OBJECT DETECTION AND CLASSIFICATION USING FASTER RCNN,FAST RCNN

Project submitted to the

SRM University – AP, Andhra Pradesh

for the partial fulfillment of the requirements to award the degree of

Bachelor of Technology

In

Computer Science and Engineering

School of Engineering and Sciences

Submitted by

U.Venkata Naga Sai Abhinav(AP21110010915)
K.Ganesh Sesha Sai Akhil(AP21110010931)
Md.Ameen Naimuddin(AP21110010948)
D.Sai Amruth(AP21110010962)
P.Venkata Arun Kumar(AP21110010970)

Under the Guidance of

Dr.Shuvendu Rana

SRM University-AP

Neerukonda, Mangalagiri, Guntur

Andhra Pradesh - 522 240

December 2023

Certificate

Date: 01-Dec-23

This is to certify that the work present in this Project entitled "**OBJECT DETECTION USING FASTER RCNN,FAST RCNN**" has been carried out by

U.Venkata Naga Sai Abhinav-AP21110010915

under my supervision. The work is genuine, original, and suitable for submission to the SRM University – AP for the award of Bachelor of Technology in **School of Engineering and Sciences**.

Supervisor

(Signature)

Dr.Shuvendu Rana,

SRM University-AP.

Acknowledgement

This undergraduate research program focuses on "Object detection and classification using Fast Rcnn and Faster Rcnn". It was one of the most enlightening and exciting projects we have ever worked on. It was a fantastic learning experience in which we learned a lot and tried out new skills. This research initiative supplied theoretical as well as practical information. This project is an honest attempt to provide whatever we have learned as useful experience that will undoubtedly help us advance up the learning curve towards the route we have selected.

We would like to express our gratitude to Dr.Shuvendu Rana, our mentor for the final project, for providing proper guidance to complete our research successfully. We are thankful to the people who encouraged us directly or indirectly in the completion of this project.

Table of Contents

Abstract

In order to identify, classify, and localize objects in pictures as a unique solution to Computer Vision Real World Domain specific challenges, the generalized object detection framework Faster R-CNN is based on a CNN. Unlike previous object identification algorithms that perceive it as a classification problem, it views object identification as a single regression problem. Due to its efficiency in simultaneously detecting, classifying, and localizing many items from various classes, this complex object detection method has gained popularity. The goal of this project is to build a dataset with various picture formats to implement Faster R-CNN for object detection, classification, and localization. The chosen approach would include detecting, classifying, and localizing persons and objects on the road using a pretrained CNN. This has the potential to be used in active safety systems for driverless cars as well. Additionally, it can be implemented differently to assist with data collection and categorization.

Keywords: - R-CNN, OBJECT DETECTION AND RECOGNITION

Abbreviations

CNN	Convolutional Neural Network
ROI	Region Of interest
RPN	Region Proposal Network
SSD	Single Shot Multibox Detection
IOU	Intersection Over union
GPU	Graphics Processing Unit
NMS	Non - Maximum suppression

List of Figures

1. Introduction

The problem of object detection in the area of computer vision requires detecting and locating certain items within an image or video. It has several real-world uses, including autonomous driving, security systems, and image analysis. Deep learning has significantly improved accuracy and performance in the field of object detection during the past several years.

Convolutional neural networks is frequently the foundation of deep learning models for image recognition because they are excellent at extracting valuable characteristics from photos. These models study the structures and patterns seen in a sizable collection of labelled photos to learn to recognise items.

The region based cnn family of deep learning models is the most important and commonly used deep learning model for object identification. The two primary steps of R-CNN models are object categorization and region proposal. During the region proposal stage, region proposals—also referred to as possible object regions—are created using selective search or other techniques. In order to extract characteristics, these area suggestions are then input into a CNN. The characteristics are used to categorise the items that are present in the region suggestions at the end of the object classification step.

The enhanced accuracy and efficiency of the RCNN family are demonstrated by the algorithms Fast RCNN, Faster RCNN, and Mask RCNN. By streamlining the processes of feature extraction and region proposal generation, these models improve speed and accuracy.For example, faster RCNN creates an integrated model that is faster and more efficient by using a rpn that shares convolutional layers with the object classification network.

The singleshot multibox detector is yet another interesting image identification technique. SSD models combine the object classification and area proposal phases into a single network, eliminating the need for separate proposal creation. Consequently, inference times are reduced without sacrificing accuracy.

Commonly utilised big annotated datasets for object detection training include COCO (Common Objects in Context) and Pascal VOC. These datasets include a wide variety of photos with bounding box annotations for different items, allowing the models to develop excellent object detection and classification skills.

By consumers and experts alike, the terms "object recognition" and "object detection" are frequently used interchangeably. It might be difficult for newcomers to the field of computer vision to tell apart many related activities.

We may use an example to illustrate the differences between computer vision tasks like object detection and object recognition.

Image Classification: Making assumptions about the kind or class of an object depicted in a picture allows for this.
Input:a picture that just shows one thing, like a snapshot of only one thing

Output: A class label

Object Detection: This is done by locating items in a picture, drawing bounding boxes around them, and figuring out what kinds or classes of objects they belong to.
Input: a picture that includes one or more objects, such as a snapshot of several objects.

Output: a number of bounding boxes, each with a class label associated with it.

Drawing bounding boxes around individual items in an image is the process of object localisation, whereas a complete picture is classified. These two tasks are combined in object detection, which is more difficult since it calls for both categorising and localising objects using bounding boxes. All these activities are together refered as "object recognition".

Object recognition refers to the technique of recognizing things in digital pictures as a whole. Object localization and recognition problems were addressed through the development of a class of techniques known as region-based convolutional neural networks (R-CNNs). In particular, the Faster R-CNN method is renowned for its effective performance.

Faster R-CNN is a two-stage methodology that involves region proposal creation followed by object categorization and refining to achieve better accuracy. Its approach makes it well-suited for scenarios where high accuracy is crucial. While YOLO is recognized for its real-time performance, Faster R-CNN provides a balance between accuracy and speed.

In conclusion, deep learning-based object detection, particularly through Faster R-CNN, has developed into a crucial and effective method in computer vision. These advancements enable very precise object recognition in a variety of contexts, significantly advancing areas like autonomous driving, surveillance, and visual comprehension.

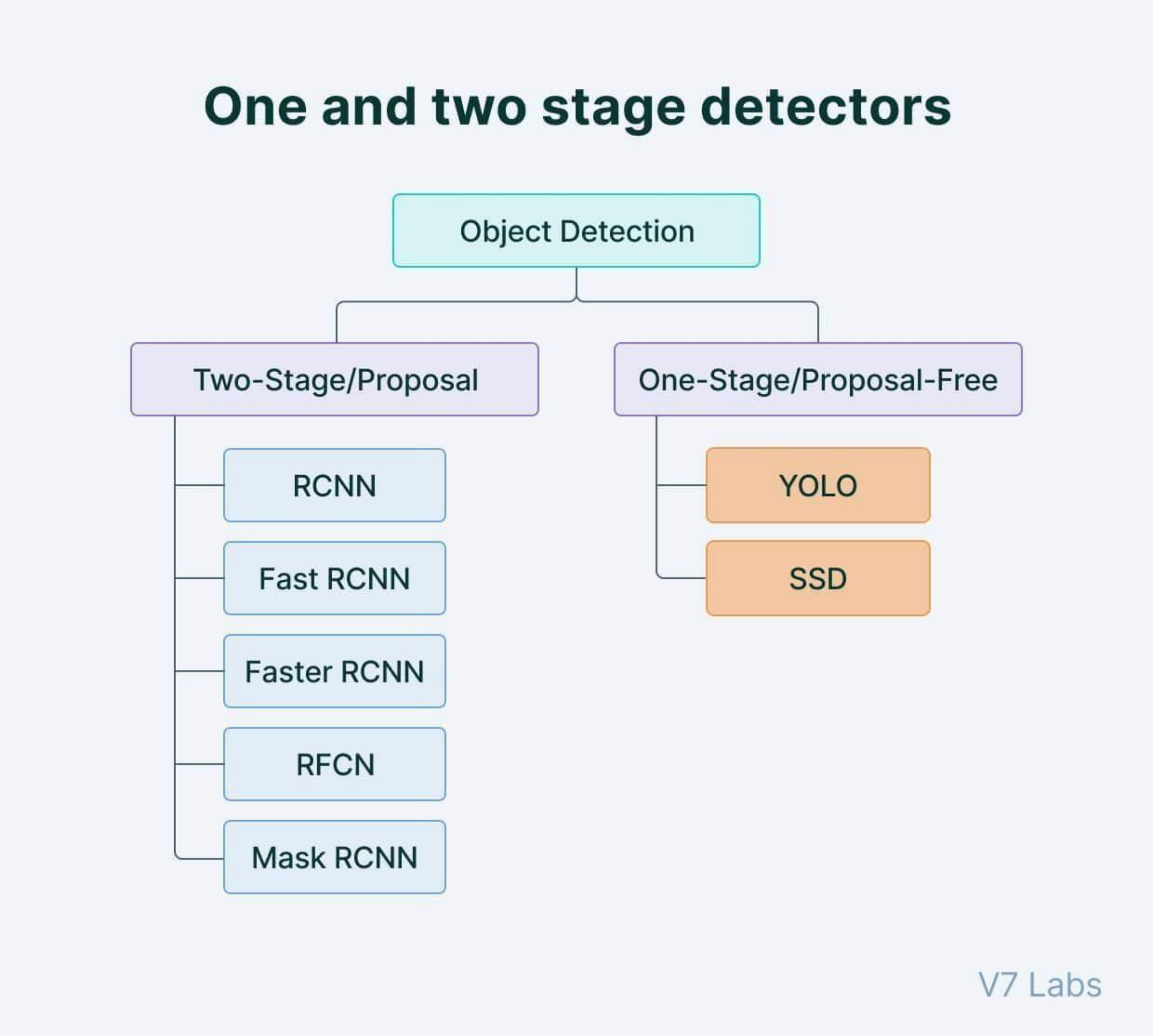


Figure 1 : Categories of Object detection

2. Methodology

2.1 CNN: Convolutional neural networks are DL algorithms that are capable of feature detection on images that are given as input.CNNs can assign weights to different process characteristics. Contrary to manual instruction required for fundamental algorithms, CNNs are capable of learning filters and features on their own. As a result, less preprocessing is required.

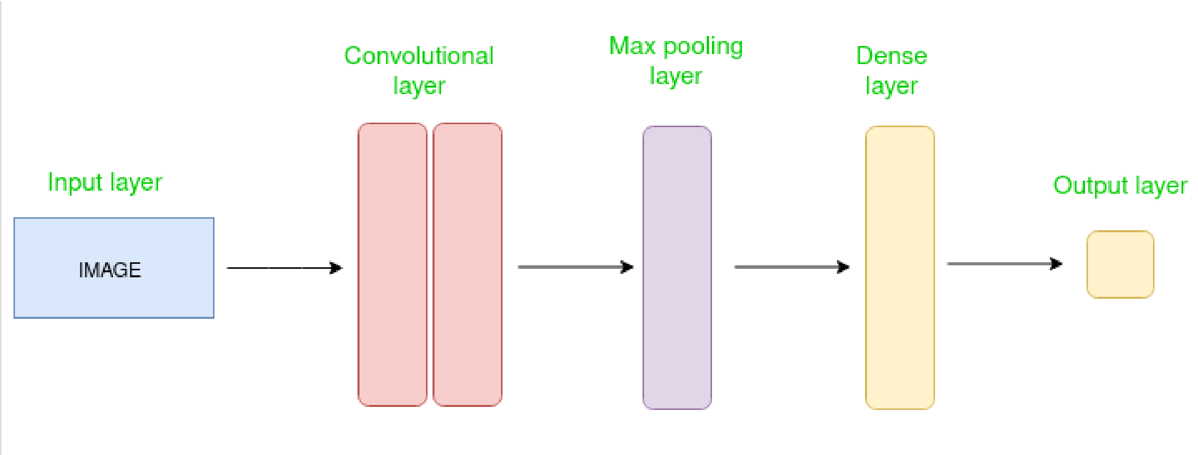


Figure 2 : Basic CNN Architecture

Convolutional layers, pooling layers, and fully linked layers are all crucial parts of the ConvNet's basic design, which is shown in Figure 2

Given that CNNs are essentially neural networks, the organisation of the neurons in the brain's visual cortex affects how information is sequenced. Unique neurons have responses that are dependent on the receptive field in which they function and are activated. The whole region of the visual cortex is made up of these neurons, which are found in bundles. Convolutional layers—the number of layers depends on how deep the network is—make up the majority of CNNs. ii. Activation layers are employed to keep the math correct and avoid achieving improbable results or random excitations. Common activation layers are ReLU , Leaky ReLU. Pooling layers are to minimise tensor size.

2.2 FASTER R-CNN:

Faster RCNN is an improved version of Fast RCNN that uses a rpn to produce ROIs. This approach employs feature maps as its input. The result is a score for abasement.

The CNN is fed an input picture by the Faster R-CNN, which then uses it to create an image's feature map.

On those maps, RPN is applied, and the abjectness score and object recommendation are returned.

Over these object maps and 'k', the RPN employs sliding windows. Different-sized anchor boxes are created. Each anchor box is projected to do two things:

1. The probability of the anchor holding objects (it doesn’t classify the objects at this step)

2. A bounding box regressor to better adjust the anchor to fit the objects better inside the box.

The suggestions are then brought in and each one is cropped to include an object. To reduce all of these proposals to the same size, a ROI pool layer is used. For each anchor, ROI pooling layers extract a fixed size pooling map.

The completely linked layer that contains the "SoftMax" layer receives the shrunk suggestions at the end. To categorise and output the bounding boxes, there is a layer of linear regression on top.

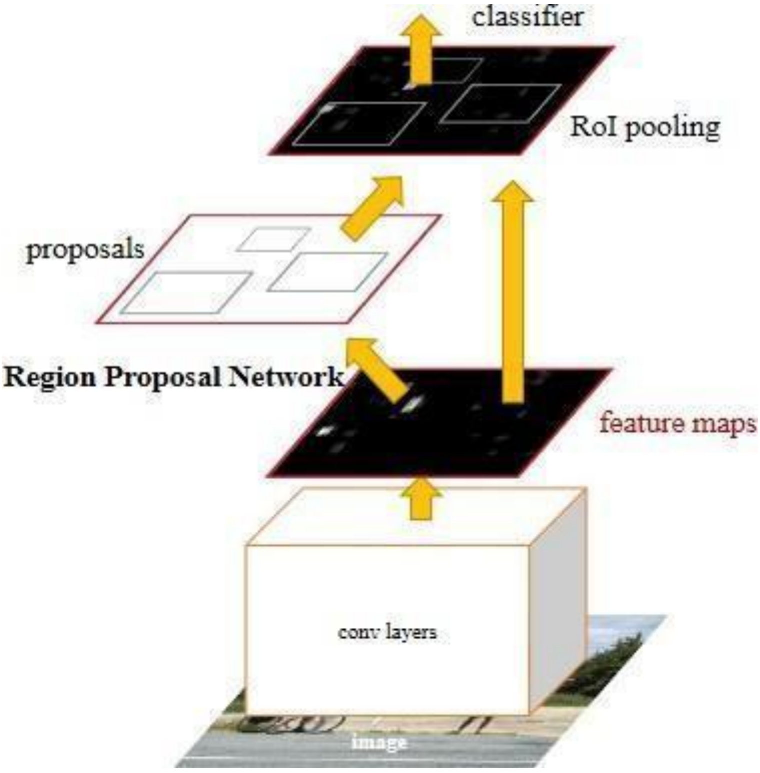


Figure 3 : Architecture of Faster RCNN

A two-stage object identification model using deep learning is called the Faster R- CNN. Regions of interest (ROIs) within a picture are found in the initial step. After that, a convolutional neural network (CNN) receives these ROIs. A support vector machine (SVM) is then used to further process the generated feature maps for categorization. Regression is also used to determine how anticipated bounding boxes and ground truth bounding boxes relate to one another. This general architecture summarizes the overall approach of the Faster R-CNN.

2.3 Fast RCNN :

A model for object detection called Fast R-CNN (Region-based Convolutional Neural Network) expands on the original R-CNN strategy and enhances both accuracy and speed. By making numerous significant changes to the architecture and training procedure, fast R-CNN overcomes some of the drawbacks of R-CNN, including its sluggish training and inference times.

The major elements and characteristics of Fast R-CNN are listed below:

1. **Region Proposal:** Like R-CNN, Fast R-CNN starts by generating region recommendations in order to identify likely object regions in the input image. Rather than using selective search or a similar method to generate a large number of region ideas, Fast R-CNN uses a single region proposal network (RPN) to generate fewer high-quality region proposals.
2. **CNN Backbone:** Fast R-CNN uses a convolutional neural network (CNN) as its base to extract features from the entire input image. The CNN backbone is typically pre-trained on a large-scale dataset (like ImageNet), and can be a variant of VGG, ResNet, or another design.
3. **Region Of Interest(ROI) :** For each region suggestion, Fast R-CNN incorporates RoI pooling to derive fixed-sized feature maps from the CNN backbone. RoI pooling makes it possible to efficiently and consistently extract features from areas of different sizes while preserving the region-level data.
4. **Fully Connected layers:** In a sequence of fully connected layers that conduct object classification and bounding box regression, the RoI-pooled feature maps are fed. Both the item category and the projected bounding box coordinates are captured by the fully linked layers.
5. **Multi-task loss:** The classification loss and the bounding box regression loss, two independent components, are combined in the multi-task loss function that Fast R-CNN uses. Whereas the classification loss is usually calculated using sigmoid or softmax activation functions, the bounding box regression loss is frequently calculated using regression techniques such as smooth L1 loss.
6. **Training:** The two phases of quick R-CNN training are pre-training and fine-tuning. A large dataset is used by the pre-trained CNN backbone to help it learn general characteristics. Next, using backpropagation on the target dataset, the fully connected layers and the region proposal network (RPN) are jointly adjusted.
7. **Interface:** Fast R-CNN runs the full picture through the CNN backbone during inference to extract features. Utilising the RPN, region proposals are created,

and each region proposal's features are extracted via RoI pooling. The fully connected layers then get the region characteristics to forecast class probabilities and improve the bounding box coordinates. The NMS technique is used to get rid of overlapping and redundant detections.

Fast R-CNN improves R-CNN by reducing feature computation redundancy and speeding up both training and inference by sharing the convolutional features among region proposals. It also addresses the problem of localization accuracy by introducing ROI pooling and simultaneously optimizing the classification and bounding box regression tasks.

An overview of the Fast R-CNN (Region-based Convolutional Neural Network) object identification technique is given below:

1. **Input:** The input for the Fast R-CNN algorithm is a picture.
2. **CNN Backbone:** A convolutional neural network (CNN) backbone, such as VGG, ResNet, or another design, processes the input picture. From the complete image, the CNN backbone retrieves hierarchical characteristics.
3. **Region Proposal:** Based on the characteristics retrieved by the CNN backbone, a region proposal network (RPN) creates probable object regions, or region proposals. Each area proposal's possibility of having an item is predicted by the RPN.
4. **RoI Pooling:** For each proposed region, RoI (Region of Interest) pooling is used to align and retrieve fixed-sized feature maps from the CNN backbone. Regardless of the size of the region, RoI pooling makes sure that the characteristics derived from each region proposal have uniform spatial dimensions.
5. **Fully Connected Layers:** The RoI-pooled features are input into a sequence of fully connected layers that carry out the bounding box regression and object classification tasks. The completely linked layers assign each region proposal's objects to one of several categories and forecast precise bounding box coordinates.
6. **Multi-task Loss:** The classification and bounding box regression tasks are concurrently trained using Fast R-CNN using a multi-task loss function. The classification loss and the bounding box regression loss are combined in the loss function. The bounding box regression loss is often estimated using methods like smooth L1 loss, whereas the classification loss is typically computed using softmax or sigmoid activation functions.
7. **Training:** Fast R-CNN is trained during the course of two stages. First, a big dataset, like ImageNet, is used to pre-train the CNN backbone on general characteristics. Then, using backpropagation, the RPN and fully connected layers are adjusted together on the target dataset. The network parameters for

object classification and bounding box regression are optimised throughout the training phase.

8. **Inference:** Fast R-CNN runs the full input picture through the CNN backbone during inference to extract features. Utilising the RPN, region proposals are created, and each region proposal's features are extracted via RoI pooling. The fully connected layers then get the region characteristics to forecast class probabilities and improve the bounding box coordinates. In order to remove overlapping and redundant detections, non-maximum suppression (NMS) is used. This produces the final collection of identified objects, together with the class labels and bounding box coordinates that go with each one.

By adding the RoI pooling layer, which enables more effective and precise feature extraction from area suggestions, Fast R-CNN enhances the original R-CNN method. By doing away with costly per-region calculation, training and inference are accelerated. Additionally, the accuracy of localization is increased by jointly training the classification and bounding box regression tasks.

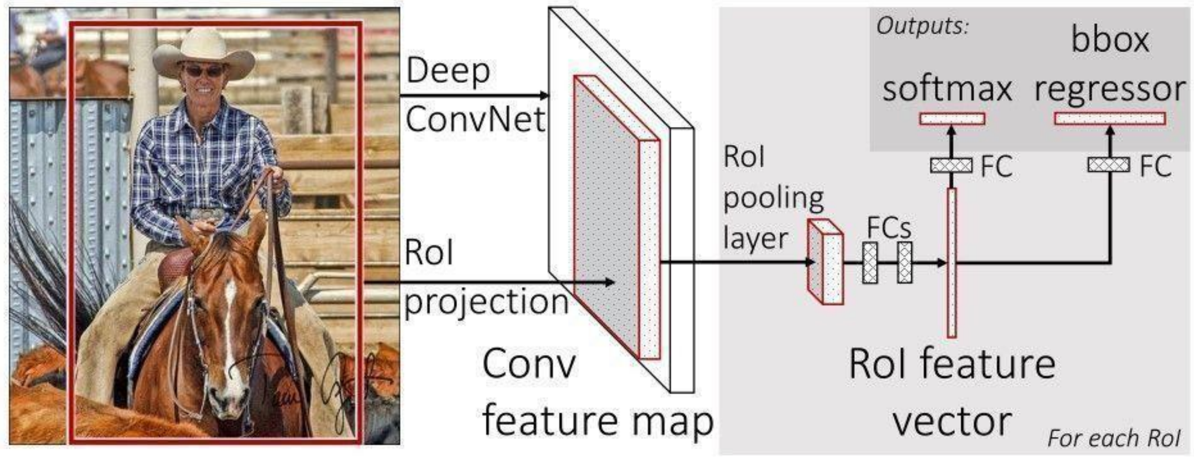
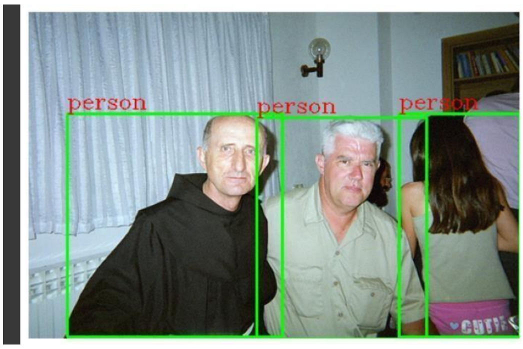


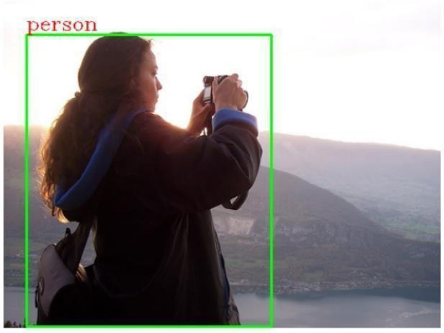
Figure 4 : Architecture of fast Rcnn

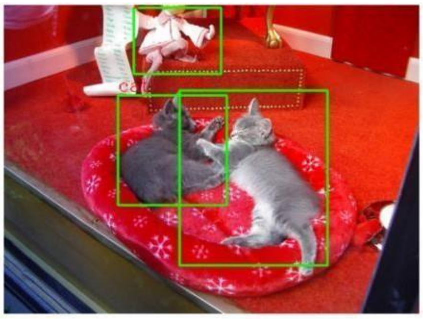
Below is a presentation of Fast R-CNN's general structure. This model employs a single stage, as opposed to the three stages used by R-CNN. It accepts an image as input and outputs the probability for various classes along with the bounding box coordinates for the recognised items.

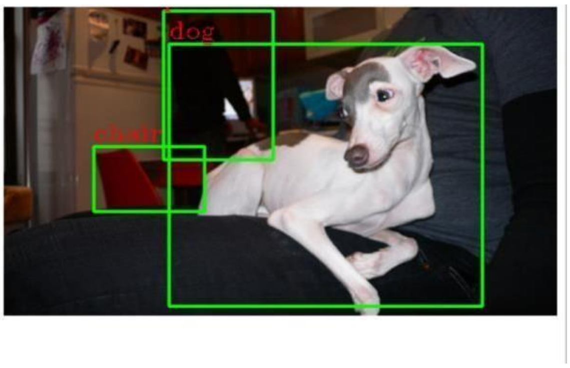
3. RESULTS

In this project, we have implemented FASTER RCNN, and FAST RCNN algorithms, and to train these algorithms, we have the PASCAL VOC2012 dataset. We have done this algorithm using jupyter notebook, and below are the results.









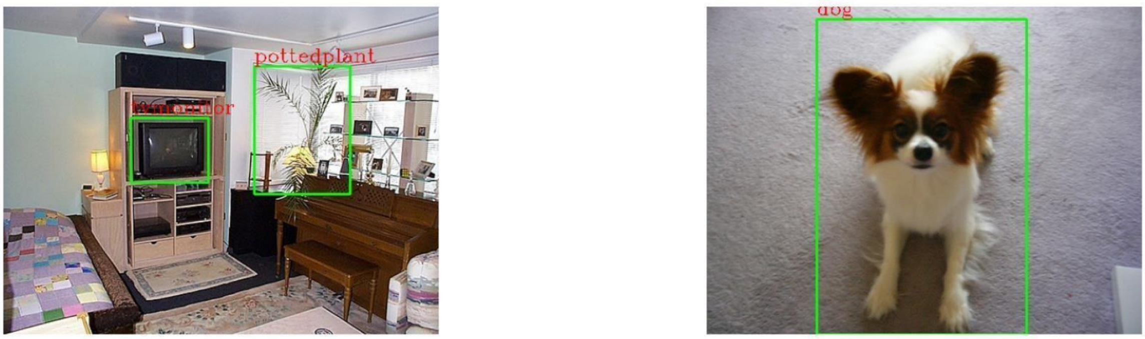


Figure 5 : Faster Rcnn detection outputs

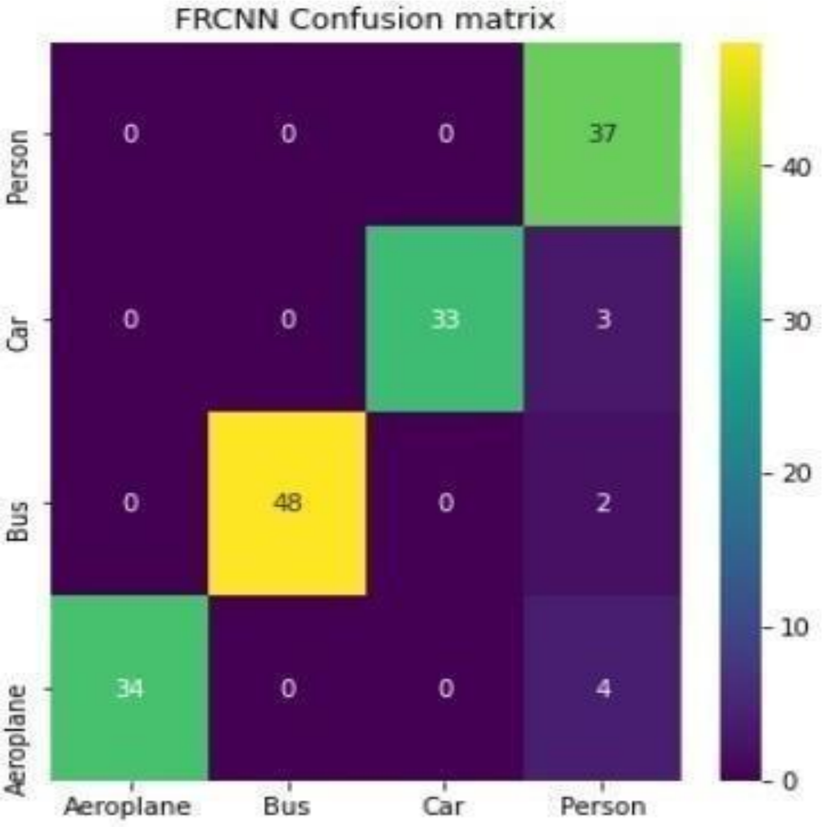
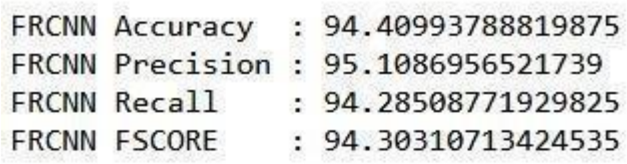


Figure 6 : Confusion Matrix for FRCNN



```
FRCNN Accuracy  : 94.40993788819875
FRCNN Precision : 95.1086956521739
FRCNN Recall    : 94.28508771929825
FRCNN FSCORE    : 94.30310713424535
```

Figure 7 : Accuracy and Precision

About Figure 6 and Figure 7 :

Using the previously described FRCNN, we were able to achieve a high accuracy of 95% in the evaluation. Several metrics, such as accuracy, precision, recall, and F-score, were used to assess the performance. The confusion matrix graph's x-axis displays the expected labels, and the y-axis displays the actual labels. The diagonal boxes in different colors display the accurate forecast counts, while the blue boxes represent the few incorrect guesses. In the ROC graph, the True Positive Rate is displayed by the y-axis, and the False Positive Rate is represented by the the y-axis. True positive predictions are shown by the blue line crossing the orange line; erroneous positive predictions are indicated by the blue line crossing the orange line. Only a few of the predictions on the ROC graph in question were inaccurate or mispredicted.

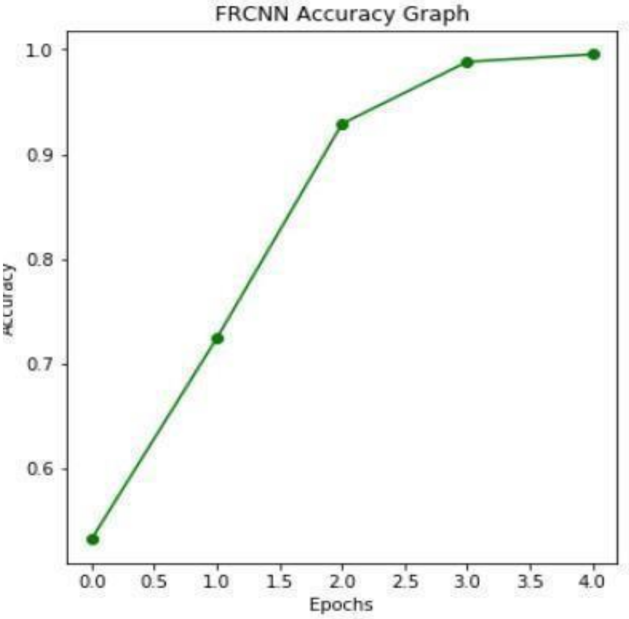


Figure 8 : FRCNN Accuracy Graph

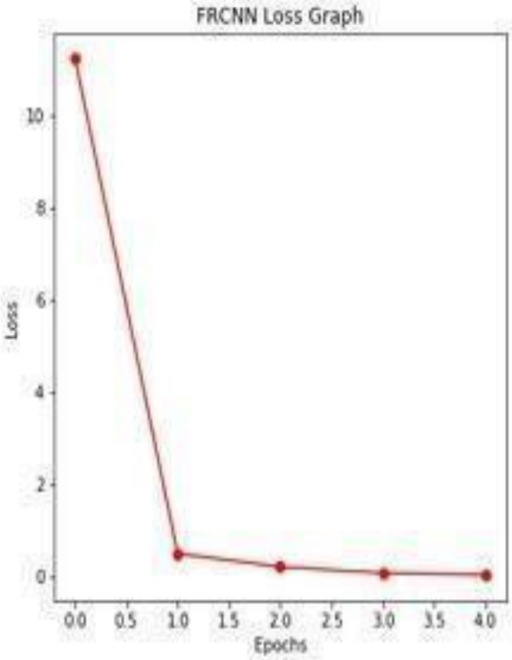


Figure 9 : FRCNN Loss graph

The training accuracy and loss of the FRCNN model are shown in the graphs above. The y-axis displays the appropriate accuracy and loss values, while the x-axis displays the number of training epochs. The graph clearly shows that as the number of epochs rises, the model's accuracy rises and the loss falls.

4. Conclusion

In conclusion, object detection, which includes locating and recognizing items inside digital photos or videos, is a critical computer vision problem.

Faster R-CNN, on the other hand, employs a two-stage methodology to get more accuracy. It initially creates a number of area ideas, which are then categorized and improved using a different region classification network. Although it can be computationally more expensive, this technique often performs better in situations where exact object localization is important.

Faster R-CNN has been a significant contributor to object identification, finding widespread application in various fields. Its strengths lie in object tracking, aerial picture analysis, and medical imaging, owing to its enhanced accuracy. Unlike YOLO, which excels in real-time applications, Faster R-CNN's impact is more pronounced in scenarios where precision is paramount.

Anticipated advancements in Faster R-CNN and related methodologies are likely to center on refining occlusion handling, bolstering adaptability across diverse scenarios, and leveraging contextual cues. These developments aim to further enhance its capabilities in perceiving complex environments. Ultimately, these strides will aid in augmenting how robots interpret and engage with the visual world.

5. Future Work

The goal of future work on object detection utilising YOLO models and Faster R-CNN is to solve current constraints and investigate new directions to further improve performance. The following are some prospective study directions:

1. Improved Accuracy : Despite the outstanding results that YOLO and Faster R- CNN have produced, there is still space for accuracy growth, particularly in difficult situations like recognising tiny or strongly obstructed objects. Future research can concentrate on improving the training approaches and architectures to improve the detection and classification precision.
2. Real time performance improvements : Even though YOLO is renowned for its real-time processing skills, there is always room to improve the speed and effectiveness of the system. To obtain further quicker inference times, future studies may examine architectural improvements, model compression strategies, or hardware acceleration. Similar improvements may be made to Faster R-CNN's computational performance to allow real-time applications on devices with limited resources.
3. Hybrid Architectures : Combining the strengths of YOLO and Faster R-CNN is an area of ongoing research. Future research might look at hybrid architectures that combine the benefits of the two approaches, such as YOLO-like prediction heads integrated into Faster R-CNN or region proposal networks within the YOLO framework, with the goal of improving the trade-off between accuracy and speed.
4. Real-time and video object detection: For real-time applications and video object recognition, YOLOv5 can be optimised. In order to attain even quicker inference times while keeping high accuracy, this entails enhancing its speed and efficiency. To improve object recognition in video sequences, techniques including temporal consistency, motion estimation, and tracking can also be incorporated.
5. Small object detection: It is a huge problem to increase the performance of tiny item identification. By combining multi-scale training, investigating innovative anchor mechanisms, or using contextual information, future work can concentrate on improving the algorithms' capacity to reliably recognise and categorise tiny items.

In terms of accuracy, efficiency, robustness, and application to various object detection settings, these future scopes seek to solve a number of issues and push the limits of YOLOv3, YOLOv5, Faster R-CNN, and Fast R-CNN. making significant contributions to the field of computer vision and advancing the practical applications of these techniques.

6. References

1. R-CNN: Girshick, R., et al. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
2. Fast R-CNN: Girshick, R. (2015). Fast R-CNN. In Proceedings of the IEEE International Conference on Computer Vision (ICCV).
3. Faster R-CNN: Ren, S., et al. (2015). Faster R-CNN: Towards real-time object detection with region proposal networks. In Advances in Neural Information Processing Systems (NeurIPS).
4. Girshick, R.; Donahue, J.; Darrelland, T.; Malik, J. Rich feature hierarchies for object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 24–27 June 2014. [Google Scholar]
5.] S. Ren, K. He, R. Girshick, X. Zhang, and J. Sun. Object detection networks on convolutional feature maps. arXiv:1504.06066, 2015.
6. M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional neural networks. In ECCV,2014
7. K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman. Return of the devil in the details: Delving deep into convolutional nets. In BMVC, 2014.
8. D. Erhan, C. Szegedy, A. Toshev, and D. Anguelov. Scalable object detection using deep neural networks. In CVPR, 2014.
9. T. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick,J. Hays, P. Perona, D. Ramanan, P. Dollar, and C. L. Zit- nick. Microsoft COCO: common objects in context. arXive-prints, arXiv:1405.0312 [cs.CV], 2014.
10. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. arXiv 2015, arXiv:1506.01497.
11. Cai, Z.; Vasconcelos, N. Cascade R-CNN: High quality object detection and instance segmentation. IEEE Trans. Pattern Anal. Mach. Intell. 2019, 43, 1483– 1498.
12. Pinle, Q.; Chuanpeng, L.; Jun, C.; Chai, R. Research on improved algorithm of object detection based on feature pyramid. Multimed. Tools Appl. 2019

13. https://media.geeksforgeeks.org/wp-content/uploads/20210805213534/max.png

14. https://assets-global.website-files.com/5d7b77b063a9066d83e1209c/60d31e388536752a275673aa_machine-learning-infographic.jpg

15. https://media.geeksforgeeks.org/wp-content/uploads/20200219125702/faster-RCNN.png

16. https://media.geeksforgeeks.org/wp-content/uploads/20200219160147/fast-RCNN1.png