INTRODUCTION:

This project, vehicle parking management system using imag processing aims to create a better environment for a vision-based vacancy parking area detection; providing a modern and innovative solution for temporary parking place.

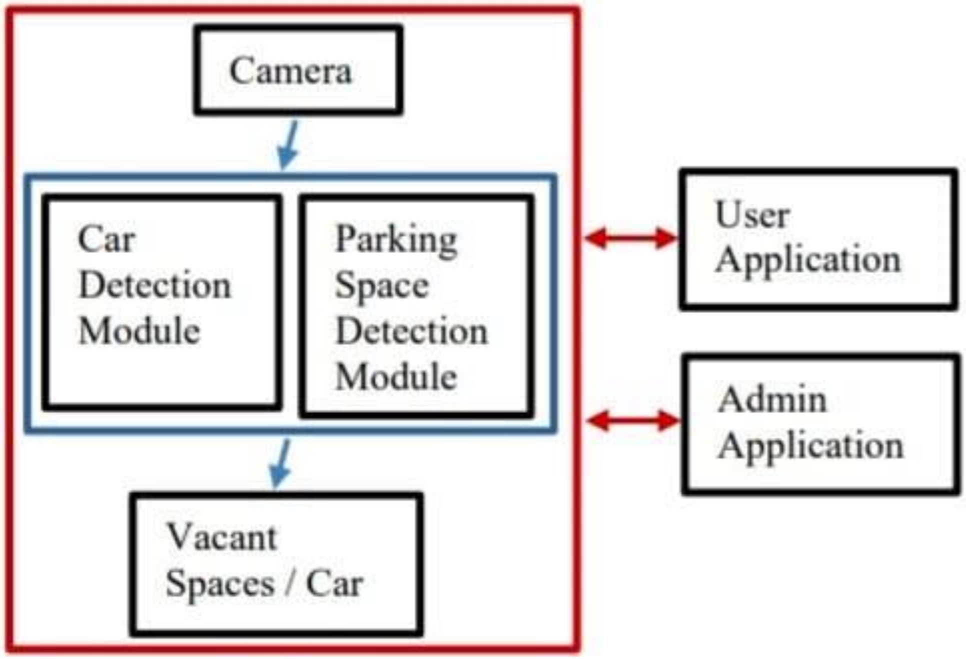
The images obtained from the live stream are then fed to the processing module, which detects the region of interest (ROI) consisting of the area to be covered for parking spaces. A car detection Module is used to detect the cars within ROI using Neural Network.

The parking space detection module generates the virtual lines for parking which will be visible to user on an app assisting in vehicle parking. Due to human error if someone parked the vehicle between the lines or parked the vehicle in wrong manner, the output will be shown at the admin side of Application with the count of wronged parked Car.

Two application interfaces have been developed, the user interface and the admin interface. Whenever the user opens the application, he will be able to see the image of the parking lot and number of vacant spaces into the parking lot.

In order to provide ease to the user; help screen is provided which tell the user how to check the status of the parking lot. The Admin application have an extensive data where the admin can see total number of parking slots, total number of cars, vacant space, occupied space as well as No.of Incorrectly parked cars. Python Programming language is used to train the models and do processing. Android Studio is used to develop the mobile applications for user and admin. Firebase is the real-time database where images are stored along with other information like total number of parking slots, vacant parking slots, occupied slots, number of cars correctly or wrongly parked. Annotation are done using Vgg Annotator.parking slots, occupied slots, number of cars correctly or wrongly parked. Annotation are done using Vgg Annotator.

Fig. 1 Block diagram summarizing the overall process and working modules



ROI Detection :

The camera has been mounted on roof top of a building from where the parking lot is completely visible. First challenge is to detect the ROI in the image obtained from camera. For this purpose markers are placed on the image corners using distinguishing colour. All images are resized to consistent dimensions and are then passed through two convoluting filters. The convoluted image is then made to pass through Gaussian blur, where the image is made slightly blurred. This process converts the original image to filtered image. The smooth and unique colour of markers is then detected using masking technique, based on the range of pixel values in the HSV colour space. In order to find out the coordinates of the mask, contours are located, which can be defined as the line of pixels whose values are same or in other word the edges in image are found. Since the camera is mounted on to roof top at some Angle, the ROI obtained is not a square, so the perspective of the image needs to be changed, to make it square because the car near to the camera is of different size as compare to the car placed far to the camera. Once ROI has been marked properly.

Application :

Camera

Car

Detection

Module

Parking

Space

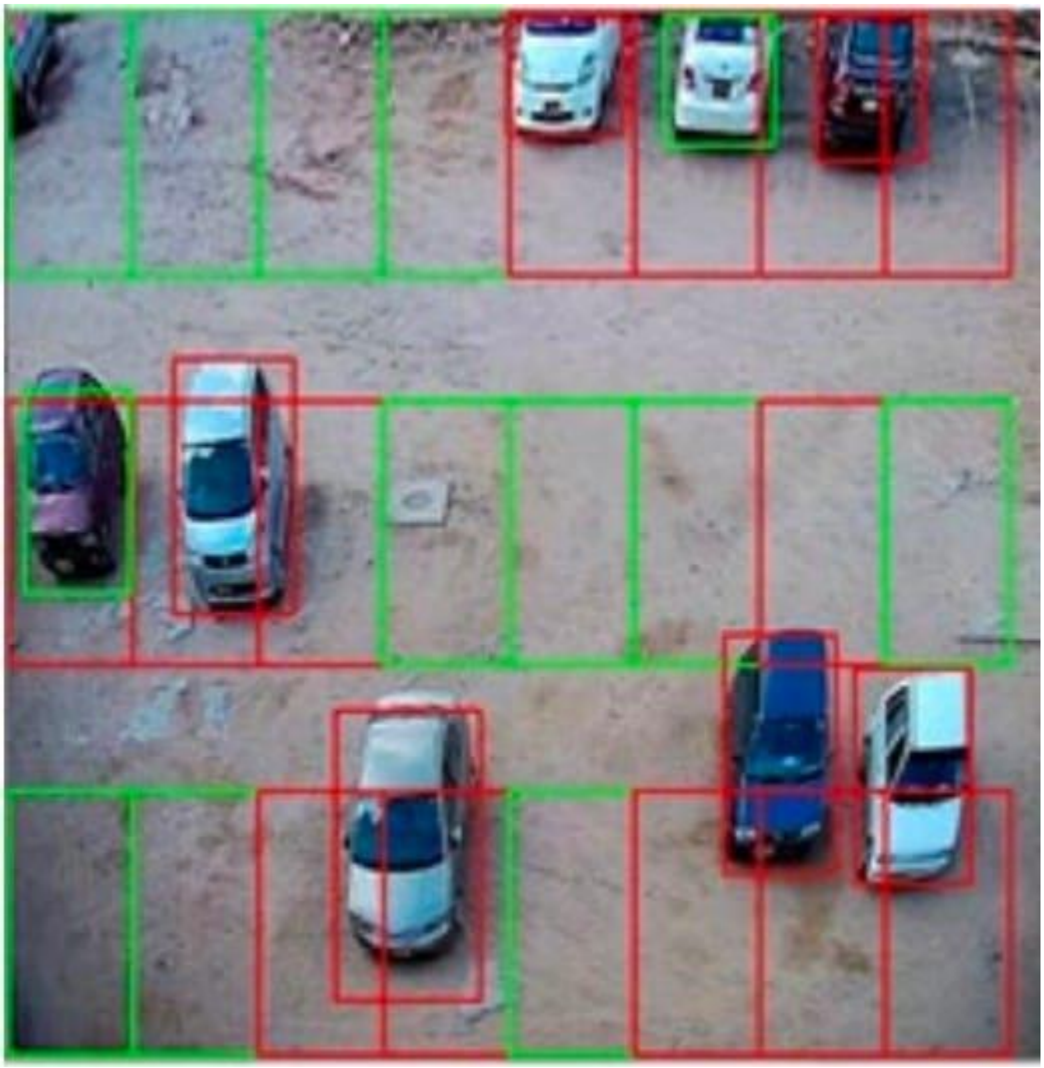
Detection

Module

Vacant

Spaces / Car

Admin

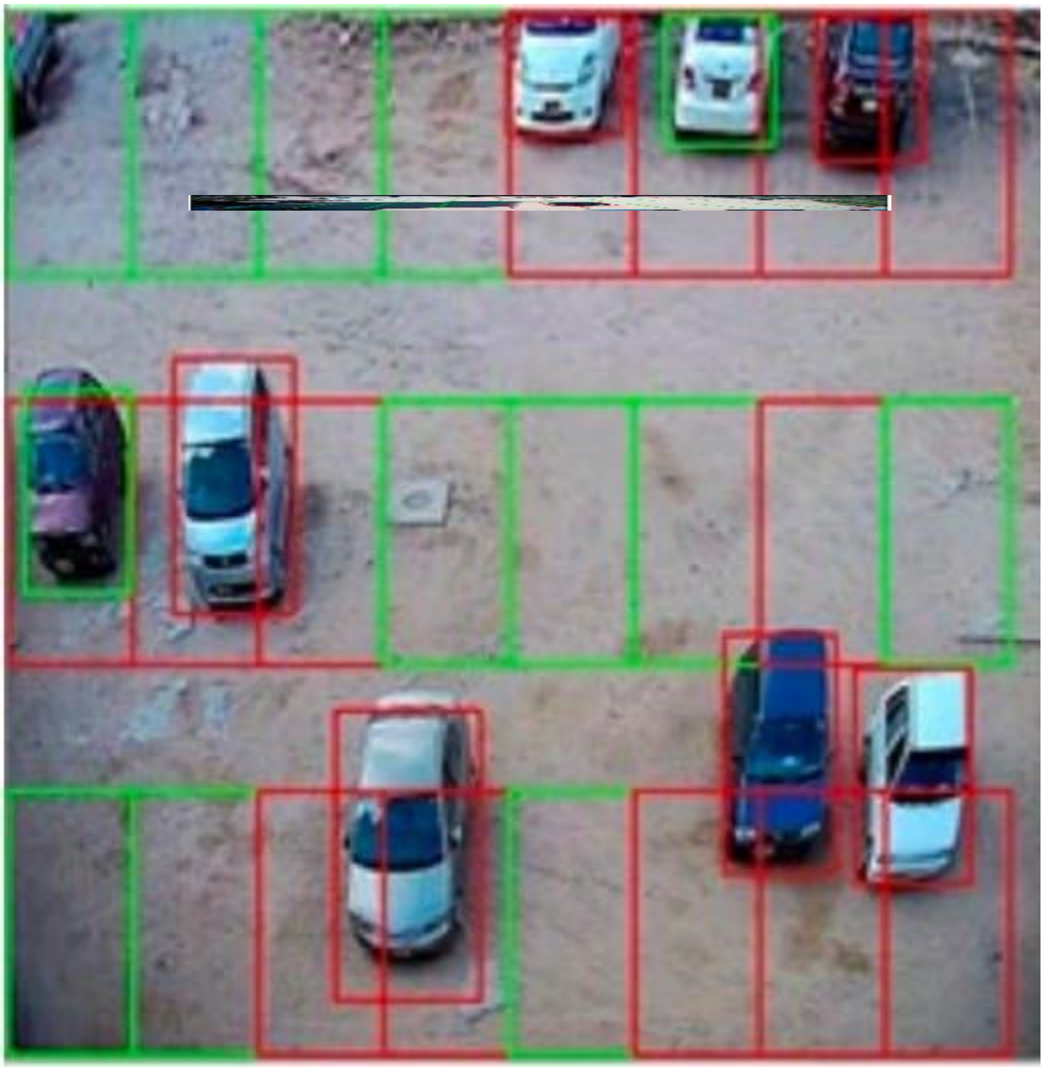


This process converts the original image to filtered image. The smooth and unique colour of markers is then detected using masking technique, based on the range of pixel values in the HSV colour space.

Since the camera is mounted on to roof top at some Angle, the ROI obtained is not a square, so the perspective of the image needs to be changed, to make it square because the car near to the camera is of different size as compare to the car placed far to the camera.

Car Detection :

Once the model being trained the weights are being saved in H5 format that can be used for detecting a car in an image. The prepared images are fed to the neural network. The chosen model architecture for training is Mask RCNN.



Drawing Virtual Parking LinesAfter detecting the cars, next virtual parking lines are drawnin the image. Since in the image the unit of distance measurement is a pixel whereas actual distances are in the unit of feet

The resulting model detects the boundary of every car. While defining the architecture to load the weights, the confidence value is set to 0.9 that means an object which have 90% confidence to be a car is detected while objects having lesser confidence values are rejected.

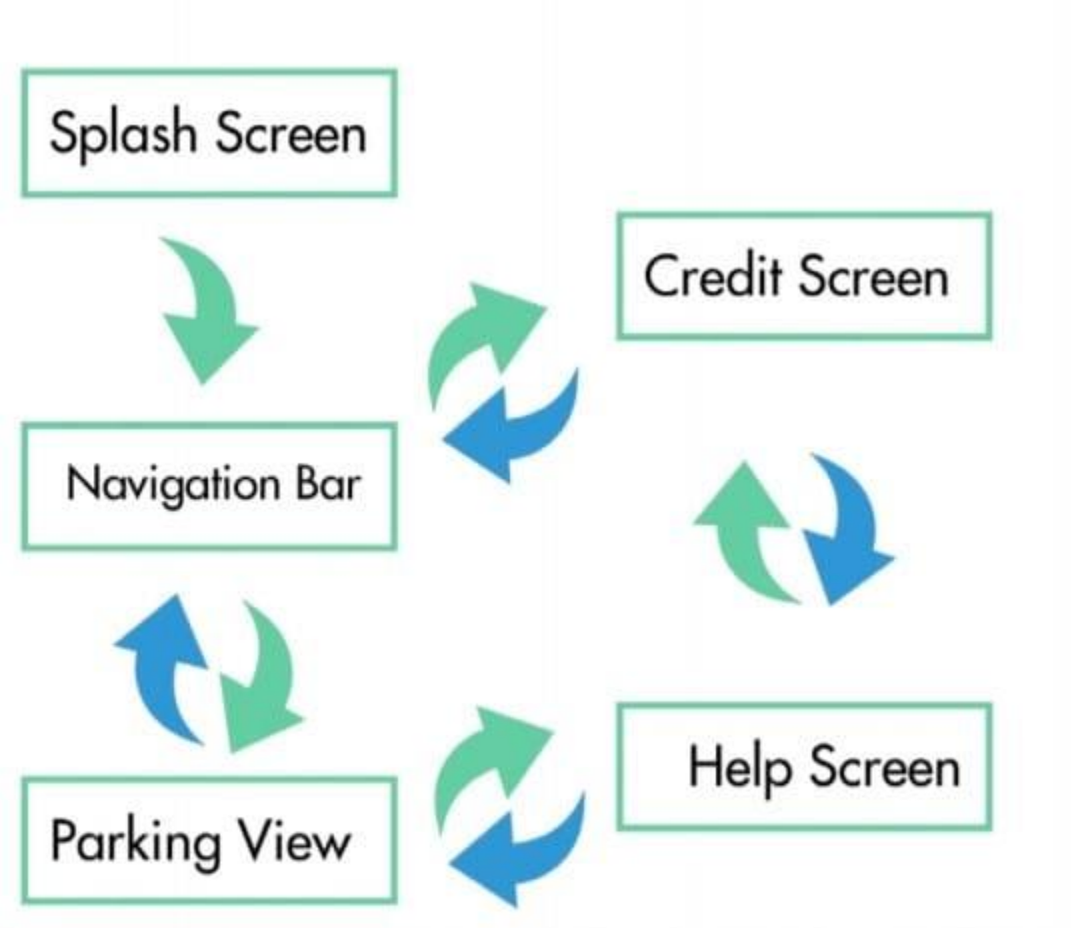
Finding Car Intersection :

The imaginary parking lines are made to achieve maximum parking, since the cars are not parked in correct way, car area needs to be calculated which resides inside a proper parking slot. To handle with this, three classes have been defined, namely, the car class, the parking class and the intersect class.

User Interface :

The user interface is the most important aspect for which the mobile application is used to provide a real-time parking space detection to facilitate the users.

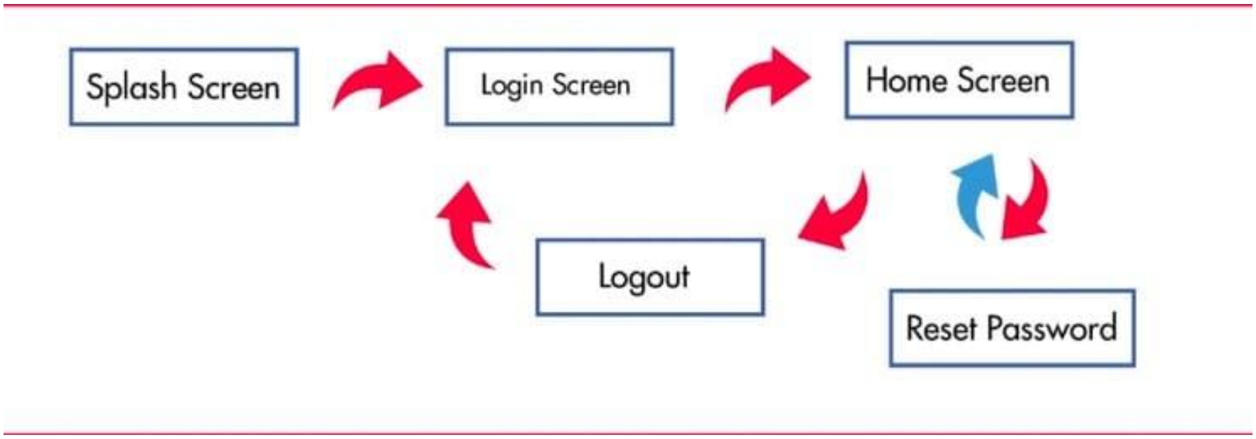
On home screen user will be able to see the parking lot with Imaginary lines through which the user will be able to determine whether any slot is vacant or not and also the total number of empty parking slots. The red box indicates the parking slot is occupied; if green then the parking slot is vacant.



Admin Interface :

In order to provide an ease to the parking management and admin personals at the parking lot, an admin application interface has been provided.

The admin application interface provides extensive data related to the parking lot, containing information like total number of car parking slots, number of the cars parked, number of correctly and incorrectly parked cars



Conclusion :

The Final Version of our Project is a mobile Application offering a start-up product in management sector that aims to address the parking difficulty issues at some mega-events where vehicles have to be parked in temporary parking area. The vision-based parking management system features to have maximum parking within the ROI and to facilitate the user with the best.