# A SURVEY ON VARIOUS AVAILABLE OBJECT DETECTION MODELS AND APPLICATION IN AUTOMATIC LICENCE PLATE DETECTION

**Aditya Kulkarni[1], Manali Munot[2] , Sai Salunkhe[3] Shubham Mhaske[4] Nilesh Korade[5]**

*[1] Student at Pimpri Chinchwad College of Engineering and Research, Ravet, Pune, India-412101*

*[2] Student at Pimpri Chinchwad College of Engineering and Research, Ravet, Pune, India-412101*

*[3] Student at Pimpri Chinchwad College of Engineering and Research, Ravet, Pune, India-412101*

*[4] Student at Pimpri Chinchwad College of Engineering and Research, Ravet, Pune, India-412101*

*[5] Asst. Professor at Pimpri Chinchwad College of Engineering and Research, Ravet, Pune, India-412101*

***Abstract:*** *With the development in technologies right from serial to parallel computing, GPU, AI, and deep learning models a series of tools to process complex images have been developed. The main focus of this research is to compare various algorithms(pre-trained models) and their contributions to process complex images in terms of performance, accuracy, time, and their limitations. The pre-trained models we are using are CNN, R-CNN, R-FCN, and YOLO. These models are python language-based and use libraries like TensorFlow, OpenCV, and free image databases (Microsoft COCO and PAS-CAL VOC 2007/2012). These not only aim at object detection but also on building bounding boxes around appropriate locations. Thus, by this review, we get a better vision of these models and their performance and a good idea of which models are ideal for various situations.*

***Keywords: Digital Image Processing, OpenCV, License Plate Detection, License Plate Recognition,YOLO,OCR.***

## 1. INTRODUCTION

Many systems use vehicle plate identification and recognition, including travel time calculation, highway car counting, traffic violation detection, and surveillance applications. As the population grows, so does the number of vehicles on the road.As a result, finding a parking spot for a large number of students and faculty at Educational Institutions has become increasingly difficult. The majority of parking lots are handled manually by security guards who may or may not keep track of the vehicles parked there.As a result, the vehicle driver must continue to walk the parking lot in search of a parking spot. In the absence of security guards, car robberies and quarrels between drivers over parking spaces can occur.Automated License Plate Recognition (ALPR) is another name for Automated Number Plate Recognition (ANPR).Automatic Number Plate Recognition, or ANPR, is a technology that 'reads' vehicle number plates using pattern recognition. Simply put, ANPR cameras 'photograph' the number plates of vehicles that violate the rules as they drive by.This 'photograph' is then fed into a computer system, which extracts information about the vehicle's driver and owner, as well as information about the vehicle itself. ANPR is made up of ccs that are regulated by a computer.When a car passes, ANPR 'reads' Vehicle Registration Marks – also known as number plates – from digital images captured by cameras mounted on a mobile unit or in traffic monitoring vehicles.In the identification of licence plates, computer vision and character recognition, as well as algorithms for licence plate recognition, play a critical role. As a result, they are the foundation of every ANPR scheme.A static camera, a framer, a monitor, and specially built applications for image processing, analysis, and recognition are all part of the framework for automatic car licence plate recognition.

## 2. RELATED WORK

Due to the exponential growth of automobiles, bikes, and other vehicles, automatic number plate recognition is a well-known proposal in today's world. For vehicle identification, this automated number plate recognition device employs image processing technology.This device can be used in densely populated and restricted areas to quickly identify vehicles that have broken traffic rules, as well as retrieve the owner's name, address, and other information.

P. Meghana, S. SagarImambi, P. Sivateja, and K. Saira collaborated on the design and implementation of the proposed concept.

## 3. APPROACH

The solution for these issues can be provided with the help of already feasible and efficient CCTV camera networks. Most of the residential society, tolls, business complexes and parking spaces have CCTV surveillance systems implemented.The idea is SIMPLE:To use existing CCTV systems to cope with the problems.For residential societies : If an unknown vehicle enters the premise, the authorities and security will be notified.Tracking and storing the vehicles passing through a checkpost. Based on this data critical analysis can be performed.Number plate recognition on real time data : This ensures that the whole video is processed in real time and only the license plate data is stored. Web-based portal for manual monitoring and operating the system.. Methodology of the project is given below :

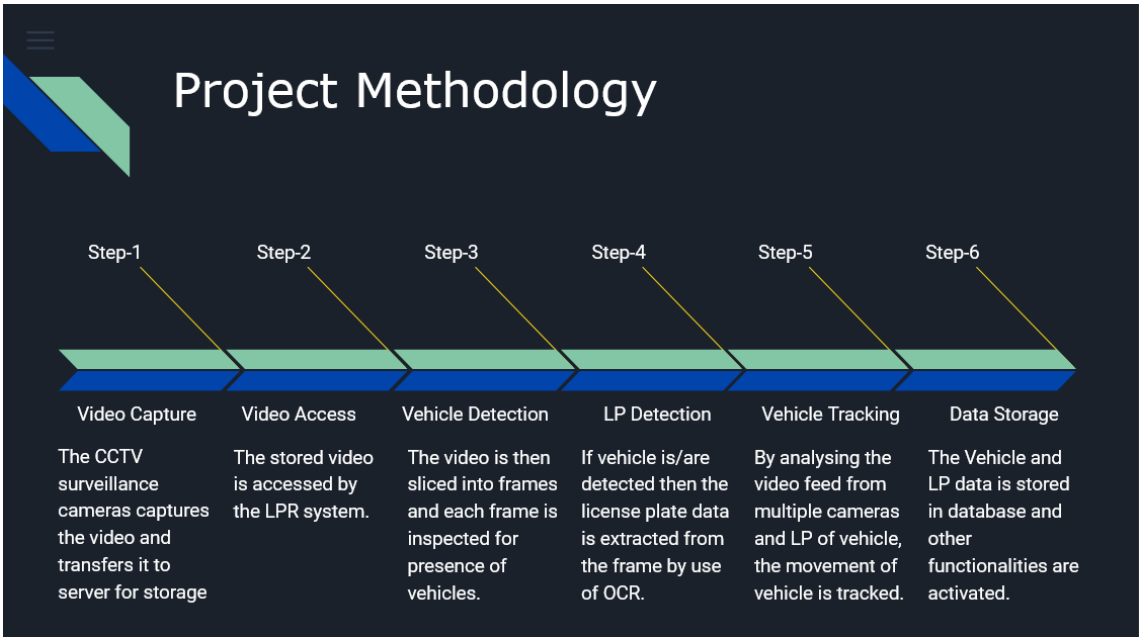


**Figure 1: Project Methodology**

## 4. METHODOLOGY

### 4.1 Convolutional Neural Network (CNN)

Convolution Neural Network(CNN) is a specialized neural network for processing the data that has input in the form of a 2D matrix; like image.The most important property of a convolution neural network is that the algorithm itself assigns importance in the form of learn-able weights and biases to various aspects and objects in the image. This means that CNN is able to recognize the more accurate features and differentiate important features from unimportant ones.

CNN's are specifically used for image detection and classification. Seldom use of CNN is noted in pattern detection and matching. Most of the use of CNN is on data in the form of an image. Images are a 2D matrix of pixels. We can use CNN to either recognize the image or classify it.

The convolution neural network model can be defined in form of three layers:

1. Convolution Layer: The purpose of the convolution layers is to extract different features of the input. The first convolution layer extracts low-level features like edges, lines, and corners. Higher-level layers extract higher-level features like objects and shapes. The input is a 3D matrix of size N x N x D and is convolved with H kernels, each of size k x k x D separately. Convolution of input with one kernel generates one output feature, thus a total of H output features are generated. The Kernel starts from the top-left corner of the input, each kernel is moved from left to right, one element at a time. Once the top-right corner is reached, the kernel is moved one element in a downward direction, and again the kernel is moved from left to right, one element at a time. This process is repeated until the kernel reaches the bottom-right corner. For each position of the kernel in a sliding window process, k x k x D elements of input and k x k x D elements of the kernel are element-by element multiplied and accumulated. So to create one element of one output feature, k x k x D multiply-accumulate operations are required.
2. Pooling Layer: The pooling layer, also called the subsampling layer decreases the resolution of the features. It makes the features immune to noise and distortion. There are two ways to do pooling: max pooling and average pooling. In both cases, the input is divided into non-overlapping two-dimensional spaces.Consider the following 4x4 matrix:

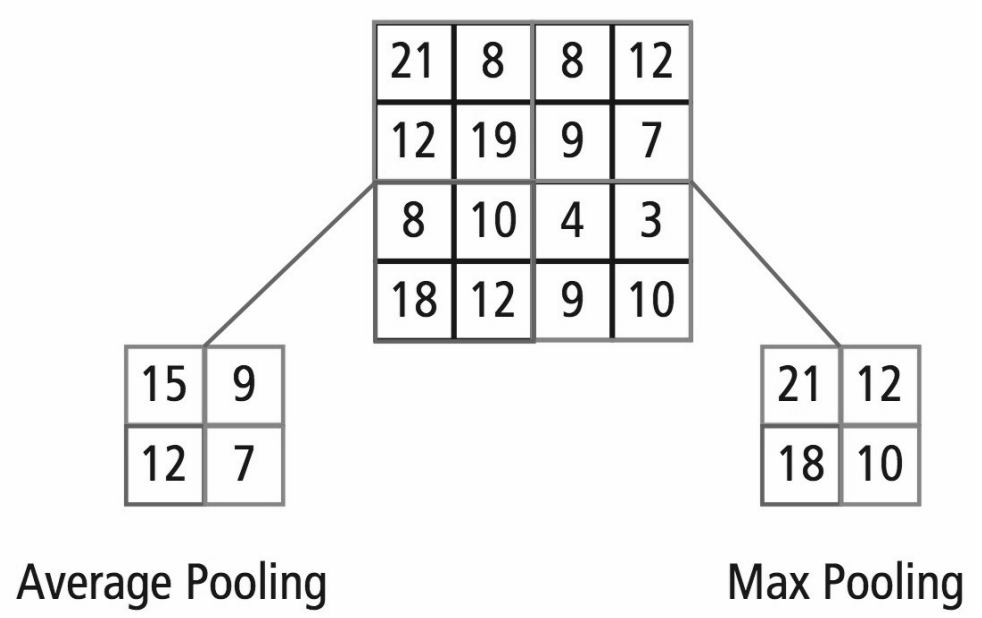


**Figure 2: Example of max pooling and average pooling**

- Average Pooling:In this pooling method, the average of the values in a region is calculated and used to represent the region. For instance in the above matrix top-left region has values [21,8,12,19]. So on average pooling, this region is represented by the average of these numbers i.e 15.
- Max Pooling: In this pooling method, the maximum of the values in a region is used to represent the region. For instance in the above matrix top-left region has values [21,8,12,19]. So on max pooling, this region is represented by the maximum number among these numbersi.e 21.

3. Fully Connected Layer: Fully connected layers are often used as the final layers of a CNN. These layers mathematically sum a weighting of the previous layer of features, indicating the precise mix of “ingredients” to determine a specific target output result. In the case of a fully connected layer, all the elements of all the features of the previous layer get used in the calculation of each element of each output feature.

- ReLU-Rectified Linear Units: Neural networks in general and CNNs, in particular, rely on a non-linear "trigger" function to signal distinct identification of likely features on each hidden layer. CNN may use a variety of specific functions —such as rectified linear units (ReLUs) and continuous trigger (non-linear) functions—to efficiently implement this nonlinear triggering. ReLU implements the function y = max(x,0), so the input and output sizes of this layer are the same. It increases the nonlinear properties of the decision function and the overall network without affecting the receptive fields of the convolution layer.

**4.2 Region Based Convolutional Neural Networks(R-CNN)**

To overcome the problem of selecting a very high number of regions, Ross Girshick et al. proposed a method called region proposals. In region proposals, we use selective search to only take out 2000 regions from the image and work only with these 2000 regions. Selective search uses hierarchical grouping to find out regions that have similar regions based on color, text, size, and shape. Selective search algorithm is :

1. Generate potential regions from the initial segmentation.
2. Combine similar regions to form larger regions recursively using the Greedy algorithm.
3. Use these regions to produce final candidate regions. [12]

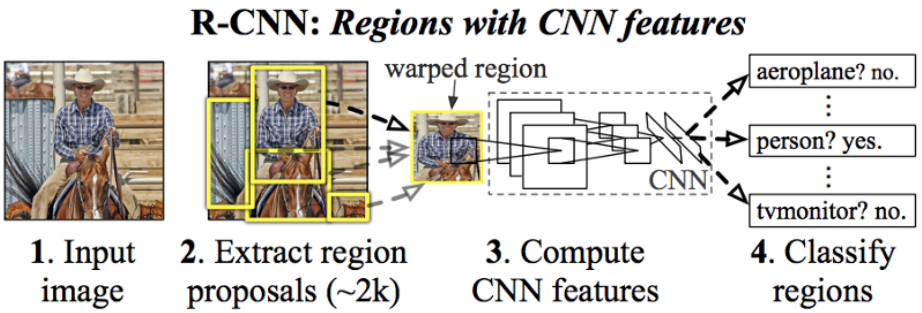


**Figure 3: Regions with CNN features-RCNN [12]**

Now, these 2000 region proposals are then fed to CNN as input by arranging them into a square pattern. CNN acts as a feature extractor and produces a 4096-dimensional vector output. These output layers consist of features that are further fed to SVM to classify the object present in the candidate region. Furthermore, the algorithm will also predict four values which are offset values to increase the precision of bounding boxes. R-CNN also has a few limitations. It takes up to 47 seconds for each image to be processed which is not possible to be implemented in real life. It also takes much time to train as 2000 regions must be extracted from each image. [12]

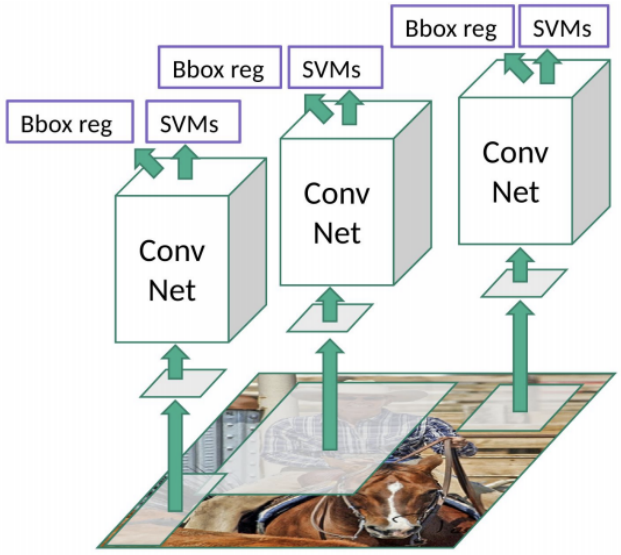


**Figure 4: RCNN [3]**

### 4.3 Fast Region Based Convolutional Neural Networks(F-RCNN)

The same author of R-CNN solved some of the issues of R-CNN and developed Fast-RCNN, A faster object detection algorithm.

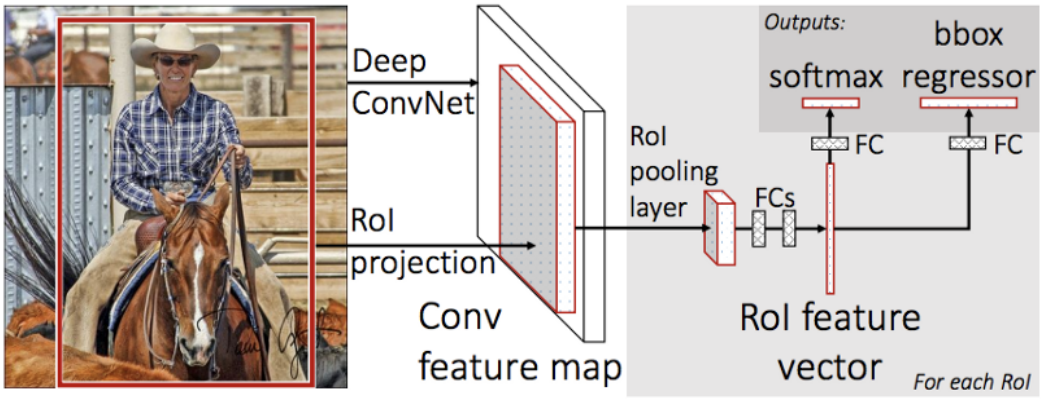


**Figure 5: F-RCNN [13]**

The approach of Fast-RCNN is comparable to R-CNN but we feed the input image to CNN to generate a convolutional feature map instead of region proposals. From this feature map, we recognize the various regions of proposals and warp them into squares. Further, we reshape them into a fixed size using the ROI pooling layer to feed them to a fully connected layer. From the ROI feature vector, the class of the proposed region and also the offset values for the bounding box can be predicted using a softmax layer.

Since you don't have to feed 2000 region proposals to the convolutional neural network every time, “Fast R-CNN” is faster than R-CNN.Instead, a feature map is created from the convolution operation, which is done only once per image. [13]

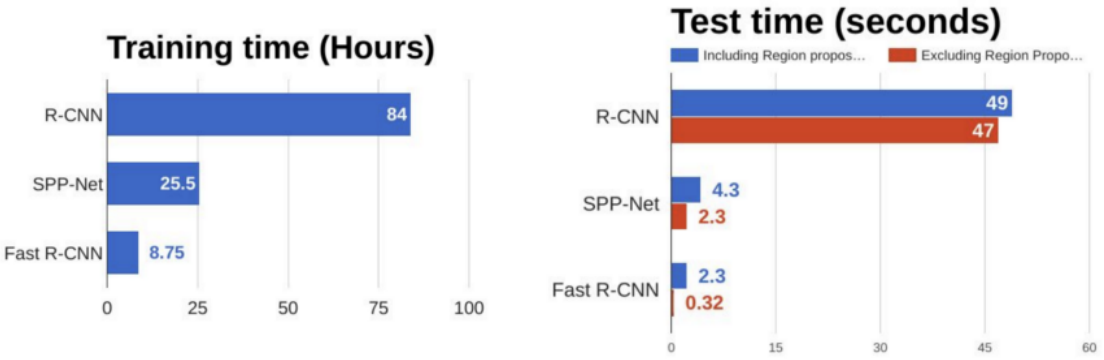


**Figure 6: Comparison of Training and Testing time of RCNN and F-RCNN [3]**

Fast R-CNN is substantially faster in training and research sessions than R-CNN, as seen in the graphs above.When comparing the output of Fast R-CNN during research, using region proposals significantly slows down the algorithm compared to not using region proposals.As a result, area proposals become bottlenecks in the Fast R-CNN algorithm, slowing it down.

### 4.4 Faster Region Based Convolutional Neural Networks

To find area proposals, both of the above algorithms (R-CNN and Fast R-CNN) use selective search.Selective search is a slow and time-consuming process that degrades network efficiency.As a result, Shaoqing Ren et al. devised an object detection algorithm that does away with the selective search algorithm and allows the network to learn region proposals.

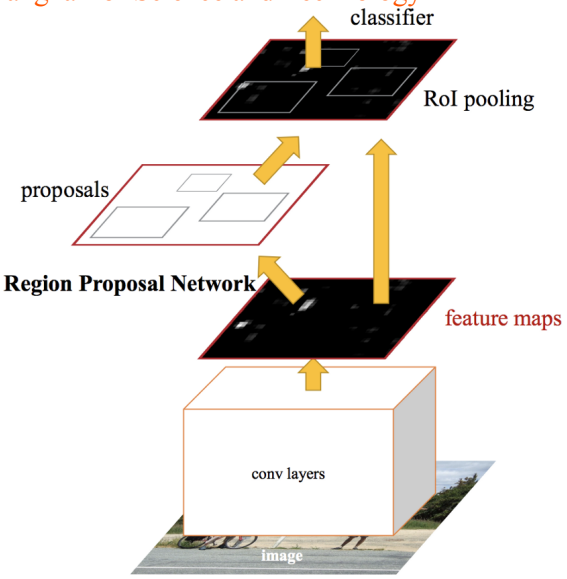


**Figure 7 : Faster R-CNN [1]**

The image is fed into a convolutional network, which outputs a convolutional feature map, similar to Fast R-CNN.A separate network is used to predict the region proposals instead of using a selective search algorithm on the function map to find the region proposals. The predicted region proposals are reshaped using the ROI pooling layer and the further image is classified within the proposed region and predicts the offset values for the bounding boxes.[1]

Faster R-CNN is clearly faster than its predecessors, as seen in the graph below.As a result, it can also be used for real-time object detection.

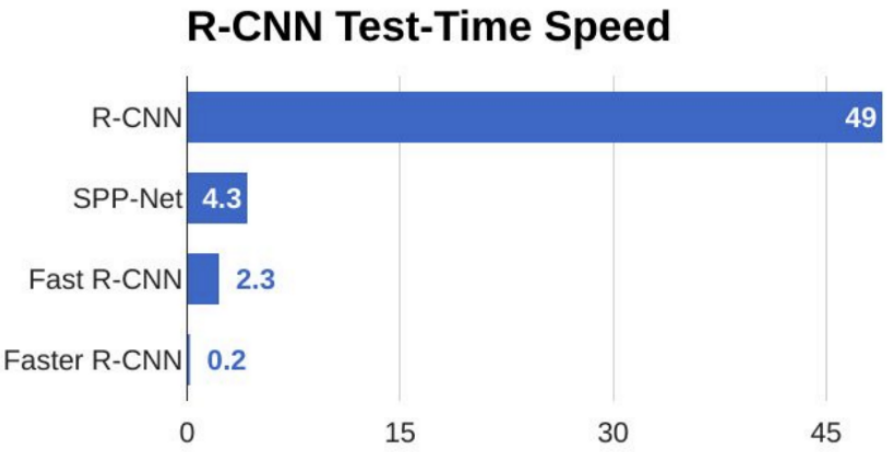


**Figure 8: Comparison of test-time speed of object detection algorithms [3]**

### 4.5 You Only Look Once(YOLO)

YOLO is a real-time object detection algorithm that is much faster and accurate than all other models. All other algorithms excluding YOLO use multiple parts of the image which have a good probability of containing the object to detect objects. They do not take a complete view of the image. On the Contrary, YOLO uses only a single neural network which will then divide the image into various regions to find bounding boxes and probabilities of the image. This saves a lot of time as thousands of neural networks required by all other models are reduced to just one neural network in YOLO which can make the algorithm up to 100 times faster than all other algorithms. YOLO has the capability of processing images at the rate of 45 frames per second.[2]

YOLO divides the image into SxS grids. For each grid formed we take m bounding boxes. Then the bounding boxes are processed by the network giving us output as class probability and offset values for each bounding box. The bounding boxes which have probability above the baseline are then selected for image detection. The problem with this approach is that this approach will contend resolutely with small objects as they have very less detectable features. [2]

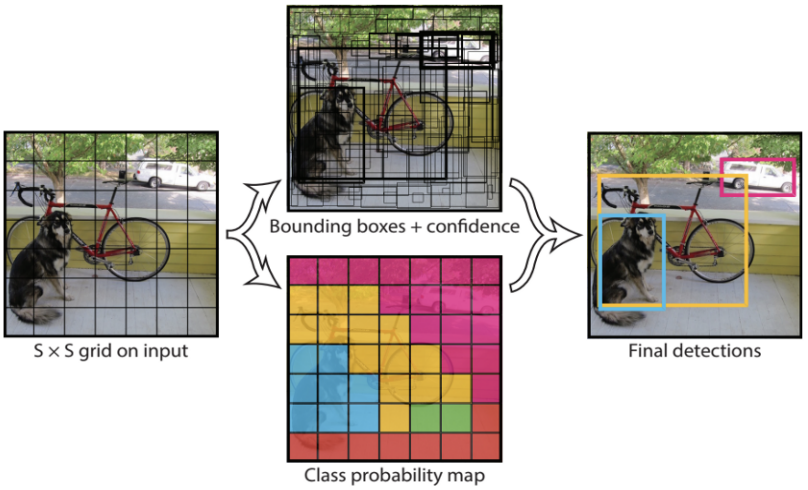


**Figure 9: Working of YOLO [2]**

B bounding boxes and confidence scores for these boxes are predicted in each grid cell. These confidence scores reflect the model's assumption that the box contains an object as well as the accuracy with which it assumes the box it predicts is. Confidence is defined as: Pr(Object) *IOUtruth pred in formal terms. The confidence scores should be zero if no entity exists in that cell. Otherwise, the confidence score should be equal to the intersection over union (IOU) between the predicted and ground truth boxes.

There are five predictions in each bounding box: x, y, w, h, and confidence. The (x, y) coordinates represent the box's centre in relation to the grid cell's borders. The width and height are calculated in terms of the entire image. Finally, the IOU between the expected box and any ground truth box is expressed by the confidence prediction.

Each grid cell also predicts:

$$Pr(Class_i|Object) \cdot Pr(Object) \cdot IOU(truth, pred) = Pr(Class_i) \cdot IOU(truth, pred)$$

This gives us confidence scores for each box depending on the class. These scores reflect the likelihood of that class appearing in the box as well as the accuracy with which the expected box matches the item.

## 5. RESULTS AND DISCUSSION

### 5.1 Comparison

It is quite difficult to find the comparison between various object detection algorithms.In real life, we can use speed and accuracy to compare them. There are numerous other factors that impact their performance right from feature extractors, image resolution to training sets, platforms used, etc.

Performance Results :

1. Faster R-CNN:

The results of the PASCAL VOC 2012 test set are shown below. The last three rows, which reflect the Faster R-CNN output, are what we're interested in.The number of RoIs made by the region proposal network is shown in the second column.The training dataset is described in the third column.In measuring accuracy, the fourth column is the mean average precision (mAP).

#### 5.1.1 Results on PASCAL VOC 2012 test set:

| | training data | mAP (%) | test time (sec/img) |
|---|---|---|---|
| Faster R-CNN [9] | 07++12 | 73.8 | 0.42 |
| Faster R-CNN +++ [9] | 07++12+COCO | **83.8** | 3.36 |
| **R-FCN** multi-sc train | 07++12 | 77.6† | 0.17 |
| **R-FCN** multi-sc train | 07++12+COCO | **82.0**‡ | 0.17 |

**Figure 10: VOC 2012 for Faster R-CNN.[1]**

| method | proposals | training data | COCO val | | COCO test-dev | |
|---|---|---|---|---|---|---|
| | | | mAP@.5 | mAP@[.5, .95] | mAP@.5 | mAP@[.5, .95] |
| Fast R-CNN [2] | SS, 2000 | COCO train | - | - | 35.9 | 19.7 |
| Fast R-CNN [impl. in this paper] | SS, 2000 | COCO train | 38.6 | 18.9 | 39.3 | 19.3 |
| Faster R-CNN | RPN, 300 | COCO train | 41.5 | 21.2 | 42.1 | 21.5 |
| Faster R-CNN | RPN, 300 | COCO trainval | - | - | **42.7** | **21.9** |

**Figure 11: COCO for Faster R-CNN**

| model | system | conv | proposal | region-wise | total | rate |
|---|---|---|---|---|---|---|
| VGG | SS + Fast R-CNN | 146 | 1510 | 174 | 1830 | 0.5 fps |
| VGG | RPN + Fast R-CNN | 141 | **10** | 47 | **198** | **5 fps** |
| ZF | RPN + Fast R-CNN | 31 | **3** | 25 | **59** | **17 fps** |

**Figure 12: Timing on a K40 GPU in millisecond with PASCAL VOC 2007 test set.[1]**

### 5.1.2 Fast-RCNN

| | training data | mAP (%) | test time (sec/img) |
|---|---|---|---|
| Faster R-CNN [9] | 07++12 | 73.8 | 0.42 |
| Faster R-CNN +++ [9] | 07++12+COCO | **83.8** | 3.36 |
| **R-FCN** multi-sc train | 07++12 | 77.6† | 0.17 |
| **R-FCN** multi-sc train | 07++12+COCO | **82.0**‡ | 0.17 |

**Figure 13: VOC 2012 for R-FCN [8]**

| | training data | test data | AP@0.5 | AP | AP small | AP medium | AP large | test time (sec/img) |
|---|---|---|---|---|---|---|---|---|
| Faster R-CNN [9] | train | val | 48.4 | 27.2 | 6.6 | 28.6 | 45.0 | 0.42 |
| **R-FCN** | train | val | 48.9 | 27.6 | 8.9 | 30.5 | 42.0 | 0.17 |
| **R-FCN** multi-sc train | train | val | 49.1 | 27.8 | 8.8 | 30.8 | 42.2 | 0.17 |
| Faster R-CNN +++ [9] | trainval | test-dev | **55.7** | **34.9** | 15.6 | 38.7 | 50.9 | 3.36 |
| **R-FCN** | trainval | test-dev | 51.5 | 29.2 | 10.3 | 32.4 | 43.3 | 0.17 |
| **R-FCN** multi-sc train | trainval | test-dev | 51.9 | 29.9 | 10.8 | 32.8 | 45.0 | 0.17 |
| **R-FCN** multi-sc train, test | trainval | test-dev | **53.2** | **31.5** | 14.3 | 35.5 | 44.2 | 1.00 |

**Figure 14: COCO for R-FCN [8]**

### 5.1.3 YOLO

| Detection Frameworks | Train | mAP | FPS |
|---|---|---|---|
| Fast R-CNN [5] | 2007+2012 | 70.0 | 0.5 |
| Faster R-CNN VGG-16[15] | 2007+2012 | 73.2 | 7 |
| Faster R-CNN ResNet[6] | 2007+2012 | 76.4 | 5 |
| YOLO [14] | 2007+2012 | 63.4 | 45 |
| SSD300 [11] | 2007+2012 | 74.3 | 46 |
| SSD500 [11] | 2007+2012 | 76.8 | 19 |
| YOLOv2 288 × 288 | 2007+2012 | 69.0 | 91 |
| YOLOv2 352 × 352 | 2007+2012 | 73.7 | 81 |
| YOLOv2 416 × 416 | 2007+2012 | 76.8 | 67 |
| YOLOv2 480 × 480 | 2007+2012 | 77.8 | 59 |
| YOLOv2 544 × 544 | 2007+2012 | **78.6** | 40 |

**Figure 15: VOC 2007 for YOLOv2 [9]**

| Method | data | mAP | aero | bike | bird | boat | bottle | bus | car | cat | chair | cow | table | dog | horse | mbike | person | plant | sheep | sofa | train | tv |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Fast R-CNN [5] | 07++12 | 68.4 | 82.3 | 78.4 | 70.8 | 52.3 | 38.7 | 77.8 | 71.6 | 89.3 | 44.2 | 73.0 | 55.0 | 87.5 | 80.5 | 80.8 | 72.0 | 35.1 | 68.3 | 65.7 | 80.4 | 64.2 |
| Faster R-CNN [15] | 07++12 | 70.4 | 84.9 | 79.8 | 74.3 | 53.9 | 49.8 | 77.5 | 75.9 | 88.5 | 45.6 | 77.1 | 55.3 | 86.9 | 81.7 | 80.9 | 79.6 | 40.1 | 72.6 | 60.9 | 81.2 | 61.5 |
| YOLO [14] | 07++12 | 57.9 | 77.0 | 67.2 | 57.7 | 38.3 | 22.7 | 68.3 | 55.9 | 81.4 | 36.2 | 60.8 | 48.5 | 77.2 | 72.3 | 71.3 | 63.5 | 28.9 | 52.2 | 54.8 | 73.9 | 50.8 |
| SSD300 [11] | 07++12 | 72.4 | 85.6 | 80.1 | 70.5 | 57.6 | 46.2 | 79.4 | 76.1 | 89.2 | 53.0 | 77.0 | 60.8 | 87.0 | 83.1 | 82.3 | 79.4 | 45.9 | 75.9 | 69.5 | 81.9 | 67.5 |
| SSD512 [11] | 07++12 | 74.9 | 87.4 | 82.3 | 75.8 | 59.0 | 52.6 | 81.7 | 81.5 | 90.0 | 55.4 | 79.0 | 59.8 | 88.4 | 84.3 | 84.7 | 83.3 | 50.2 | 78.0 | 66.3 | 86.3 | 72.0 |
| ResNet [6] | 07++12 | 73.8 | 86.5 | 81.6 | 77.2 | 58.0 | 51.0 | 78.6 | 76.6 | 93.2 | 48.6 | 80.4 | 59.0 | 92.1 | 85.3 | 84.8 | 80.7 | 48.1 | 77.3 | 66.5 | 84.7 | 65.6 |
| YOLOv2 544 | 07++12 | 73.4 | 86.3 | 82.0 | 74.8 | 59.2 | 51.8 | 79.8 | 76.5 | 90.6 | 52.1 | 78.2 | 58.5 | 89.3 | 82.5 | 83.4 | 81.3 | 49.1 | 77.2 | 62.4 | 83.8 | 68.7 |

**Figure 16: VOC 2012 for YOLOv2 [9]**

| | backbone | AP | $AP_{50}$ | $AP_{75}$ | $AP_S$ | $AP_M$ | $AP_L$ |
|---|---|---|---|---|---|---|---|
| *Two-stage methods* | | | | | | | |
| Faster R-CNN+++ [16] | ResNet-101-C4 | 34.9 | 55.7 | 37.4 | 15.6 | 38.7 | 50.9 |
| Faster R-CNN w FPN [20] | ResNet-101-FPN | 36.2 | 59.1 | 39.0 | 18.2 | 39.0 | 48.2 |
| Faster R-CNN by G-RMI [17] | Inception-ResNet-v2 [34] | 34.7 | 55.5 | 36.7 | 13.5 | 38.1 | 52.0 |
| Faster R-CNN w TDM [32] | Inception-ResNet-v2-TDM | 36.8 | 57.7 | 39.2 | 16.2 | 39.8 | **52.1** |
| *One-stage methods* | | | | | | | |
| YOLOv2 [27] | DarkNet-19 [27] | 21.6 | 44.0 | 19.2 | 5.0 | 22.4 | 35.5 |
| SSD513 [22, 9] | ResNet-101-SSD | 31.2 | 50.4 | 33.3 | 10.2 | 34.5 | 49.8 |
| DSSD513 [9] | ResNet-101-DSSD | 33.2 | 53.3 | 35.2 | 13.0 | 35.4 | 51.1 |
| **RetinaNet** (ours) | ResNet-101-FPN | 39.1 | 59.1 | 42.3 | 21.8 | 42.7 | 50.2 |
| **RetinaNet** (ours) | ResNeXt-101-FPN | **40.8** | **61.1** | **44.1** | **24.1** | **44.2** | 51.2 |

**Figure 17: COCO for YOLOv2 [9]**

| | backbone | AP | $AP_{50}$ | $AP_{75}$ | $AP_S$ | $AP_M$ | $AP_L$ |
|---|---|---|---|---|---|---|---|
| *Two-stage methods* | | | | | | | |
| Faster R-CNN+++ [3] | ResNet-101-C4 | 34.9 | 55.7 | 37.4 | 15.6 | 38.7 | 50.9 |
| Faster R-CNN w FPN [6] | ResNet-101-FPN | 36.2 | 59.1 | 39.0 | 18.2 | 39.0 | 48.2 |
| Faster R-CNN by G-RMI [4] | Inception-ResNet-v2 [19] | 34.7 | 55.5 | 36.7 | 13.5 | 38.1 | 52.0 |
| Faster R-CNN w TDM [18] | Inception-ResNet-v2-TDM | 36.8 | 57.7 | 39.2 | 16.2 | 39.8 | **52.1** |
| *One-stage methods* | | | | | | | |
| YOLOv2 [13] | DarkNet-19 [13] | 21.6 | 44.0 | 19.2 | 5.0 | 22.4 | 35.5 |
| SSD513 [9, 2] | ResNet-101-SSD | 31.2 | 50.4 | 33.3 | 10.2 | 34.5 | 49.8 |
| DSSD513 [2] | ResNet-101-DSSD | 33.2 | 53.3 | 35.2 | 13.0 | 35.4 | 51.1 |
| RetinaNet [7] | ResNet-101-FPN | 39.1 | 59.1 | 42.3 | 21.8 | 42.7 | 50.2 |
| RetinaNet [7] | ResNeXt-101-FPN | **40.8** | **61.1** | **44.1** | **24.1** | **44.2** | 51.2 |
| YOLOv3 608 × 608 | Darknet-53 | 33.0 | 57.9 | 34.4 | 18.3 | 35.4 | 41.9 |

**Figure 18: COCO for YOLOv3 [10]**

### 5.2 Conclusion from Comparisons

Although the R-FCN model is faster on average, it cannot match the precision of the Faster R-CNN if speed is not a factor.R-CNN that is faster needs at least 100 milliseconds per image. Detection accuracy is negatively impacted when only low-resolution feature maps are used.The resolution of the input image has a huge effect on accuracy. When imagesize is reduced by half in width and height, accuracy is reduced by 15.88 percent on average, but inference time is reduced by 27.4 percent on average.If only one IoU is used to measure mAP, use mAP@IoU=0.75 .

### 5.3 Application in Licence Plate Recognition

YOLO (You Only Look Once) algorithm is used to detect vehicles and the number plate respectively from the videos and images.As compared to other algorithms, YOLO algo-rithms operate much faster. Because of its algorithm and the way it is trained, YOLOis most likely a highly generalised network. It processes each frame at a rate ranging from 45 frames per second (fps) for a larger network to 150 frames per second (fps) for a smaller network, which is better than real-time. Hence, YOLO improves detection efficiency by training on complete images.

### 5.4 Conclusions

We proposed a device in this project that would detect when a vehicle entered the premises.The number plate will then be detected, the text will be retrieved, and the corresponding owner information will be obtained from the database. To detect vehicles and licence plates, the YOLO V3 is used. As a result, putting this device in place has the potential to alter the current trend and aid protection in a variety of ways.

## ACKNOWLEDGMENT

We would like to extend our special thanks to Prof. Dr. Archana Chaugule, Head, Department of Computer Engineering, and Prof. Jameer Kotwal,, Department of Computer Engineering, Pimpri Chinchwad College of Engineering and Research for their encouragement and useful critiques for this research work. We would also like to thank our internal guide, Prof. Nilesh Korade for guiding us through this project.

## REFERENCES

*[1] Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun*

*[2] You Only Look Once:Unified, Real-Time Object Detection Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi University of Washington, Allen Institute for AI, Facebook AI Research*

*[3]http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture11.pdf*

*[4] A. K. Alexander Mordvintsev, "Face detection using haar cascades,"2013. [Online].*

*[5] T.datascience,"What'sthedifferencebetweenhaar-featureclassifiers and convolutional neural networks?"2018.[Online].*

*[6] Z. Yu and C. Zhang, "Image based static facial expression recognition with multiple deep network learning,"Proceedings of the 2015 ACM on International conference on Multimodal Interaction - ICMI 15, 2015.*

*[7] C. Shan, A comprehensive study,"Image and VisionComputing, vol. 27, no. 6, p. 803–816, 2009*

*[8] R-FCN: Object Detection via Region-based Fully Convolutional Networks Jifeng Dai Microsoft Research, Yi Li Tsinghua University, Kaiming He Microsoft Research, Jian Sun Microsoft Research*

*[9] YOLO9000:Better, Faster, Stronger Joseph Redmon, Ali Farhadi University of Washington, Allen Institute for AI*

*[10] YOLOv3: An Incremental Improvement Joseph Redmon, Ali Farhadi University of Washington*

*[11] Sharifara, M. S. M. Rahim, and Y. Anisi, "A general review of human face detection including a study of neural networks and haar feature-based cascade classifier in face detection,"2014 International Symposium onBiometrics and Security Technologies (ISBAST), 2014.*

*[12] Rich feature hierarchies for accurate object detection and semantic segmentationTech report (v5) Ross Girshick Jeff Donahue Trevor Darrell Jitendra Malik UC Berkeley*

*[13] Fast R-CNN Ross Girshick Microsoft Research*

*[14] M. Nielsen,Neural Networks and Deep Learning.[5] R. K. Samer Hijazi and C. Rowen, "Using convolutional neural networks for image recognition," 2015.*

*[15]S.University,"Introduction To Convolutional Neural Networks,"2018*