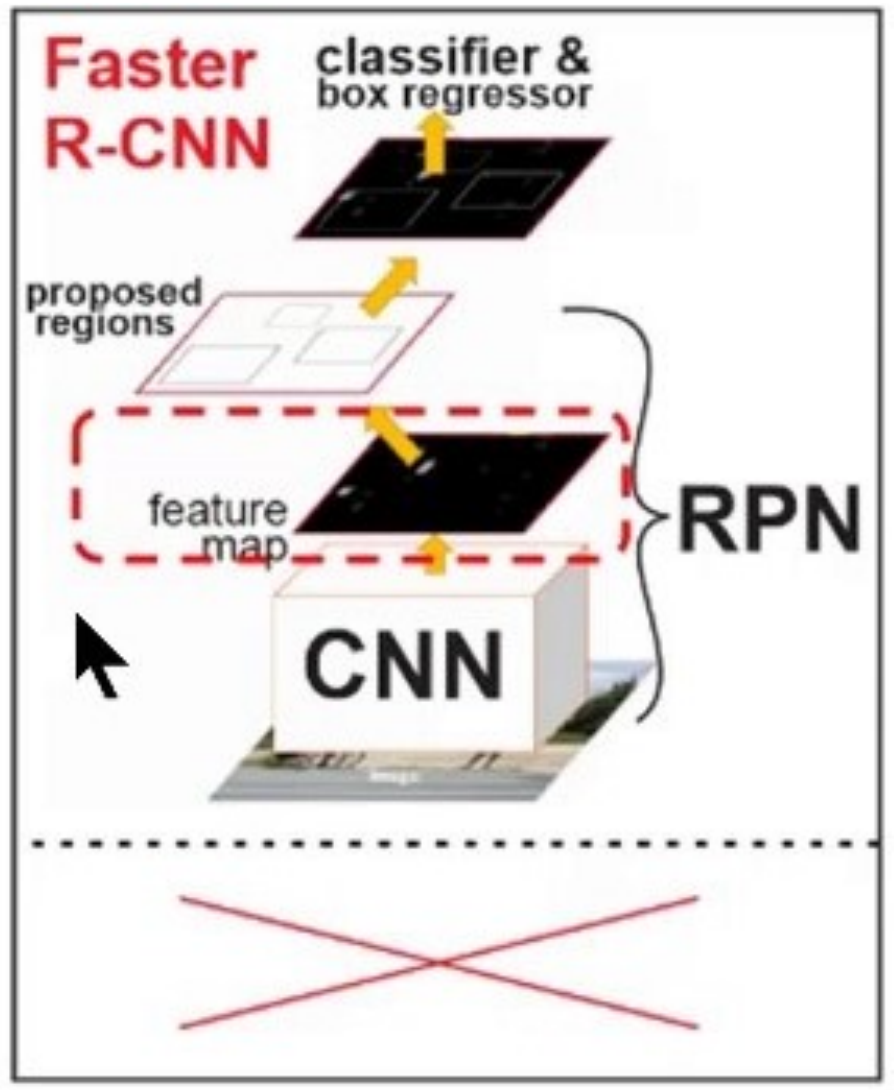




Differences between R-CNN, Fast R-CNN and Faster R-CNN





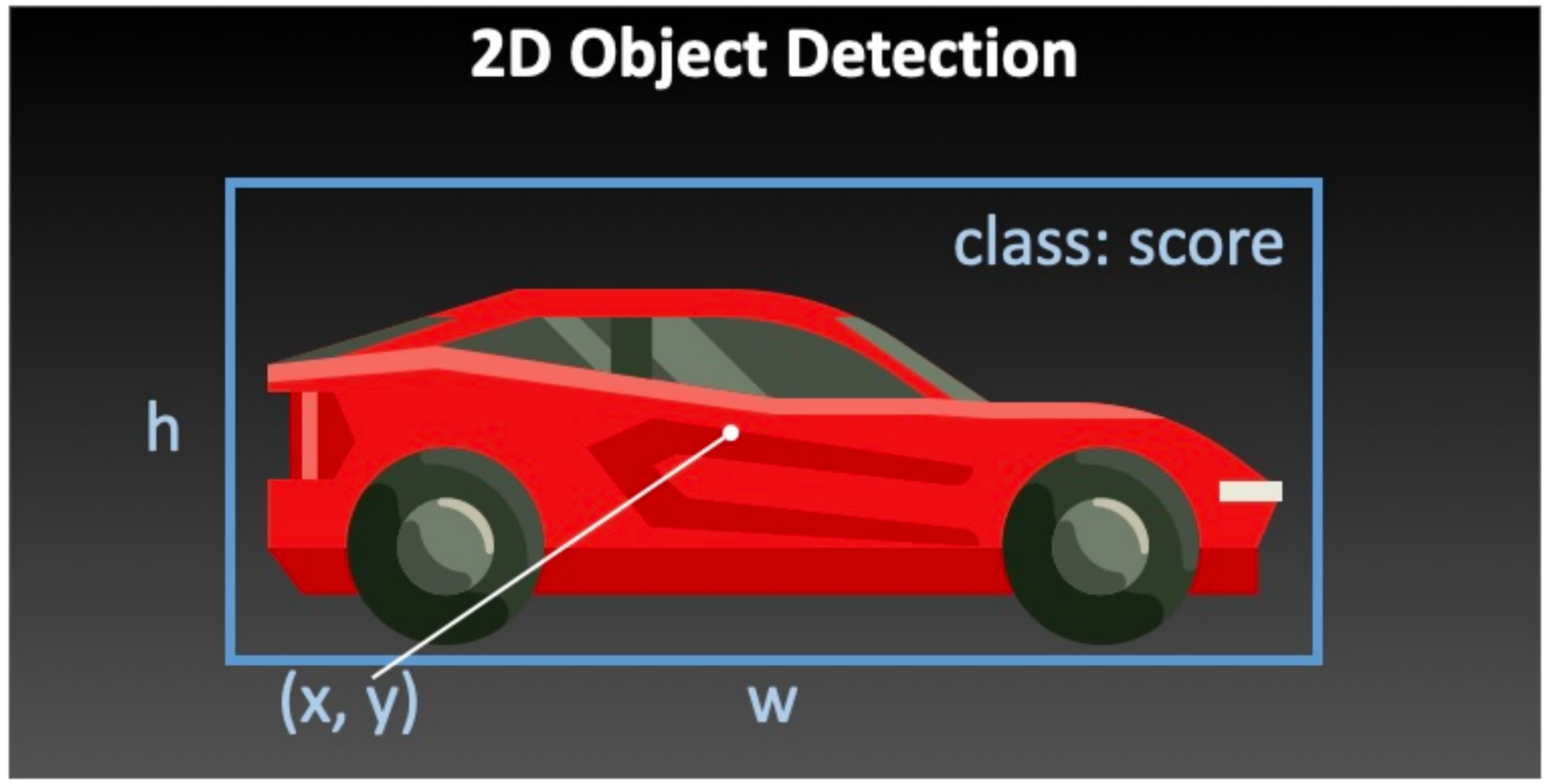


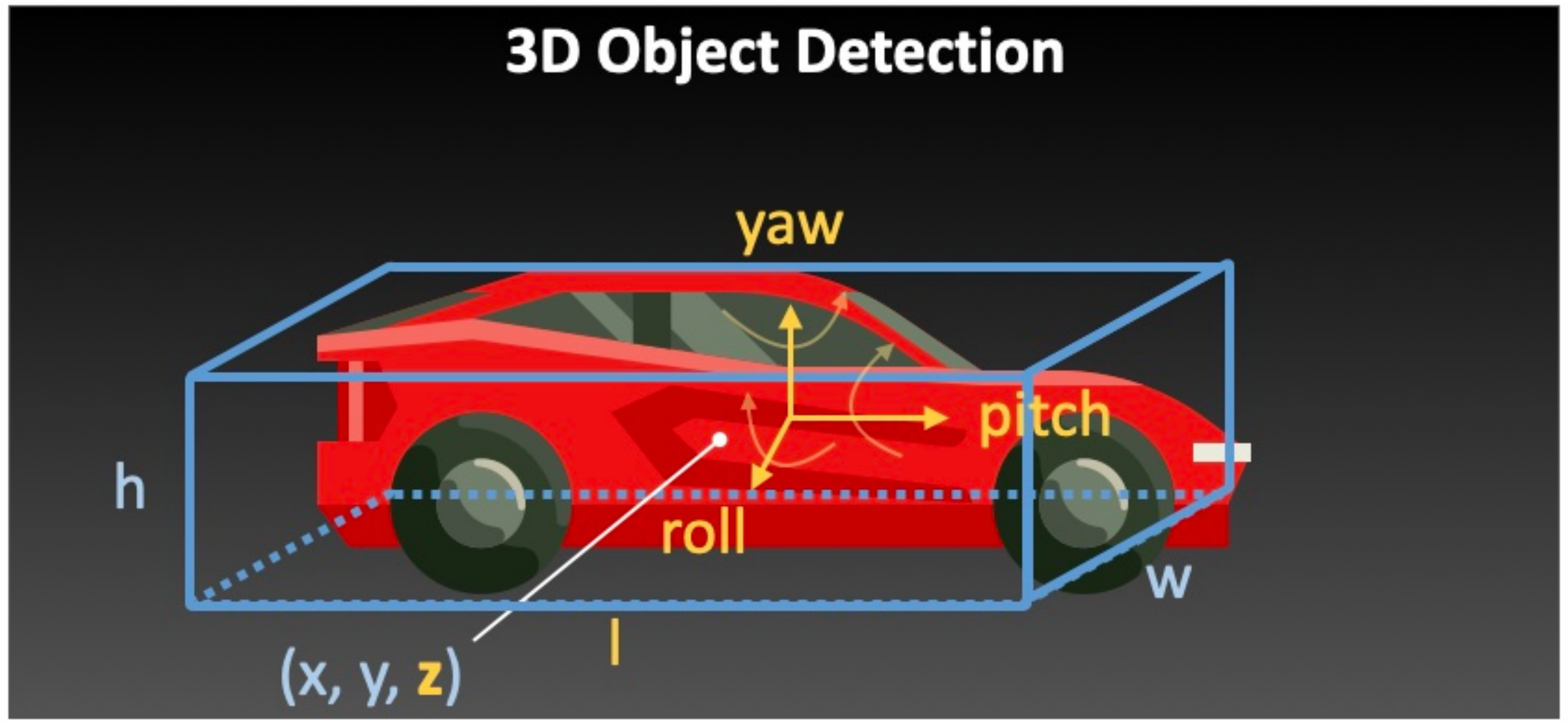
	R-CNN	Fast R-CNN	Faster R-CNN
Test time per image	50 seconds	2 seconds	0.2 seconds
Speed-up	1x	25x	250x
mAP (VOC 2007)	66.0%	66.9%	66.9%

2D and 3D Object Detection

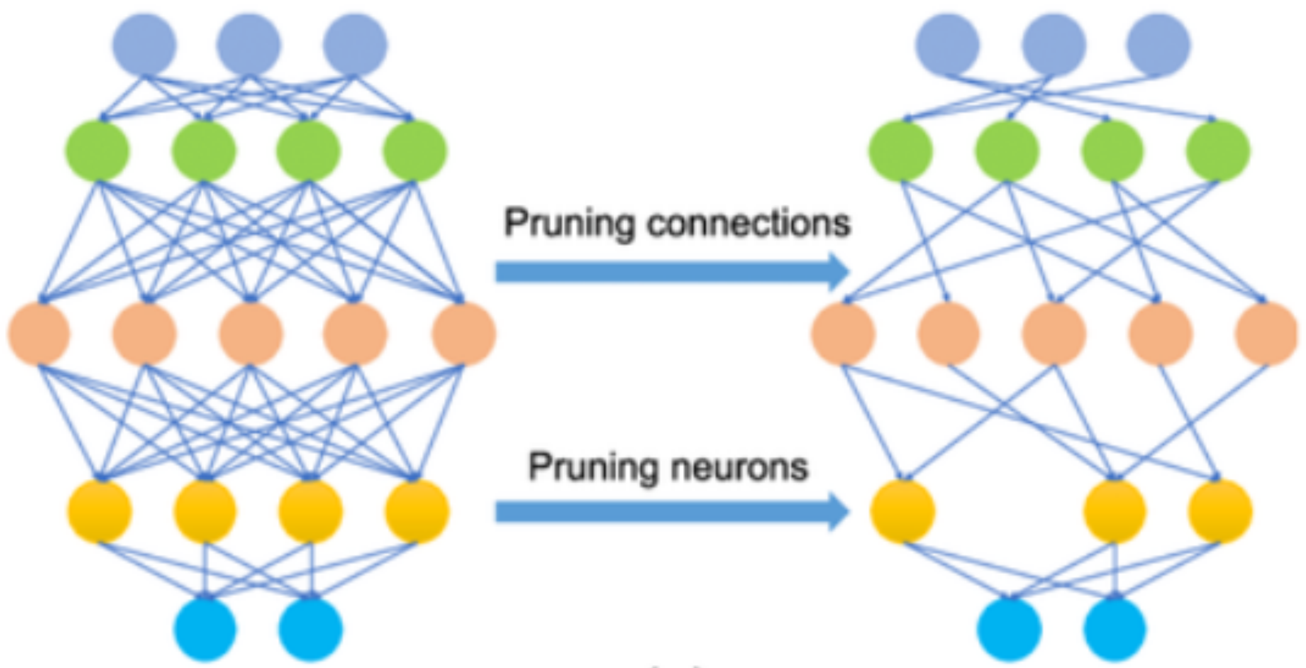
We reduce the complexity of object detection in 3D by using pointnets

Pointnets are used with RGB images, where 2D bounding boxes are obtained using RGB images. Then these boxes are used as ROIs for 3D object detection which reduces the search effort for 3D detection.

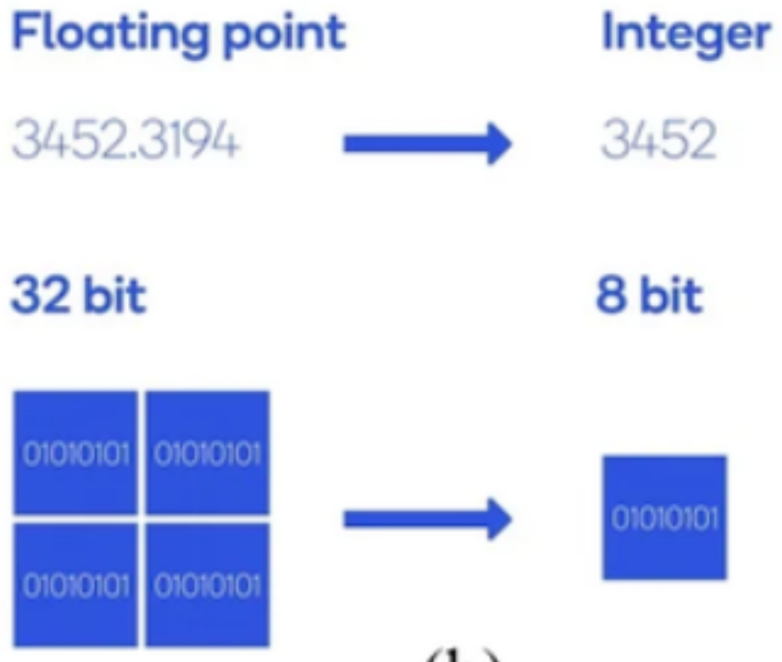




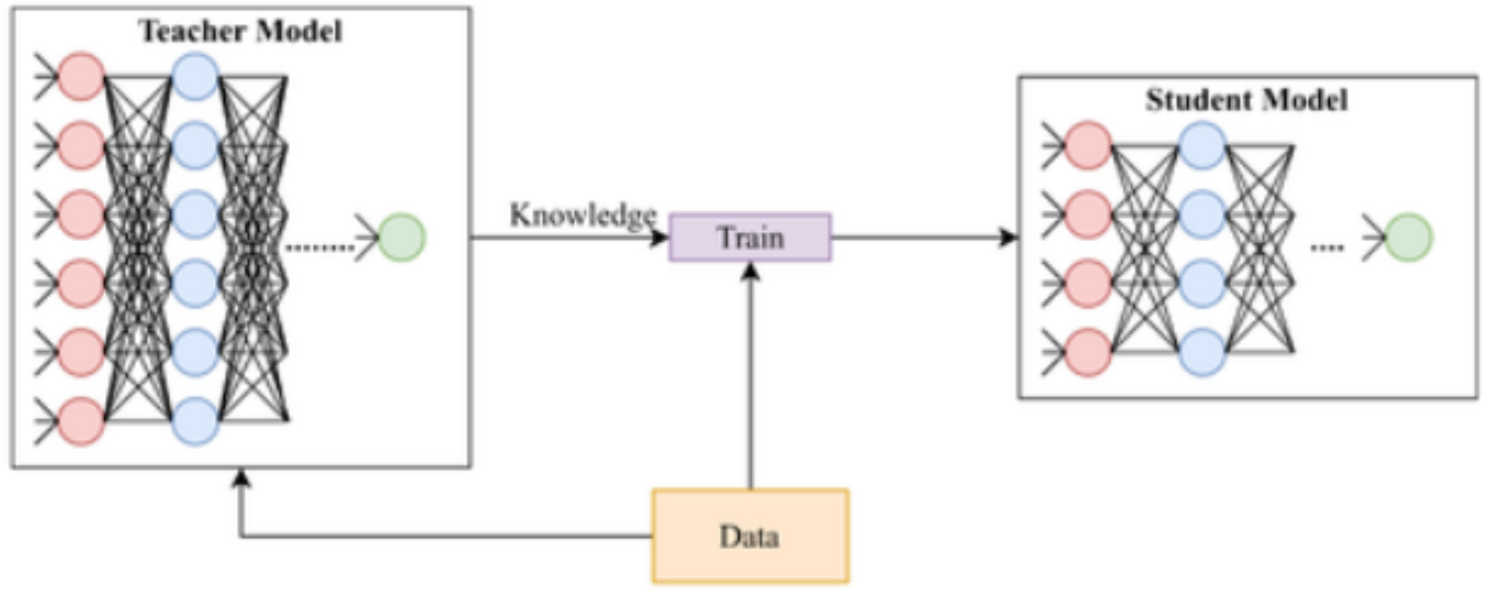
Techniques to reduce the complexity of models for AV



(a)



(b)



(c)

(a) Pruning involves removing redundant weights and creating sparsity in the model by training the model with various regularization techniques

(b) Quantizing involves converting the baseline 32-bit parameters to fewer (e.g., 16 or 8) bits, to achieve lower memory footprint, without significantly reducing model accuracy.

(c) Knowledge Distillation involves transferring learned knowledge from a larger model to a smaller, more compact model.

Other techniques for improving the object detection performance

1. Neural Architecture Search (NAS) - Automatically determines the best structure for OD.
2. Real time processing – Correlations between consecutive frames in a video can increase the efficiency of the object detection in videos.
3. Sensor fusion – Use more than one sensor and combine all the data such that the model doesn't fail any condition in AVs.
4. Time series – Use data from previous and current time instances at specific intervals.
5. Semi supervised OD – Use supervised machine learning to detect objects.
6. Open datasets – Use many datasets to train the model.
7. Use techniques like Pruning, Quantization and Knowledge Distillation to reduce complexity of the model.

Small-object detection based on YOLOv5 in autonomous driving systems

ABSTRACT

Imagine a self-driving car navigating a busy intersection. It flawlessly detects large objects like cars and trucks. But what about the small, yet crucial details: a faded stop sign partially hidden by a tree branch, or a distant yellow school bus light signaling caution? Here's why improved small object detection is critical for autonomous vehicles:

Traffic Signs and Lights: The Unsung Heroes: These small objects hold immense power. A missed stop sign or a misidentified traffic light can lead to disastrous consequences. By accurately detecting these objects, autonomous vehicles can ensure they obey traffic rules and navigate intersections safely.

Understanding the Bigger Picture: Small objects like pedestrians, cyclists, and even animals on the road edges often appear as tiny pixels in camera images. But they are vital components of the driving scene. Improved detection allows the car to understand the complete picture and make informed decisions, like slowing down for a crossing pedestrian or safely maneuvering around a cyclist.

Adverse Conditions: Nature Throws Curveballs:** Rain, fog, or low-light can significantly reduce the detail in camera images, making small objects even harder to see for standard object detection systems. Improved detection algorithms can compensate for these challenges and ensure the car's "vision" remains sharp even in bad weather.

The proposed iS-YOLOv5 is like giving self-driving cars a vision upgrade. It allows them to not only see large objects clearly but also pay attention to the finer details on the road. This enhanced perception translates to safer navigation, improved decision-making, and ultimately, a crucial step towards reliable autonomous driving.

1)INTRODUCTION

Challenges in Small Object Detection for Autonomous Driving

This work investigates challenges in object detection for autonomous vehicles, particularly focusing on small objects like traffic signs and lights.

Current Issues:

- Existing detectors struggle with small objects due to limited resolution, background complexity, and insufficient contextual information.
- Balancing detection accuracy for small and large objects remains a challenge.
- Real-time applications often prioritize speed over accuracy, sacrificing small object detection performance.

Proposed Approach:

This paper proposes an improved YOLOv5 architecture (iS-YOLOv5) specifically designed for small object detection in autonomous driving scenarios.

Key Benefits:

- Improved small object detection accuracy without compromising large object detection or computational cost.
- Maintains real-time processing speed with a slight increase in complexity.

Evaluation and Results:

- The iS-YOLOv5 model achieves significant improvements in small object detection accuracy on the BDD100K dataset.
- The model demonstrates effectiveness in various weather conditions, highlighting its robustness.
- Empirical results on additional datasets (TT100K and DTLD) validate the model's performance for traffic sign and light detection.

RELATED WORK

Related Work in Small Object Detection for Autonomous Driving

YOLO for Real-Time Object Detection:

- YOLO models (v1-v4) have been widely used for real-time object detection in autonomous driving, with a focus on speed over accuracy.

Architectural Modifications for YOLO:

- Some studies explore modifying YOLOv4 for better accuracy in limited scenarios, but these methods increase complexity and inference time.
- YOLOv5 offers a balance of performance and accessibility for real-world applications.

YOLO for Small Object Detection:

- Existing research utilizes YOLOv3-v5 for small object detection (traffic signs, lights, cones) in autonomous driving.
- Optimizations often involve minor adjustments to the original YOLO structure or adding external modules, impacting inference speed or resource usage.

Limitations of Current Approaches:

- Most YOLOv5-based systems lack evaluation of computational cost, crucial for autonomous driving.
- Reported accuracy improvements for small objects rely on techniques like normalization, not architectural changes.

Our Contribution:

This work proposes architectural modifications to YOLOv5 (iS-YOLOv5) for enhanced small object detection in autonomous driving, focusing on minimal cost and avoiding additional techniques. We aim to improve detection performance, especially in challenging weather conditions.

METHODOLOGY

Methodology

This section details our approach to improve small object detection in autonomous driving using YOLOv5.

Motivation:

- Existing methods for small object detection in autonomous driving often rely on normalization, additional modules, or increased model complexity.
- Architectural modifications have shown promise for improving detection performance.
- Accurate small object detection is crucial for autonomous vehicles to make informed decisions in complex environments.

Challenges:

- Small object detection is difficult due to limited visual cues, foreground-background imbalance, and perspective distortion.
- Balancing detection speed and accuracy is essential for real-time applications.

Our Approach:

We propose architectural modifications to the YOLOv5 model (iS-YOLOv5) specifically designed to enhance small object detection in autonomous driving. This approach aims to:

- Improve small object detection accuracy
- Maintain fast inference speed
- Minimize increase in model complexity

Our Contribution:

To our knowledge, this work is the first to propose architectural changes to YOLOv5 for autonomous driving that improve small object detection accuracy without sacrificing large object detection or significantly impacting inference speed or model complexity.

EXPLAINING YOLO

What is YOLO?

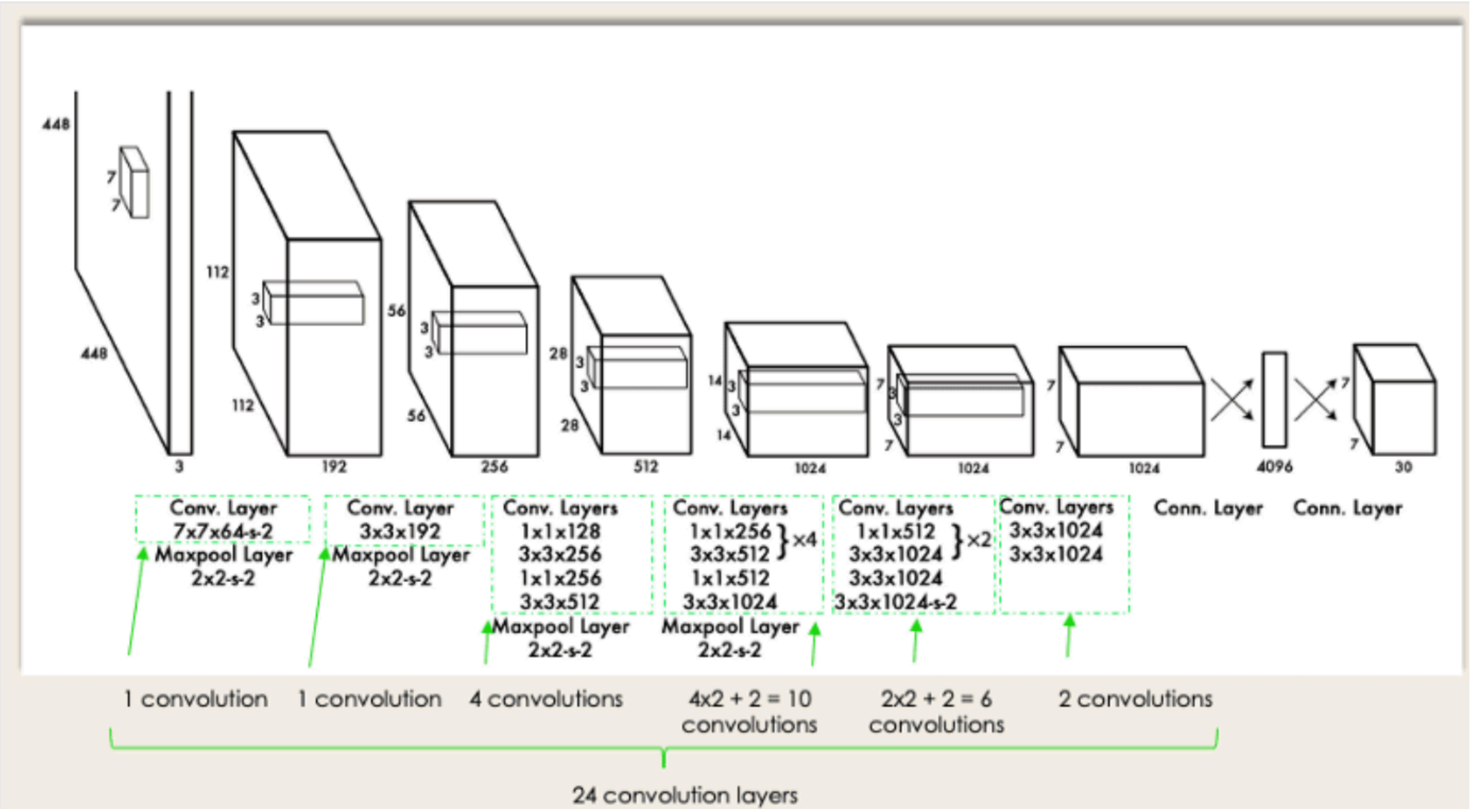
You Only Look Once (YOLO) is a state-of-the-art, real-time object detection algorithm introduced in 2015 by **Joseph Redmon**, **Santosh Divvala**, **Ross Girshick**, and **Ali Farhadi** in their famous research paper "**You Only Look Once: Unified, Real-Time Object Detection**".

What Makes YOLO Popular for Object Detection?

Some of the reasons why YOLO is leading the competition include its:

- Speed
- Detection accuracy
- Good generalization
- Open-source

YOLO ARCHITECTURE



YOLO ARCHITECTURE

The standard YOLOv5 architecture prioritizes generic object detection tasks, and its information flow within the network isn't optimized for extracting features from small objects. The study identifies limitations in three key areas:

1. **PANet Structure:** The current PANet structure struggles to exchange information between non-adjacent layers, leading to a loss of detail during the feature aggregation process. This is detrimental for small object detection where preserving fine-grained information is critical.
2. **SPP Module:** The max-pooling operations within the SPP module discard too much spatial information, hindering the model's ability to precisely locate small objects.
3. **Information Paths:** The current routing of information within YOLOv5 isn't ideal for extracting features relevant to small objects. The convolutional layers that capture these features are not emphasized enough in the original architecture.

To address these shortcomings, the study proposes several modifications:

- An improved SPP module that utilizes dilated convolution instead of max-pooling. Dilated convolution preserves spatial resolution while expanding the receptive field, allowing the model to capture multi-scale information necessary for small object detection.
- An enhanced PANet structure with cross-layer connections. These connections facilitate information exchange between non-adjacent layers, ensuring better utilization of semantic information and shallow location details crucial for small object recognition.
- Optimized information paths within the network. This involves introducing a new functional block (N-CSP) to streamline feature extraction for small objects, along with strategic use of the Hard Swish activation function for efficient information processing. Additionally, a dedicated detection head is incorporated to specifically handle small objects identified in high-resolution feature maps.

These modifications, collectively called iS-YOLOv5 (Improved Scaled YOLOv5), aim to significantly improve small object detection accuracy without sacrificing the real-time processing capabilities essential for autonomous driving applications. By addressing the limitations of the original YOLOv5 architecture, iS-YOLOv5 offers a promising solution for object detection tasks in self-driving vehicles.

YOLOv5 is a popular object detection model known for its balance between speed and accuracy. It achieves this through a multi-stage architecture. The first stage, the backbone, extracts features from the input image. Here, the specific convolutional layers used can vary depending on the chosen YOLOv5 variant. Common building blocks include Conv Blocks for basic feature extraction and BottleneckCSP blocks for improved efficiency.

Following the backbone is the SPP module, which is crucial for small object detection. The standard SPP module uses max-pooling to capture features from various image regions at different scales. However, this can discard some spatial details, making it difficult to precisely locate small objects. The research paper you referred to addresses this by proposing an improved SPP module that utilizes dilated convolution instead. Dilated convolution expands the receptive field of the filters without losing resolution, allowing the model to capture multi-scale features from a larger area while preserving crucial spatial information. This improvement is particularly important for tasks like self-driving cars, where accurate detection of small objects like pedestrians and traffic signs is essential.

After the SPP module, the information is processed by the neck network, which combines features from different stages of the backbone. In YOLOv5, this typically involves a combination of Path Aggregation Network (PANet) and Feature Pyramid Network (FPN) architectures. Finally, the head network takes the fused features and generates the final detections, including bounding boxes and class probabilities for objects within the image. Overall, by understanding the components of YOLOv5 architecture, particularly the improvements made to the SPP module for small object detection, you can effectively utilize this model for various object detection tasks.

EXPERIMENTAL RESULTS

This section talks about how researchers improved YOLOv5 for self-driving cars. They tested their method on datasets with images of traffic lights and signs. Here's a breakdown:

Datasets:

- They mainly used BDD100K, a dataset with 100,000 images from various environments.
- They also used TT100K (traffic signs) and DTLD (traffic lights) for fine-tuning.
- In BDD100K, they classified objects based on size (small, medium, large) depending on the number of pixels they occupied in the image.

Data Augmentation:

- This is a technique to artificially increase the amount of data the model is trained on.
- They used various methods like image shifting, scaling, flipping, blurring, and adding noise to the images.
- This helps the model learn the characteristics of objects in different conditions.

Training Setup:

- They trained the model on a computer with an Intel i9 CPU and an Nvidia RTX 5000 GPU.
- They used PyTorch software and an optimizer called SGD to adjust the model parameters during training.
- They set specific values for learning rate, batch size, and other hyperparameters.

Evaluation Metrics:

- They mainly used mean Average Precision (mAP) to measure how well the model detects objects.
- mAP considers both precision (correct detections) and recall (detecting all objects).
- They also tracked how many calculations the model needed (FLOPs) and how long it took to process an image (inference time).

Effectiveness of the Proposed Model (iS-YOLOv5):

- Their improved model, iS-YOLOv5, achieved better mAP scores on the traffic sign and light datasets compared to the original YOLOv5.
- It also improved accuracy for small objects (traffic signs and lights) in the BDD100K dataset.
- Notably, the accuracy improvement for small objects came without sacrificing accuracy for larger objects.
- They visualized the training process and the model's performance on precision-recall curves.
- The model showed strong performance even in challenging weather conditions like fog and rain.
- They showed examples where iS-YOLOv5 could detect traffic signs and lights in heavy traffic scenes where regular YOLOv5 struggled.

Importance of Different Parts of the Model (Ablation Study):

- They tested the contribution of different components in their model by removing them one at a time and measuring the effect.
- Each component improved accuracy or speed to some degree.
- This shows that all the parts of their model worked together effectively.

Comparison with Other Detectors:

- Their iS-YOLOv5 outperformed other object detection models in terms of accuracy, speed, and computational cost.
- This makes it well-suited for real-time object detection tasks in self-driving cars.

In summary, the researchers improved YOLOv5 for autonomous driving by focusing on small object detection and using various techniques during training and evaluation. Their iS-YOLOv5 model showed significant improvements and is a promising approach for self-driving car applications.

CONCLUSION

Conclusion: Our Improved YOLOv5 for Self-Driving Cars

In our project, we aimed to improve YOLOv5, a popular object detection model, for autonomous vehicles. We specifically focused on enhancing the detection of small objects like traffic signs and lights, without compromising the accuracy for larger objects.

Here's a breakdown of what we achieved:

- **Optimized Information Flow:** We modified the way information travels within the YOLOv5 architecture. This helped the model better understand and process features of small objects.
- **iS-YOLOv5 Model:** We introduced our improved model, iS-YOLOv5. It boasts better detection accuracy and speed compared to regular YOLOv5, and all this without significantly increasing the model's complexity.
- **Extensive Testing:** We thoroughly tested iS-YOLOv5 on challenging datasets containing images from real-world driving scenarios.
- **Generalizability in Bad Weather:** We also evaluated how well iS-YOLOv5 performs in harsh weather conditions like rain and fog. The results were promising!

Impact:

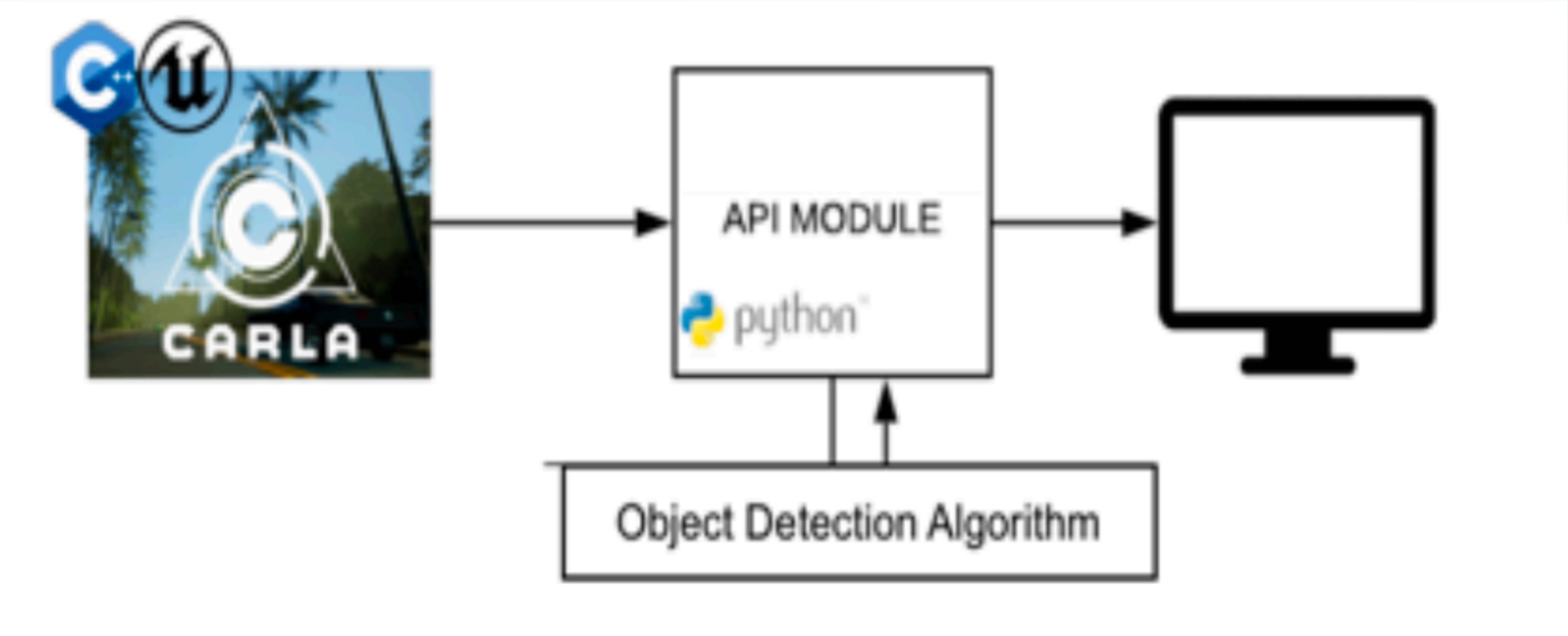
Our work can significantly improve the perception capabilities of self-driving cars. By accurately detecting small objects like traffic signs and lights, even in difficult situations, iS-YOLOv5 can help autonomous vehicles make better decisions and navigate roads more safely. This paves the way for more robust and reliable self-driving technology.

Future Work:

- We can further explore different network architectures to see if even better detection accuracy can be achieved for small objects.
- We can test iS-YOLOv5 on a wider range of datasets and real-world scenarios to ensure its generalizability across diverse situations.
- We can investigate how to integrate iS-YOLOv5 with other components of a self-driving car's decision-making system for a more holistic approach.

CARLA Simulator

- CARLA simulator (CAR Learning to Act) help us train and test models and to gain insights into autonomous driving with ease.



- The choice of the algorithm to be used is done by a tradeoff between Mean Average Precision (mAP) and the speed of the algorithm

.

CARLA Simulator

- CARLA (Car Learning to Act) simulator is an open-source simulator built for autonomous driving research. It consists of two main modules:
  A. CARLA simulator
  B. CARLA Python API
- It was designed with the purpose of democratizing autonomous driving research by providing a platform which models real world driving scenarios and use cases.
- It is built on Unreal Engine 4 and allows users to control the simulations through the Python API.

Features of CARLA simulator:

1. Features of CARLA include integration of various sensor packages for autonomous driving including multi-camera, LIDAR and GPS.
2. It also has a flexible API to control simulator.aIt scenario Runner allows the user simulate different traffic situation based on modular behaviour.
3. Fast execution via fast simulation for planning and control, and users can generate the custom maps from tools like Road runner.

DESIGN AND IMPLEMENTATION

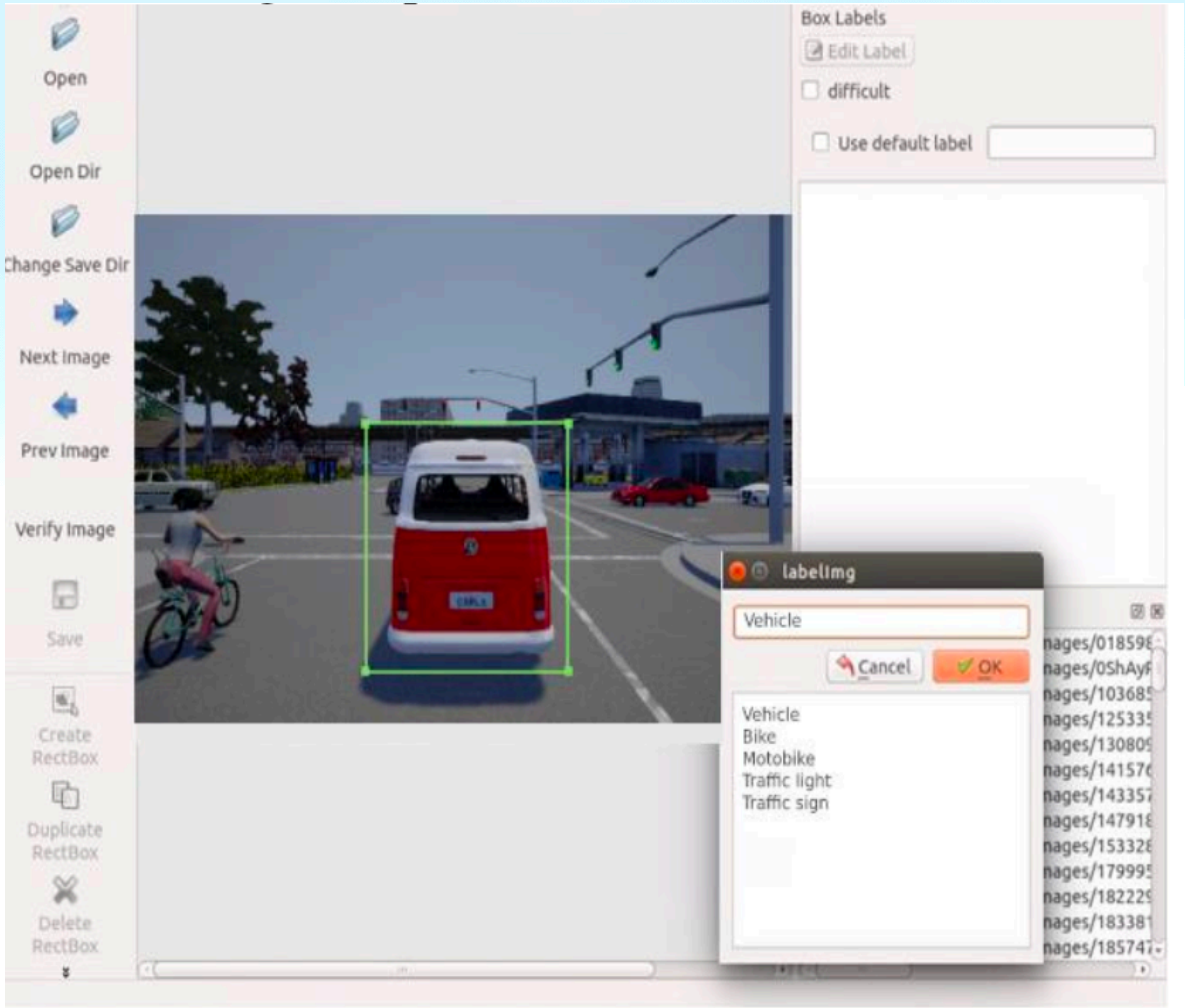
- The dataset was created from different scenarios on the CARLA simulator and Five scenarios from different town and weather settings were considered. The images were later labelled for training purposes. The classes of objects considered were: Vehicles (cars and trucks), Bikes, Motorbikes, Traffic Lights and Traffic signs.

- The dataset was created by capturing images from CARLA simulator at certain intervals. A total of 1028 images of 640x380 pixels were taken.
- The images were taken from 5 different town settings in CARLA simulator with different kinds of weather conditions (by altering percentage of cloudiness, precipitation and sun altitude angle). Different traffic conditions are also present in each image.
- The 1028 images were split into the test and train dataset with 208 test and 820 training(20-80 split).

DESIGN AND IMPLEMENTATION

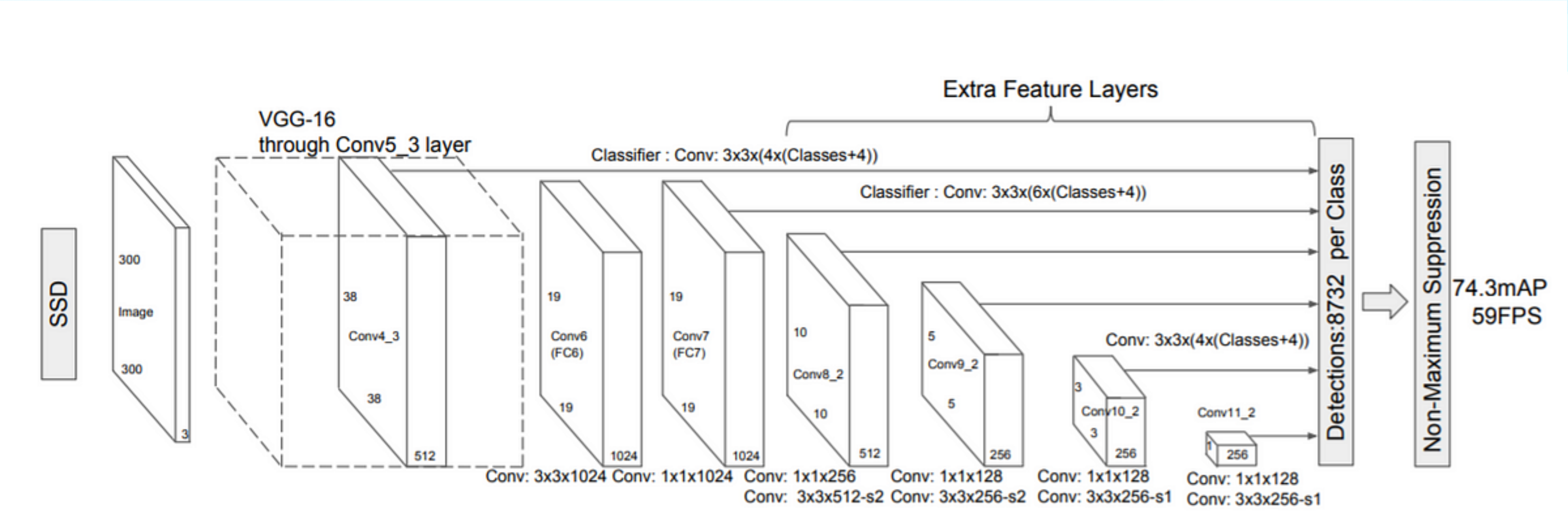
- Image annotation was done with a Python GUI tool called labelmg as shown, identified 5 classes of objects for our object detection problem: Vehicles (Cars, Trucks), Bike (Cycle) Motorbike, Traffic light and Traffic Sign. For each image an associated label .xml file in the pascal VOC format is hence created. SSD MobileNet (Single Shot Multi-box Detector) algorithm was used for object detection.

SSD MobileNet V1: Design Specifications	
Initial Learning Rate	0.004
Activation Function	ReLU (Rectified Linear Unit)
Batch Size	10
Regulariser	L2 Regulariser
Epochs	4000
Loss Function	RMSE (Root Mean Squared error)

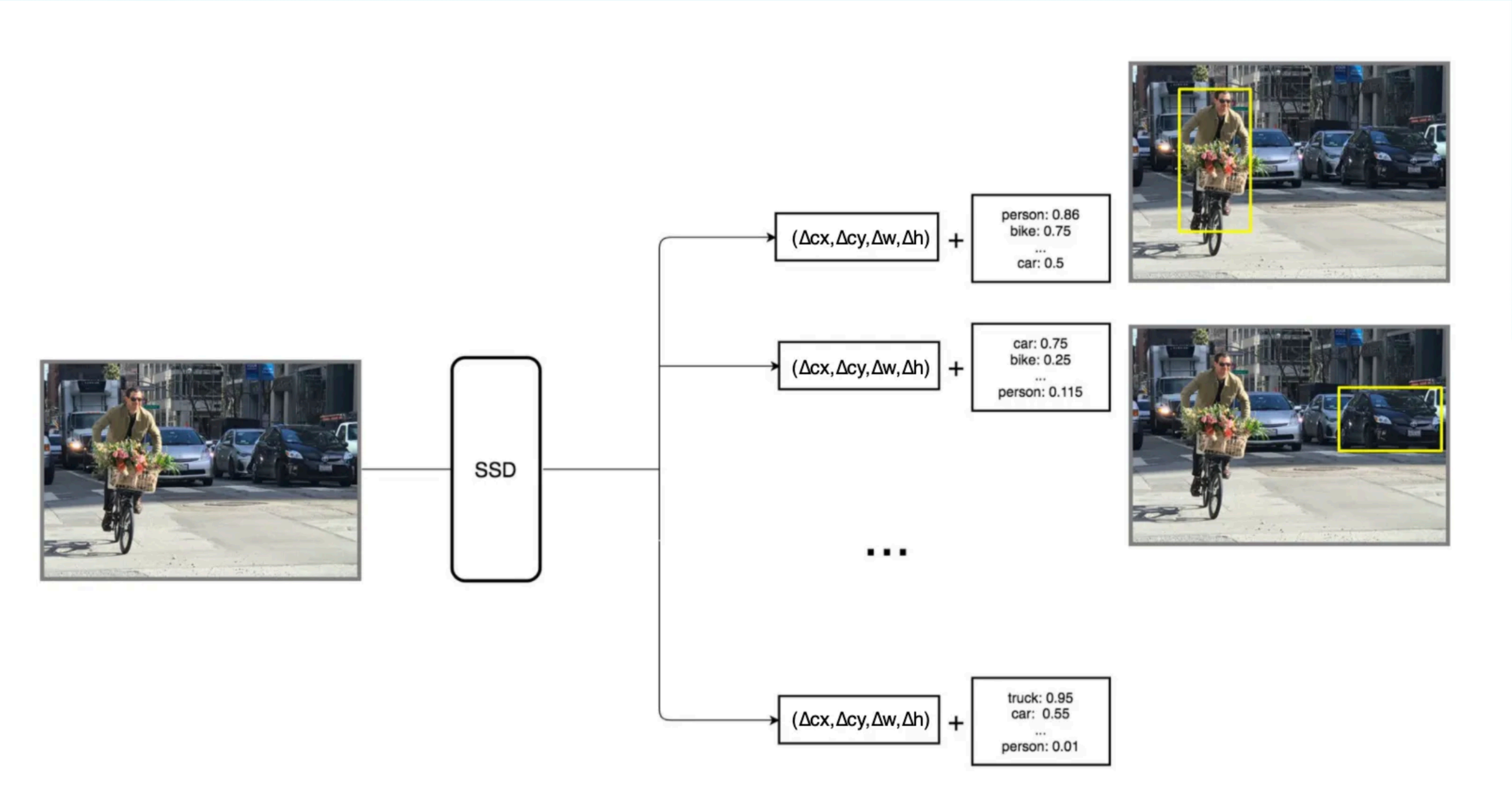


Single Shot Multi-box Algorithm

- **Single Shot:** this means that the tasks of object localization and classification are done in a *single forward pass* of the network
- **MultiBox:** this is the name of a technique for bounding box regression developed by Szegedy et al
- **Detector:** The network is an object detector that also classifies those detected objects



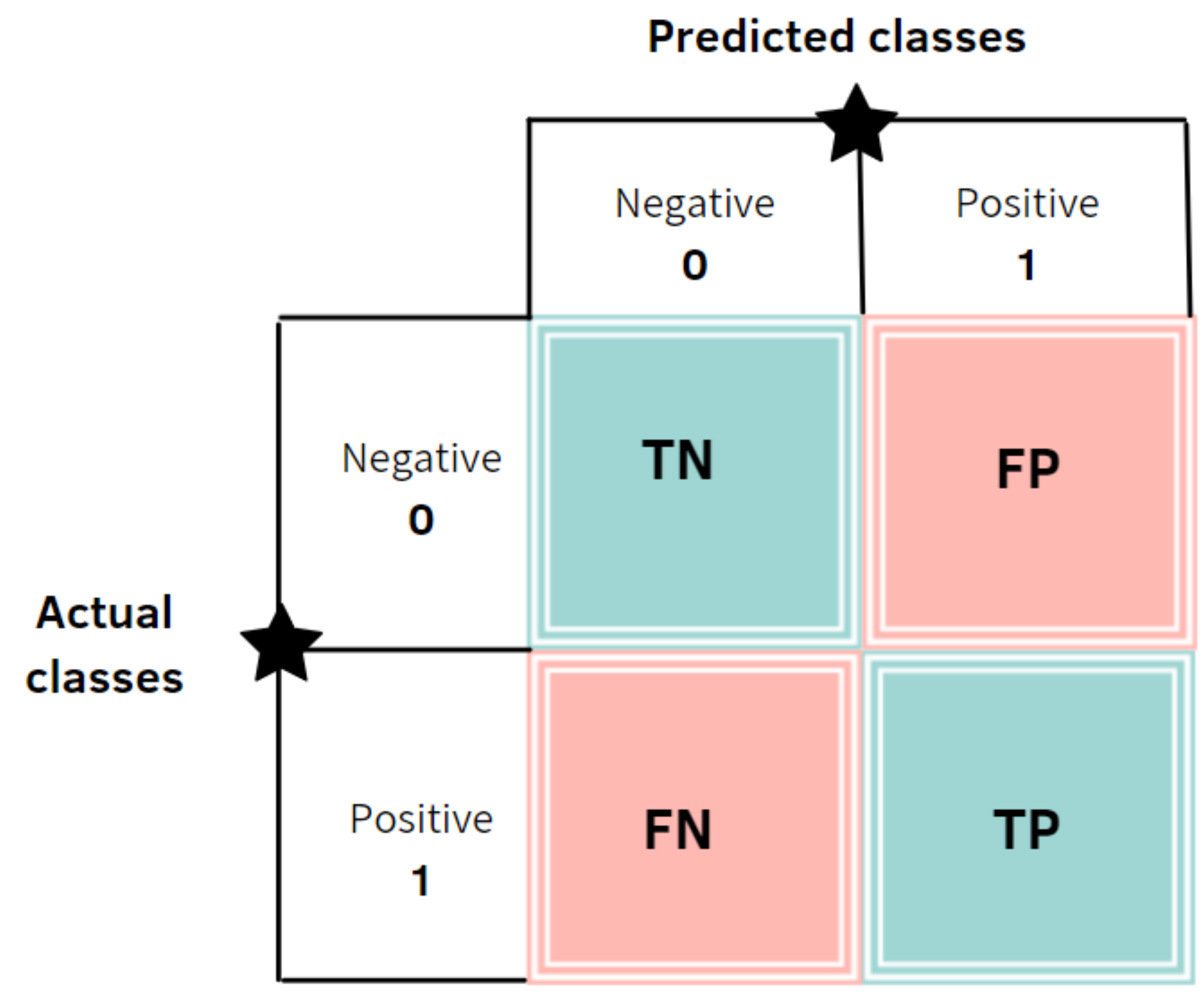
Single Shot Multi-box Algorithm



https://jonathan-hui.medium.com/ssd-object-detection-single-shot-multibox-detector-for-real-time-processing-9bd8deac0e06

PERFORMANCE METRICS

Criteria	Abbreviations	Formulas
Accuracy Rate	ACC	(TP+TN)/(N+P)
Sensitivity - True Positive Rate	TPR	TP/(TP+FN)
Specifity - True Negative Rate	TNR	TN/(TN+FP))
False Positive Rate	FPR	FP/(FP+TN)
False Negative Rate	FNR	FN/(FN+TP)
Positive Pred Value - Precision	PPV	TP/(TP+FP)
F Score	F	2*((PPV*TPR)/(PPV+TPR))
Error Rate	ERR	(FP+FN)/(N+P)



Thank You

Team Members :

Abhijit rao.S	SE21UARI003	N.Gagan	SE21UARI094
Varshini.V	SE21UARI183	Sri Ashritha	SE21UARI153
Srinivas Gurram	SE21UARI190		