Lord’s Eye

Team : Parabellum

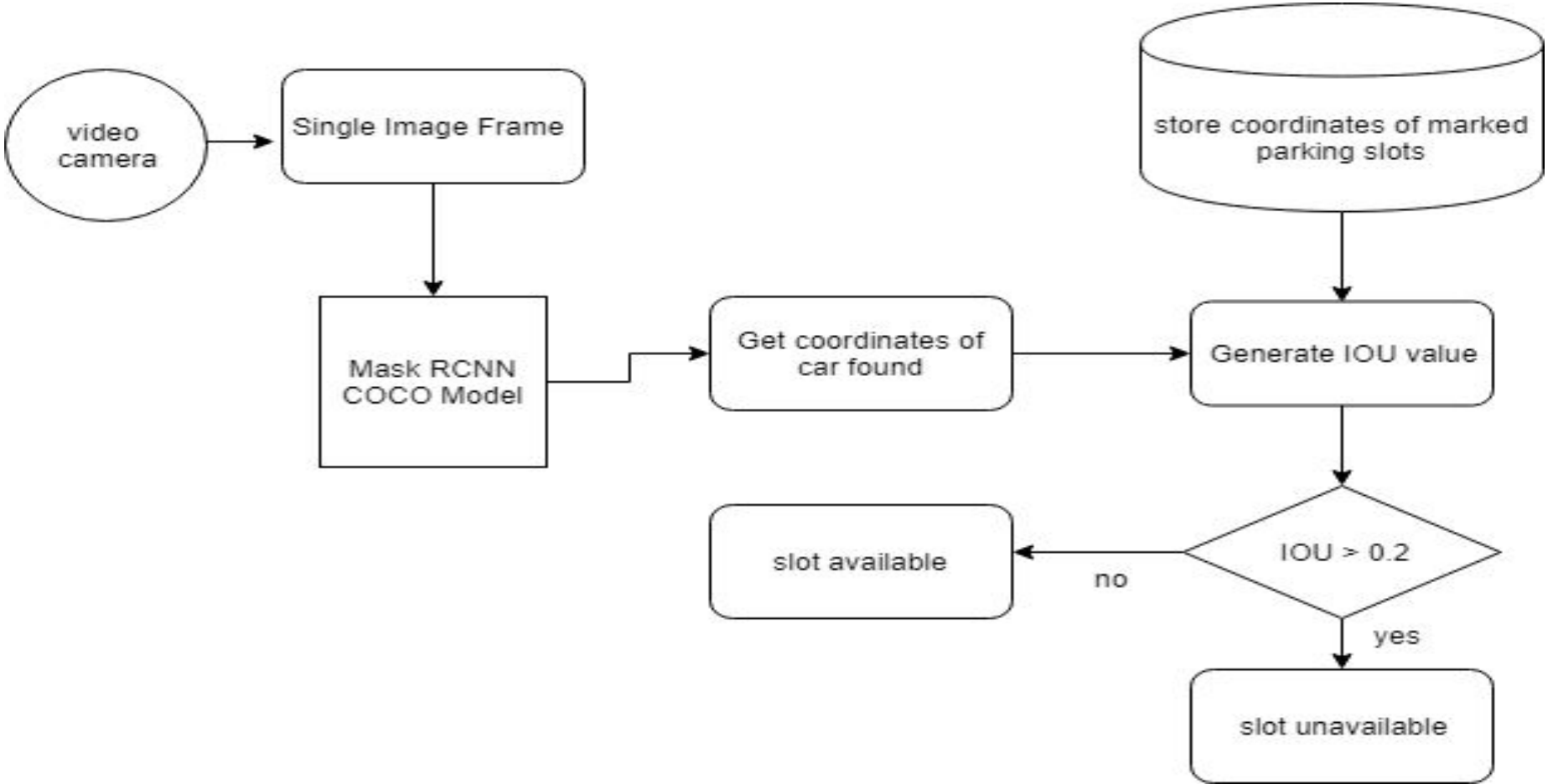
Problem Statement

- **AIR POLLUTION & ROAD TRAFFIC :** Average time motorists spend looking up to a total of 90.5 hours spent searching for a parking spot over the year.
- Currently in **malls and cinemas, hardware solutions** are been used like **Infrared or weight sensor** to check the presence of car at a particular slots & congestion.
- The approach is **very costly and isn't self sustaining.**

Proposed Solution - **Lord's Eye**

- ➔ The idea to build an **object detection model** that looks beyond the **parking space hash marks** drawn on the plots.Since all the parking space line markers are really small and hard to see from a distance, so they will also be *hard to detect with a computer.*
- ➔ Our project Lord's Eye comes into play that helps in detecting the parking slots by marking the slot based on **detected cars and its time limit.**
- ➔ Once marked, the **coordinates will be then stored** and used throughout system's life to find if spots stored are empty or not.

Flow Diagram of Proposed Solution



Approach

We pass each frame of the video captured by the camera through a pipeline, one frame at a time

1. Detect all parking spaces in the area
2. Detect all the cars in the area
3. Determine which spaces are occupied by a car and which are empty using steps 1 and 2

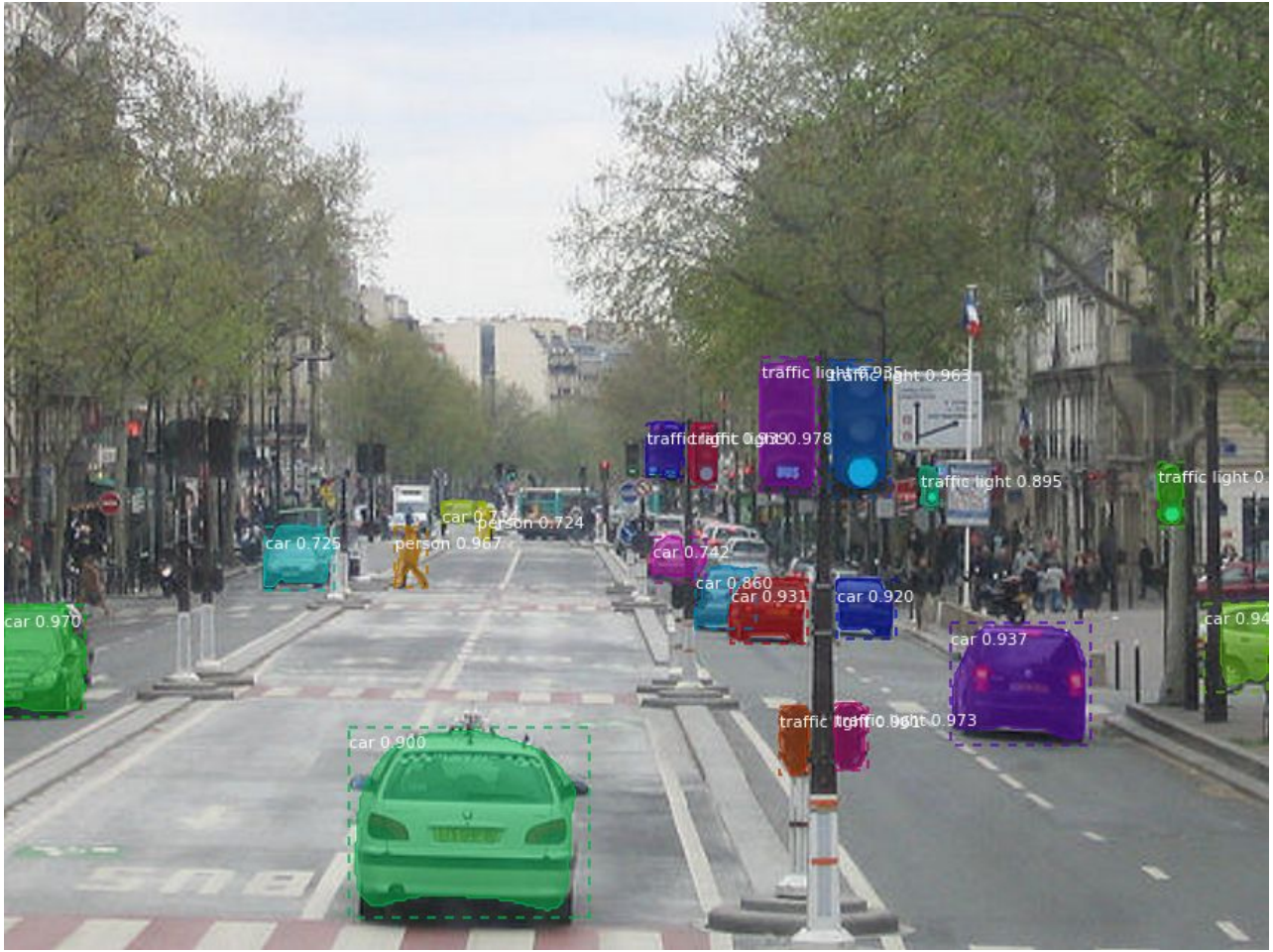
- There are many methods for detecting a parking space
  - Hardcoding the locations of each parking spot - inefficient because if we move the camera to a different location we'll need to hardcode everything again
  - Detect parking spots using parking line marks on the ground - unreliable as not all lines will be clear in the camera and model may not detect them
  - Best approach - detect cars that don't move for a long time, these denote parking spaces
- So our problem boils down to detection of cars in all the frames - using RCNN COCO model and then finding out which parking slots are occupied and which are free

Module 1 : Marking the parking slots using Mask R-CNN.

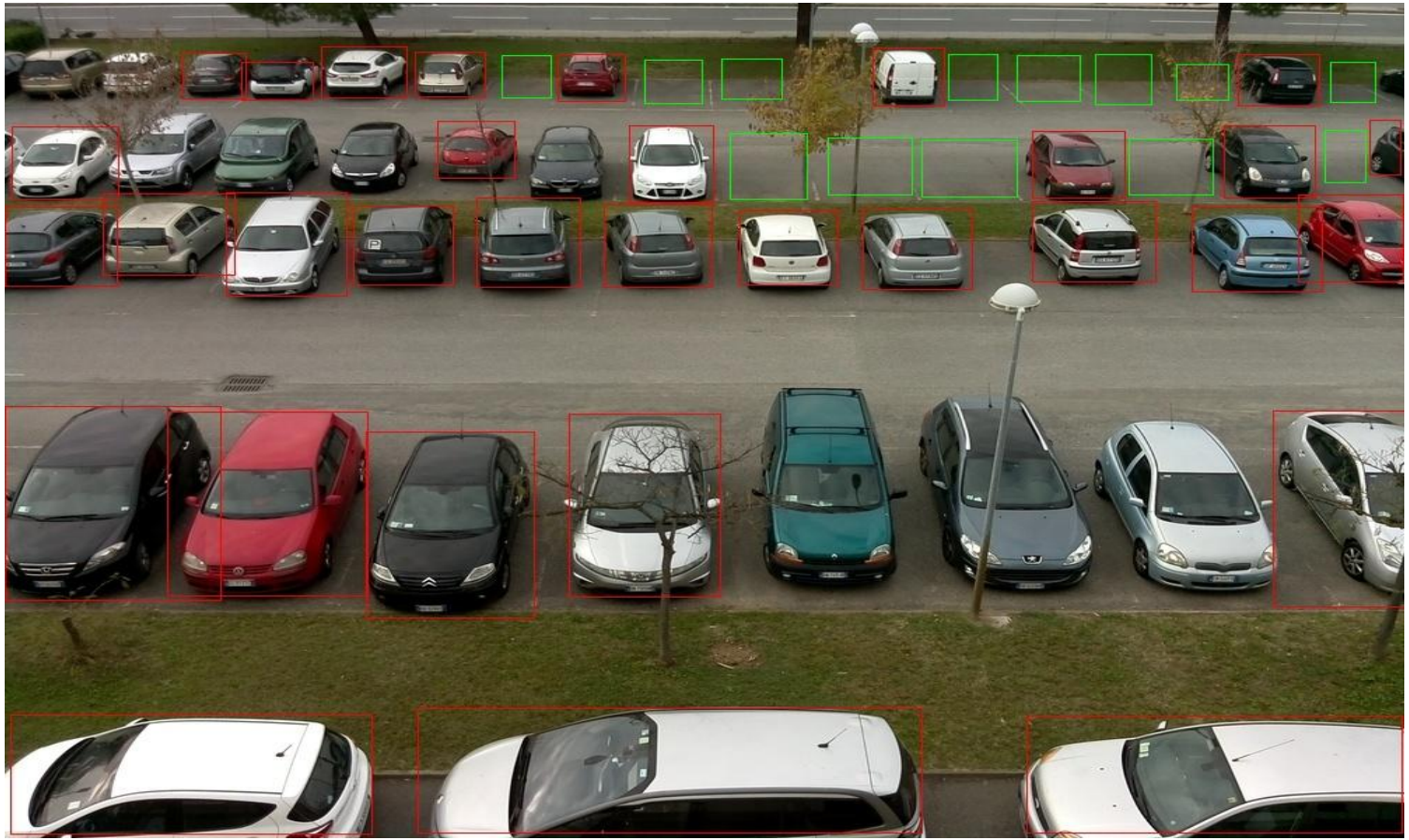
- We use the Mask R-CNN model with the COCO dataset
- We basically obtain bounding box of each object along with its location and also an object outline(mask)
- We use the COCO (Common Objects In Context) dataset that has images annotated with object masks
- We use a great open source Mask R-CNN implementation from Matterport which comes with a pre-trained model.
- For each object detected in the image, we get back four things from the Mask R-CNN model:
  a. The type of object that was detected (as an integer). The pre-trained COCO model knows how to detect 80 different common objects like cars and trucks.
  b. A confidence score of the object detection. The higher the number, the more certain the model is that it correctly identified the object.
  c. The bounding box of the object in the image, given as X/Y pixel locations.
  d. A bitmap “mask” that tells which pixels within the bounding box are part of the object and which aren’t. With the mask data, we can also work out the outline of the object.
- We get the pixel coordinates of each detected car printed which we then store in a CSV

  Cars found in frame of video:
  Car: [492 871 551 961]
  Car: [450 819 509 913]
  Car: [411 774 470 856]
- We then use IOU (Intersection over Union) method to detect available parking spots

Mask R-CNN on COCO test images


car 0.970
car 0.725
person 0.967
person 0.724
car 0.742
car 0.860
car 0.931
car 0.920
car 0.937
car 0.900
traffic light 0.963
traffic light 0.978
traffic light 0.895
traffic light 0.973

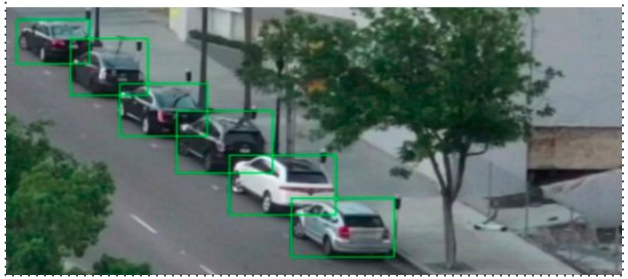
Screenshots for Module 1

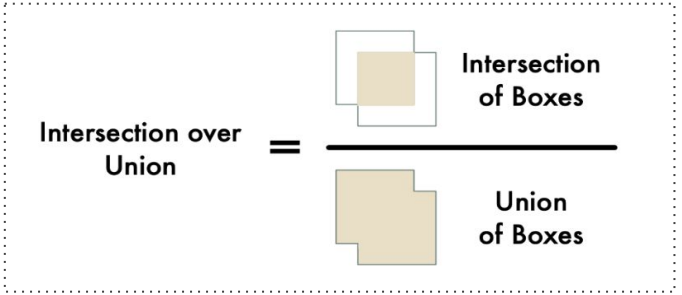


Screenshots for Module 1

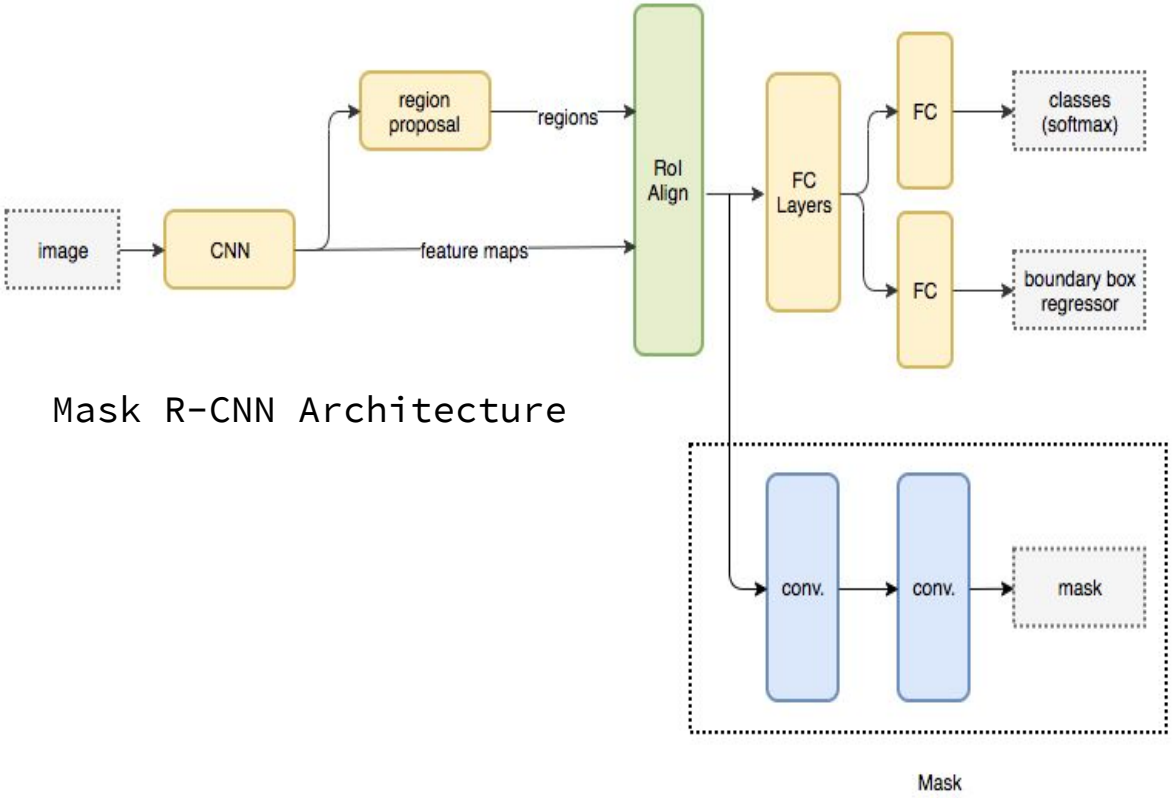
Module 2 : Checking the available parking slots using IOU method

- ➔ Coordinates of "**marked parking slots**" are stored in database.
- ➔ Our model detects the presence of the car and retrieves its coordinate and **compares with all the coordinates data available in the db** and checks in which slot the car is present.
- ➔ To measure we have used ***Intersection Over Union or IoU*** which is calculated by finding the amount of pixels where two objects *overlap and dividing it by the amount of pixels* covered by both objects, like the below figure.
- ➔ It returns the measure of how much a **car's bounding box is overlapping a parking spot bounding box.**
- ➔ With this, we can easily work out if a car is in a parking space or not.
- ➔ If the IoU measure is low, like 0.2, that means the car isn't really occupying much of the parking space.
- ➔ But if the measure is high, like 0.6, that means the car is occupying the majority of the parking space area so we can be sure that the space is occupied.





Technology Stack



Mask R-CNN Architecture

Keras + TensorFlow

Pandas

OpenCV

NumPy

scikits learn

COCO
Common Objects in Context

ADD ONS : Planning to Build before Finals

- ➔ To detect the cars which are **not parked perfectly** and store its **number plate value in database by using OCR** on detected car.
- ➔ To **find the nearest available parking slots** for driver for ease in parking.
- ➔ To check the number of **available parking spaces and traffic congestion** at a particular area and display it on to smart screens or on mobile apps using IOT.
- ➔ System that checks the presence of **vehicle at real-time basis** in order to provide the insights about the traffic congestions as well as the available parking slots that helps in **minimizing the traffic as well as pollution and gas consumption rates.**
- ➔ **Performance and predictive analysis** based on cars parked in each month and time to analyse shopping trends.

More about Team Members :

We Believe : ***The world is hungry for inspiration. It craves stories of success in dealing with complex issues, especially in technological world. Chopping of origin roots of problem is the best and self-sustainable approach to solve various challenges.***

Kavish Gandhi

Team Leader
Computer Engineering (2019 batch)

I am an energetic, enthusiastic person, with a will for constant improvement and lifelong learning. I have been bright academically and my desire to solve problems has led me to the field of computer science and engineering. I've completed my computer engineering from VJTI college, Mumbai.
I believe that technology is a great leveller. With technology, comes the responsibility to do good for the society. I wish to apply my knowledge in the field of ML and AI for social good.

Shruti Rampure

Team Member - 1
Third Year - Computer Engineering (2021 batch)

I am a person who is positive & energetic about every aspect of life. I'm Currently pursuing my Computer engineering degree from Sardar Patel Institute of Technology college, Mumbai.
The rapidly-evolving field of computer engineering offers me an unique opportunities to make a real difference by designing innovative software solutions for solving global and day-to-day problems that we face in our life using various upcoming and highly disruptive technologies like AI and ML.

Bhavya Meghnani

Team Member - 2
Fourth Year - Computer Engineering (2020 batch)

My objective is to secure a job in the IT industry where I can utilize my knowledge for the organization's growth.
I have completed my diploma in computer engineering and currently pursuing degree from "Sardar Patel Institute of technology, Mumbai".
I have build many computer related projects and professionally have implemented few of them for solving real-life challenges.
I consider myself as a good public speaker and my dream is to become a social entrepreneur.