FAST ENHANCED UNIDENTIFIABLE OBJECT DETECTION USING DEEP LEARNING ALGORITHM

A PROJECT REPORT

Submitted by

MONISHA.V [REGISTER NO:211414104153]
RAMYA.S [REGISTER NO:211414104219]
RAMYA.S [REGISTER NO:211417104220]

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

PANIMALAR ENGINEERING COLLEGE, CHENNAI-600123.

ANNA UNIVERSITY: CHENNAI 600 025

AUGUST 2021

BONAFIDE CERTIFICATE

Certified that this project report **“FAST ENHANCED UNIDENTIFIABLE OBJECT DETECTION USING DEEP LEARNING ALGORITHM”** is the bonafide work of **MONISHA.V[211417104153] RAMYA.S[211417104219] and RAMYA.S[211417104220]** who carried out the project work under my supervision.

SIGNATURE	**SIGNATURE**
Dr.S.MURUGAVALLI,M.E.,Ph.D.,	**M.MAHESWARI,M.E.,**
HEAD OF THE DEPARTMENT	**SUPERVISOR**
	ASSOCIATE PROFESSOR
DEPARTMENT OF CSE,	DEPARTMENT OF CSE,
PANIMALAR ENGINEERING COLLEGE,	PANIMALAR ENGINEERING COLLEGE,
NASARATHPETTAI,	NASARATHPETTAI,
POONAMALLEE,	POONAMALLEE,
CHENNAI-600 123.	CHENNAI-600 123.

Certified that the above candidate(s) was/ were examined in the Anna University Project Viva-Voce Examination held on ….5.8.2021….

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

We express our deep gratitude to our respected Secretary and Correspondent **Dr.P.CHINNADURAI, M.A., Ph.D.** for his kind words and enthusiastic motivation, which inspired us a lot in completing this project.

We would like to extend our heartfelt and sincere thanks to our Directors **Tmt.C.VIJAYARAJESWARI**, **Thiru.C.SAKTHIKUMAR,M.E.,** and **Tmt. SARANYASREE SAKTHIKUMAR B.E.,M.B.A.,** for providing us with the necessary facilities for completion of this project.

We also express our gratitude to our Principal **Dr.K.Mani, M.E., Ph.D.** for his timely concern and encouragement provided to us throughout the course.

We thank the HOD of CSE Department, **Dr. S.MURUGAVALLI,M.E.,Ph.D.,** for the support extended throughout the project.

We would like to thank my **Project Guide, Mrs.M.MAHESWARI,M.E**., and all the faculty members of the Department of CSE for their advice and suggestions for the successful completion of the project.

MONISHA.V

RAMYA.S

RAMYA.S

ABSTRACT

Over the years the number of sensors, cameras, and cognitive pieces of equipment placed in the wilderness has been growing exponentially. However, the resources (human) to leverage these data into something meaningful are not improving at the same rate. Our framework detects all the objects based on training set provided to it. Main view of this project is to increase the accuracy rate of the detection. it recognize the object even in blur stage or under less brightness.In recent days the detection of objects is done with RCNN. Our framework takes full advantage of extracting from FCN providing more advanced representation at each layer, this property is used for segment detection. It has advantages in terms of efficiency ie ,0.08 second per image, effectiveness and simplicity over existing algorithms. It can be analyzed on the role of training data on performance, the experimental results provide a more reasonable and powerful training set for future research.Over 60-65k images can be trained for detection. This framework can be constructed using deep learning, which is a subset of machine learning in AI that has network capability of learning unsupervised from data that is unstructured or unable.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATION

S.NO	ABBREVATION	EXPANSION
1.	CNN	Convolution Neural networks
2.	ROI	Region Of Interest
3.	SVM	Support Vector Machine
4.	COCO	Common Objects in Context
5.	DNN	Deep Neural Network

CHAPTER 1

INTRODUCTION

1. INTRODUCTION

1.1 OVERVIEW

A few years ago, the creation of the software and hardware image processing systems was mainly l imited to the development of the user interface, which most of the programmers of each firm were engaged in. The situation has been significantly changed with the advent of the Windows operating system when the majority of the developers switched to solving the problems of image processing itself.

Object recognition is to describe a collection of related computer vision tasks that involve activities like identifying objects in videos. Image classification involves activities such as predicting the class of one or more objects in an image. Object localization is refers to identifying the location of one or more objects in an image and drawing an abounding box around their extent. Object detection does the work of combines these two tasks and localizes and classifies one or more objects
in an image.

Image classification also involves assigning a class label to an image, whereas object localization involves drawing a bounding box around one or more objects in an image. Object detection is always more challenging and combines these two tasks and draws a bounding box around each object of interest in the image and assigns them a class label. Together, all these problems are referred to as object.

Object recognition refers to a collection of related tasks for identifying objects in videos.Region-based Convolutional Neural Networks, or R-CNNs, is a family of techniques for addressing object localization and recognition tasks, designed for model performance. You Only Look Once, or YOLO is known as the second family of techniques for object recognition designed for speed and real-time use.

1.2 PROBLEM STATEMENT

The project "Fast Enhanced Unidentifiable Object Detection Using Deep Learning Algorithms” detects objects efficiently based on YOLO algorithm and FRCNN algorithm and apply the algorithm on image data and video data to detect objects.

Our framework detects all the objects based on training set provided to it. Main view of this project is to increase the accuracy rate of the detection. It recognize the object even in blur stage or under less brightness,Over 60-65k images can be trained for detection.

object detection aims at locating and classifying existing objects in any one image, and labeling them with rectangular bounding boxes to show the confidences of existence. The frameworks of generic object detection methods can mainly be categorized into two types .One follows traditional object detection pipeline, generating region proposals at first and then classifying each proposal into different object categories. The other regards object detection as a regression or classification problem, adopting a unified framework to achieve final results (categories and locations)directly.

CHAPTER 2

LITERATURE SUVERY

1.TITLE : Deep Convolutional Network based Species Recognation For Wild Animal Monitoring.

AUTHOR: Guoben chen et al

YEAR: 2015

DESCRIPTION:

Supposedly, this paper presents the first attempt to automate wild life monitoring using Deep Convolutional Neural Network on camera trap image.In this paper, the performance of DNN is benchmarked against Visual Bag of Words automation method.The DNN model was trained on 14346 images and tested upon 9530 images.It achieved a better prediction performance over the visual bag of words method

MERITS:

This method provides an automated method to classify wild animals.

DEMERITS:

Moreover, the method discussed in the paper heavily depends on training data annotated through other methods including manual efforts. But that could be considered as an intial constraint rather than throughout the whole process

2.TITLE: Animal Identification in low quality camera-trap images using very deep convolutional neural networks and confidence thresholds**.**

AUTHOR : Gomez et al

YEAR :2016

DESCRIPTION:

Successfully applied DNNs to distinguishing birds vs mammals in a small dataset of 1572 images and distinguish two mammal sets in a dataset of 2597 images .This method achieved better testing accuracy than Chen et al’s.

MERITS:

This method achieved better testing accuracy

DEMERITS:

Even though, this method provides an automated method to classify wild animals with better accuracy, the test dataset used performance evaluation was very small

3.TITLE: Towards automatic wild animal monitoring: Identification of animal species in camera-trap images using very deep CNN.

AUTHOR: GOMEZ ,DIEZ G ET AL

YEAR:2018

DESCRIPTION:

This paper focuses on evaluating deep convolutional neural networks on the open source Snapshot Serengeti dataset which is much larger dataset with more than 3 million images.

MERITS:

Having much larger dataset with more than 3 million images.

DEMERITS:

The estimated accuracy of their models seem to be only around 57 %. And moreover, this paper only focuses on evaluating 26 species out of 48 species available in the snapshot Serengeti dataset.

4.TITLE: You Only Look Once: Unified, Real-Time Object Detection.

AUTHOR : Joseph Redmon

YEAR: 2016

DESCRIPTION:

A fast and simple approach to detecting real time images was introduced in this paper as You Only Look Once. The model was built to detect images accurately, fast and to differentiate between art and real images.

MERITS:

It detects images accurately, fast and to differentiate between art and real images.

DEMERITS:

In comparison with Object detection techniques that came before YOLO, like R-CNN, YOLO introduced a single unified architecture for regression.

5.TITLE: Identification and Detection of Automotive Door Panel Solder Joints based on YOLO.

AUTHOR: ZHIMIN MO,LIDING CHEN

YEAR:2019

DESCRIPTION:

A method for identifying the solder joints of automotive door panels based on YOLO algorithm.

MERITS:

This method achieved better testing accuracy and detects image accurately.

DEMERITS:

The YOLO algorithm, proposed identifies the position of the solder joints accurately in real time.

CHAPTER 3

SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

Due to object detection's close relationship with video analysis and image understanding, it has attracted much research attention in recent years. Traditional object detection methods are built on handcrafted features and shallow trainable architectures. Their performance easily stagnates by constructing complex ensembles which combine multiple low-level image features with high-level context from object detectors and scene classifiers. With the rapid development in deep learning, more powerful tools, which are able to learn semantic, high-level, deeper features, are introduced to address the problems existing in traditional architectures. These models behave differently

in network architecture, training strategy and optimization function, etc. In this paper, we provide a review on deep learning based object detection frameworks. Our review begins with a brief introduction on the history of deep learning and

its representative tool, namely Convolutional Neural Network(CNN). Then we focus on typical generic object detection architectures along with some modifications and useful tricks to improve detection performance further. As distinct specific detection tasks exhibit different characteristics, we also briefly survey several specific tasks, including salient object detection, face detection and pedestrian detection. Experimental analyses are also provided to compare various methods and draw some meaningful conclusions. Finally, several promising directions and tasks are provided to serve as guidelines for future work in both object detection and relevant neural network based learning systems.

3.2 PROPOSED SYSTEM

The proposed method uses these multilayer convolutional neural networks to develop a system model which consists of multilayers to classify the given objects into any of the defined classes. The schemes then use multiple images and detect the objects from these images, labeling them with their respective class label. To speed up the computational performance, the proposed algorithm is applied along with the multilayer convolutional neural network which uses a larger number of default boxes and results in more accurate detection. The accuracy in detecting the objects is checked by different parameters such as loss function, frames per second (FPS), mean average precision (mAP), and aspect ratio. Experimental results confirm that our proposed improved SSD algorithm has high accuracy

3.3 TECHNOLOGY STACK

HARWARE REQUIREMENTS

Processor	: intel core i5
Hard disk	: 500 GB
RAM	: 8 GB (minimum)

SOFTWARE REQUIREMENTS

- Python 3.6.4 Version

SOFTWARE SPECIFICATION

Machine learning

Machine learning (**ML**) is the study of computer algorithms that improve automatically through experience and by the use of data. It is seen as a part of artificial intelligence. Machine learning algorithms build a model based on sample data, known as "training data", in order to make predictions or decisions without being explicitly programmed to do so. Machine learning algorithms are used in a wide variety of applications, such as email filtering and computer vision, where it is difficult or unfeasible to develop conventional algorithms to perform the needed tasks.

The types of machine learning algorithms are mainly divided into four categories:

- **Supervised learning,**
- **Un-supervised learning,**
- **Semi-supervised learning,**
- **Reinforcement learning**.

Supervised learning

Supervised learning algorithms build a mathematical model of a set of data that contains both the inputs and the desired outputs. The data is known as training data, and consists of a set of training examples. Each training example has one or more inputs and the desired output, also known as a supervisory signal. In the mathematical model, each training example is represented by an array or vector, sometimes called a feature vector, and the training data is represented by a matrix. Through iterative optimization of an objective function, supervised learning algorithms learn a function that can be used to predict the output associated with new inputs. An optimal function will allow the algorithm to correctly determine the output for inputs that were not a part of the training data. An algorithm that improves the accuracy of its outputs or predictions over time is said to have learned to perform that task.

CLASSIFICATION

As the name suggests, Classification is the task of "classifying things" into sub- categories. But, by a machine. If that doesn't sound like much, imagine your computer being able to differentiate between you and a stranger. Between a potato and a tomato. Between an A grade and a F. In Machine Learning and Statistics, Classification is the problem of identifying to which of a set of categories (sub populations), a new observation belongs to, on the basis of a training set of data containing observations and whose categories membership is known.

TYPES OF CLASSIFICATION

Classification is of two types:

- Binary Classification
- Multiclass Classification

Binary Classification

When we have to categorize given data into 2 distinct classes. Example – On the basis of given health conditions of a person, we have to determine whether the person has a certain disease or not.

Multiclass Classification

The number of classes is more than 2. For Example

– On the basis of data about different species of flowers, we have to determine which specie does our observation belong to

Fig 2 : Binary and Multiclass Classification. Here x1 and x2 are our variables upon which the class is predicted.Suppose we have to predict

whether a given patient has a certain disease or not, on the basis of 3 variables, called features. Which means there are two possible outcomes:

1. The patient has the said disease. Basically a result labelled “Yes” or “True”.

2. The patient is disease free. A result labelled “No” or “False”.

This is a binary classification problem.We have a set of observations called training data set, which comprises of sample data with actual classification results. We train a model, called Classifier on this data set, and use that model to predict whether a certain patient will have the

1. X : pre-classified data, in the form of a N*M matrix. N is the no. of observations and M is the number of features

2. y : An N-d vector corresponding to predicted classes for each of the N observations.

3. Feature Extraction : Extracting valuable information from input X using a series of transforms.

4. ML Model : The “Classifier” we’ll train.

5. y’ : Labels predicted by the Classifier.

6. Quality Metric : Metric used for measuring the performance of the model.

7. ML Algorithm : The algorithm that is used to update weights w’, which update the model and “learns” iteratively.

Types of Classifiers (Algorithms)

There are various types of classifiers. Some of them are :

- Linear Classifiers : Logistic Regression

- Tree Based Classifiers : Decision Tree Classifier
- Support Vector Machines
- Artificial Neural Networks
- Bayesian Regression
- Gaussian Naive Bayes Classifiers
- Stochastic Gradient Descent (SGD) Classifier
- Ensemble Methods : Random Forests, AdaBoost, Bagging Classifier, Voting Classifier, ExtraTrees Classifier

Practical Applications of Classification

- Google's self driving car uses deep learning enabled classification techniques which enables it to detect and classify obstacles.
- Spam E-mail filtering is one of the most widespread and well recognized uses of Classification techniques.
- Detecting Health Problems, Facial Recognition, Speech Recognition, Object Detection, Sentiment Analysis all use Classification at their core.

REGRESSION

A regression problem is when the output variable is a real or continuous value, such as "salary" or "weight". Many different models can be used, the simplest is the linear regression. It tries to fit data with the best hyper-plane which goes through the points.

❖ **Un-supervised learning**

Un-supervised learning algorithms take a set of data that contains only inputs, and find structure in the data, like grouping or clustering of data points. The algorithms, therefore, learn from test data that has not been labeled, classified or categorized. Instead of responding to feedback, unsupervised learning algorithms identify commonalities in the data and react based on the presence or absence of such commonalities in each new piece of data. A central application of unsupervised learning is in the field of density estimation in statistics, such as finding the probability density function. Though unsupervised learning encompasses other domains involving summarizing and explaining data features.

CLUSTERING

It is basically a type of unsupervised learning method. An unsupervised learning method is a method in which we draw references from datasets consisting of input data without labelled responses. Generally, it is used as a process to find meaningful structure, explanatory underlying processes, generative features, and groupings inherent in a set of examples. Clustering is the task of dividing the population or data points into a number of groups such that data points in the same groups are more similar to other data points in the same group and dissimilar to the data points in other groups. It is basically a collection of objects on the basis of similarity and dissimilarity between them.For example, the data points in the graph below clustered together can be classified into one single group. We can distinguish the clusters, These data points are clustered by using the basic concept that the data point lies within the given constraint from the cluster center. Various distance methods and techniques are used for calculation of the outliers.

Clustering is very much important as it determines the intrinsic grouping among the unlabeled data present. There are no criteria for a good clustering. It depends on the user, what is the criteria they may use which satisfy their need. For instance, we could be interested in finding representatives for homogeneous groups (data reduction), in finding “natural clusters” and describe their unknown properties (“natural” data types), in finding useful and suitable groupings (“useful” data classes) or in finding unusual data objects (outlier detection). This algorithm must make some assumptions which constitute the similarity of points and each assumption make different and equally valid clusters.

3.6.5.1.1 Clustering Methods :

1. **Density-Based Methods :** These methods consider the clusters as the dense region having some similarity and different from the lower dense region of the space. These methods have good accuracy and ability to merge two clusters.Example DBSCAN

(Density-Based Spatial Clustering of Applications with Noise) , OPTICS (Ordering Points to Identify Clustering Structure) etc.

2. **Hierarchical Based Methods :** The clusters formed in this method forms a tree type structure based on the hierarchy. New clusters are formed using the previously formed one. It is divided into two category

- Agglomerative (bottom up approach)
- Divisive (top down approach) .

3. **Partitioning Methods :** These methods partition the objects into k clusters and each partition forms one cluster. This method is used to optimize an objective criterion similarity function such as when the distance is a major parameter example K-means, CLARANS (Clustering Large Applications based upon randomized Search) etc.

4. **Grid-based Methods :** In this method the data space are formulated into a finite number of cells that form a grid-like structure. All the clustering operation done on these grids are fast and independent of the number of data objects example STING (Statistical Information Grid), wave cluster, CLIQUE (CLusteringIn Quest) etc.

Clustering Algorithms:

- K-Means Clustering.
- Mean-Shift Clustering for a single sliding window.
- The entire process of Mean-Shift Clustering.

- DBSCAN Smiley Face Clustering.
- EM Clustering using GMMs.
- Agglomerative Hierarchical Clustering.

❖ **Semi-supervised learning**

Semi-supervised learning falls between unsupervised learning (without any labeled training data) and supervised learning (with completely labeled training data). Some of the training examples are missing training labels, yet many machine-learning researchers have found that unlabeled data, when used in conjunction with a small amount of labeled data, can produce a considerable improvement in learning accuracy.

❖ **Reinforcement learning**

Reinforcement learning is an area of machine learning concerned with how software agents ought to take actions in an environment so as to maximize some notion of cumulative reward. Due to its generality, the field is studied in many other disciplines, such as game theory, control theory, operations research, information theory, simulation-based optimization, multi-agent systems, swarm intelligence, statistics and genetic algorithms. In machine learning, the environment is typically represented as a Markov decision process (MDP). Many reinforcement learning algorithms use dynamic programming techniques. Reinforcement learning algorithms do not assume knowledge of an exact mathematical model of the MDP, and are used when exact models are infeasible. Reinforcement learning algorithms are used in autonomous vehicles or in learning to play a game against a human opponent.

ANACONDA

Anaconda is a free and open source distribution of the Python and R programming languages for data science and machine learning related applications (large-scale data processing, predictive analytics, scientific computing), that aims to simplify package management and deployment. Package versions are managed by the package management system conda. Anaconda Distribution is used by over 6 million users, and it includes more than 250 popular data science packages suitable for Windows, Linux, and MacOS.

Python is a high-level programming language devised by Guido van Rossum & first released in 1991. It's the most popular coding language used by software developers to build, control, manage and for testing. It is also an interpreter which executes Python programs. The python interpreter is called python.exe on Windows.

Python Packages

Packages or additional libraries help in scientific computing and computational modelling. In Python, the packages are not the part of the Python standard library. Few major packages are –

numpy (NUMeric Python): matrices and linear algebra
scipy (SCIentific Python): many numerical routines
matplotlib: (PLOTting LIBrary) creating plots of data
sympy (SYMbolic Python): symbolic computation
pytest (Python TESTing): a code testing framework

Together with a list of Python packages, tools like editors, Python distributions include the Python interpreter. Anaconda is one of several Python distributions.

Anaconda is a new distribution of the Python and R data science package. It was formerly known as Continuum Analytics. Anaconda has more than 100 new packages.

This work environment, Anaconda is used for scientific computing, data science, statistical analysis, and machine learning. The latest version of Anaconda 5.0.1 is released in October 2017.The released version 5.0.1 addresses some minor bugs and adds useful features, such as updated R language support. All of these features weren't available in the original 5.0.0 release.

This package manager is also an environment manager, a Python distribution, and a collection of open source packages and contains more than 1000 R and Python Data Science Packages.

IPYTHON NOTEBOOKS

IPython is a command shell for interactive computing in multiple programming languages, originally developed for the Python programming language, that offers introspection, rich media, shell syntax, tab completion, and history. IPython provides the following features:

Interactive shells (terminal and Qt-based).

• A browser-based notebook interface with support for code, text, mathematical expressions, inline plots and other media.

• Support for interactive data visualization and use of GUI toolkits.

• Flexible, embeddable interpreters to load into one's own projects.

• Tools for parallel computing.

IPython is based on an architecture that provides parallel and distributed computing. IPython enables parallel applications to be developed, executed, debugged and monitored interactively. Hence, the I (Interactive) in

IPython.[3]This architecture abstracts out parallelism, which enables IPython to support many different styles of parallelism[4]including:
With the release of IPython 4.0, the parallel computing capabilities have been made optional and released under the ipyparallel python package.IPython frequently draw from SciPy stack[5] libraries like NumPy and SciPy, often installed alongside from one of many Scientific Python distributions. IPython provide integration some library of the SciPy stack like matplotlib, like inline graph when in used with the Jupyter notebook. Python libraries can implement IPython specific hooks to customize object Rich object display. SymPy for example implement rendering of Mathematical Expression as rendered LaTeX when used within IPython context.
Other features:

IPython also allows non-blocking interaction with Tkinter, PyGTK, PyQt/PySide and wxPython (the standard Python shell only allows interaction with Tkinter). IPython can interactively manage parallel computing clusters using asynchronous status call-backs and/or MPI. IPython can also be used as a system shell replacement. Its default behaviour is largely similar to Unix shells, but it allows customization and the flexibility of executing code in a live Python environment. Using IPython as a shell replacement is less common and it is now recommended to use Xonsh which provide most of the IPython feature with better shell integrations

CHAPTER 4

4.1 Flowchart Diagram

4.2 Sequence Diagram

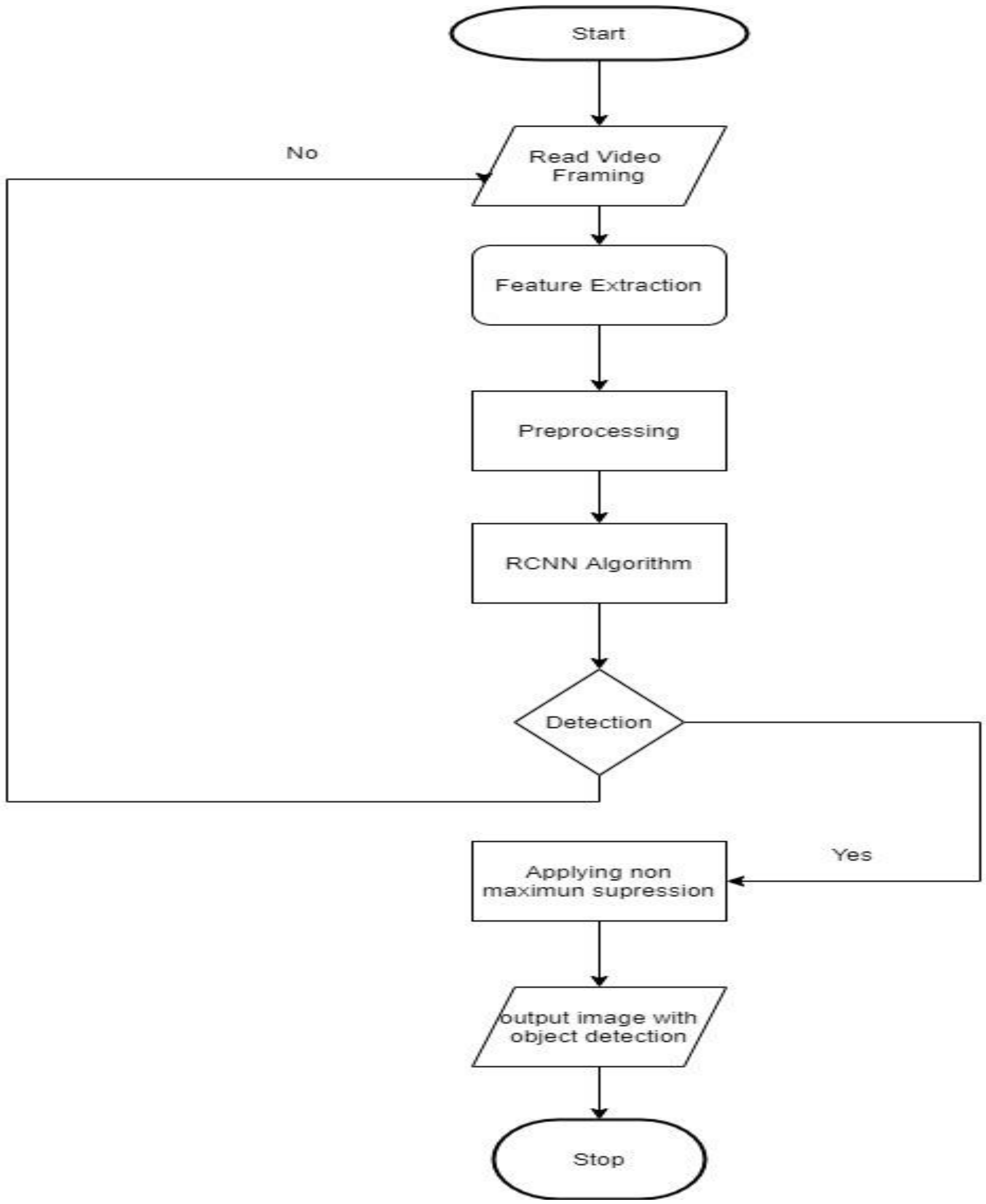


Fig 4.1 Flow chart Diagram

4.2 Sequence Diagram

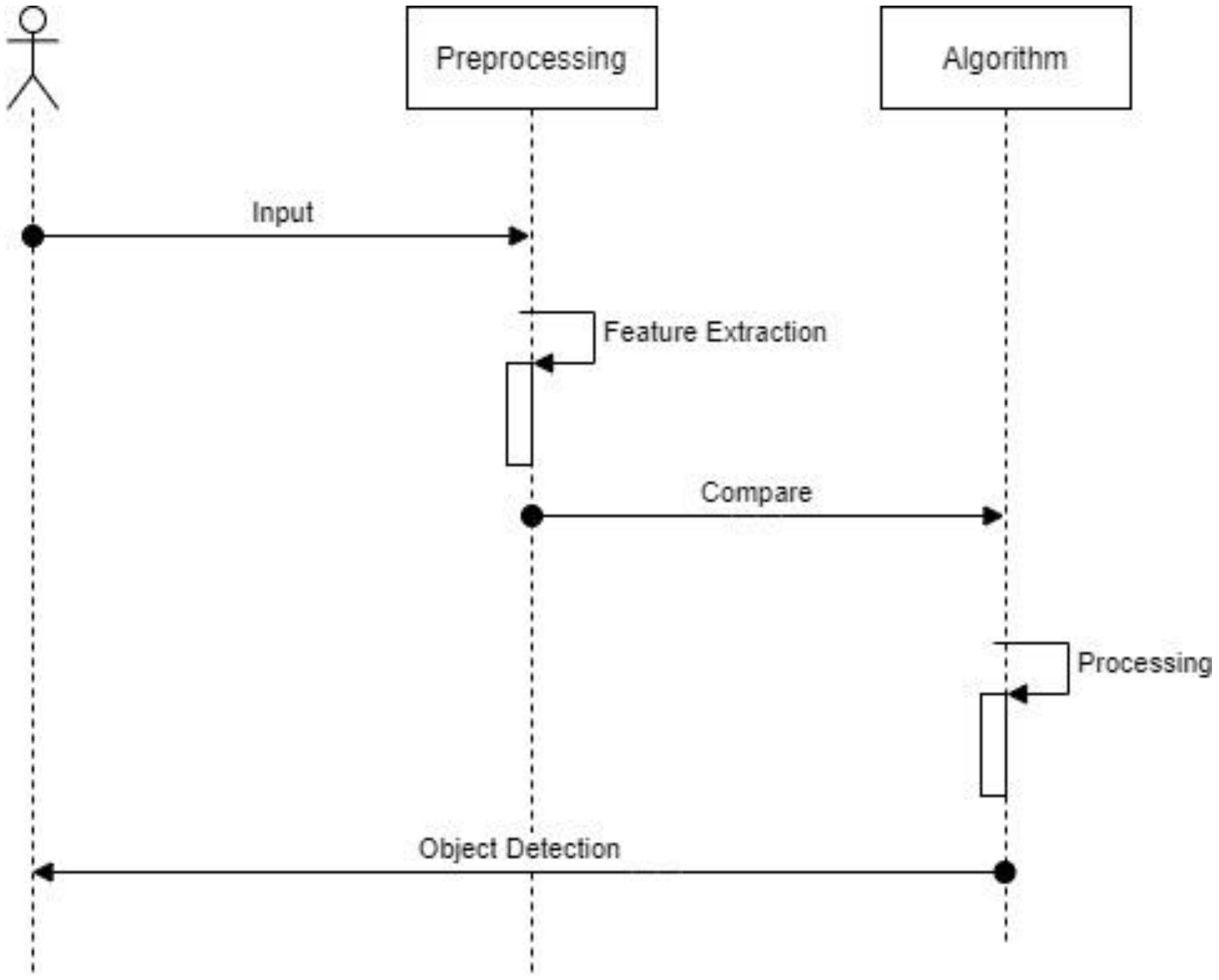


Fig 4.2 Sequence diagram

CHAPTER 5

ARCHITECTURE

5.1 SYSTEM ARCHITECTURE

System architecture is the conceptual model that defines the structure, behavior, and more views of a system. An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structures and behaviors of the system.

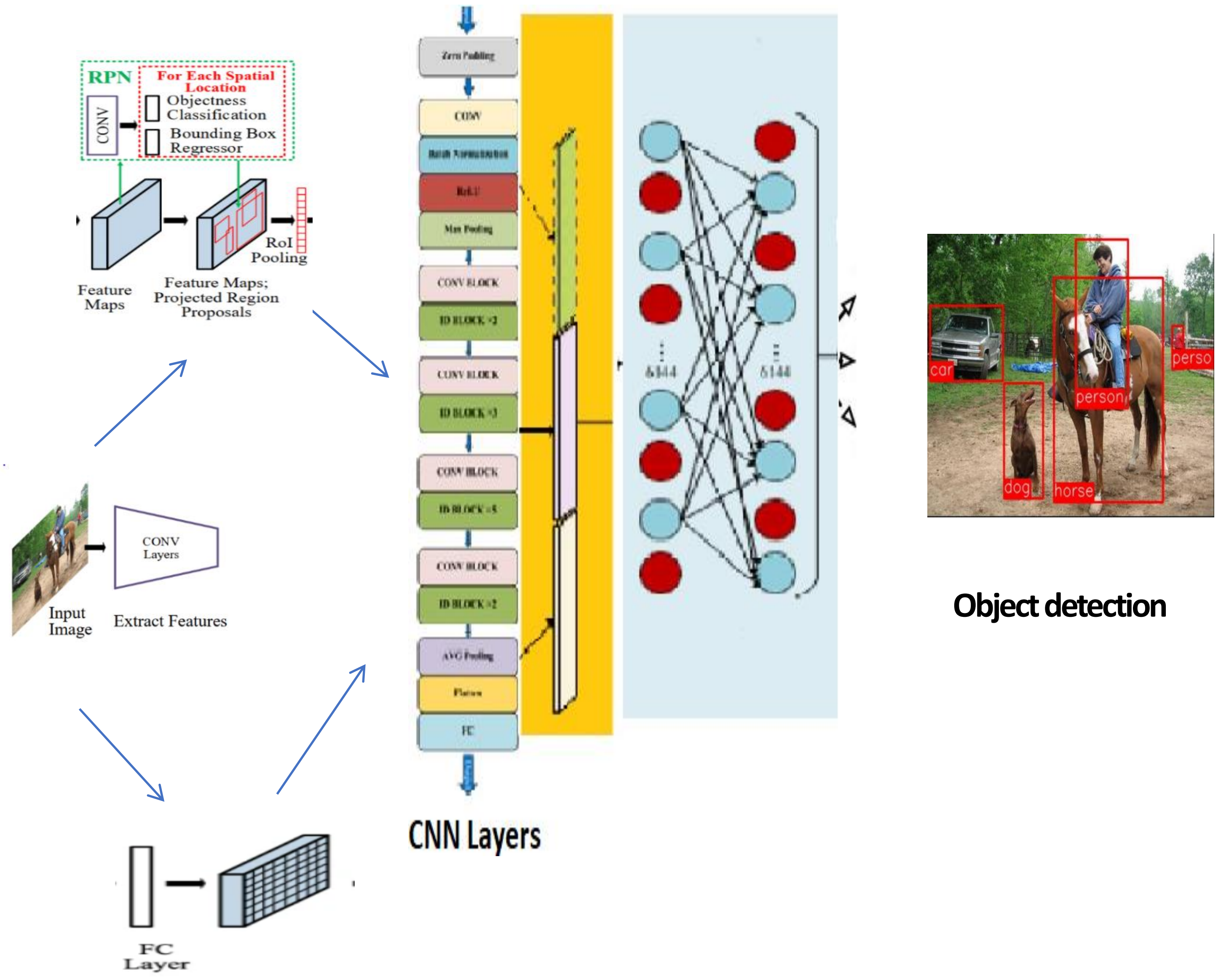


Fig.5.1 System Architect

SYSTEM MODULE

5.2 MODULE

- OBJECT DATA COLLECTION
- DATA PREPROCESSING

5.2.1MODULE DESCRIPTION

Object data collection:

Real time data collected from kaggle- Coco dataset.

1. Collection of data is one of the major and most important tasks of any machine learning projects.
2. Because the input we feed to the algorithms is data. So, the algorithms efficiency and accuracy depends upon the correctness and quality of data collected. So as the data same will be the output.
3. Common Objects in Context (COCO): COCO is a large-scale object detection, segmentation, and captioning dataset. It contains around 330,000 images out of which 200,000 are labelled for 80 different object categories.

DATA PREPROCESSING

- Data collected from various means will be in an unorganized format and there may be lot of null values, in-valid data values and unwanted data.
- Cleaning all these data and replacing them with appropriate or approximate data and removing null and missing data and replacing them with some fixed alternate values are the basic steps in pre processing of data.
- Even data collected may contain completely garbage values. It may not be in exact format or way that is meant to be.

All such cases must be verified and replaced with alternate values to make data meaning meaningful and useful for further processing. Data must be kept in an organized format.

5.3 Program Design language

5.3.1 FAST REGIONAL CONVOLUTIONAL NEURAL NETWORK(FRCNN) ALGORITHM

Ross Girshick, the author of RCNN, came up with this idea of running the CNN just once per image and then finding a way to share that computation across the 2,000 regions. In Fast RCNN, we feed the input image to the CNN, which in turn generates the convolutional feature maps. Using these maps, the regions of proposals are extracted. We then use a RoI pooling layer to reshape all the proposed regions into a fixed size, so that it can be fed into a fully connected network.

Let's break this down into steps to simplify the concept:

1. As with the earlier two techniques, we take an image as an input.
2. This image is passed to a ConvNet which in turns generates the Regions of Interest.
3. A RoI pooling layer is applied on all of these regions to reshape them as per the input of the ConvNet. Then, each region is passed on to a fully connected network
4. A softmax layer is used on top of the fully connected network to output classes. Along with the softmax layer, a linear regression layer is also used parallely to output bounding box coordinates for predicted classes.

So, instead of using three different models (like in RCNN), Fast RCNN uses a single model which extracts features from the regions, divides them into different classes, and returns the boundary boxes for the identified classes simultaneously.

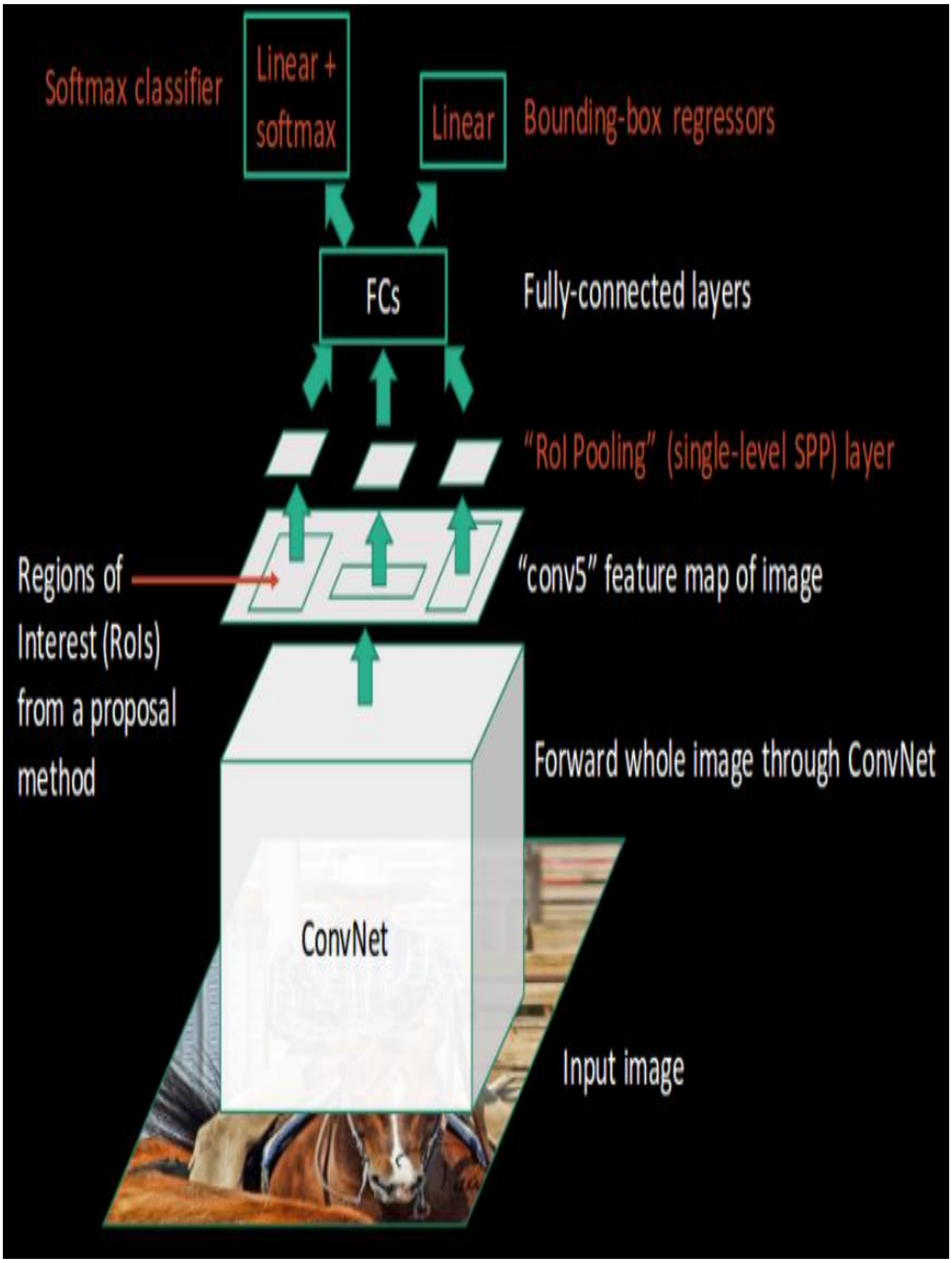
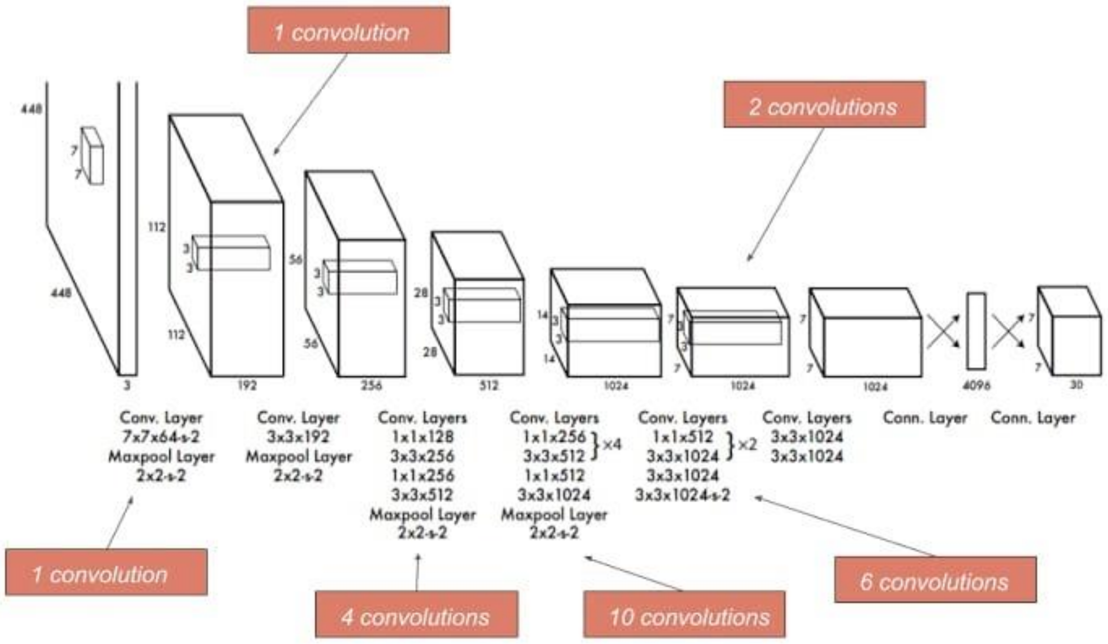

Softmax classifier
Linear + softmax
Linear
Bounding-box regressors
FCs
Fully-connected layers
"RoI Pooling" (single-level SPP) layer
Regions of Interest (RoIs) from a proposal method
"conv5" feature map of image
Forward whole image through ConvNet
ConvNet
Input image

Fig.5.3.1 FRCNN Architecture

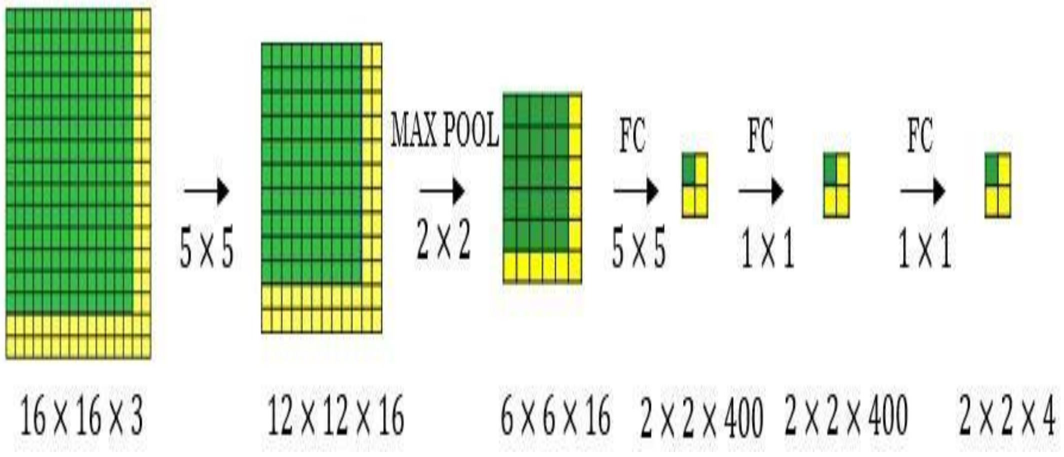
5.3.2 YOU ONLY LOOK ONCE(YOLO) ALGORITHM

- The previous object detection algorithms use regions to localize the object within the image. The network does not look at the complete image. Instead, parts of the image which have high probabilities of containing the object. YOLO or You Only Look Once is an object detection algorithm much different from the region based algorithms. In YOLO a single convolutional network predicts the bounding boxes and the class probabilities for these boxes.
- How YOLO works is that we take an image and split it into an SxS grid, within each of the grid we take m bounding boxes. For each of the bounding box, the network outputs a class probability and offset values for the bounding box. The bounding boxes having the class probability above a threshold value is selected and used to locate the object within the image.
- YOLO is orders of magnitude faster(45 frames per second) than other object detection algorithms. The limitation of YOLO algorithm is that it struggles with small objects within the image, for example it might have difficulties in detecting a flock of birds. This is due to the spatial constraints of the algorithm.



1 convolution
2 convolutions
448
7
7
448
3
112
3
3
112
192
56
3
3
56
256
28
3
3
28
512
14
3
3
14
1024
7
3
3
7
1024
7
7
1024
4096
7
7
30
Conv. Layer 7x7x64-s-2 Maxpool Layer 2x2-s-2
Conv. Layer 3x3x192 Maxpool Layer 2x2-s-2
Conv. Layers 1x1x128 3x3x256 1x1x256 3x3x512 Maxpool Layer 2x2-s-2
Conv. Layers 1x1x256 3x3x512 }×4 1x1x512 3x3x1024 Maxpool Layer 2x2-s-2
Conv. Layers 1x1x512 3x3x1024 }×2 3x3x1024 3x3x1024-s-2
Conv. Layers 3x3x1024 3x3x1024
Conn. Layer
Conn. Layer
1 convolution
4 convolutions
10 convolutions
6 convolutions

Fig.5.3.2 Convolution Layer



CHAPTER 6
SYSTEM IMPLEMENTATION

6.1 coding

```
import numpy as np

import cv2

confidenceThreshold = 0.5

NMSThreshold = 0.3

modelConfiguration = 'cfg/rcnn.cfg'

modelWeights = 'rcnn.weights'

labelsPath = 'coco.names'

labels = open(labelsPath).read().strip().split('\n')

np.random.seed(10)

COLORS = np.random.randint(0, 255, size=(len(labels), 3), dtype="uint8")

net = cv2.dnn.readNetFromDarknet(modelConfiguration, modelWeights)

outputLayer = net.getLayerNames()

outputLayer = [outputLayer[i[0] - 1] for i in net.getUnconnectedOutLayers()]

video = cv2.VideoCapture('video.mp4')

writer = None

(W, H) = (None, None)

try:

    prop = cv2.CAP_PROP_FRAME_COUNT

    total = int(video.get(prop))
    print("[INFO] {} total frames in video".format(total))
except:
    printf("Could not determine no. of frames in video")
count = 0
while True:
    (ret, frame) = video.read()
    if not ret:
        break
    if W is None or H is None:
        (H,W) = frame.shape[:2]
    blob = cv2.dnn.blobFromImage(frame, 1 / 255.0, (416, 416), swapRB = True,
crop = False)
    net.setInput(blob)
    layersOutputs = net.forward(outputLayer)
    boxes = []
    confidences = []
    classIDs = []
    for output in layersOutputs:
        for detection in output:
            scores = detection[5:]
            classID = np.argmax(scores)
            confidence = scores[classID]
            if confidence > confidenceThreshold:
                box = detection[0:4] * np.array([W, H, W, H])
                (centerX, centerY,  width, height) = box.astype('int')
                x = int(centerX - (width/2))
                y = int(centerY - (height/2))
                boxes.append([x, y, int(width), int(height)])
                confidences.append(float(confidence))
                classIDs.append(classID)
    #Apply Non Maxima Suppression
    detectionNMS = cv2.dnn.NMSBoxes(boxes, confidences,
confidenceThreshold, NMSThreshold)
    if(len(detectionNMS) > 0):
        for i in detectionNMS.flatten():
            (x, y) = (boxes[i][0], boxes[i][1])
            (w, h) = (boxes[i][2], boxes[i][3])
            color = [int(c) for c in COLORS[classIDs[i]]]
            cv2.rectangle(frame, (x, y), (x + w, y + h), color, 2)
            text = '{}: {:.4f}'.format(labels[classIDs[i]], confidences[i])
            cv2.putText(frame, text, (x, y - 5), cv2.FONT_HERSHEY_SIMPLEX,
0.5, color, 2)
            if writer is None:
                fourcc = cv2.VideoWriter_fourcc(*'MJPG')
        writer = cv2.VideoWriter('video_output.avi', fourcc, 30, (frame.shape[1], frame.shape[0]), True)
  if writer is not None:
    writer.write(frame)
    print("Writing frame" , count+1)
    count = count + 1
writer.release()
video.release()
```

CHAPTER 7

PERFORMANCE ANALYSIS

7.1 UNIT TESTING

Testing is a process of executing a program with the intent of finding an error. A good test case is one that has a high probability of finding an as-yet – undiscovered error.A successful test is one that uncovers an as-yet-undiscovered error. System testing is the stage of implementation, which is aimed at ensuring that the system works accurately and efficiently as expected before live operation commences. It verifies that the whole set of programs hang together. System testing requires a test consists of several key activities and steps for run program, string, system and is important in adopting a successful new system. This is the last chance to detect and correct errors before the system is installed for user acceptance testing. The software testing process commences once the program is created and the documentation and related data structures are designed. Software testing is essential for correcting errors. Otherwise the program or the project is not said to be complete. Software testing is the critical element of software quality assurance and represents the ultimate the review of specification design and coding. Testing is the process of executing the program with the intent of finding the error. A good test case design is one that as a probability of finding an yet undiscovered error. A successful test is one that uncovers an yet undiscovered error. Any engineering product can be tested in one of the two ways:

WHITE BOX TESTING

This testing is also called as Glass box testing. In this testing, by knowing the

specific functions that a product has been design to perform test can be conducted that demonstrate each function is fully operational at the same time searching for errors in each function. It is a test case design method that uses the control structure of the procedural design to derive test cases. Basis path testing is a white box testing.

Basis path testing:

- Flow graph notation
- Kilometric complexity
- Deriving test cases
- Graph matrices Control

BLACK BOX TESTING

In this testing by knowing the internal operation of a product, test can be conducted to ensure that "all gears mesh", that is the internal operation performs according to specification and all internal components have been adequately exercised. It fundamentally focuses on the functional requirements of the software.

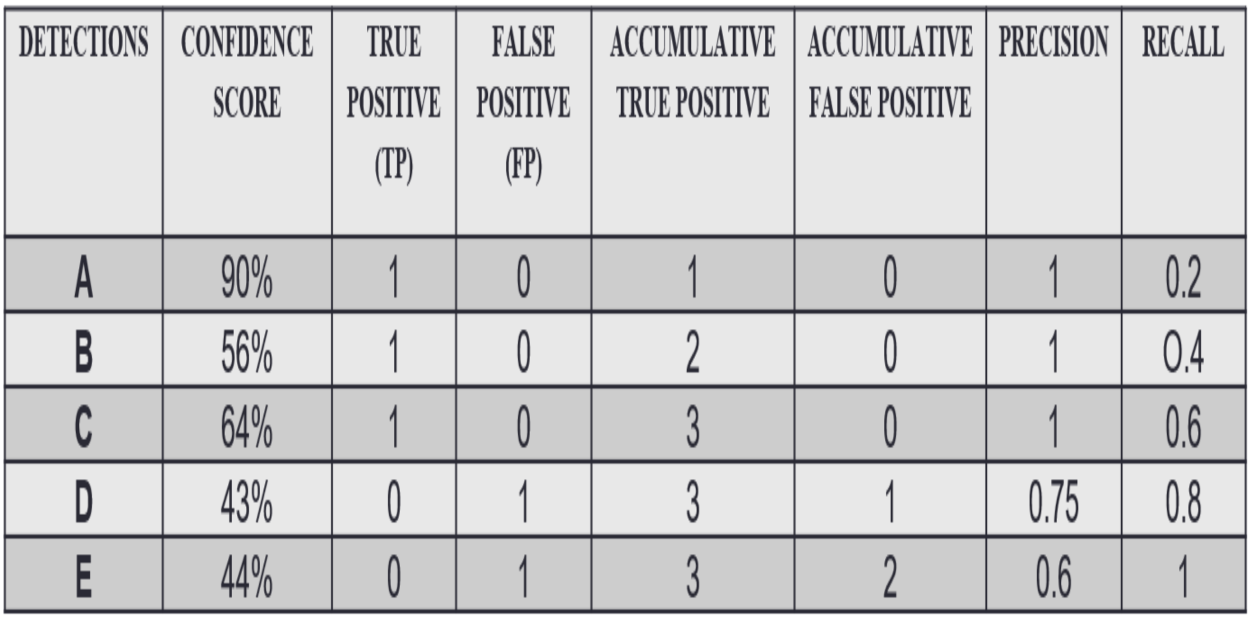
7.2 INTEGRATION TESTING

It is defined as a type of testing where software modules are integrated logically and tested as a group. A typical software project consists of multiple software modules, coded by different programmers. The purpose of this level of testing is to expose defects in the interaction between these software modules when they are integrated.Moreover, integration testing is a component of **extreme programming**, which is a pragmatic method of software development that takes meticulous approach to build a product by means of continual testing and revision.

7.3 EXPERIMENTAL RESULTS

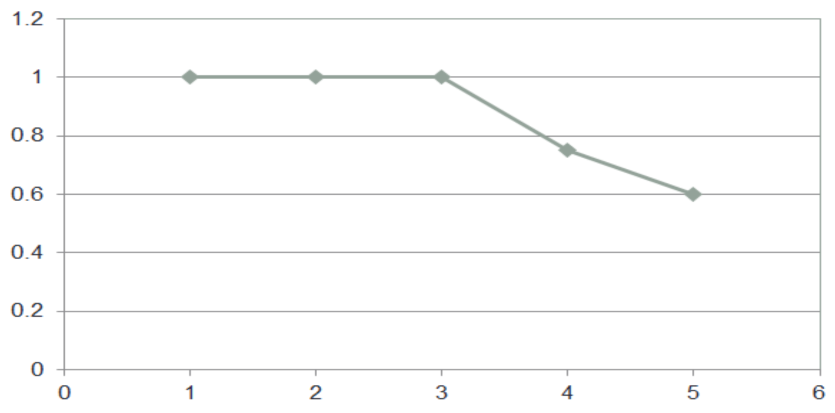
Table1 Precision and Recall Calculation by finding true positive and false positive:

- True Positive = 1(confidence score > 50%)
- False Positive = 1(confidence score < 50%)
- Precision = (True Positive + False Positive)/True Positive
- Recall = (True Positive + False Positive)/No. of. Bounding Boxes



DETECTIONS	CONFIDENCE SCORE	TRUE POSITIVE (TP)	FALSE POSITIVE (FP)	ACCUMULATIVE TRUE POSITIVE	ACCUMULATIVE FALSE POSITIVE	PRECISION	RECALL
A	90%	1	0	1	0	1	0.2
B	56%	1	0	2	0	1	0.4
C	64%	1	0	3	0	1	0.6
D	43%	0	1	3	1	0.75	0.8
E	44%	0	1	3	2	0.6	1

- Using the below graph the average precision can be calculated using 11 point interpolation method.



Precision-Recall Curve

TABLE 2 Comparing the Average Precision Value

ALGORITHMS	**AVERAGE PRECISION(AP)**
YOLO	21.6
FRCNN	34.1
Combined(YOLO & FRCNN)	38.8

- By comparing the Average Precision of the three algorithms we come to know that the AP is greater when both the algorithms are combined since accuracy depends on AP, it is obvious that we can achieve greater accuracy when both the algorithms are combined.

CHAPTER 8

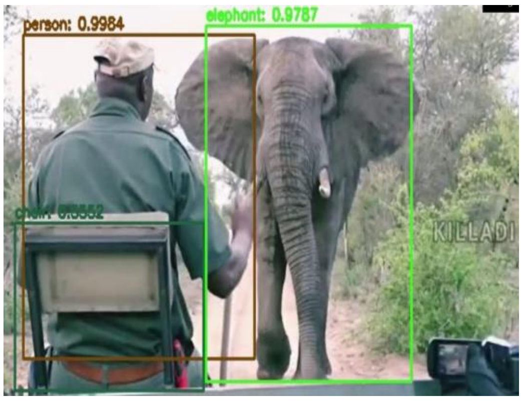
CONCLUSION AND FUTURE ENHANCEMENTS

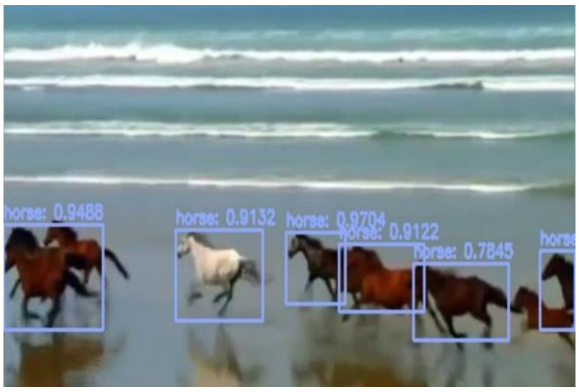
By using this thesis and based on experimental results we are able to detect object more precisely and identify the objects individually with exact location of an object in the picture in x ,y axis. This paper also provide experimental results on different methods for object detection and identification and compares each method for their efficiencies.

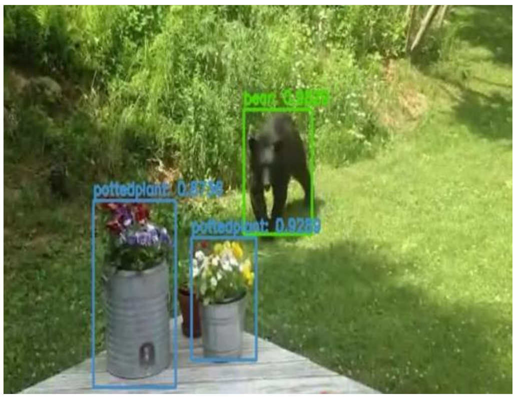
In Future, it will be enhanced by achieving alert signals by detecting the objects.

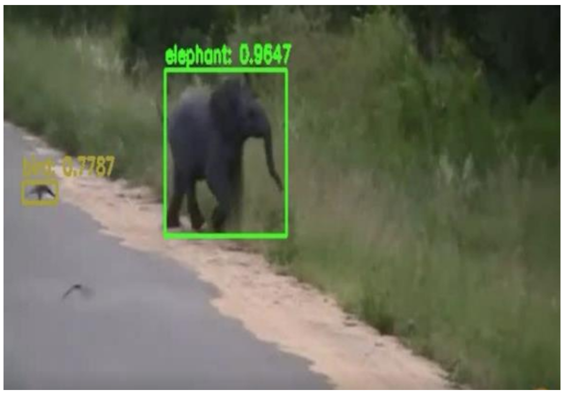
APPENDICES

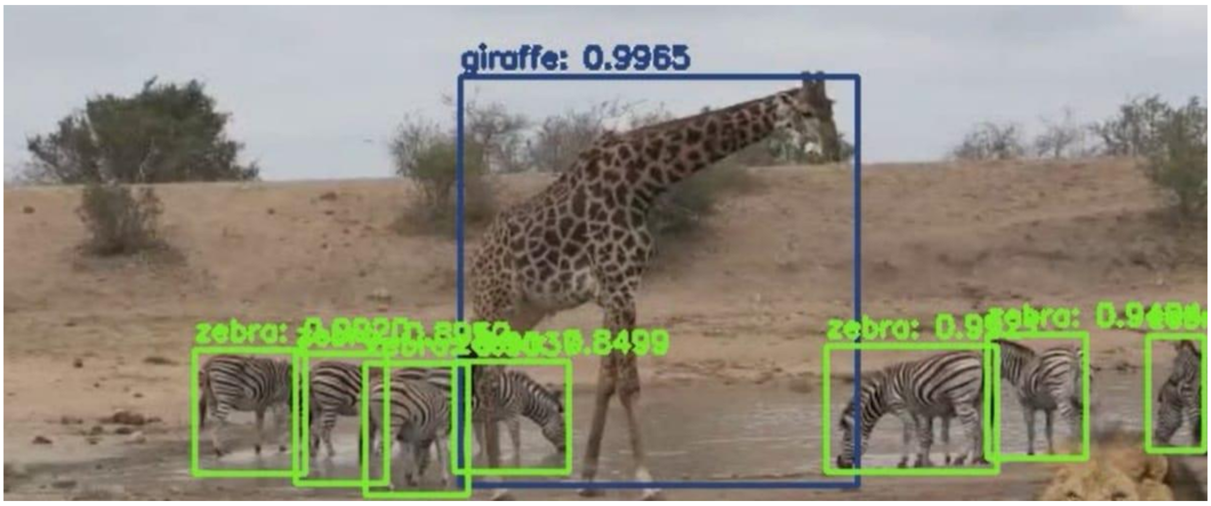
A.1 SAMPLE SCREENS









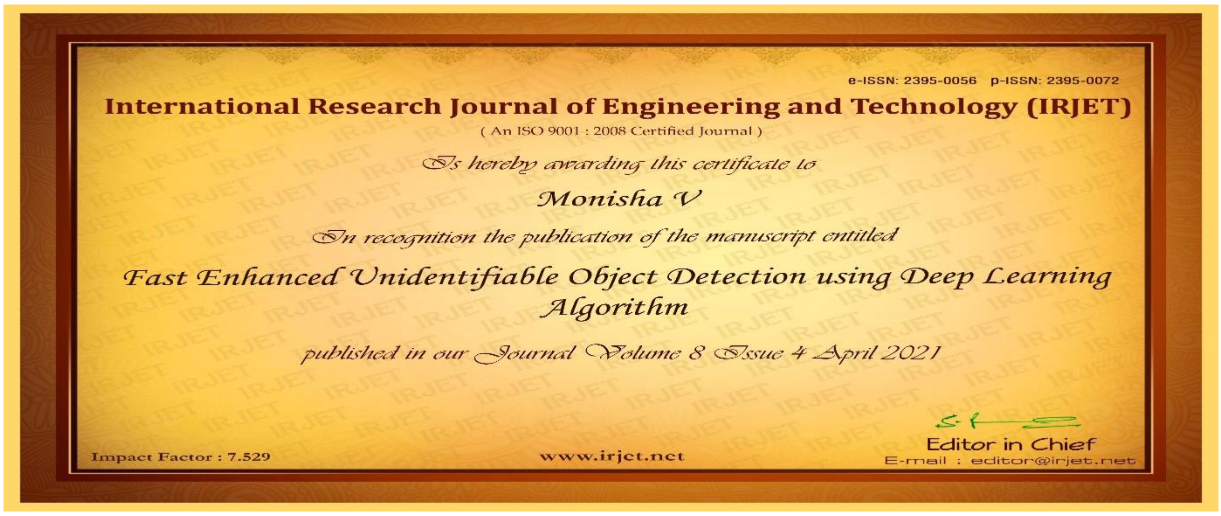


A.2 PUBLICATION

Journal name -International Research Journal of Engineering and Technology (IRJET)

Paper title -Fast Enhanced Unidentifiable Object Detection using Deep Learning Algorithm

Publication details - Volume 8, Issue 4, April 2021 ,S.No:460



e-ISSN: 2395-0056 p-ISSN: 2395-0072

International Research Journal of Engineering and Technology (IRJET)

(An ISO 9001 : 2008 Certified Journal)

Is hereby awarding this certificate to

Monisha V

In recognition the publication of the manuscript entitled

Fast Enhanced Unidentifiable Object Detection using Deep Learning Algorithm

published in our Journal Volume 8 Issue 4 April 2021

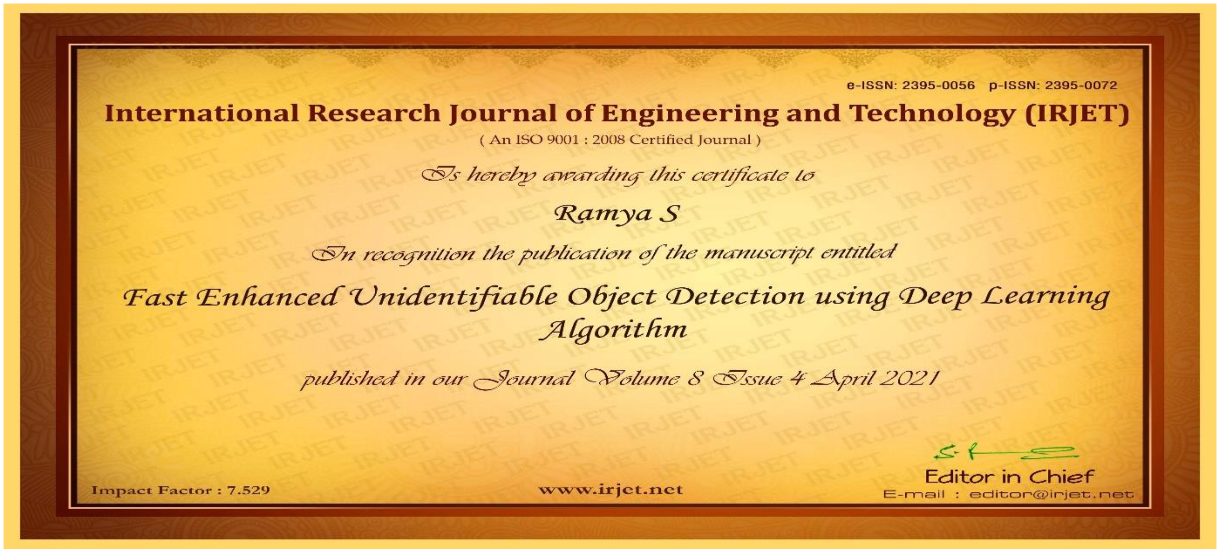
Impact Factor : 7.529

www.irjet.net

[illegible]

Editor in Chief

E-mail : editor@irjet.net



e-ISSN: 2395-0056 p-ISSN: 2395-0072

International Research Journal of Engineering and Technology (IRJET)

(An ISO 9001 : 2008 Certified Journal)

Is hereby awarding this certificate to

Ramya S

In recognition the publication of the manuscript entitled

Fast Enhanced Unidentifiable Object Detection using Deep Learning Algorithm

published in our Journal Volume 8 Issue 4 April 2021

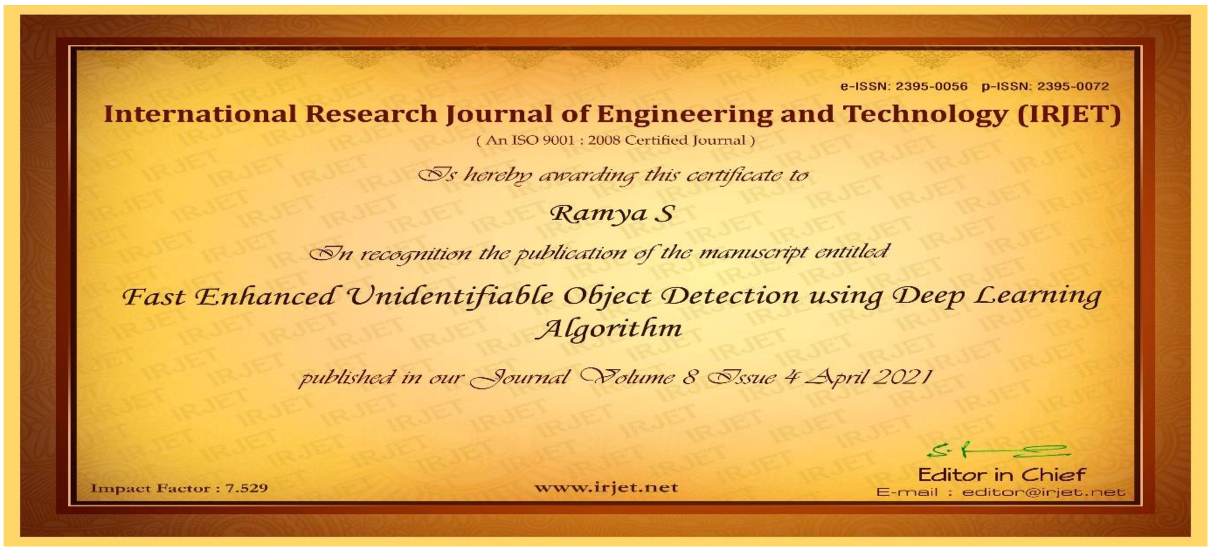
Impact Factor : 7.529

www.irjet.net

[illegible]

Editor in Chief

E-mail : editor@irjet.net



e-ISSN: 2395-0056 p-ISSN: 2395-0072

International Research Journal of Engineering and Technology (IRJET)

(An ISO 9001 : 2008 Certified Journal)

Is hereby awarding this certificate to

Ramya S

In recognition the publication of the manuscript entitled

Fast Enhanced Unidentifiable Object Detection using Deep Learning Algorithm

published in our Journal Volume 8 Issue 4 April 2021

[illegible]

Editor in Chief

Impact Factor : 7.529

www.irjet.net

E-mail : editor@irjet.net

REFERENCE

[1] P. F. Felzenszwalb, R. B. Girshick, D. Mcallester, and D. Ramanan, “Object detection with discriminatively trained part-based models,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 32, no. 9, p. 1627, 2010.

[2] K. K. Sung and T. Poggio, “Example-based learning for view-based human face detection,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 20, no. 1, pp. 39–51, 2002.

[3] C. Wojek, P. Dollar, B. Schiele, and P. Perona, “Pedestrian detection: An evaluation of the state of the art,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 34, no. 4, p. 743, 2012.

[4] H. Kobatake and Y. Yoshinaga, “Detection of spicules on mammogram based on skeleton analysis.” IEEE Trans. Med. Imag., vol. 15, no. 3, pp. 235–245, 1996.

[5] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, “Caffe: Convolutional architecture for fast feature embedding,” in ACM MM, 2014.

[6] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in NIPS, 2012.

[7] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, “Realtime multi-person 2d pose estimation using part affinity fields,” in CVPR, 2017.

[8] Z. Yang and R. Nevatia, “A multi-scale cascade fully convolutional network face detector,” in ICPR, 2016.

[9] C. Chen, A. Seff, A. L. Kornhauser, and J. Xiao, “Deepdriving: Learning affordance for direct perception in autonomous driving,” in ICCV, 2015.

[10] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia, “Multi-view 3d object detection network for autonomous driving,” in CVPR, 2017.

[11] A. Dundar, J. Jin, B. Martini, and E. Culurciello, “Embedded streaming deep neural networks accelerator with applications,” IEEE Trans. Neural Netw. & Learning Syst., vol. 28, no. 7, pp. 1572–1583, 2017.

[12] R. J. Cintra, S. Duffner, C. Garcia, and A. Leite, “Low-complexity approximate convolutional neural networks,” IEEE Trans. Neural Netw. & Learning Syst., vol. PP, no. 99, pp. 1–12, 2018.

[13] S. H. Khan, M. Hayat, M. Bennamoun, F. A. Sohel, and R. Togneri, “Cost-sensitive learning of deep feature representations from imbalanced data.” IEEE Trans. Neural Netw. & Learning Syst., vol. PP, no. 99, pp. 1–15, 2017.

[14] A. Stuhlsatz, J. Lippel, and T. Zielke, “Feature extraction with deep neural networks by a generalized discriminant analysis.” IEEE Trans. Neural Netw. & Learning Syst., vol. 23, no. 4, pp. 596–608, 2012.

[15] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in CVPR, 2014.