A

Project Report

on

# Real-Time Object Detection Using Deep Learning

Submitted in partial fulfillment of the requirements
for the award of the degree of

## Bachelor of Technology

in

## Computer Science and Engineering


by

Nitin Singh 1809710065

Aryan Yadav 1809710027

Ashutosh Prajapati 1809710028

Under the Supervision of

Mr. Ajeet Kumar Bhartee

Assistant Professor



Galgotias College of Engineering & Technology

Greater Noida, Uttar Pradesh

India-201306

Affiliated to




Dr. A.P.J. Abdul Kalam Technical University

Lucknow, Uttar Pradesh,

India-226031


May, 2022

**GALGOTIAS COLLEGE OF ENGINEERING & TECHNOLOGY**
**GREATER NOIDA, UTTAR PRADESH, INDIA- 201306.**

# CERTIFICATE

This is to certify that the project report entitled Real Time Object Detection using Deep Learning submitted by **Mr. Nitin Singh 1809710065, Mr. Aryan Yadav 1809710027, Mr. Ashutosh Prajapati 1809710028** to the Galgotias College of Engineering & Technology, Greater Noida, Uttar Pradesh, affiliated to Dr. A.P.J. Abdul Kalam Technical University Lucknow, Uttar Pradesh in partial fulfillment for the award of Degree of Bachelor of Technology in Computer science & Engineering is a bonafide record of the project work carried out by them under my supervision during the year 2021-2022.

**Mr. Ajeet Kumar Bhartee**
**Assistant Professor**
**Deptt. of CSE**

**Dr. Vishnu Sharma**
**Professor and Head**
**Deptt. of CSE**

# ACKNOWLEDGEMENT

In this initiative, we have put forth effort. However, without the kind support and assistance of many individuals and organizations, it would not have been feasible. All of them deserve our heartfelt gratitude.

We are highly indebted to **Mr. Ajeet Kumar Bhartee** for his guidance and constant supervision. Also, we are highly thankful to them for providing necessary information regarding the project & also for their support in completing the project.

We are extremely indebted to **Dr. Vishnu Sharma**, HOD, Department of Computer Science and Engineering, GCET and **Dr. Jaya Sinha**, Project Coordinator, Department of Computer Science and Engineering, GCET for their valuable suggestions and constant support throughout my project tenure. We would also like to express our sincere thanks to all faculty and staff members of Department of Computer Science and Engineering, GCET for their support in completing this project on time.

We also like to thank our parents for their aid and encouragement with our initiative. Our gratitude and appreciation extends to our project's developers as well as everyone who has volunteered their time and skills to assist me.

Nitin Singh

Aryan Yadav

Ashutosh Prajapati

# ABSTRACT

The field of computer vision known as real-time object detection is big, dynamic, and difficult. Image localization refers to the detection of a single object in an image, while Object Detection refers to the detection of many objects in an image. In digital photos and videos, this recognizes the semantic objects of a class. Tracking objects, video surveillance, pedestrian recognition, people counting, self-driving cars, face detection, ball tracking in sports, and many other applications use real-time object detection. Convolution neural networks is a Deep Learning tool for detecting objects with OpenCV (Opensource Computer Vision), which is a set of programming functions primarily for real-time computer vision.

**KEYWORDS:** Computer vision, Deep Learning, Convolution Neural Networks.

# CONTENTS

# LIST OF TABLES

# LIST OF FIGURES

# ABBREVIATIONS

| | |
|---|---|
| CNN | Convolutional Neural Network |
| CV | Computer Vision |
| YOLO | You Only Look Once |
| SIFT | Scale Invariant Feature Transformation |
| HOG | Histogram of Oriented Gradients |
| IOU | Intersection Over Union |

# CHAPTER 1

# INTRODUCTION

Until recently, the bulk of each firm's programmers were devoted to the creation of the user interface for software and hardware image processing systems. The situation changed dramatically with the advent of the Windows operating system, and the majority of engineers moved their concentration to addressing image processing difficulties. However, there has been no major improvement in tackling basic tasks such as face recognition, vehicle numbers, road signs, understanding remote and medical pictures, and so on. A variety of technical and scientific teams overcome each one of these "eternal" difficulties via trial and error. Because of the high cost of contemporary technical solutions, the task of automating the manufacture of software tools for dealing with intellectual issues has been developed and fiercely handled overseas. The necessary tool kit in the field of image processing should make it easier for typical programmers to analyze and recognize previously unknown content photographs, as well as design applications effectively. Similarly, the Windows toolkit facilitates the generation of interfaces to tackle a variety of real issues.

## 1.1 Motivation

Recognition system refers to a class of vision - based tasks that involve identifying items in digital pictures. Picture classification includes recognizing the class of a single item in an image. Object localization is the process of identifying the position of things in a picture and creating a frame around their extent. Object detection includes two important tasks by finding and categorizing one or more objects in a picture. When a user or practitioner says "objects recognition," they're usually referring to "object detection." Beginners may find it difficult to discriminate between various computer vision tasks.

With this example, we can differentiate between these three computer vision tasks:

To accomplish image classification, determine the type or category of an object in a picture. A single-object image, such as a photograph, is used as an input and output is a class label.

Object localization is accomplished by detecting the existence of objects in a picture and indicating their location using a bounding box. An image made up of one or more objects, such as a photograph, is used as an input and one or more frame boxes as an output.

Object detection is performed by employing a bounding box to locate the presence of items in a picture and determining the kinds or categories of the objects discovered. As input, an image composed of one or more items, such as an image, is utilized, and as output, one or more enclosing containers with a class name for each bounding box.

Object segmentation, also known as "object instance segmentation" or "semantic segmentation," is a subset of computer vision tasks in which instances of identified objects are communicated by highlighting particular pixels of the object rather than a coarse bounding box.

We may extrapolate from this that object recognition refers to a set of tough computer vision tasks.


Fig 1.1 Vehicles Stuck in Traffic Jams [1]

For example, image classification is simple, but the distinctions between object localization and object detection might be puzzling, particularly when all required tasks are referred to as object recognition.

Humans can notice and identify objects in images. The human vision system is capable of doing complex tasks such as identifying several objects and recognizing obstructions with minimal conscious thinking. Because of the availability of massive amounts of data, faster GPUs, and better algorithms, we can now quickly train computers to recognize and categorize multiple elements inside a picture with high accuracy. We'll go over terms like object recognition, object localization, and object recognition and clustering loss functions, as well as a "You only look Once" object detection technique (YOLO).

Object localization is the process of drawing a bounding box around one or more objects in an image, whereas image classification is the process of assigning a class label to an image. Object detection is usually more challenging since it combines these two jobs by constructing a bounding box around each object of interest in the picture and assigning it a class name. All of these challenges are referred to as object recognition.

Object recognition is a term that refers to a group of tasks that are used to recognize things in digital pictures. R-CNNs, or Region-based Convolutional Neural Networks, are a set of algorithms for solving object recognition and localization tasks that are optimized for model performance. YOLO is the second family of object identification systems that are designed for speed and real-time use.

Deep Learning transformed compute power and improved the way apps are developed forever. Applications are becoming smarter, faster, and more capable of performing work that were earlier too difficult for computers to handle. Classifying and identifying items in a scene, analyzing enormous volumes of text, answering passage questions, making art, and competing against and defeating human players at complicated games like Chess are just a few of the more difficult jobs. Artificial Intelligence aspires to mimic the human brain's approach to processing enormous

amounts of data including numerous patterns, as well as how the brain can recognize these patterns, reason about them, and then take action using Deep Learning.

Emergency trucks are essential in every situation. a life-or-death crisis Over 20% of the time is spent stuck in traffic. The patient is housed in an ambulance, but when the patient's condition deteriorates, The risk of patient death has grown dramatically. These are times when an emergency patient requires immediate attention. must get to the hospital very once, and the ambulance arrived We're stuck in a traffic gridlock. In the event of an emergency, this scenario is dangerous. Patients with heart problems who needed to be taken to the hospital as soon as possible. In many people do not bother to give way in traffic congestion. Both the emergency truck and the traffic cops are unable to see which they should make room for the ambulance in the lane. There are several patients die before they reach the hospitals. For firefighting squads, traffic congestion is also a significant obstacle. People in the United States expect a firefighter squad to arrive within four minutes in at least 90% of circumstances where one is required.


Fig 1.2 Detecting words on vehicles [1]

Deep Learning is capable of understanding data of varying patterns at an accurate rate. That is why it is used in most the innovations in understanding language and images. Research is moving forward at a fast pace with Deep Learning. Deep Learning has a major impact on object detection. Recognizing items is a word which referred a range in vision of device tasks that include activities such as recognizing item in photographs. Picture classification entails prediction the class of a single object in a photograph.

Items localize is the process to determined the positions of objects into picture, and creating a bound boxes around them. Items detector concatenate these two work by detecting and classifying single or multi item into pictures. When someone mentions "item-recognition," they usual mean "item-detection." Starting person may have trouble distinguishing between different types of computer vision tasks. Object detection is performed by employing a bound to locate the available of items into imagex, and determining the classes types of the objects identified. For inputs, a photograph made up of one or more objects is used, and as an output-value, box bound with a classes labeled for each boxes. Humans have capability, ability of finding and identified objects in images. The human being eye system can do complex work, differentiating many small objects and recognizing obstructions with their small conscious thought. Because of large quantity of data present, GPUs, and well formed algorithms, Now we quickly trained computers appliances to detect it and classified multiple item into pictures with higher accuracy. We'll looking terminology same to detect object, localize item, and elements detect and localize functional loss, as well an object finder algorithm "You Only Look Once" (YOLO).

Items locating is way to draw a box around multi-objects in an photo. Whereas image differentiation is the best path of assigning a label to an items of class to which it belong. Finding items in input pictures is always hard, as it concatenate these 2 jobs by constructing a box boundary surrounded each interested items in the photograph and assigning it a class name. Problems are referred to as "item recognition."

The phrase "item-recognition" referred to a set of jobs that can use to distinguish item into images. RCNNs, or Region based- networks, are a set of methods that are tuned

for model performance when addressing object recognition and localization problems. YOLO is the second generation of real-time object recognition systems.

Fig 1.3 Various Objects in Real World [2]

Deep Learning completely revolutionized the computation power and led to a fundamental change in how applications are being created. Applications becoming intelligent, fast, and capable of performing those tasks which were initially very complex and out of reach of the computer to perform. There are various complex tasks include classifying and detecting objects in a scene, summarization of large amounts of text, answering passage questions, generating art, and competing and defeating human players at complex games like Chess. Artificial Intelligence aims to replicate the approach used by the human brain to process a large amount of data containing various patterns, how the brain can identify these patterns and reasons about them then take some actions through Deep Learning. Deep Learning is capable of understanding data of varying patterns at an accurate rate. That is why it is used in most the innovations in understanding language and images. Research moving forward at a fast pace with Deep Learning. Object detection is one of the fields which is widely affected by Deep Learning in a substantial way.

Blind folks have their own way of doing things and lead normal lives. They do,

however, confront difficulties as a result of inaccessible infrastructure and social issues. Navigating around locations is the most difficult task for a blind person, especially one who has lost all eyesight. Obviously, blind persons can navigate their homes without assistance since they are familiar with their surroundings. Blind persons have a difficult difficulty locating objects in their environment. As a result, we decided to create an OBJECT DETECTION SYSTEM IN REAL TIME. After reading a few publications in this field, we became interested in this topic. As a result, we're extremely motivated to create a system that can recognize objects in real-time.

## 1.2 Description of Deep Learning and its concepts

Deep learning is a type of machine learning. It teaches a computer to learn how to predict and classify information by filtering inputs through layers. Images, writing, and music can all be used to express observations. The way the human brain filters information is the source of inspiration for deep learning. Its goal is to produce some actual magic by simulating how the human brain functions. There are approximately 100 billion neurons in the human brain. Each neuron is connected to around 100,000 other neurons. We're re-creating it, but in a way and at a level that machines can understand.

A neuron has a body, dendrites, and an axon in our brains. The signal from one neuron goes down the axon to the dendrites of the following neuron. A synapse is the connection through which the signal travels. Neurons aren't particularly useful on their own. When there are a lot of them, however, they may create some major magic. That's how a deep learning algorithm works! You collect data from observations and combine it into a single layer. That layer produces an output, which becomes the input for the following layer, and so on. This continues till you reach your final output signal!

The neuron (node) receives one or more signals (input values) that pass through it. The output signal is delivered by that neuron. Consider the input layer to be your senses: what you see, smell, and feel, for instance. For a single observation, these are independent variables. This data are split into integers and binary bits that a computer can understand. To bring these variables into the same range, you'll need to

standardize or normalize them. For feature extraction and transformation, they employ multiple layers of nonlinear processing units. The output of the previous layer is used as the input for the next layer. What they learn is organized into a hierarchy of ideas. Each level in this hierarchy learns to turn its incoming data into a more abstract and composite representation as it progresses.


Fig 1.4 Detecting Ambulance at High Speed [3]

The output of the previous layer is used as the input for the next layer. What they learn is organized into a hierarchy of ideas. Each level in this hierarchy learns to turn its incoming data into a more abstract and composite representation as it progresses. This means that the input for an image, for example, may be a matrix of pixels. The first layer may encode the borders and pixel composition. The following layer could be an arrangement of edges. A nose and eyes could be encoded in the next layer. The next layer may detect the presence of a face in the image, and so on.

**CNN**

They're made up of neurons with weights and biases that can be learned. Each neuron takes some inputs, does a dot product, and then executes a non-linearity if desired.

Convolutional Neural Networks (CNNs) are similar to classic artificial neural networks (ANNs) in that they are made up of neurons that learn to optimize

themselves. Each neuron will still receive an input and conduct an action (such as a scalar product followed by a non-linear function), which is the foundation of innumerable artificial neural networks. The entire network will still express a single perceptual scoring function from the input raw picture vectors to the final output of the class score (the weight). The last layer will contain the loss functions related with the classes, and all of the standard ANN tips and tactics will still apply.

The main significant distinction between CNNs and standard ANNs is that CNNs are mostly utilized in picture pattern recognition. This enables us to encode image-specific properties into the architecture, making the network better suited for image-focused tasks while also lowering the number of parameters needed to set up the model. Traditional variants of ANN struggle with the computational complexity required to compute picture data, which is one of its biggest drawbacks. Due to its relatively tiny picture dimensions of only 28 x 28, common machine learning benchmarking datasets like the MNIST database of handwritten digits are suitable for most versions of ANN. With this dataset, a single neuron in the first hidden layer will have 784 weights (28281, where 1 is the number of black and white values in MNIST), which is doable for most types of ANN. When a larger colorful picture input of 64x64 is used, the number of weights on a single neuron in the first layer grows significantly to 12, 288. Consider that in order to cope with this size of input, the network must be much larger than the one used to identify color-normalized MNIST digits, and you will see the limitations of utilizing such models.

A neuron has a body, dendrites, and an axon in our brains. The signal from one neuron goes down the axon to the dendrites of the following neuron. A synapse is the connection through which the signal travels. Neurons aren't particularly useful on their own. When there are a lot of them, however, they may create some major magic. That's how a deep learning algorithm works! You collect data from observations and combine it into a single layer. That layer produces an output, which becomes the input for the following layer, and so on. Each neuron will still receive an input and conduct an action (such as a scalar product followed by a non-linear function), which is the foundation of innumerable artificial neural networks. The entire network will still express a single perceptual scoring function from the input raw picture vectors to the

final output of the class score (the weight). The last layer will contain the loss functions related with the classes, and all of the standard ANN tips and tactics will still apply.

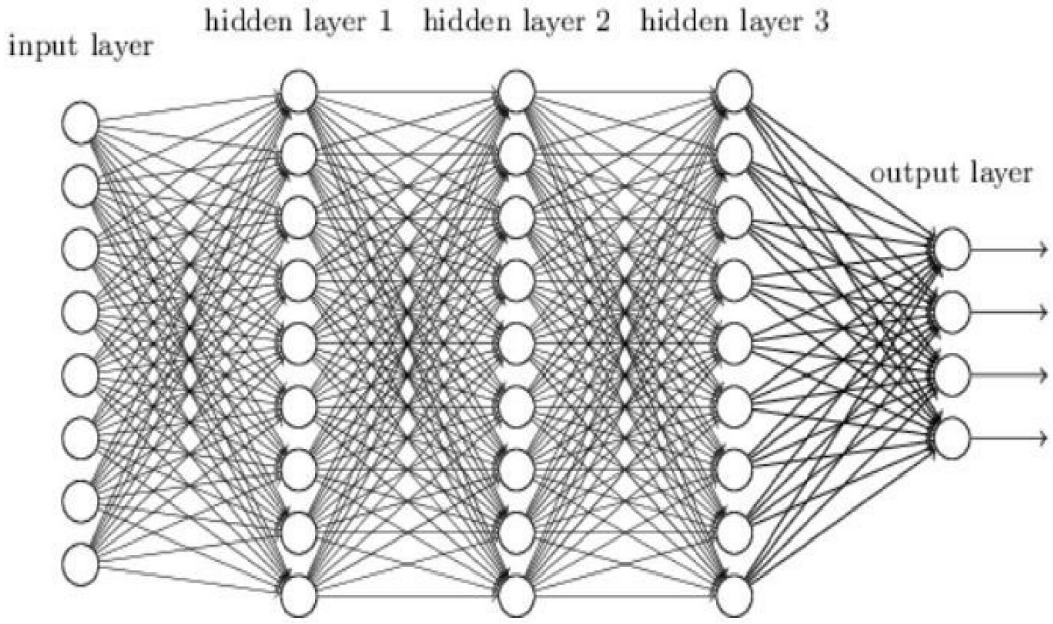


Fig 1.5 Convolutional Neural Networks

There are three sorts of layers in CNNs. Convolutional layers, pooling layers, and fully-connected layers are the three types. A CNN architecture is generated when these layers are stacked. Figure 1 shows a simplified CNN architecture for MNIST classification. Figure 1: An easy-to-understand CNN architecture with only five levels.

The core functionality of the CNN in the example above may be divided into four distinct parts.

1. The input layer, as in other types of ANN, will store the image's pixel values.
2. The convolutional layer will calculate the scalar product between their weights and the region related to the input volume to identify the output of neurons connected to particular parts of the input. The rectified linear unit (abbreviated ReLu) tries to apply a 'element wise' activation function like sigmoid to the output of the previous layer's activation.

3. The pooling layer will next execute down sampling along the input's spatial dimensionality, lowering the number of parameters inside that activation even more.

4. The fully-connected layers will then attempt to produce class scores from the activations, which will be utilized for classification, in the same way that traditional ANNs do. ReLu could also be applied between these layers to boost performance.

CNNs can alter the original input layer by layer utilizing convolutional and down sampling techniques to provide class scores for classification and regression using this simple method of transformation. It's worth noting, however, that simply comprehending the overall architecture of CNN architecture isn't enough.

## R-CNN

To get around the challenge of selecting a large number of regions, Ross Girshick et al. devised an approach in which the selective search is used to extract only 2000 regions from the image, which he calls region proposals. We may use a soft max layer to forecast the class of the proposed region as well as the offset values for the bounding box using the RoI feature vector. Because you don't have to feed 2000 area proposals to the convolutional neural network every time, "Fast R-CNN" is faster than R-CNN. Instead, only one convolution operation is performed per image, and a feature map is generated as a result.

As a result, instead of trying to classify a large number of locations, you may focus on simply 2000. The selective search algorithm described below was used to generate these 2000 region ideas.

Searching with Care:

1. Create the initial sub-segmentation, which includes a large number of prospective regions.

2. Recursively combines comparable sections into larger ones using the greedy approach.

3. Create the final candidate region proposals using the created regions.

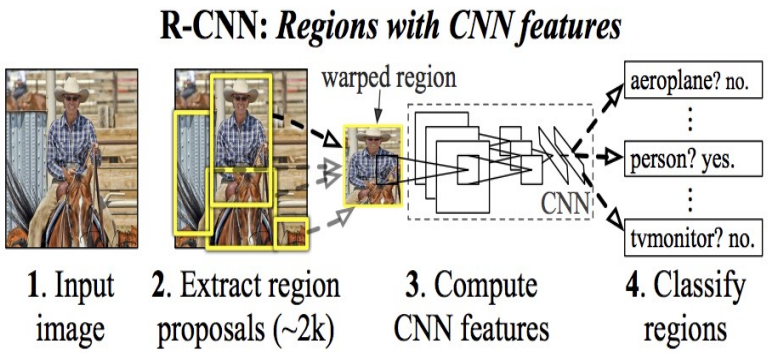


Fig 1.6 Regional Convolutional Neural Networks [4]

These 2000 proposed candidate regions are twisted into a squares & input into a convolution neurals net_work, which outputs a 4096-dimensional feature_vector. A synapse is the connection through which the signal travels. Neurons aren't particularly useful on their own. When there are a lot of them, however, they may create some major magic. That's how a deep learning algorithm works! You collect data from observations and combine it into a single layer. That layer produces an output, which becomes the input for the following layer, and so on The CNN acts as a feature_maps extractor, and the output dense_layer contains the features collected from the picture, which are input into an SVM to classify the presence of the object within the candidate region suggestion.

In addition to detecting the occurrence of an item inside the recommended region, the approach predicts four values that are offset values for increasing the accuracy of the bounding box. For example, based on the area suggestion, the algorithm may have anticipated the presence of a person, but the face of that individual inside that feature map may have been halved. As a consequence, the offset values offered help in changing the bounding box of the region proposal. Although the system anticipated the presence of an object, the face of the person inside that region proposal may have been cut in half. As a result, the offset values aid in altering the bounding box of the

region proposal; moreover, the algorithm predicts four offset values to boost the precision of the bounding box.

**Drawback of R-CNN**

1. To train the network, you'd have to classify 2000 region proposals every image, which would take a long time.
2. It cannot be implemented in real time because each test image takes roughly 47 seconds.
3. A fixed algorithm is the selected search algorithm. As a result, no learning takes place at that point. This could result in a slew of bad candidate region suggestions.

## FAST R-CNN

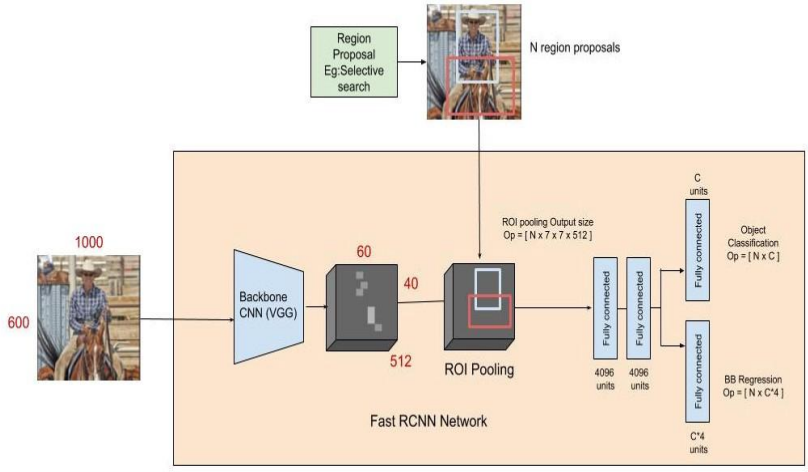


Fig 1.7 Fast R- Convolutional Neural Networks [4]

The same author of the previous study (R-CNN) fixed some of R-shortcomings CNN's to create Fast R-CNN, a quicker object detection technique. The method is comparable to the R-CNN algorithm. Instead of feeding the CNN the region proposals, we give the CNN the input image to produce a convolutional feature map.

We can identify the region of the proposals using the convolutional feature map and warp them into squares, which we then reshape using a RoI pooling layer into a fixed size that can be input into a fully connected layer.

We may use a soft max layer to forecast the class of the proposed region as well as the offset values for the bounding box using the RoI feature vector. Because you don't have to feed 2000 area proposals to the convolutional neural network every time, "Fast R-CNN" is faster than R-CNN. Instead, only one convolution operation is performed per image, and a feature map is generated as a result.

In training and testing sessions, Fast R-CNN is much faster than R-CNN. When comparing the performance of Fast R-CNN during testing, using region proposals considerably slows down the algorithm compared to not utilising region proposals. As a result, the proposed regions become bottlenecks in the Fast R-CNN method, lowering its performance.

## FASTER R-CNN

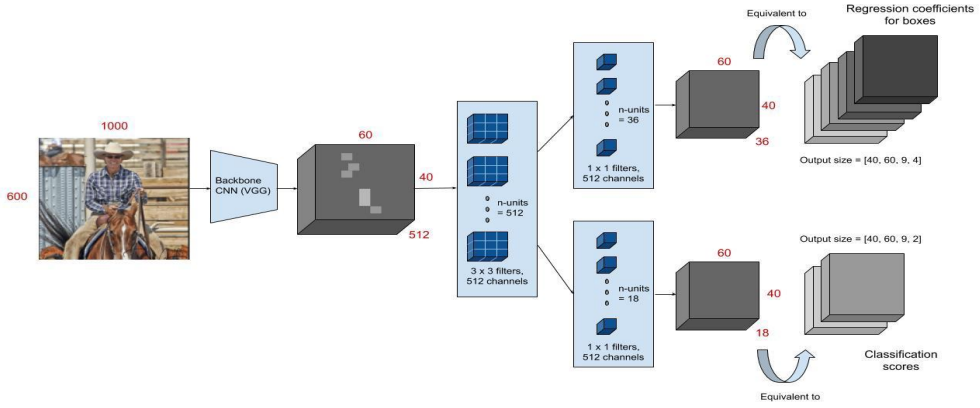


Fig 1.8 Faster R- Convolutional Neural Networks [5]

To find region proposals, both of the above algorithms (R-CNN and Fast R-CNN) use selective search. Selective search is a slow and time-consuming operation that has an impact on network performance.

The image is fed into a convolutional network, which outputs a convolutional feature map, similar to Fast R-CNN. Rather than using the feature map's selective search technique to discover area proposals, a separate network is employed to forecast region proposals. The proposed region is subsequently reshaped using a ROI pooling layer, which classifies the image within the proposed region and predicts the bounding box offset values.

Similar to Fast R-CNN, the picture is fed into a convolution network, which outputs a convolution feature_map. Instead of utilizing a selective_search algorithm on the feature space to discover region suggestions, a separate network is utilized to anticipate region proposals. The projected region suggestions are then reshaped using a RoI pooling layer, which is then used to categorize the image within the proposed region and forecast the offset values for the bounding boxes.

## YOLO

To locate the object within the image, all previous object detection techniques utilized areas. The network does not examine the entire picture. Rather, portions of the image with a high likelihood of containing the object. You Only Look Once, or YOLO, is an object detection algorithm that differs significantly from the region-based algorithms discussed previously. The bounding boxes and class probabilities for these boxes are predicted by a single neural network in YOLO.

Object finding is problem of regression done in YOLOv3 and probable class are provide of such noticed photos. Neural interconnected layers which are convolution are use inside YOLOv3 algorithm for find real-time items. To finding items, through a neural layer interconnect this way just took a forward propagation which is single, as the name implies. This means that whole picture get analyze in single time. It get use to show multiple probable class as well as bounded line at similar time. ResNet and FPN were the inspiration for this algorithm.

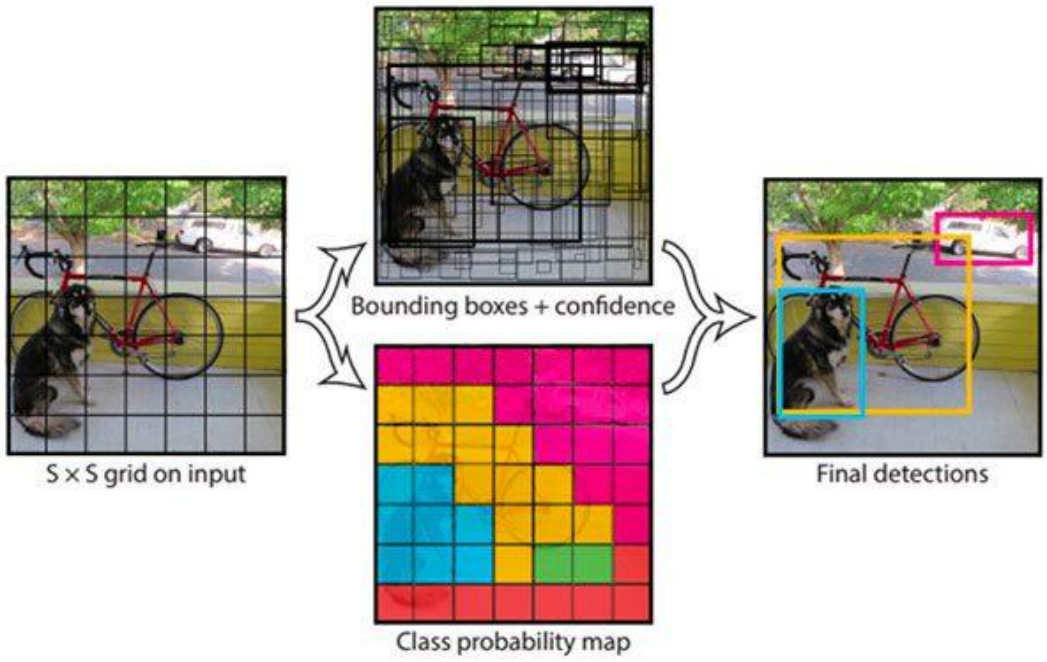


Fig 1.9 YOLO working diagram [6]

YOLO works by splitting an image into a SxS grid and creating m bounding boxes within each grid. The network outputs a class probability and offset values for each bounding box for each bounding box. The bounding boxes with a class probability greater than or equal to a threshold value are chosen and utilized to locate the object within the image.

YOLO is orders of magnitude faster than any other object detection technique (45 frames per second). The YOLO algorithm's drawback is that it has trouble identifying small things in images; for example, it might have trouble distinguishing a flock of birds. This is owing to the algorithm's spatial restrictions.

YOLO is orders of magnitude faster than conventional object detection algorithms (45 frames per second). The YOLO algorithm's drawback is that it has trouble detecting small things in images; for example, it might have trouble detecting a flock of birds. This is owing to the algorithm's spatial restrictions.

## Python

Python is general purposed programming language that is high-level and interpreting. Its design concepts prioritized program code readability through extensive indentation.

Python uses garbage collection and dynamic typing. It supports a wide range of programming paradigms, including structured (particularly procedural), object-oriented, and functional programming. It is sometimes referred to as a "batteries included" language due to its vast standard library.

Python is a programming language that is extensively used for online and software development, task automation, data analysis, and data visualization. Python's relative simplicity of learning has led to its adoption by many non-programmers, such as mathematicians and scientists, for a range of typical activities, such as financial planning.

"Writing programs is a really creative and satisfying endeavor," argues Charles R Severance of the University of Michigan and Coursera in his book Python for Everyone. "You can build programs for a variety of purposes, including making a living, solving a challenging data analysis challenge, having fun, and assisting others in solving problems."

Guido van Rossum began developing Python as a successor for the ABC programming language in the late 1980s, and Python 0.9.0 was released in 1991. Python 2.0, released in 2000, added list comprehensions, cycle-detecting garbage collection, reference counting, and Unicode support. Python 3.0, released in 2008, was a major change that was not completely backwards compatible with prior versions. Python 3 took the place of Python 2.

# CHAPTER 2

# LITERATURE REVIEW

The goal of remote sensed target finding is to pinpoint the exact location of an interested target in a remote sensed image and classify it. Since 2012, krizhevsky has sparked a surge of in-depth research in academia. The frequently used target identification algorithms based on convolutional neural networks can now be classified into two groups. At 2016, Zou et al. introduced a DCNN that combines D_CNN and machine_learning SVM and has shown to be effective in detecting ships. Yao et al. suggested a multiple-architectured neural_network that can recognise three different types of distant sensing targets: large, medium, and small. Following up on the research, it was shown that when the mean accuracy (map) of the two is around 76 percent, detection can be as high as 90 percent. YOLOV3 uses the logistics function instead of the usual soft max function and uses the greater accuracy darknet53 as the feature extraction network. Darknet53 is similar to resnet-152 in terms of accuracy, however it is faster. Furthermore, the detection effect of yolov3 on small objects has been greatly increased, thanks to the network's new top-down structure. The scale of the entire model will be substantially simplified, and the quantity of processing will be greatly reduced, thanks to YOLOv3 Mobilenet. Mobilenets has 53 convolutions, just 1 convolution, and 13 deep separable convolutions when compared to darknet53. One-to-one convolution kernels and channels are used, and 11 point wise convolution is used to complete the integration of Depth wise and Point wise characteristic graphs. K-means is a maximal expectation-based unsupervised clustering algorithm.[1]

Setting a prior bounding box size has the goal of improving the intersection of the prediction box and the ground truths over the (IOU) findings. IoU stands for the correlation between the prediction frame and the ground truths in target detection. K-means employs an update approach after calculating the mean value each cluster iteration, making it more sensitive to outliers and noise. There will be a few super huge or super small targets in remote sensing image detection due to large variances in terms of satellite shooting height, camera resolution, and real object size. The median has a strong anti-interference to noisy points or outliers, which reduces the

impact of aberrant size and increases detection accuracy.

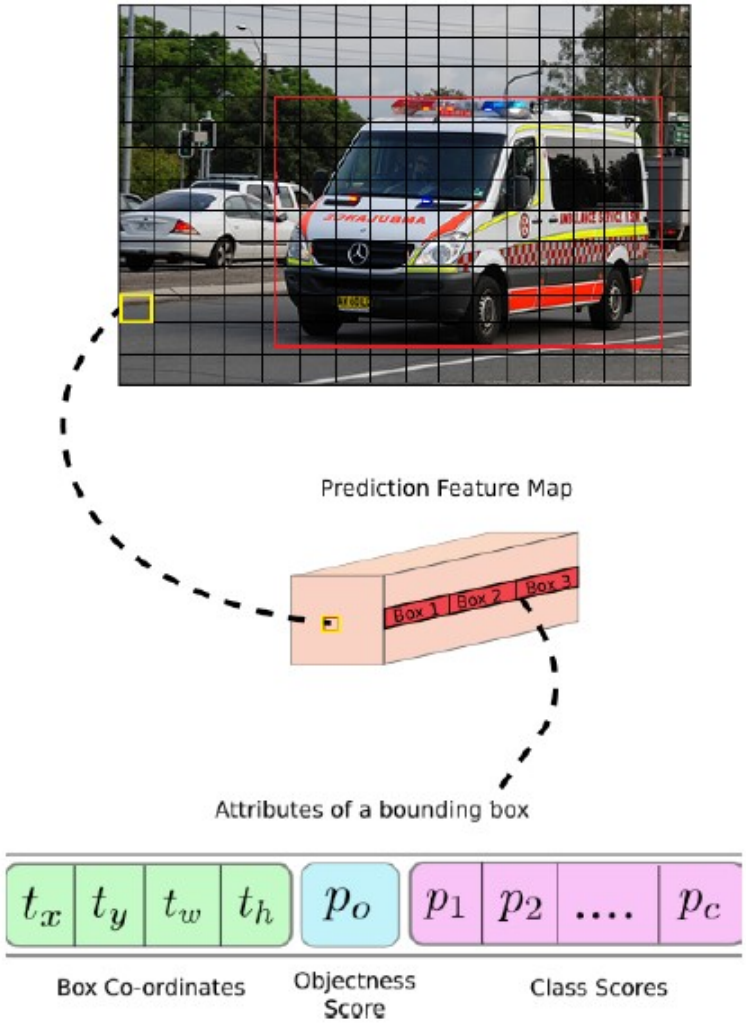


Fig 2.1 Prediction of maps_feature of a sample picture.[7]

It predicts three boxes per matrix. Each box has box_coordinates, objectness_score and class_prediction. The data set utilized in the experiment is an augmentation of the NWPU-VHR10 set of photographs. There are 650 original photos in the database and 398 extracted images. Instead of its predecessor, box_coordinates, objectness_score and class_prediction the YOLOv3-Mobilenet satellite is equipped with a lightweight network called Mobilenet, which decreases the database and 398 extracted images the model scale and computation load while keeping the same detection accuracy.

medians technique is developed for predicting the target's position in real-time. The experiment sets the groundwork for future algorithm simulations on the embedded platform.

Object detection is a technique for identifying meaningful things in digital photos and movies. Multiple item detection from an image is one of its real-time applications. The car, motorcycle, and pedestrian are the most prevalent objects to detect in this program. Object Localization is used to locate the items in the image. Our algorithm processes 45 frames per second in real time. For estimating accurate bounding boxes from a picture, the YOLO method is utilized. By estimating the bounding box for each grid and class probabilities, the image divides into S x S grids.[2]

Anchor boxes are also utilized to improve object detection accuracy. To solve the shortcomings of the sliding window method, they employed the bounding box method for object localization. An image is captured and divided into three 3 x 3 matrixes. Each grid is labeled, and picture classification and object localization procedures are used to each grid. The first grid in our example image has no suitable object, so it is rendered as. The bounding boxes of the object in the sixth grid are bx, by, bh, and bw. If the same object appears in two or more grids, the object's centre point is determined, and the grid with that point is chosen. If grid's object is an automobile, the classes are (0,1,0). The matrix for the fifth grid will be somewhat similar, with varying bounding boxes based on the position of the objects in the associated grid. IoU uses the formulas IoU = Area of Intersection / Area of Union to calculate the IoU of two boxes using the actual and expected bounding box values. For a single image, any number of anchor boxes can be utilized to detect various objects. In our case, we used two anchor boxes, one for the people and one for the vehicle.

The label Y comprises 16 values in this type of overlapping object detection, i.e. the values of both anchor boxes. To forecast a (7, 7, 30) tensor, the YOLO technique employs a Convolutional neural network. It does a linear regression to construct a 7x7x2 bounding box prediction using two fully connected layers. The system predicts numerous bounding boxes for a single grid cell. Only the box that is in charge of the object is counted.

The YOLO algorithm was created with the goal of detecting objects with only one neural network. The algorithm is simple to construct and may be trained on an entire image. When estimating borders, it uses the complete image and predicts less false positives in background areas. This approach is substantially more efficient and faster to utilize in real time than previous classifier algorithms.

The human visual system is quick and accurate, allowing us to do complicated tasks like driving with minimal effort. Fast, accurate object identification algorithms could allow computers to drive cars without the use of specialized sensors. They may also make it possible for assistive gadgets to provide real-time scene information to humans. Object detection is reframed as a single regression issue, with picture pixels being converted to bounding box coordinates and class probabilities. You only need to glance at an image once to forecast what objects are present and where they are using our method. On a Titan XGPU, YOLO runs at 45 frames per second with no batch processing. YOLO detects the complete image and encodes object context.[3]

When compared to Fast R-CNN, YOLO makes half as many background mistakes. It can quickly recognize things in photos, but it still falls short of cutting-edge detection technologies. End-to-end training and real-time speeds are possible thanks to the YOLO design, which maintains excellent average precision. To forecast each bounding box, our network incorporates characteristics from the entire image. It also predicts all bounding boxes for an image across all classes at the same time. The GoogLeNet model for image classification inspired our network architecture. In addition, we train a fast version of YOLO to test the limits of quick object identification. Our network's final output is a tensor of predictions with a value of 7x 7x 30.

We used the ImageNet 1000-class competition dataset to pre-train our convolutional layers. We train this network for about a week and attain 88 percent single crop top-5 accuracy. This is analogous to Caffe'sModel Zoo's GoogLeNet models. Per grid cell, YOLO predicts numerous bounding boxes. For each object, we assign one predictor to be "responsible." Each predictor improves its ability to anticipate specific sizes, aspect ratios, or object classifications. As a result, the bounding box predictors become more specialized. The YOLO network predicts 98 bounding boxes and class

probabilities for each box for each image. The grid design ensures that the bounding box forecasts are spatially diverse. When compared to R-CNN or DPM, non-maximal suppression improves mAP performance by 2% to 3%.

Because each grid cell only predicts two boxes, YOLO places strict spatial limits on bounding box predictions. Small items that occur in groups, such as flocks of birds, pose a problem for our model. Our model struggles to generalize to objects in new or unexpected configurations since it learns to predict bounding boxes from data. We compare the YOLO detection method to a number of other popular detection systems, pointing out major parallels and differences. Instead of using static features, the network trains and optimizes the features in real time for the detection task. Our unified architecture produces a model that is both faster and more accurate than Deformable Parts models.

R-CNN and YOLO have certain commonalities. Using convolutional features, each grid cell offers candidate bounding boxes and rates them. Rather than attempting to optimize individual components of a complex detection pipeline, YOLO completely discards the pipeline and is designed to be fast. YOLO is a multi-object detector that learns to detect a variety of items at the same time. For many items of multiple classes in an image, YOLO predicts both bounding boxes and class probabilities. MultiBox can also conduct single object detection by substituting a single class prediction for the confidence prediction.

YOLO is the world's fastest general-purpose object detector, pushing the boundaries of real-time object detection. YOLO may be trained on a loss function that directly reflects its performance, making it perfect for applications that require reliable, rapid, and reliable object detection. There are many different types of feature descriptors available for detection. The (SIFT) approached developed by Lowe allowing us to extracted distinct in variant characteristics from item we currently intent to detecting. A database have varied key-point featured can be created used a collection of trained photos. Object detector can be accomplished comparing featured key-points in a test image dictionary have varied key-points taken from trained photos for the items we seeked.

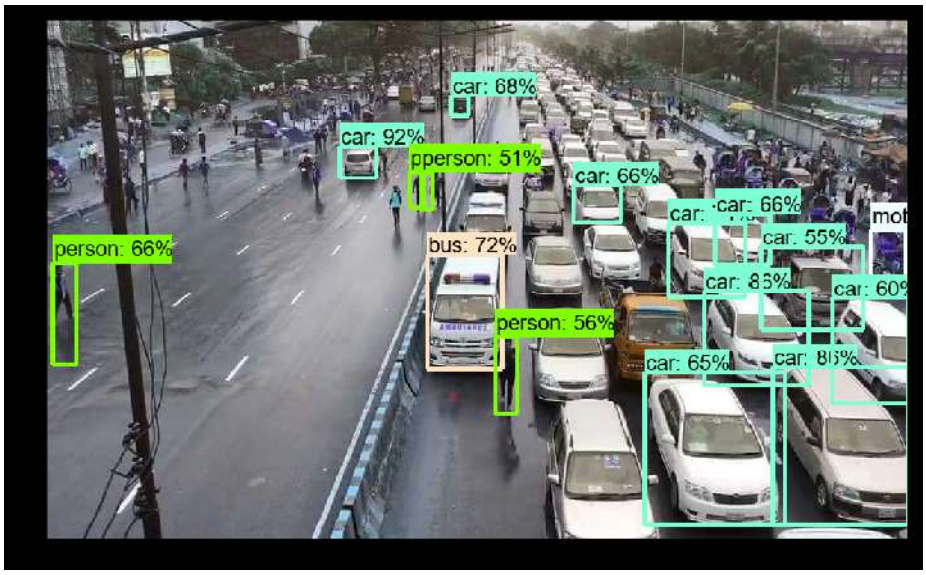


Fig 2.2 Example of detection by YOLO object detector [8]

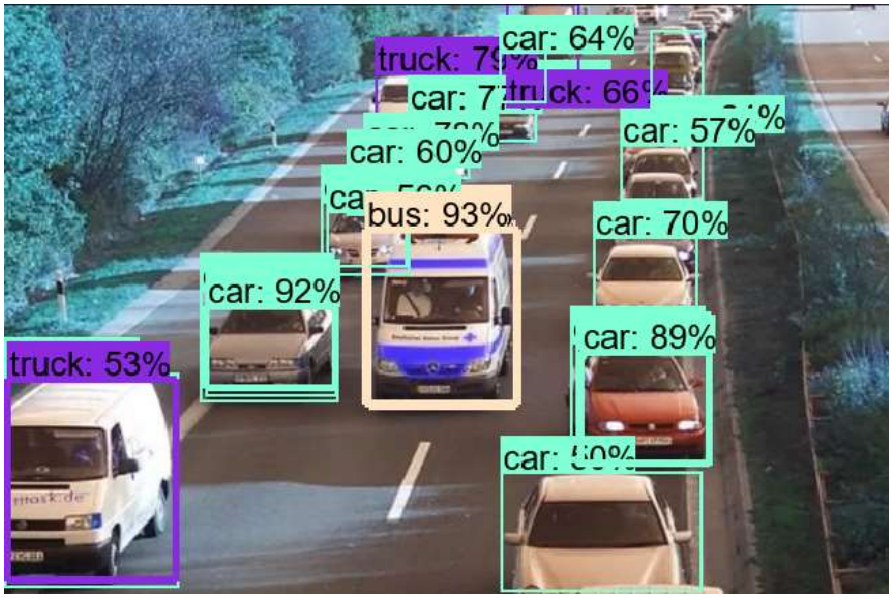


Fig 2.3 Example of detection of vehicle by YOLO object detector [8]

Traditional aerial imaging approaches are unable to process high-resolution aerial photos, and algorithm efficiency is low. Instead of manually extracting features from the inputs, deep learning algorithms have been presented. These algorithms select areas in which the objects may appear based on characteristics such as shape, color, and texture. Sliding window searching is one of the most widely used region-based

approaches for object detection, but we created a novel method that employs fewer windows and has a higher recall rate. Instead of the serial feature extraction procedure used by the previous approach, Faster RCNN, the Fast RCNN method can execute feature extraction on the input image using a neural network. The last layer of the network's features is used for regression and classification in the traditional RCNN method. These features contain high-level semantic information. Low-level structural features and high-level semantic features can be fused using the multi-scale feature. fusion procedure to improve detection accuracy for small objects. DOTA, a big dataset for object detection in aerial photos, was used in this paper as the aerial image dataset. The Caffe platform with Ubuntu 16.04, OpenCV 2.4, Anaconda 2, and Python 2.7.16 is used in our investigations. The improved approach has the potential to produce superior visual outcomes. At the moment, our key goal is to improve the algorithm's accuracy, which will serve as the foundation for future studies. The revised method improves detection accuracy slightly, and the total result is acceptable. The current hardware can fully meet the requirements for detection rate, and the increase in computing time has little impact on practical applications.[4]

The Faster RCNN model processes fused features using a multi-scale feature fusion approach, as well as a residual module and pooling layer. To improve the usefulness and robustness of the suggested model, further study could entail gathering more data and expanding the categories of items.

There are many different types of feature descriptors available for detection. The (SIFT) approached developed by Lowe allowing us to extracted distinct in variant characteristics from item we currently intent to detecting. A database have varied key-point featured can be created used a collection of trained photos. Object detector can be accomplished comparing featured key-points in a test image dictionary have varied key-points taken from trained photos for the items we seeked.

This Shape Context Belongiee method involves sampled the edged of an item into interest points and captured the distributed of the sampled on the form with regard to specified points of body. relationship between a body point and other points was explained using distance and angle measurements. To make a histogram, you can bin distance and angles into multiple buckets. The similarity p and other points the form

will be captured by the histogram. Matched two body is analogous to comparing spots each form having same boy context. These ways works well for finding text, fingerprint matcher, and other similar tasks.

Another method presented is to utilize a Histogram of Orienting Gradients (H-O-G) as description templates to recognize humans. Normal intensity gradients (edges) are utilized to efficiently describe the local object appearance and shape information.

The region proposals in a CNN are defined by the identification of these regions of interest from the input frames (Convolutional neural network) To reduce the complexity and enhance the speed, these regions are wrapped and fed into CNN layers. SVM is used in RCNN to recognize different classes of cars for recognition purposes. 18 Convolutional layers, sub sampling layers, and fully linked layers were combined to compute features for 128 x 128 input frames. Convoluted images are created by moving a kernel through the entire image and producing convolved images, which are then utilized to identify information from the input frames. The R-CNN's Region Proposals assisted in achieving a recognition accuracy of 91.3 percent for a variety of vehicle kinds, with a performance metric of. The suggested traffic surveillance system's competency with state-of-the-art approaches is demonstrated by the performance metric.

By the help of hyper-parameter Fast RCNN gets updated. It is made up of a CNN which is also called backbone, a final pooling layer called "ROI pooling," an FC layer called (a(K+1) soft max layer, and bounding box regression. For Fast R-CNN, weight values with backpropagation are critical in all training networks.

Sliding window searching is one of the most widely used region-based approaches for object detection, but we created a novel method that employs fewer windows and has a higher recall rate. Instead of the serial feature extraction procedure used by the previous approach, Faster RCNN, the Fast RCNN method can execute feature extraction on the input image using a neural network. The last layer of the network's features is used for regression and classification in the traditional RCNN method. These features contain high level semantic information. The Faster RCNN model processes fused features using a multi-scale feature fusion approach, as well as a

residual module and pooling layer. To improve the usefulness and robustness of the suggested model.

It is one of the best and faster item detector for casual need in world, pushing the limits of real-time object detection. YOLO may be train along loss function which direct reflects its speed, makes it ideals for appliances requiring speed, accurate item detector. It would be multiple item finders that learned to detect many items at once. YOLO can find both bounding line and category probabilities for many items of various classes in a picture. If predicting confidence is replaced by single category predict then single item detect can also be perform by multi-box.


Fig 2.4 Detecting other details with the ambulance [9]

The data set utilized in the experiment is an augmentation of the NWPU-VHR10 set of photographs. There are 650 original photos in the database and 398 extracted images. Instead of its predecessor, the YOLOv3- Mobilenet satellite is equipped with a weight less in a network called Mobilenet, which decreases the model scale and computation load having equal detection accuracy. In addition, an IoU K-medians technique is developed for predicting the target's position in real-time. The experiment sets the groundwork for future algorithm simulations on the embedded platform

From a background of working in event or festival management, traffic control and auto parking arrangements are hard concerns. Vehicle classification and counting are

techniques that are used to improve the analysis and management of events such as religious ceremonies, recreational, social, sporting, and fundraising events. Vehicle detection from Event surveillance monitoring video streaming relies heavily on image processing. Buses, vehicles, motorcycles, and lorries are examples of event traffic. Vehicle categorization and counting on video streaming from a Myanmar wedding event are used in this system, which uses hyper-parameter optimization on a Fast R-CNN.[5]

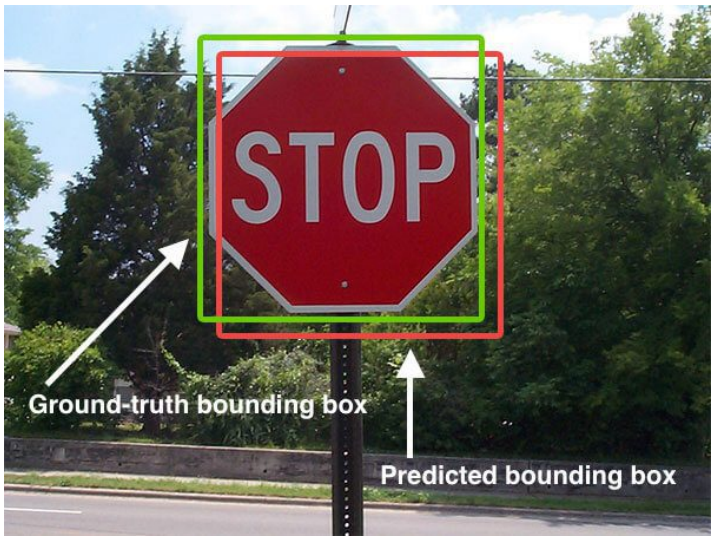


Fig 2.5 Bounding Boxes comparison [10]

In the city of Karlsruhe, Germany, Fast RCNN was used to detect and classify vehicles in traffic regions. Section II of the study explains the overall landscapes of the system and proposes methods such as quick RCNN and Nelder-Mead optimization. Section-IV summarizes the experimental findings and results based on real-life wedding event streaming video, concluding with a positive outcome. To categorize Vehicles and count them, this system used a Fast RCNN model that was updated via hyper-parameter tuning. It is made up of a backbone CNN, a final pooling layer called "ROI pooling," an FC layer called (a(K+1) softmax layer, and bounding box regression. For Fast R-CNN, weight values with back-propagation are critical in all training networks. The system is trained with openCV2, Caffe model construction to get training picture set and included hyperparameters optimizing the objective function by repeating its evaluation vertices on an AWS instance with 8 core vCPUs,

16 GB RAM, and a strong NVIDIA GPU with 1,536 cores (4GB of video RAM). This study developed fresh Myanmar Cars datasets by collecting and organizing 5043 vehicle photos from the Facebook used car auctions page. Using deep learning, hyper-parameter optimization, and a new relevant environment dataset, a group of researchers built a categorization and counting method for Myanmar cars. The model is more robust in terms of handling a big dataset of automobiles and performs well in deep neural learning. Cars Image datasets require additional data to categorize and train to cover more varieties of Myanmar automobiles.

The proposed traffic surveillance system's sub-sections include frame extraction, detection, and recognition of moving vehicles, with the goal of making it easier for police and other law enforcement agencies to keep a watch on pedestrians and bikers using body-worn cameras. The proposed vehicle detection component of the proposed vehicle recognition system includes phases such as frame smoothing, estimate, and backdrop removal. A moving average filter, often known as a box filter, is a linear smoothing filter that reduces the intensity of succeeding pixels in a picture. It's similar to the convolution filter, except instead of rapid color fluctuations, it uses kernels with positive entries added up to achieve an ar value of'1'. The intended zone of interest is the result of the detection, i.e. moving vehicles were detected using background subtraction as the proposed region of interest.[6]

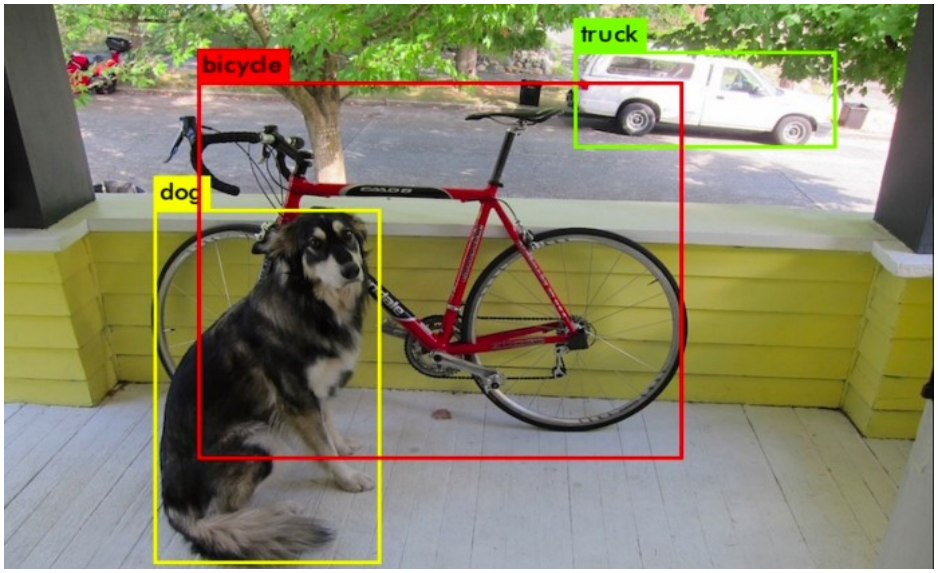


Fig 2.6 Real time objects detection [11]

The region proposals in a CNN are defined by the identification of these regions of interest from the input frames (Convolutional neural network) To reduce the complexity and enhance the speed, these regions are wrapped and fed into CNN layers. SVM is used in RCNN to recognize different classes of cars for recognition purposes. Convolutional layers, subsampling layers, and fully linked layers were combined to compute features for 128 x 128 input frames. Convoluted images are created by moving a kernel through the entire image and producing convolved images, which are then utilized to identify information from the input frames. The R-CNN's Region Proposals assisted in achieving a recognition accuracy of 91.3 percent for a variety of vehicle kinds, with a performance metric of. The suggested traffic surveillance system's competency with state-of-the-art approaches is demonstrated by the performance metric.

We propose a comprehensive feature set in this paper that allows the human shape to be distinguished clearly even in tough backdrops and under adverse lighting. We demonstrate that locally-normalized Histogram of Oriented Gradient (HOG) descriptors outperform various known feature sets, including wavelets. Our detectors perform almost flawlessly on the MIT pedestrian test set [18,17], thus we generated a more difficult collection with over 1800 pedestrian photos to test. Our detector has a simpler architecture with only one detection window, but it appears to perform substantially better than earlier methods. It operates by extracting edge pictures and utilizing chamfer distance to match them to a set of learned exemplars. This section presents a high-level overview of our feature extraction process, as seen in the image. The method employs a dense grid to evaluate well-normalized local histograms of image gradient orientations. Over the last decade, such features have been increasingly popular [4,5,12,15]. The core premise is that the distribution of local intensity gradients or local object form may often be used to characterize the appearance and shape of local objects.[7]

We put our detector to the test on two separate datasets. The first is the well-known MIT pedestrian database, which includes 509 training and 200 test photos of pedestrians in urban settings (plus left-right reflections of these). The second is 'INRIA,' a new and far more difficult data set that contains 1805 64128 photographs

of humans clipped from a variety of personal photos. A preliminary detector is trained for each detector and parameter combination, and the 1218 negative training pictures are thoroughly examined for false positives ('hard examples'). To obtain the final detector, the technique is re-trained using this augmented set (original 12180 + hard. examples). It equates to a raw error rate of roughly 0.8 false positives per 640x480 image evaluated in a multiscale detector. Because our DET curves are usually relatively shallow, even minor increases in miss rate equate to huge increases in FPPW at a constant miss rate. With only one orientation bin, we imitate this using our CHOGdescriptor.  although be aware that an accurate comparison is not achievable.

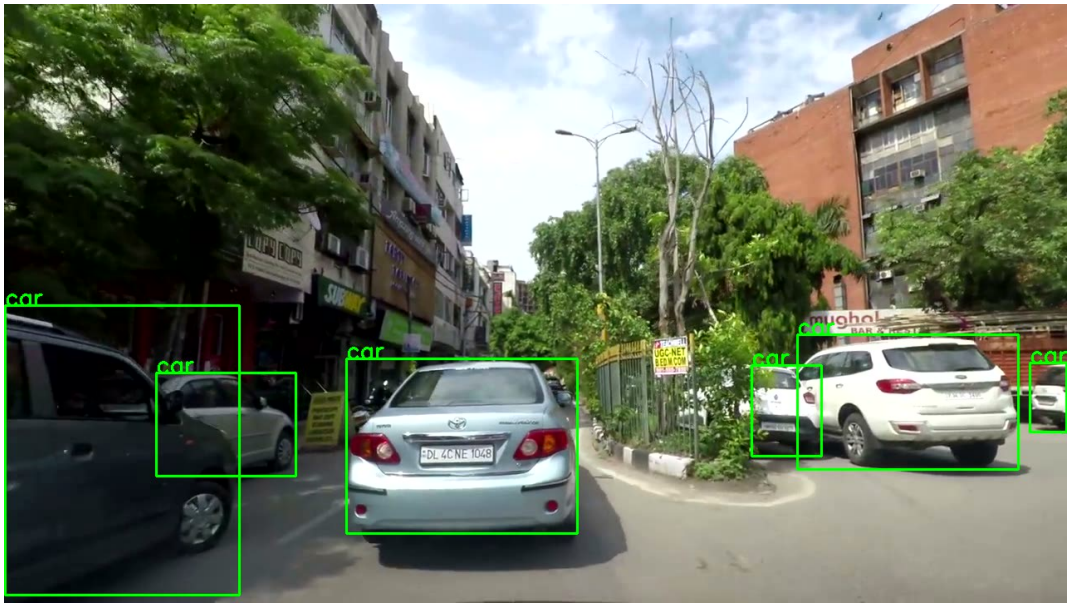


Fig  2.7 Car Detection using YOLO [12]

The human_being visualisation system is quick and accurated, allowed us to do complicated tasks like driving with minimal effort. Fast, accurate object identification algorithms could allow computers to drive cars without the use of specialized sensors. They may also make it possible for assistive gadgets to provide real-time scene information to humans. Object detection is reframed as a single regression issue, with picture pixels being converted to bounding box coordinates and class probabilities. You only need to glance at an image once to forecast what objects are present and

where they are using our method. On a Titan XGPU, YOLO runs at 45 frames per second with no batch processing. YOLO detects the complete image and encodes object context.[3]

When compared to Fast R-CNN, YOLO makes half as many background mistakes. It can quickly recognize things in photos, but it still falls short of cutting-edge detection technologies. End-to-end training and real-time speeds are possible thanks to the YOLO design, which maintains excellent average precision. To forecast each bounding box, our network incorporates characteristics from the entire image. It also predicts all bounding boxes for an image across all classes at the same time. The GoogLeNet model for image classification inspired our network architecture. In addition, we train a fast version of YOLO to test the limits of quick object identification. Our network's final output is a tensor of predictions with a value of 7x 7x 30.

Adding primitive bar detectors to R-HOG doubles the feature dimension while also improving performance. At 104 FPPW, using binary edge voting (ECHOG) instead of gradient magnitude weighted voting (CHOG) reduces performance by 5%. Fine-scale derivatives (no smoothing), a large number of orientation bins, a block spacing stride of 8 pixels, and moderately overloaded, firmly normalized, overlapping descriptor blocks are recommended for good performance. At low FPPWs, compression of each color channel increases performance by 1%, but log compression is too powerful and lowers performance by 2% at 10+4 fps. Larger masks always tend to degrade performance, and smoothing exacerbates the problem. Various 1-D point (uncentered [1, 1], centered [+1, 0, 1], and cubic corrected (33) Sobel masks were evaluated.

SIFT descriptors are comparable to R-HOG blocks, however, they are more complicated. They are dense grids of pixels with no prevailing orientation alignment at a single scale. 33% cell blocks of.666-pixel cells perform best for human detection, with a 10.4% miss rate at 104% FPPW w.r.t. The number of angular and radial bins, the radius of the central bin in pixels, and the expansion factor for following radii are the four parameters of the C-HOG pattern. Because the circular center variations have fewer spatial cells, we only show results for them. Major human contours (particularly the head and shoulders) are often found in the most critical cells in an R-

HOG detector, rather than blocks outside the contour area. There is no discernible difference in performance when using numerous alternative normalizations for each cell based on different pooling scales. At 104 FPPW, using Gaussian kernel SVM improves performance by about. r 3%, but at the cost of a substantially longer run time. For good results, strong local contrast normalization is required, and the usual center-surround technique is not necessarily the most successful.


Fig 2.8 Emergency Vehicle Detection [13]

We've shown that in a dense overlapping grid, utilizing locally normalized histogram features comparable to SIFT descriptors yields excellent results for person detection. In comparison to the best Haar wavelet-based detector, this reduces false-positive rates by more than an order of magnitude.

To address "real-time" needs for picture matching, a new SIFT algorithm is being developed. A feature descriptor's dimensions are reduced from 128 to 48, and the primary primaryr direction is not determined. The dimension performance was also examined during the feature extraction process. Digital cameras' "real-time" and matching accuracy can be considerably improved with a better SIFT algorithm. It's based on the idea that for a given picture I(x, y), the Gaussian kernel is the sole linear transformation (1, 2, 3, 4, 5). A more advanced version of the SIFT algorithm has been created. To create a feature descriptor, it employs a Gaussian and Gaussian

residual pyramid. The algorithm selects the extreme point in scale space of the Gaussian retrocessional pyramid, the enhanced approach uses a circle as the feature descriptor and four concentric sub-domains with radius of 2,4,6,8. It ensures that the neighborhood region and orientation information for the feature point are not modified. The number of concentric circles and the dimension (n) into which they are split determine the dimension of a feature.[8]

SIFT's method uses matching rate to decide what is a fair dimension, and matching rate is defined as the ratio of accuracy matching rate to computing time. He devised a method for detecting extreme points in scale space, generating feature descriptors, and reducing descriptor dimensions from 128 to 48.

Fig 2.9 Ambulance Detection in Heavy Traffic [13]

Image characteristics with several attributes useful for matching photographs of an object or scene are described in this paper. Occlusion, clutter, and noise are less likely to disturb features because they have excellent spatial and frequency localization.. Using a cascade filtering method, the cost of obtaining these features is reduced. Image data is transformed into scale_invariant coordinates relatively to local features using the Scale_Invariant_Feature_Transform (SIFT). The key point descriptors are very different, allowing a single feature to find its match in a huge database of features with a high probability.[9]

SIFT characteristics are extracted and saved in a database after being extracted from a

series of reference photos. Many features from the backdrop can cause many false matches in addition to the valid ones in an image-matching challenge. By selecting subgroups of critical points that agreed on the position, size, and alignment of the item, the right matches can be filtered out of the whole collection of matches. They were designed with stereo and short-range motion tracking in mind. They've since been applied to more challenging tasks like picture recognition, in which a feature must be matched against a massive database of images. Lindeberg has done extensive research into the subject of determining a suitable and consistent scale for feature detection (1993, 1994).

The noise sensitivity of affine frames is higher than that of scale-invariant features. Because training views are ideally taken every 30 degrees rotation in viewpoint, greater affine invariance may not be necessary. The capacity to recognise larger numbers of features and significantly more efficient feature extraction are also advantages. Affine invariance is a useful characteristic for matching planar surfaces when the view shifts dramatically.

More research into how to combine this with non-planar 3D perspective invariance is needed. Keypoints are locations and scales that can be assigned to the same object in different views. The basic picture is convolved using Gaussian to create a picture segregated in scale space by a constant factor k., which is an efficient technique to constructing D(x, y,). The component (k 1) in the equation is constant across all scales and hence has no bearing on the location of the extrema.

The frequency of sampling in the picture and scale domain that is required to properly detect the extrema is an essential issue. The number of key points recognized in a typical image as a function of the number of scale samples can be used to determine the best choices experiment. In this it demonstrate the experimentally determined sampling frequency that maximizes extrema stability.

Displays the simulation results used to see how changing the number of scales per octave at which the picture function is sampled before detection affects the result. The percentage of descriptors accurately matched to a huge database is shown on the lower line.. When sampling only three scales per octave, the best reproducibility is

achieved. Using a larger number of scale samples would be ideal for many applications, but this would increase computational expenses. Illustrates an attempt to determine how much prior smoothing is performed to each image level before constructing the scale space representation for an octave. Prior to constructing the first level of the pyramid, the image is extended using linear interpolation to make the most of the data.

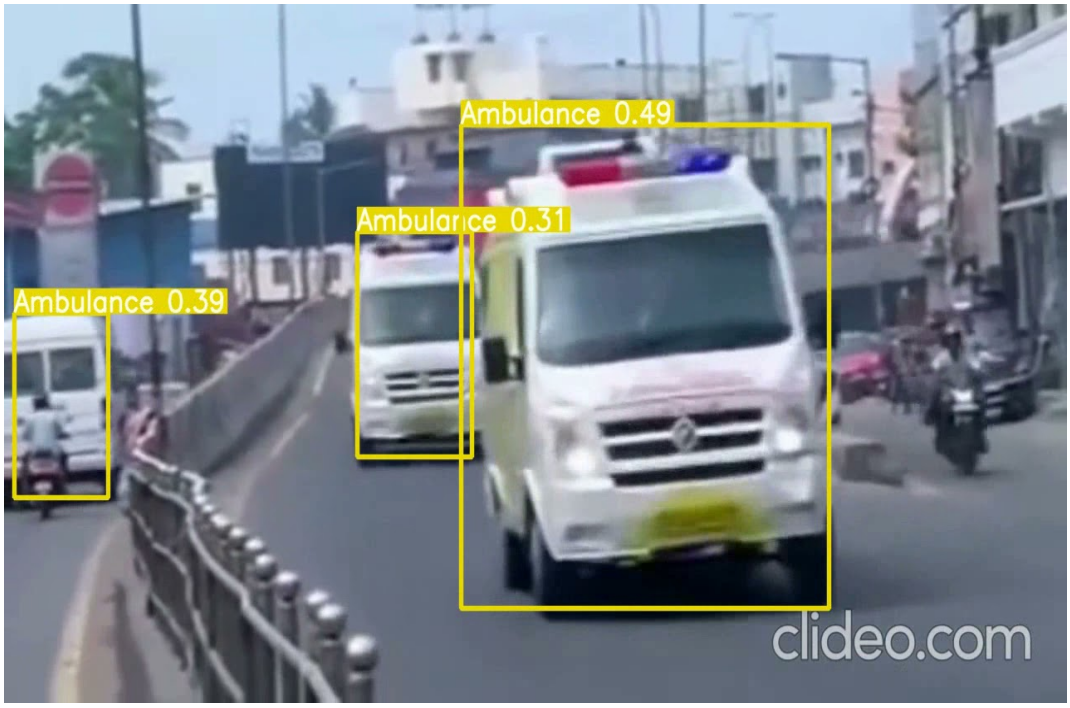


Fig 2.10 Multiple Vehicle detection [14]

The next stage is to create a descriptor for the local image region that is very distinctive while staying as invariant to remaining alterations as feasible, such as lighting or 3D viewpoint. Edelman, Intrator, and Poggio have all proved a better approach.

It shows how to calculate the key point descriptor by constructing orientation histograms spanning 4x4 sample sections, it allows for substantial shifts in gradient positions. The value of each gradient sample is distributed into nearby histogram bins using tri-linear interpolation. A 4x4 array of histograms with 8 orientation bins in each produces the greatest results.

To lessen the effects of changing illumination, the feature vector is adjusted. This indicates that matching magnitudes for big gradients is less critical, and the distribution of orientations is more significant. To change the complexity of a description, you can use two parameters. The number of orientations and the size of the array of histograms are these parameters.

As a function of rotation in depth of a plane away from a viewer, the graph displays the dependability of key point location and scale_selection, orientation_assignment, and nearest_neighbor_matching to a database. None of these methods are actually affine invariant.

Because of their uniqueness, the SIFT key points mentioned in this work are very useful. They allow a key point's correct match to be chosen from a big database of other key points. With near-real-time performance, thousands of key points may be retrieved from a common image. There are numerous avenues for additional investigation into the development of invariant and distinguishing visual properties. On data sets with full 3D viewpoint and illumination changes, systematic testing is required. The invariant local feature technique has the advantage of eliminating the need to choose only one feature type.

We used Adam [16] optimization technique to train the model instead of regular stochastic gradient descent. Adam employs an adaptable learning rate rather than a set learning rate. The learning rate is controlled by two factors, beta1 and beta2. Momentum parameters are what they're termed.

We created a basic 2-layer convolutional neural network to get a baseline accuracy before experimenting with our model. We ran a number of tests using hyper-parameters and regularization approaches. The first computer system based on these local interconnection among neurons and hierarchically arranged picture transformations is Neocognitron.. It claims that translational invariance is obtained when neurons with the same characteristics are applied to regions of the preceding layer at different places.. Later, Yann LeCun and others developed Convolutional Neural Networks that used the error gradient and achieved excellent performance in a variety of pattern recognition applications.

The non_complex model produced the quality result with dimension 128 x 128, as seen in the table above. Although larger images have more information to analyze and produce better results, a shallow convolutional neural network cannot benefit from this since it lacks the parameters and complicated structure to capture such patterns in a large image, can, and give the emergency can priority on that road. No human effort will be necessary to manually assist in such a case with this automatic approach. Our program has produced impressive results in detecting and identifying all types of emergency vehicles.

# CHAPTER 3

# PROBLEM FORMULATION

Object detection's purpose is to locate one or more objects from a still image or video data. It also has a broad application in the areas such as traffic accident prevention, surveillance systems, advanced human-computer interaction, and many more. But it is complex to do and balancing the relationship between accuracy and computing costs is a difficult task.

## 3.1 Problem with traditional methods

Traditionally manual extracting feature models are used for object detection. HOG, SIFT, is the classical algorithms based on grayscale are used for feature representation and then SVM is used as a classifier. These traditional models are only able to extract low-level feature information and not good in case of multiple object detection in a scene. However deep learning-based models like R-CNN(region-based convolution neural networks), YOLO(You only look once) are well Known for computing these complex tasks. Deep learning models are extracting detailed features and are also able to obtain high-level information.

## 3.2 Problem description

We propose a multi-scale convolutional network model to deal with and balance the relationship between speed and accuracy in object detection. The model can generate image features sensitive to object deformation information and also the ability to detect objects that have geometrical deformation is improved. In the final detection feature, fusion operations are performed on the multi-scale feature map. Then the information of different scaled image features maps is used for classification and object position prediction. This model ensures and improves the accuracy of object detection, detection speed and enhances the target information of small information. As a function of rotation in depth of a plane away from a viewer, the graph displays the dependability of key point location and scale selection, orientation assignment, and nearest neighbor matching to a database.

## 3.3 You Only Look Once (YOLO) - Network Architecture

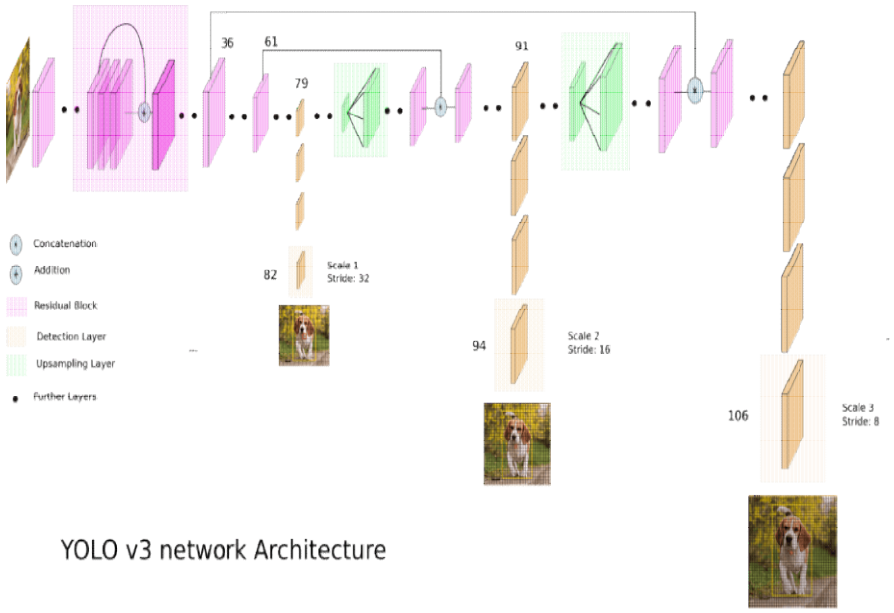


Fig 3.1 YOLO V3 Network Architecture [15]

1. The inputs are a collection of images.
2. This image is sent to a convolutional neural network by YOLO v3 (CNN).
3. The output volume of (19, 19, 425) is obtained by flattening the final two dimensions of the above output:
4. A 19 × 19 grid returns 425 numbers in each cell.
5. The number of anchor boxes for each grid is 5, hence 425 = 5 * 85.
6. 85 = 5 + 80, where 5 represents (pc, bx, by, bh, bw) and 80 represents the number of classes to detect.

The result is a list of bounding boxes together with the classes that have been identified. Six numbers are assigned to each bounding box (pc, bx, by, bh, bw, c). When c is expanded into an 80-dimensional vector 85 values represent each bounding box. Finally, to prevent selecting overlapping boxes, we use the IoU (Intersection over Union) and Non-Max Suppression techniques.

## 3.4 Objectives

1. Improving object classification and localization**:** Models can classify as well as locate the position of objects with high accuracy. The fact that object detection has a secondary aim is the first big problem. The object localization challenge entails classifying visual items as well as determining their locations. To tackle this problem, researchers usually use a multi-task loss function to penalize both misclassifications and localization errors.

2. Improving Speed for real-time detection : To fulfill the real-time needs of video processing, object detection algorithms must not only reliably classify and localize relevant objects, but they must also be extremely fast at prediction time. Over the years, several major improvements have improved the performance of these algorithms, bringing the test time from 0.02 frames per second (fps) for R-CNN to an astonishing 155 fps for Fast YOLO.

3. Building a multiple scale object detection model : Items of interest in various object detection applications may come in a variety of sizes and aspect ratios. Practitioners apply a variety of ways to ensure that detection algorithms can capture things at various scales and angles.

4. Removing class imbalance problem: In most classification issues, the class imbalance is a concern, and object detection is no exception. Take a look at a typical photograph. The snapshot most likely only features a few key things, with the rest of the image being filled with background. Remember that R-selective CNN's search generates 2,000 potential ROIs in each image–imagine how many of these regions empty and so considered negatives are!

# CHAPTER 4

# PROPOSED WORK

We employed deep learning techniques to recognize items in real time. Object detection is widely employed in a variety of fields, such as auto driving cars, video surveillance, and so on. The purpose of this research is to reduce the time spent on real-time object detection and develop an efficient model for it. Convolutional neural networks (CNNs) are used in Deep Learning-based approaches for unsupervised end-to-end object detection that does not require any predefined features. Deep learning is the most recommended approach for object detection due to its high accuracy. We gather a variety of image data featuring a variety of distinct things in order to train our model for accuracy.

## 4.1 Introduction

In this experiment we use yolov3 as a feature extractor by removing the detection layer from its architecture. Instead of detection at 3 different scale in YOLO we add a block of layers at these 3 different scale which consist of a flatten layer then 3 dense layer and two normalization layer between dense layers. Then all the output of the different scale were concatenated and then pass through 1 dense, 1 normalization and at the end soft max layer which used for detection purpose. Flatten layer convert the multidimensional feature in to a single dimensional feature map which is further given as a input to dense layers. Normalization layer are for normalizing the feature map value in the range of 0-1. At the end we are using a soft max layer which can detect 10 different class of vehicle.

In this experiment we use yolov3 as a feature extractor by removing the detection layer from its architecture. Instead of detection at 3 different scale in YOLO we add a block of layers at these 3 different scale which consist of a flatten layer then 3 dense layer and two normalization layer between dense layers. Then all the output of the different scale were concatenated and then pass through 1 dense, 1 normalization and at the end soft max layer which used for detection purpose. Flatten layer convert the multidimensional feature in to a single dimensional feature map which is further given

as a input to dense layers. Normalization layer are for normalizing the feature map value in the range of 0-1. At the end we are using a soft max layer which can detect 10 different class of vehicle.

Real-time object detection is a large, dynamic, and challenging topic of computer vision. The detection of a single object in an image is referred to as image localization, whereas the detection of several objects in an image is referred to as object detection. This recognizes semantic objects of a class in photographs and videos. Real-time object detection is widely available and uses by multiple applications, have items track feature, surveillance videos, counting persons, self drive automobiles, face matcher, and track sports ball. With OpenCV, a set of tools for write program primarily for real time vision in computer, convolution neural networks is a Deep Learning technique for recognizing objects. In this we do an extension of well know detection algorithm Yolov3 to detect 10 different classes of objects in a screen. We use Yolov3 as a feature extractor and added a dense layer block after it. Which uses yolov3 learning to detect respective classes.

## 4.2 Proposed Methodology/Algorithm

Deep Learning a subset of machine learning which a subset of artificial intelligence is has networks which are capable of learning things from the unstructured or unlabeled data. The approach utilized in this project is YOLO (You Only Look Once).

Yolo is a great example of single stage detectors which is less accurate than two stage detectors but is significantly faster. It introduce in 2015 by Redmon etal. Number of different iterations has gone by YOLO and it is capable of detecting over 9000 objects.

YOLO performs feature learning as a regression problem and returns the posterior distribution of the discovered photos. To identify location of an object, the YOLO method leverages convolution_neural networks (CNN). To identify objects, the strategy requires just one forward passage through a neural network, as the name implies.. This suggests that the complete image is predicted in a single algorithm run.

The CNN is used to forecast several posterior distribution and bounding boxes at the same time.

YOLO is orders of magnitude faster than any other object detection technique (45 frames per second). The YOLO algorithm's drawback is that it has trouble identifying small things in images; for example, it might have trouble distinguishing a flock of birds. This is owing to the algorithm's spatial restrictions.

Feature pyramids are employed in object identification networks to produce probability of non identical attribute scale or the concatenation of several measure of information. YOLO v3 predicts on three separate scales with strides of 32, 16, and 8. In other words, given a 416 x 416 input image, it generates predictions on 13-X-13, 26-X- 26, and 52-X-52 scales. At each level, all features are rescaled and their channel counts are changed. Assume we've got a 416 × 416 input image, and that we need to concatenate attribute at level 2 (attribute map size is 26-X-26 and also the channel count is 512) with attribute at level 3 of high resolve (where map of attribute sized is 26-X-26 and that's why the channel count is 512). (Have resolution of 52-X-52, channel count 256). The layer_ is get downscaled to 26-X-26 pixels, with the amount increase in channel to 512. The lower resolutions attribute at level 1 with 13-X-13 resolve, channel count 1024, on the opposite hand, be sampled up to 26-X-26 and also the channel counting lowered till 512. Up scaling is accomplished by first compressing the amount of channels of features with a 1-X-1 layers of convolve, so up scaling with interpolation. A 3 by 3 layers of convolve have stride of two employed to alter the channels amounts and also the resolution at identical time for down-sampling with a 1/2 ratio. Before the 2-stride convolution, a two stride layers of max pooling is used for dimensions ratio of 1/4.

Yolo is great examples of single stage detectors which is less accurate than two stage detectors but are significantly faster. It introduce in 2015 by Redmon etal. Number of different iterations has gone by YOLO and it is capable of detecting over 9000 objects.

YOLO performs feature learning as a regression problem and returns the posterior distribution of the discovered photos. To identify location of an object, the YOLO

method leverages convolution_neural networks (CNN). To identify objects, the strategy requires just one forward passage through a neural network, as the name implies.. This suggests that the complete image is predicted in a single algorithm run. The CNN is used to forecast several posterior distribution and bounding boxes at the same time.

YOLO is orders of magnitude faster than any other object detection technique (45 frames per second). The YOLO algorithm's drawback is that it has trouble identifying small things in images; for example, it might have trouble distinguishing a flock of birds. This is owing to the algorithm's spatial restrictions.

In this experiment we use yolov3 as a feature extractor by removing the detection layer from its architecture. Instead of detection at 3 different scale in YOLO we add a block of layers at these 3 different scale which consist of a flatten layer then 3 dense layer and two normalization layer between dense layers. Then all the output of the different scale were concatenated and then pass through 1 dense, 1 normalization and at the end soft max layer which used for detection purpose. Flatten layer convert the multidimensional feature in to a single dimensional feature map which is further given as a input to dense layers. Normalization layer are for normalizing the feature map value in the range of 0-1. At the end we are using a soft max layer which can detect 10 different class of vehicle.

The detection of a single object in an image is referred to as image localization, whereas the detection of several objects in an image is referred to as object detection. This recognizes semantic objects of a class in photographs and videos. Real-time object detection is widely available and uses by multiple applications, have items track feature, surveillance videos, counting persons, self-drive automobiles, face matcher, and track sports ball. With OpenCV, a set of tools for write program primarily for real time vision in computer, convolution neural networks is a Deep Learning technique for recognizing objects. In this we do an extension of well know detection algorithm Yolov3 to detect 10 different classes of objects in a screen. We use Yolov3 as a feature extractor and added a dense layer block after it. Which uses yolov3 learning to detect respective classes.

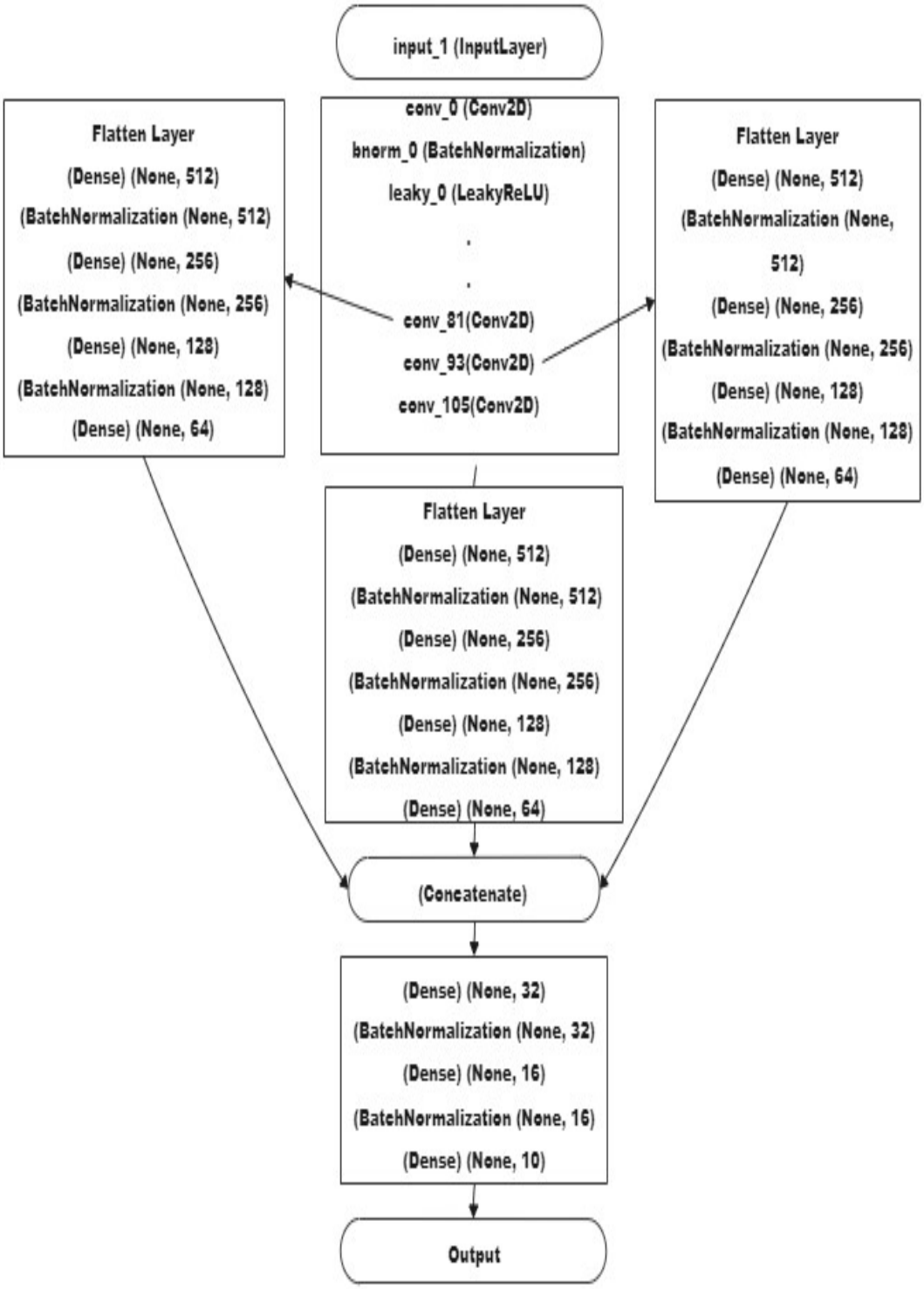


Fig 4.1 Functional Flow of each layer

Transfer learning is a method in which we can use the learning of a predefined model which is trained over well optimize large datasets. So this process helps to utilize the learning of pre trained model and we have no need to train a new model from scratch which reduce training time. In this process we use the pre model as feature extractor and freeze all the layers of that model so that its weight cannot be changed. Then we attached the extended layers and these layers are unfreeze layers and their weight get updated at the time of training.

Transfer learning in machine learning refers to the employment of a previously trained model on a new task. In transfer learning, a computer leverages past assignment expertise to improve prediction about a new task.

During transfer learning, the information of a previously trained machine learning model is transferred to a separate but closely related task. If you trained a basic classifier to predict if a picture contains a backpack, you might utilize the model's training experience to identify additional items like sunglasses.

Transfer learning has several advantages, the most prominent of which are shorter training times, enhanced neural network performance (in most cases), and the lack of a huge amount of data.

A large amount of information is gradually required to trained a neural_network from start, but access to that data isn't always feasible — this is when transfer learning comes in useful. Transfer learning may build an effective machine learning model with relatively minimal training data since the model has previously been pre-trained. This is particularly beneficial in natural language processing, where large tagged datasets need a great deal of expert knowledge. Furthermore, training time is reduced because constructing a deep neural network from the beginning of a challenging job might take days or even weeks.

When there is insufficient labeled information to build our model. When a pre-trained model which has been trained on similar information and tasks is available. If you trained the original model with TensorFlow, you could made the process it and retrain certain layers for your job.

## 4.3 Description of each step

Depiction of Algorithm with the help of flowchart and give description of all steps

YOLO algorithm works using following three techniques:

1. Residual Blocks

   The image is first separated into several grids. S x S is the dimensions of each grid. The graphic below shows how a grid is created from an input image.

   There are several grid cells of identical size in the image above. Objects that appear within grid cells will be detected by each grid cell. If an item center emerges within a specific grid cell, for example, that cell will be responsible for detecting it.

2. Bounding box regression

   Outline that highlights an object in an image is termed as bounding box.

   In YOLO single bounding box regression to predict the height, width, center, and class of objects.

   Every bounding box consist of following attributes:

   - Width
   - Height
   - Class
   - Bounding box center

3. Intersection over union (IOU)

   It is a phenomenon in object detection that describes overlapping of boxes. YOLO uses IOU in order to find the best fitting boundary box and eliminating all other less feasible overlap boxes.

Bounding boxes and their confidence score are predicted for each grid cell. After evaluation of IOU if it is equal to 1 then the bounding box is the same as real box. This can eliminate all the box not equal to real box.

$$IOU = \frac{\text{Area of intersection regions of real and predicted box}}{\text{Area of Union regions of real and predicted box}}$$

The image is first subdivided into grid cells. B bounding boxes are forecasted in each grid cell, along with their confidence scores. To determine the class of each object, the cells estimate the class probability.

We can see at least three types of objects, for example, a car, a dog, and a bicycle. A single convolutional neural network is used to make all of the predictions at the same time.

The predicted bounding boxes are equal to the true boxes of the objects when intersection over union is used. This phenomenon gets rid of any extra bounding boxes that don't fit the objects' properties (like height and width). The final detection will be made up of distinct bounding boxes that exactly suit the objects.

The above methods are the method which is used to detect object using yolo but we did a architectural change in yolo model. The yolo is a model or algorithm which is a multiscale model and provide out at three different layers these layers. So instead of using the detection layer of yolo we only use the feature map provided by yolo from these three different layer individually and then pass these feature maps to the newly added dense layer.

We also use flatten layer before dense layer because the feature map provided by YOLO layers are in 2 dimensional state so flatten layer will convert that into a single dimension which can be used as an input for the dense layers. The dense layer after each flatten layer is consist of 512 nodes which is fully connected and the we also use batch normalization layers between these dense layers so that it can normalize the

input in range of 0-1 to reduce the computation of model and it also helps to increase the model speed.

Then all the output from three different layers are get concatenated at a single point by using a concatenation layer then the output of concatenate layer is used as an input for the next dense layer consist of 32 fully connected nodes. The activation function which is used throughout the process is RELU which is a efficient activation function for the object detection purpose. The output from the dense layer consist 16 nodes are then pass to the final output layer which is a soft max layer consisting 10 nodes which can be used to detect and classify 10 different class of vehicle. These class are ambulance, fire fighter vehicle, Army trucks, police car, JCB, police bike, Road Roller , Government official vehicle, armor trucks.

# CHAPTER 5

# SYSTEM DESIGN

The end-to-end deliverable includes a frontend with the primary goal of accepting video streams as input data and then the data is sent to the backend. The backend processes the data before sending it to the deep learning object detection module. Using the frontend, the object detection model analyses the data in real-time and shows discovered objects as bounding boxes with name labels.

## 5.1 Functional Specification of System

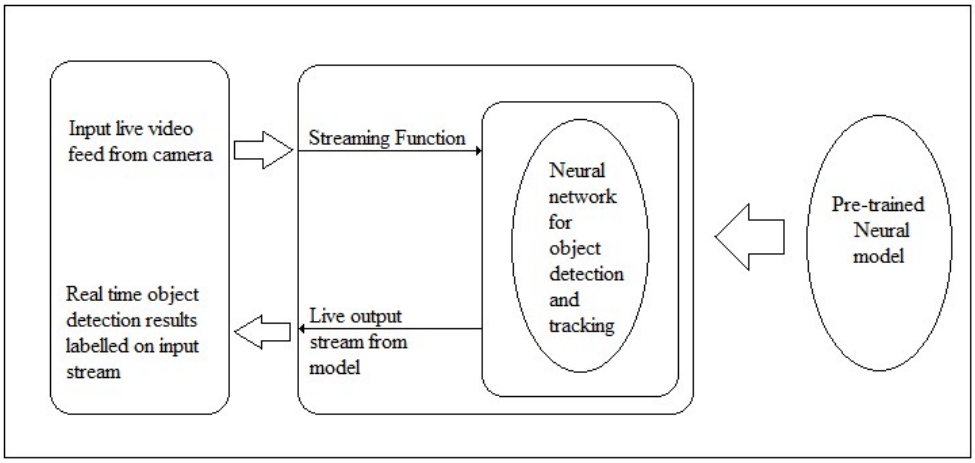


Fig 5.1 Functional Structure of Real-Time Detection System

The predicted bounding boxes are equal to the true boxes of the objects when intersection over union is used. This phenomenon gets rid of any extra bounding boxes that don't fit the objects' properties (like height and width). The final detection will be made up of distinct bounding boxes that exactly suit the objects. We also use flatten layer before dense layer because the feature map provided by YOLO layers are in 2 dimensional state so flatten layer will convert that into a single dimension which can be used as an input for the dense layers.

## 5.2 Structural and Dynamic Modeling of System

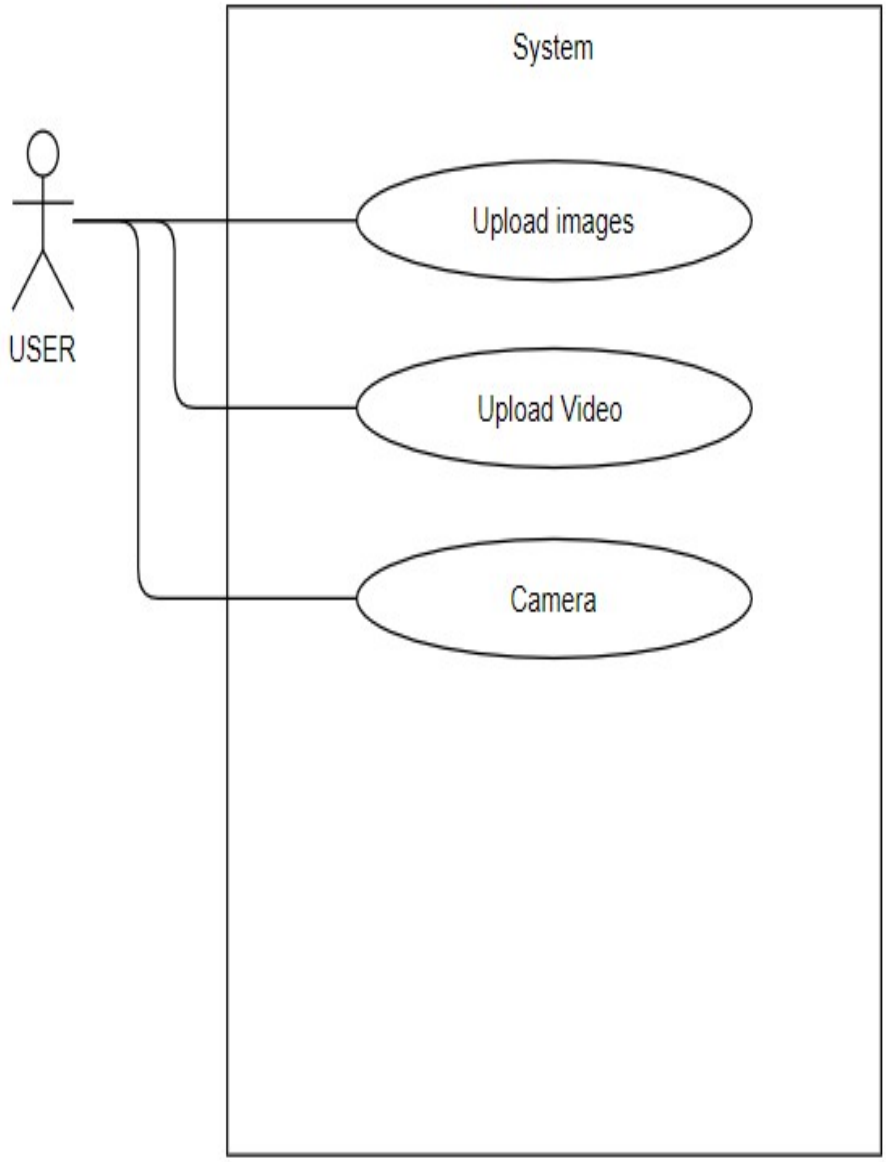


Fig 5.2 Use case Diagrams Real-Time Detection System

YOLO predicts both bounding boxes and class probabilities. MultiBox can also conduct single object detection by substituting a single class prediction for the confidence prediction.

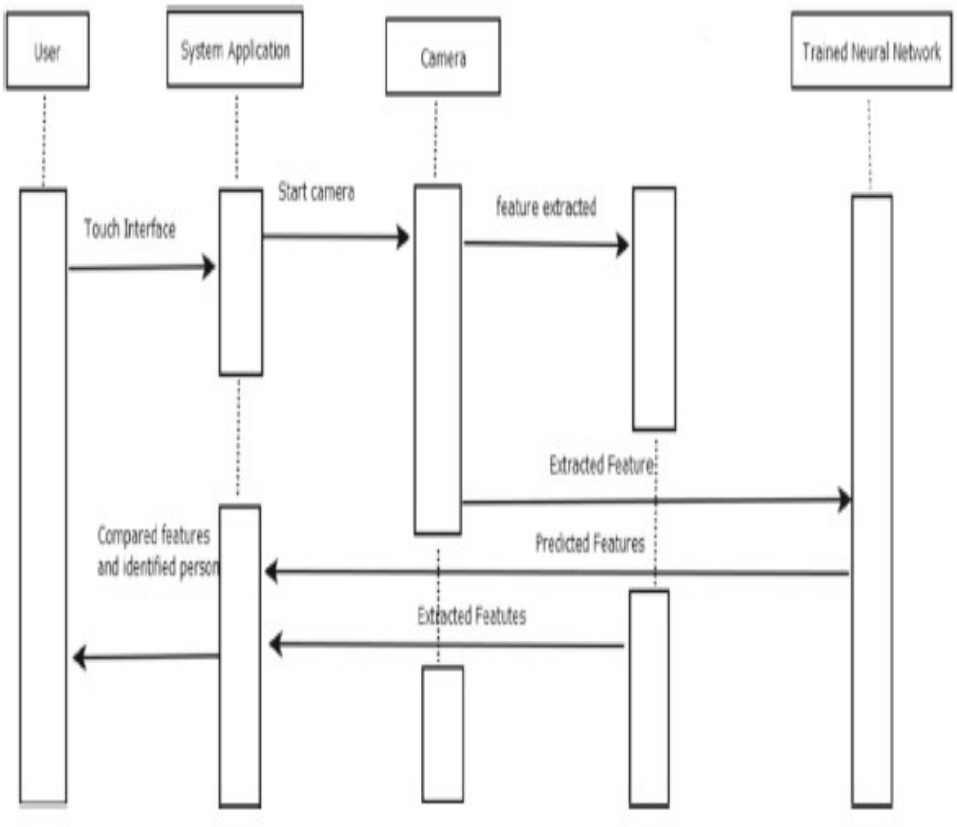


Fig 5.3 Sequence Diagram Real-Time Detection System

The model can generate image features sensitive to object deformation information and also the ability to detect objects that have geometrical deformation is improved. In the final detection feature, fusion operations are performed on the multi-scale feature map. Then the information of different scaled image features maps is used for classification and object position prediction. This model ensures and improves the accuracy of object detection, detection speed and enhances the target information of small information. As a function of rotation in depth of a plane away from a viewer, the graph displays the dependability of key point location and scale selection, orientation assignment, and nearest neighbor matching to a database. It also has a broad application in the areas such as traffic accident prevention, surveillance systems, advanced human-computer interaction, and many more. But it is complex to do and balancing the relationship between accuracy and computing costs is a difficult task.

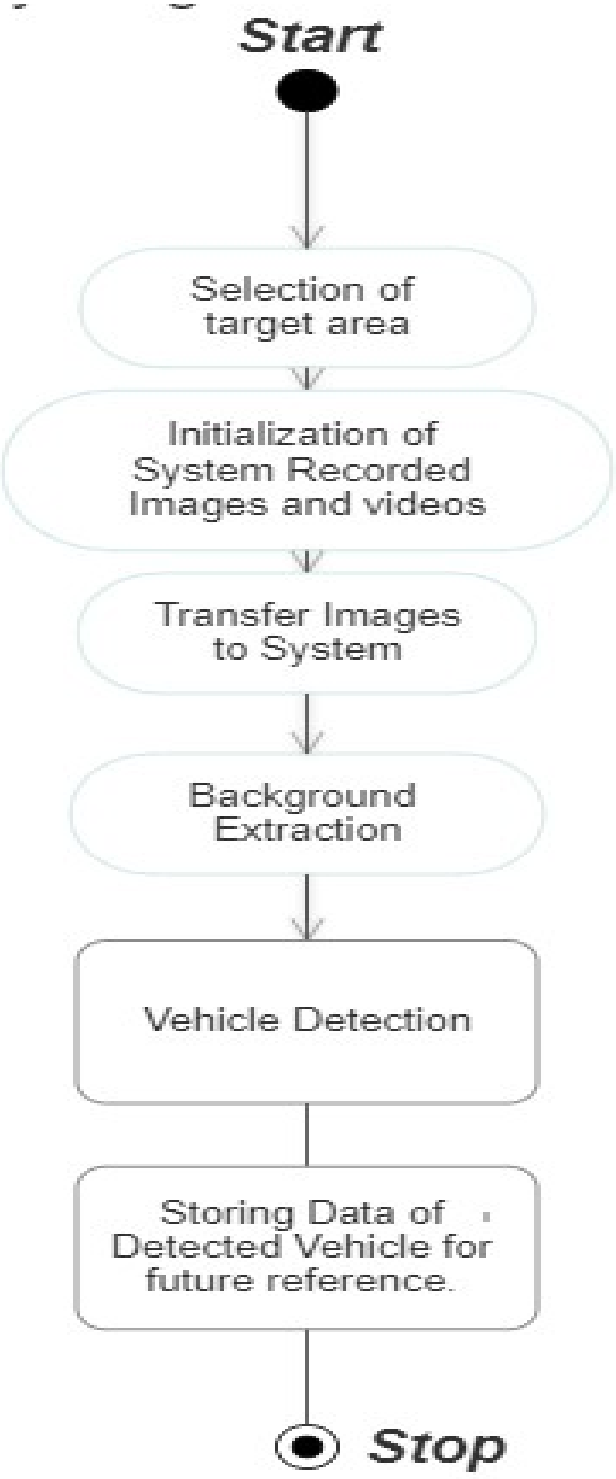


Fig 5.4 Activity Diagrams Real-Time Detection System

# CHAPTER 6

# IMPLEMENTATION

YOLO is the world's fastest general-purpose object detector, pushing the boundaries of real-time object detection. YOLO may be trained on a loss function that directly reflects its performance, making it perfect for applications that require reliable, rapid, and reliable object detection. YOLO is a multi-object detector that learns to detect a variety of items at the same time. For many items of multiple classes in an image, YOLO predicts both bounding boxes and class probabilities. MultiBox can also conduct single object detection by substituting a single class prediction for the confidence prediction.[7]

## 6.1 Experimental Setup

In our project experimental setup consists of algorithms used, software used and datasets to build our project. Experimental setup is described in below sections.

### 6.1.1 Algorithms/techniques used

YOLO is an algorithm that recognizes and detects different things in an image (in real-time). Object detection in YOLO is done as a regression problem, and the identified photos' class probabilities are provided. Convolutional neural networks (CNN) are used in the YOLO method to recognize objects in real time. To detect objects, the approach just takes a single forward propagation through a neural network, as the name suggests. This indicates that a single algorithm run is used to forecast the entire image. The CNN is used to forecast multiple bounding boxes and class probabilities at the same time.

### 6.1.2 Software tools used

Python is a powerful, interpretive, dynamic, and entity-oriented scripting language. Python is designed to be a convenient programming language. It employs English terminology rather than punctuation and has limited vocabulary features than in other languages.

OpenCV is a cross-platform framework for developing authentic computer vision applications. Its primary functions are image enhancement, recorded video, and evaluation, including features such as face identification and object detection.

Google Colab is the platform which is used for training and testing the model on the custom dataset. GPU( Graphic processing unit ) used in this process is Nvidia it saves time in training the model. It is Free GPU provide by the Google Colab platform.

LabelImg is a tool which can be used to label the images in the form of input format required by the YOLO model which consist of the class number, width and height of image and the value of x and y coordinate. It is an open source software tool which is freely available on github we can clone its repository and use it. It provide a easy to use user friendly interface.

Tensorflow is a python library which we used in this project. This library consist of multiple pre-trained optimize models which can be directly used for experimental purpose and there is no need to make model from scratch. It's also providing multiple functionality which we can use for detection of objects and manipulating data.

## 6.2 Dataset Description

Dataset consists of MS COCO dataset which has multiple classes of objects. Custom dataset consists of few classes like mobile phones, ambulance, etc. In this experiment we use two different custom dataset 1$^{st}$ consist of only a single class which is a cell phone dataset and train our custom YOLO model to see change in various accuracy measure change with the number of iteration. There are 109 mobile phone photos in the Custom Dataset. It is a single-class cell phone that uses a labelImg tool to label photographs in the Yolo input format.

### 6.2.1 Source of Dataset

MS COCO is a large-scale object detection dataset that tackles three basic scene recognition analytic problems. Object detection of non-iconic scenes (or non-canonical views), contextual reasoning inside objects, and precise 2D localization.

In this experiment we use two different custom dataset 1st consist of only a single class which is a cell phone dataset and train our custom yolo model to see change in various accuracy measure change with the number of iteration. There are 109 mobile phone photos in the Custom Dataset. It is a single-class cell phone that uses a labelImg tool to label photographs in the Yolo input format.

The 2nd dataset is also a custom dataset which consist of 10 different vehicle classes consist 800 training images. We use this dataset for training out yolov3 extended model which use yolov3 as a feature extractor and with some additional dense and normalization layer and it uses the learning of yolov3 to make a detection of new objects.

Fig 6.1 Custom dataset of 10 different class of vehicle [16]

### 6.2.2 Size (No. of Samples) and description of attributes

We use the Coco dataset, which is a preprocessed dataset, as well as a new dataset in this experiment. The custom dataset is made up of a single cell phone class. This class has over 100 photos for training the Yolov3 model, and Coco is large-scale object identification and segmentation dataset that Microsoft released in 2015. The Common

Things in Context (COCO) dataset contains photos of everyday objects collected in daily contexts. There are 80 different types of objects in this collection.

In this experiment we use dataset is a custom dataset which consist of 10 different vehicle classes consist 800 training images. We use this dataset for training out yolov3 extended model which use yolov3 as a feature extractor and with some additional dense and normalization layer and it also uses the learning of yolov3 to make a detection of new objects. The vehicle class it detects are ambulance, fire fighter vehicle, Army trucks, police car, JCB, police bike, Road Roller , Government official vehicle, armor trucks.

# CHAPTER 7

# RESULT ANALYSIS

Result Analysis for our project consists of Precision, Recall, F1-Measure, AVG-IOU and MAP which helps us to understand our model accuracy with the datasets. These performance measures are described in below sections. YOLO model is computationally fast, inexpensive and well suited for the large databases. The model require a high computing device for training so its performance is get affected in case of non-gpu devices.

## 7.1 Performance Measures

Performance measures are as follows

1. Precision

   Precision = TP/ (TP + FP), the number of genuine positives as a percentage of the total number of true positives and false positives To put it another way, precision is the percentage of correctly detected things.

2. Recall

   Recall = TP/ (TP + FN), the number of genuine positives as a percentage of the total number of true positives and false negatives The recall is the percentage of things accurately detected out of all those that should have been detected.

3. F1-Measure

   F-Measure = $(1 + b^2)$ x (Precision * Recall) / ($b^2$ x Precision + Recall), where $b^2$ is a non-negative real valued weighting factor (here b = 1). The F-measure provides a rough assessment of the system's accuracy. F-Measure produces a unique rating that incorporates accuracy and memory issues into a single value.

4. AVG-IOU

   When computing mAP, intersection over union (IoU) is employed. It's a value between 0 and 1 that indicates how much the expected and ground truth bounding boxes overlap.

5. MAP

   The average mean perfection (mAP) is often used to determine the accuracy when a series of object detection systems from a model is compared to ground-truth object labels in a collection.

## 7.2 Result Analysis

The result shows the detected image which is an ambulance by the trained yolov3 model on the custom dataset consisting single ambulance class.

### 7.2.1 Result for dataset consisting 10 class of vehicle using extended yolov3.

Fig 7.1 represent the variation of accuracy with the number of epochs upto which the model is trained and we can se that we can achieve a accuracy of around 65.92% under 30 epocs by using our extended yolov3 model and transfer learning on out testing dataset.

Table 7.1 shown the result given by the extended yolov3 model on the dataset2 consist 10 different classes of vehicle. In which we are using transfer learning approach and use yolov3 as a feature extractor.

We freeze the yolov3 portion at the time of training and train the newly added layers only. This model ensures and improves the accuracy of object detection, detection speed and enhances the target information of small information So that we can use the learning of yolov3 model to detect our classes. Which can reduce time for training and we can make a detector by training it on small dataset also. By this we can achieve a descent accuracy and a fast trained model for detection. This model ensures and improves the accuracy of object detection, detection speed and enhances the target information of small information.

| Approaches | No. of epochs | Recall | F1-Score | AVG-IOU | MAP |
|---|---|---|---|---|---|
| YOLOv3 | 800 | 0.33 | 0.47 | 55.77 % | 49.34 % |
| YOLOv3 | 900 | 0.26 | 0.41 | 71.11 % | 42.07 % |
| YOLOv3 | 1000 | 0.26 | 0.37 | 41.29 % | 33.30 % |
| YOLOv3 | 2000 | 0.36 | 0.52 | 77.60 % | 48.82 % |
| **Proposed Method** | **30** | **0.44** | **0.59** | **79.54 %** | **65.92** |

TABLE 7.1 Accuracy Result using extended yolov3 on custom vehicle dataset.

Fig 7.2 represent the image of the output provide by the modified YOLO model which predict that the image consist of an ambulance and surrounding that ambulance by a bounding box.

Fig 7.3 is also representing the output by the model which is an image of an ambulance in the traffic and the model easily identified ambulance in the traffic.

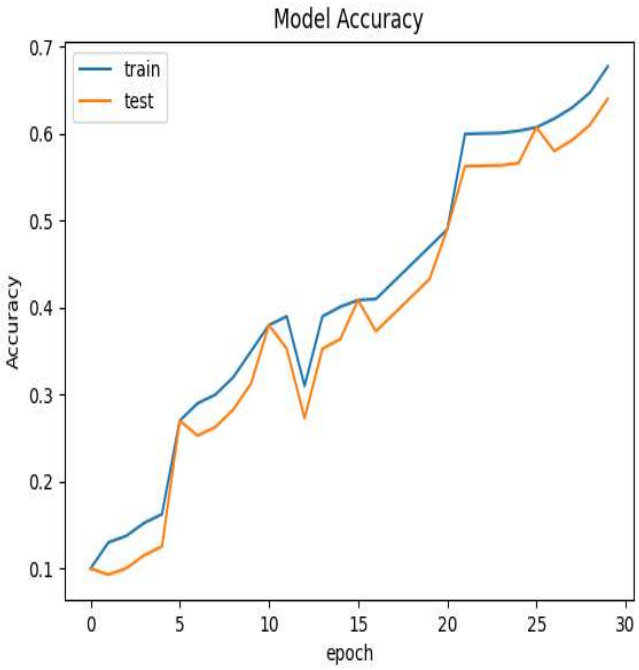


Fig 7.1 Graph shows model accuracy with respect to epochs


Fig.7.2 Predicted result by Extended model [17]

Fig 7.4 is representing the output provided by the extended or modified yolov3 model which can accurately predicted the respective vehicle images into 10 different categories like ambulance, fire fighter, Gov official vehicle and other respective classes.


Fig 7.3 Model Predict the Ambulance in the traffic [18]

```
[[8.6280698e-01 4.4364872e-04 1.2764307e-04 2.3444109e-03 1.0480608e-01
  2.4583612e-02 1.4328732e-03 8.6584267e-05 8.7538554e-04 2.4928388e-03]]
0
9
6
5
8
Ambulance
Road_roller
JCB
Government_Official_vehicle
Police_bike
```

Fig 7.4 Examples of detection of vehicle of 10 different classes.

# CHAPTER 8

# CONCLUSION, LIMITATION AND FUTURE SCOPE

When comparing to other item detection methods, this methodology produces better detection performance such as SIFT, Shape context and Hog based approach. The model is a multi-scale model which can classify and detect objects of different scales in a single scene. YOLO model is computationally fast , inexpensive and well suited for the large databases. The model require a high computing device for training so its performance is get affected in case of non-gpu devices. Accuracy measuring factors such as precision, maP, F1-Score, Recall and Avg Iou is not dependent on the number of iteration or training time they are not propositional to the number of iteration.

## 8.1 Conclusion

When compared to other object detection algorithms such as SIFT, Shape context, and Hog based approaches, this strategy delivers better detection results. The model is a multi-scale model that can recognize and classify items of various scales in a single scene. The YOLO model is computationally efficient, low-cost, and well-suited to big databases. Because the model requires a powerful computer processor for training, non-GPU devices' performance suffers. Precision, maP, F1-Score, Recall, and Avg Iou are not proportional to the number of iterations or training period, and they are not proportional to the number of iterations.

We describe a fast video object recognition technique in this paper. We highlight our algorithm's performance in terms of detection speed, with an amazing efficiency of more than video frames per second and high detection accuracy. We improve the convolution operation instead of tiny convolution based on the You Only Look Once (YOLO) network to reduce the amount of calculation and considerably speed up detection. We remove the effects of the environment and noise as the image progresses. Furthermore, we do not need to construct any size of boxes as the background's bounding boxes, reducing the background's analysis.

When compared to the other algorithm Yolov3 is a faster and precise algorithm for detection and it is already trained over a large preprocessed dataset coco so instead of making a new model from scratch we can use the learning of these highly efficient models like YOLO by use of transfer learning. This approach require less computation power and can be used with small datasets. When compared to other object detection approaches, this technique delivers better detection results SIFT, Shape context and Hog based approach.. Accuracy measuring factors such as precision, maP, F1-Score, Recall and Avg Iou is not dependent on the number of iteration or training time they are not propositional to the number of iteration.

## 8.2 Limitation

Our strategy, however, has certain drawbacks. To begin with, existing datasets only have a single category of cell phones, limiting their potential to generalize in engineering applications. We'll collect a variety of datasets, including everyday household items like bags and bottles. Second, compared to Faster R CNN, it has lower recall and higher localization error, and it struggles to detect close and small objects because each grid can only suggest two bounding boxes.

## 8.3 Future Scope

In the coming time, we will solve the issue of an not-balance between positives and negatives samples when the majority of samples are negative by using approaches such as re-sampling and voting. We will also examine multi-scale parallel processing for our model in order to increase detection speed and better optimize the network topology, as we currently have issues with a huge computational load and dense target objects.

# REFERENCES

**[1] Wang, Jintao, Wen Xiao, and Tianwei Ni**. "Efficient object detection method based on improved YOLOv3 network for remote sensing images." *2020 3rd International Conference on Artificial Intelligence and Big Data (ICAIBD)*. IEEE, 2020.

**[2] Lu, Yonghui, Langwen Zhang, and Wei Xie**. "Yolo-compact: An efficient yolo network for single category real-time object detection." *2020 Chinese Control And Decision Conference (CCDC)*. IEEE, 2020.

**[3] Redmon, Joseph, et al**. "You only look once: Unified, real-time object detection." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.

**[4] Jin, Ying-Hui, et al**. "Chemoprophylaxis, diagnosis, treatments, and discharge management of COVID-19: An evidence-based clinical practice guideline (updated version)." *Military Medical Research* 7.1 (2020): 1-33.

**[5] Htet, Khaing Suu, and Myint Myint Sein**. "Market Intelligence Analysis on Age Estimation and Gender Classification on Events with deep learning hyperparameters optimization and SDN Controllers." *2020 IEEE Conference on Computer Applications (ICCA)*. IEEE, 2020.

**[6] Murugan, V., V. R. Vijaykumar, and A. Nidhila**. "A deep learning RCNN approach for vehicle recognition in traffic surveillance system." *2019 International Conference on Communication and Signal Processing (ICCSP)*. IEEE, 2019.

**[7] Dalal, Navneet, and Bill Triggs**. "Histograms of oriented gradients for human detection." *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*. Vol. 1. Ieee, 2005.

**[8] Belongie, Serge, Jitendra Malik, and Jan Puzicha**. "Shape matching and object recognition using shape contexts." *IEEE transactions on pattern analysis and machine intelligence* 24.4 (2002): 509-522.

**[9] Lowe, David G**. "Object recognition from local scale-invariant features." *Proceedings of the seventh IEEE international conference on computer vision*. Vol. 2. Ieee, 1999

**[10] Roy, Shuvendu, and Md Sakif Rahman**. "Emergency vehicle detection on heavy traffic road from cctv footage using deep convolutional neural network." *2019 International Conference on Electrical, Computer and Communication Engineering (ECCE)*. IEEE, 2019.

**[11] Mauri, Antoine, et al.** "Real-time 3D multi-object detection and localization based on deep learning for road and railway smart mobility." *Journal of imaging* 7.8 (2021): 145.

**[12] Meng, Lingxuan, et al.** "Real-time detection of ground objects based on unmanned aerial vehicle remote sensing with deep learning: Application in excavator detection for pipeline safety." *Remote Sensing* 12.1 (2020): 182.

**[13] Kido, Daiki, Tomohiro Fukuda, and Nobuyoshi Yabuki.** "Diminished reality system with real-time object detection using deep learning for onsite landscape simulation during redevelopment." *Environmental Modelling & Software* 131 (2020): 104759.

**[14] Akyol, Gamze, et al.** "Deep Learning Based, Real-Time Object Detection for Autonomous Driving." *2020 28th Signal Processing and Communications Applications Conference (SIU)*. IEEE, 2020.

**[15] Shanahan, James G., and Liang Dai.** "Introduction to computer vision and real time deep learning-based object detection." *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2020.

**[16] Koskowich, Bradley J., Maryam Rahnemoonfai, and Michael Starek.** "Virtualot—A Framework Enabling Real-Time Coordinate Transformation & Occlusion Sensitive Tracking Using UAS Products, Deep Learning Object Detection & Traditional Object Tracking Techniques." *IGARSS 2018-2018 IEEE International Geoscience and Remote Sensing Symposium*. IEEE, 2018.

**[17] Zhang, Jinfeng, et al.** "Real-time quadrilateral object corner detection algorithm based on deep learning." *2019 Computing, Communications and IoT Applications (ComComAp)*. IEEE, 2019.

**[18] Zhao, Yiming, et al.** "Robust real-time object detection based on deep learning for very high resolution remote sensing images." *IGARSS 2019-2019 IEEE International Geoscience and Remote Sensing Symposium*. IEEE, 2019.

**[19] Hammedi, Wided, et al.** "Deep learning-based real-time object detection in inland navigation." *2019 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2019.

**[20] Vaidya, Bhaumik, Harendra Panchal, and Chirag Paunwala.** "Silhouette-Based Real-Time Object Detection and Tracking." *Proceedings of 2nd International Conference on Computer Vision & Image Processing*. Springer, Singapore, 2018.

# LIST OF PUBLICATION

**[1] N. Singh, A.K. Bhartee, A. Prajapati, A. Yadav 2022** "Real Time Object Detection using Deep Learning" – Accepted at IEEE International Conference on Advances in Computing, Communication Control and Networking (ICAC3N-22), IEEE, 2022.

# APPENDIX:

## A. CONTRIBUTION OF PROJECT

1. **Objective and Relevance of Project**

   Improving Speed for real-time detection : To fulfill the real-time needs of video processing, object detection algorithms must not only reliably classify and localize relevant objects, but they must also be extremely fast at prediction time. Over the years, several major improvements have improved the working of these algorithms, bringing the test time from 0.02 frames per second (fps) for R-CNN to an astonishing 155 fps for Fast YOLO.

   Building a multiple scale object detection model : Items of interest in various object detection applications may come in a variety in sizes and aspect_ratios. Practitioners apply a variety of ways to ensured that detection algorithms can catch things at various scales and angles.

2. **Deliverables**

   A three-part output is produced by an object detection model: If you're utilizing the COCO file format, the bounding boxes are x1, y1, width, and height. This is the bounding box's class. The probability scores for that prediction, which indicates how confident the model is that the class is the one anticipated.

   We have also published research paper on this report “Real Time Object Detection using Deep Learning” in IEEE International Conference on Advances in Computing, Communication Control and Networking (ICAC3N-22) Program Committee, IEEE, 2022 which is also accepted.

3. **Concerns Related to Project**

   **Social Relevance:**

   According to the economic growth theorist, Improving people's material well-being is

one of the broad social impacts of improving computer vision. Developing various applications of computer vision necessitates transforming perceptual and interpretation knowledge into new gadgets that broaden and deepen human capabilitie. With increasing population and unemployment, there are some worries that using computer vision to save lives and replace laborious professions will overburden civilization. such concepts are unjustified because humans are 'the ultimate resource.' Because the advances in computer vision includes both expenses and advantages, researchers who want to have a good societal influence should explore initiatives that intend to pay off in the free market often as feasible and seek private rather than government support.

**Health:**

In the last few months, as Coronavirus (COVID-19) changed how we live and work, Computer Vision technology evolve and get a new level of importance in this phase. Healthcare services around the world are scrambling to care for patients and work. All the steps are taken to make detection, testing, and tracing easier and more effective. Their detection and other computer vision techniques play a major role. Computer Vision devices are now being used for the detection of possible coronavirus lesions in the CT scan and to measure their shape, density, and volume.

A chest Computed tomography (CT) scan can be analyzed in 50 seconds which quickly detects whether or not a patient has COVID-19, depending on whether there are lesions on the lungs. Computer Vision technology is being used to create a deep learning model which differentiates Coronavirus from pneumonia and other lung diseases. It strengthens personalized medical plans, care assistance, and decision-making.

Earlier stage disease detection is another machine vision medical application. With more data points, more images, and videos being processed, it should make it easier for AI-powered systems to help doctors in detection and a better understanding of the patient's condition. When diseases are identified at an earlier stage, outcomes are also improved for patients. Object detection and computer vision make a revolutionary change in the perspective of the human towards healthcare.

**Legal and Cultural Aspects:**

The impacts of computer vision on employment in the context of its relation to the automation problem and the Concerns are developing that utilizing computer vision in surveillance and public safety systems violates safety and privacy. These are the main reason why people are not accepting it. But with time and the developing application of Computer vision in various fields and the involvement of various government bodies which provide funds for research and development in this field increases the public belief in these new technologies.

# cse_22CSE10_PREPORTajeet.pdf

ORIGINALITY REPORT

29%
SIMILARITY INDEX

PRIMARY SOURCES

| | | | |
|---|---|---|---|
| 1 | www.coursehero.com<br>Internet | 735 words | 4% |
| 2 | towardsdatascience.com<br>Internet | 517 words | 2% |
| 3 | ece.anits.edu.in<br>Internet | 492 words | 2% |
| 4 | noahsnail.com<br>Internet | 337 words | 2% |
| 5 | www.section.io<br>Internet | 278 words | 1% |
| 6 | Jintao Wang, Wen Xiao, Tianwei Ni. "Efficient Object Detection Method Based on Improved YOLOv3 Network for Remote Sensing Images", 2020 3rd International Conference on Artificial Intelligence and Big Data (ICAIBD), 2020<br>Crossref | 236 words | 1% |
| 7 | hcis-journal.springeropen.com<br>Internet | 176 words | 1% |
| 8 | jusst.org<br>Internet | 155 words | 1% |