Annexure – 1

**REAL TIME OBJECT DETECTION**

**USING DEEP LEARNING**

**A Project Report**

***Submitted in the partial fulfilment for the award of the degree of***

**BACHELOR OF ENGINEERING**

**IN**

**CSE (Hons.) Specialisation in**

**INTERNET OF THINGS (IoT)**

**Submitted by:**

**BODDU SATEESH - 20BCS4514**

**MOGULAGANI SAIKUMAR - 20BCS4547**

**CHARANAM DINESH KUMAR - 19BCS4656**

**GELLI PAVAN KUMAR – 20BCS4539**

**Under the Supervision of:**

**Dr. RANJAN WALIA – E14288**

Text

Description automatically generated

# DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING APEX INSTITUE OF TECHNOLOGY

**CHANDIGARH UNIVERSITY, GHARUAN, MOHALI - 140413, PUNJAB**

**MAY 2023**

Annexure – 2

**DECLARATION**

I, **Boddu Sateesh, Mogulagani Saikumar, Charanam Dinesh Kumar, Gelli Pavan Kumar**, student of **‘Bachelor of Engineering in CSE(Hons.) with Specialisation in Internet of Things (IoT)**, **session: 2020-2024**, Department of Computer Science and Engineering, Apex Institute of Technology, Chandigarh University, Punjab, hereby declare that the work presented in this Project Work entitled ‘**Real Time Object Detection Using Deep Learning**’is the outcome of our own bona fide work and is correct to the best of our knowledge and this work has been undertaken taking care of Engineering Ethics. It contains no material previously published or written by another person nor material which has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgment has been made in the text.

**BODDU SATEESH - 20BCS4514**

**MOGULAGANI SAIKUMAR - 20BCS4547**

**CHARANAM DINESH KUMAR - 19BCS4656**

**GELLI PAVAN KUMAR – 20BCS4539**

**Date: 05/05/2023**

**Place: Chandigarh University**

Annexure – 3

# Table of Contents

Title Page Annexure - 1

Declaration of the student Annexure - 2

Abstract Annexure - 2

Acknowledgement Annexure - 1

1. **INTRODUCTION 1**

1.1 Problem Definition 1

1.2 Project Overview/Specification\*(page-1 and 3) 2

1.3 Hardware Specification 2

1.4 Software Specification 2

1. **LITERATURE SURVEY 3**

2.1 Existing System 3 2.2 Proposed System 4 2.3 Literature Review Summary 5

1. **DESIGN FLOW / PROCESS 8**
2. **RESULTS ANALYSIS AND VALIDATION 21**
3. **CONCLUSION AND FUTURE WORK 21**
4. **OUTPUT 22**
5. **REFERENCES 23**

#### **INTRODUCTION**

**Problem Definition:**

While the human eye can instantly and accurately recognize a given image (its content, location, and nearby images) by interacting with it, robotic systems that support computer vision can sometimes be too slow and inaccurate. . Any development in this area will lead to improvements in efficiency and performance, and could pave the way for smarter, human-like systems. Therefore, systems like advanced technology that allow humans to perform tasks with little or no awareness will certainly make our lives much easier.

Identifying each object in an image or scene using computer/software is called object detection. Object detection is one of the most important problems in the field of computer vision or wireless networks. It is the basis for complex vision tasks such as object tracking and scene understanding, and is widely used in wireless networks. The task of object detection is to determine if there is an object belonging to a specified category in the image. If it exists, the next task is to identify its category and location information.

Traditional target detection algorithms are mainly aimed at detecting multiple targets, such as pedestrian detection and infrared target detection. Due to recent advances in deep learning techniques, especially after the emergence of deep convolution neural network (CNN) techniques, object detection algorithms have undergone a revolutionary development. Among these algorithms, the three main methods commonly used in this field are You Only Look Once (YOLO), Single Shot Multiple Box Detector (SSD) and Faster Region CNN (F-RCNN).

**PROBLEM FORMULATION**

There has been a rapid and successful expansion of computer vision research in recent years. Part of this success can be attributed to the adoption and adaptation of machine learning methods, while others can be attributed to the development of new representations and models for specific computer vision problems, or the development of efficient solutions. Object detection is one area that has made significant progress. There are many objects in this world that humans have identified. So, this one is to make the machines recognize them. The current work provides an overview of object detection.

Object detection, given a set of object classes, consists in determining the location and scale of all object instances, if any, present in an image. Thus, the goal of an object detector is to find all object instances of one or more given object classes regardless of scale, location, pose camera view, partial occlusions, or lighting conditions.

**Project Overview:**

However, with the emergence of 5G, the data characteristics of wireless networks such as big data, changing business, data diversification, and uneven temporal and spatial distribution of data pose serious challenges for detection. Targets in real-time environments. Additionally, real-time object detection should be performed on any device and in any environment. To meet the challenge, the project's object detection technology detects objects in real time using a model that can run on any device in any environment.

More specifically, our proposed method applies a convolution neural network to develop a model composed of several layers to classify a given object into several defined classes. Based on recent advancements in deep learning for image processing, the proposed scheme then uses multiple images and detects objects from these images and labels them with their respective class labels.

These images can come from videos fed into our prepared model and the model will be trained until the error rate is reduced to an acceptable level. To accelerate the computational performance of object detection techniques, we use an improved single-shot multi-box detector (SSD) algorithm and a faster region convolution neural network.

**Hardware Specifications:**

* CPU – Core i5 10Gen /Ryzen 5 or above
* RAM – 8Gb or above
* ROM - 500Mb or above
* Web cam (either built-in or external)

**Software Specifications:**

* Python Compilers (Jupyter Notebook, Spyder, etc.)
* Editors (Notepad, Visual Studio Code, Code Blocks, etc.)
* Python packages for Image Recognition and Deep Learning

#### 

#### **LITERATURE REVIEW**

* + 1. **Existing System:**

Among the many uses of object detection, safety is one of the most important. Of course, there are also many object detection applications using machine learning in the security field. To our knowledge, the subjects we selected for our literature review have not been the subject of publications.

To our knowledge, no one has conducted an in-depth study of the literature on the subject we have chosen. The two studies we found closest to the survey were either specific or covered specific methods rather than analyzing other articles.

* + 1. **Proposed System:**

Use deep learning models to achieve real-time object detection and recognition in dynamic environments from images and video captured by webcams. The main objective is to detect and recognize objects in real time. All things considered, we need rich data.

We have to observe different types of objects that move relative to the camera. This will help us observe and identify different cooperating and interacting objects. In this project, we focused on precision. Use deep learning models to achieve real-time object detection and recognition in images and video captured by webcams in dynamic environments.

The main objective is to detect and recognize objects in real time. All things considered, we need rich data. We have to observe different types of objects that move relative to the camera. This will help us observe and identify different cooperating and interacting objects. In this project, we focused on precision.

* + 1. **Literature Review Summary:**

[1] A study on several techniques for object detection and tracking in video surveillance footage was done in one work by Murugan et al. The paper emphasizes how video surveillance has been a technology since the 1950s and reinforces the point made in the Introduction section about how watching security cameras can be exhausting and have a negative impact on a person's mental health. It also demonstrates how, by introducing a sophisticated surveillance system, automating the procedure was the solution to that tedious chore. The primary goal of this paper is to describe the various techniques for object detection and tracking in videos. Backdrop subtraction is the first technique described in the study. According to the report, backdrop removal is one of the most popular techniques for detecting moving objects. Background subtraction involves identifying and removing the background in order to show only the pixels of the moving object. The issue with it is that the results of the procedure are impacted if the background is not static and changes as a result of illumination or specific weather conditions. There are other algorithms for background subtraction as well.

[2] A study by Flitton et al. compares different 3D interest point descriptors for CT images of baggage at airports. The main idea here is to find interesting things during baggage x-ray inspections. The paper compared five distinct methods: density, density histogram, density gradient histogram, rotation invariant feature transform, and scale invariant feature transform. Although the research was limited to a relatively narrow issue, the report does a good job of assessing each while providing key indicators. Both of the preceding studies do an excellent job of describing the various strategies, but they are too narrow in scope in comparison to our paper. Instead of focusing on machine learning and object detection separately, both publications provide a more in-depth explanation of the object detection techniques. The two articles cover fewer papers than our Systematic Literature Review does because of their narrower scopes.

[3] Bezak, P. (September - 2016) suggests a deep learning approach for item recognition in historical architecture images in Trnava. For jobs requiring object recognition, it employs deep learning architectures based on CNN (Convolution Neural Network). Activation functions and a cascade of convolution layers are used to improve architecture. It is critical to determine the number of layers and the number of neurons in each layer. The TRNAVA LeNet 10 model was created and trained for this purpose. This model is based on 460 training images and 140 validation images, which is a 3:1 ratio. The images were color photographs that were 28x28 pixels in size and encoded in jpg format. In the image of the Trnava historical building, the model correctly identified the correct item. The prediction accuracy of the proposed model increased to 98.88%.

[4] Jung, H., Lee, S. et al (Jan – 2015), suggested that instead of using manually produced characteristics, deep learning techniques should be used to recognize face expressions. Convolution neural networks (CNN) and deep neural networks (DNN) are two types of deep networks used to solve recognizing problems. Deep networks were quickly built using deep learning toolkits that support CUDA, such as I and CudaConvnet2. They also used the OpenCV library to develop the Haar-like face detection technique. The photographs were cropped and reduced in size to 64 \* 64. The 327 face photos were then divided into ten groups, one for training and the other nine for testing. The recognition rates for six emotions were high, but the disgust label had a low recognition rate. Because the FER 2013 database contained only 547 training photos for the disgust label. Over-fitting is a possibility with the DNN.

[5] Tenguria, R., Parkhedkar et al (April – 2017), Convolution neural networks have been replaced by more precise yet sophisticated approaches that can recognize things in real-time, according to the study. This paper has the potential to make significant advances in object recognition and tagging. However, progress in this area has been relatively slow. It plans to merge the fields of computer vision and robotics, with a focus on the implementation of image description applications on an embedded system platform. Depending on the data set used to train the model, a fixed number of items are allowed in the image. According to Shaoqing Ren et al., the development of the Region Proposal Network (RPN) allows for the network to share the entire image’s convolution characteristics, resulting in nearly free region proposals. In this case, the region suggestion technique directs the algorithm to find objects in the image. Second, the application of this technique in our system makes it computationally efficient and tailored to function on low-powered platforms.

[6] Etemad, E., & Gao, Q. (Sep. – 2017): In order to improve the effectiveness of current object recognition algorithms, the research presents an object localization method that makes use of image edge information as a cue to pinpoint the locations of the objects. The image’s Generic Edge Tokens (GETs) are extracted using the perceptual organization components of human vision. These edge tokens are parsed using the Best First Search method to precisely locate objects, with the detection score provided by the Deep Convolution Neural Network serving as the goal function. When the BFS is applied to the object localization and its search space, the search space is a collection of edge elements whose overlaps with the current candidate object are greater than zero. Real-time testing revealed that the model outperformed the RCNN, and there is still room for improvement by enhancing object localization by combining picture edge, color, and texture information, as well as the image’s learned properties.

[7] Mazumdar, M., Sarasvathi, V. and Kumar, A. (Aug. – 2017), suggested a technique for creating an interactive application to identify things from films; upon user input, it is also capable of identifying the specific object now displayed on the screen. A sequential frame extraction technique for films, as well as a deep learning strategy utilizing Convolution and Fully Connected Neural Networks, is used for this challenge, which has a 77% accuracy rate. Even when the item is slightly warped, translated, rotated, or partially obscured from view and can still be easily spotted by humans, computer vision work remains difficult. Using the fact that videos are made up of frames synced with some playback audio, the video can be analyzed in much greater detail by looking at the objects present in the frame images themselves, running the classifier to obtain probabilities for various classes, and then classifying the genre as well as identifying any objects in the video. By adding more datasets and optimizing the hardware setup, this model’s operational accuracy is increased,nableing faster and more accurate item categorization over a wider variety of classes.

[8] Sujana, S. R., Abisheck, S. S., Ahmed, A. T., & Chandran, K. S. (2017), suggests a method for using convolution neural networks and the idea of deep learning to identify things. Using the input video, it generates an output with a collection of recognized objects. The convolution neural network computes a confidence score for each object. It makes use of the Single Shot Multi-box Detector, which has a high accuracy rate and uses convolution networks to identify multiple items at the same time. It employs Hard Negative Mining and Non-Maximum Suppression to increase the object’s confidence score and produce only one detection for each object, respectively. Thus, combining neural networks with deep residual networks improves the computing efficiency and accuracy of item identification.

Based on the results obtained from the literature review, the following conclusions are drawn.

* From the performance results of various object detection models on the MS COCO dataset, it can be concluded that SSD and R-FCN models are faster than Faster R-CNN.
* But if accuracy is more important than speed, Faster R-CNN outperforms SSD and R-FCN models.
* Faster R-CNN is the most accurate model when using Inception ResNet, operates at 1 frame per second and meets the minimum requirements to perform real-time object detection and recognition.
* Compared to other object detection models, SSD is faster but has difficulty detecting small objects.
* The speed of Faster R-CNN increases as the number of proposals decreases, which also reduces the accuracy of the model.
* According to Redmond et al, YOLOv3 detects 10 times faster than state-of-the-art methods. Therefore, YOLOv3 and its variant Tiny-YOLOv3 were chosen for the experiments.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S.NO** | **TITLE** | **YEAR** | **AUTHOR** | **TECHNIQUES** |
| 1 | Application of deep learning in object detection | 978-1-5090-5507- 4/17/$31.00 ©2017 IEEE ICIS 2017, May 24-26, 2017, Wuhan, China 2017 | Xinyi Zhou, Wei Gong, WenLong Fu, Fengtong Du | **DATASETS:**   * ImageNet * PASCALVOC * COCO   **METHODOLOGY:**   * R-CNN * SPP-net * Fast R-CNN * Faster R-CNN |
| 2 | An Efficient approach for object detection and object tracking | 2017 Third International Conference on Science Technology Engineering and Management | B. Maga | **MODULE:**   * Kernel Method and Training * Feature method * Template Generation   **TECHNIQUE:**   * Template Matching * Later Based Tracking |
| 3 | Object detection based on deep learning of small samples | 2018 Tenth International Conference on Advanced Computational Intelligence (ICACI) March 2018, Xiamen, China | Ce Li, Yachao Zhang | **KEY STEPS:**   * Foreground objects extraction * Background selection and fusion processing * Object Sem |
| 4 | A Learning algorithm for model-based object detection | 2011,8th International Conference on Ubiquitous Robots and Ambient Intelligence | Chen Guodong, Zeyang Xia, Rongchuan Sun, Zhenhua Wang, Zhiwu Ren and Lining Sun | **KEY STEPS:**   * Object detection * Shape Matching * Image Segmentation Shape Fragment   **PROPERTIES:**   * Rotation invariance * Scale invariance * Noise robustness |
| 5 | Object detection and tracking | 2015 INTERNATIONAL CONFERENCE ON COMPUTATIONAL INTELLIGENCDE AND COMMUNICATION NETWORK | K. Rasool Reddy, K. Hari Priya, N. Neelima | **KEY STEPS:**   * Incremental Multiple principle component analysis * Frag Track * HOG - LBP Detector Generative and Discriminative Trackers Semi Supervised Support Vector Machines |
| 6 | Modelling from an object and multi-object tracking system | 2016, Global Summit on Computers and Information Technology | Afef SALHI, Yacine MORESLY, Fahmi GHOZZI, Ameni YENGUI, and Ahmed FAKHFAKH | **KEY STEPS:**   * Block-matching * KLT algorithm (Kanade Lucas Tomasi) * Meanshift algorithm (MA) Camshift Algorithm (CA) |
| 7 | Object detection in sports video | MIPRO 2018, May 21-25, 2018, Opatija Croatia | M. Buric, M. Pobar, M.IvasicKos | **METHODOLOGY :**   * Mask R-CNN * YOLO object detector Mixture of Gaussians method |
| 8 | Object Tracking Camera | IJSRD - International Journal for Scientific Research & Development| Vol. 3, Issue 03, 2015 | ISSN (online): 2321- 0613 | Priyanka Pacharne, Sanket Kotkar, Neha Darekar | **KEY STEPS :**   * Colour Model * Object Tracking * Image Acquisition Background Subtraction |
| 9 | A Survey on Object Tracking in Video | 2017, IJSRD - International Journal for Scientific Research & Development | Snehlata Raisagar, Ashish Tiwari | **KEY STEPS :**   * Video Sequence * Object Detection * Object Recognition Tracking |
| 10 | Detection and Tracking of Moving Object in Video - A Survey | 2016, || National Conference on Technological Advancement and Automatization in Engineering | Dhaval Deshpande, Nikhil Aatkare, Prof.Reena Somani | **STATISTICAL METHODS :**   * Background Subtraction * Temporal Differencing Correspondence Based Matching * Algorithm Kernel Tracking |

1. **DESIGN FLOW/PROCESS**

**Deep learning:** A technique used in artificial intelligence (AI) called deep learning teaches computers to interpret data in a manner modelled after the human brain. Deep learning models can identify intricate patterns in images, text, audio, and other types of data to generate precise analyses and forecasts.

**Convolution neural networks:** A CNN is a particular type of network design for deep learning algorithms that is utilised for tasks like image recognition and pixel data processing. Although there are other kinds of neural networks in deep learning, CNNs are the preferred network architecture for identifying and recognising objects.

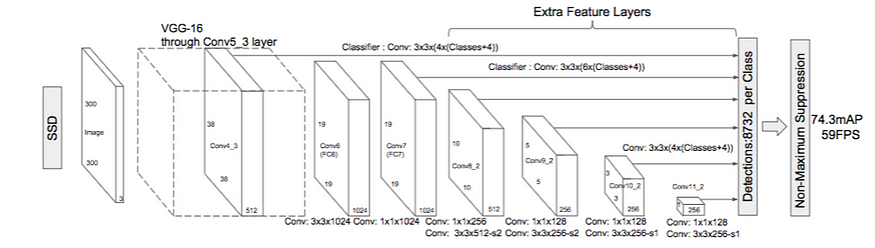
**Single shot Multi box detector:**

**Single Shot:**this means that the tasks of object localization and classification are done in a single forward pass of the network

**Multi Box:**this is the name of a technique for bounding box regression developed by Szegedy et al. (we will briefly cover it shortly)

**Detector:**The network is an object detector that also classifies those detected objects

**Architecture**



The architecture of SSD is based on the time-tested VGG-16 architecture, as seen in the diagram above, but does away with the completely connected layers. VGG-16 was chosen as the basic network due to its good performance in jobs requiring the categorization of high-quality images and its widespread application in issues where transfer learning can aid to improve outcomes. A number of auxiliary convolutional layers (starting with conv6) were added in place of the initial VGG fully connected layers, allowing for the extraction of features at various scales and a gradually smaller input size for each succeeding layer**.**

Diagram

Description automatically generated

## **MultiBox**

Szegedy's work on MultiBox, a technique for quick class-independent bounding box coordinate proposals, served as an inspiration for SSD's bounding box regression algorithm. It's interesting to note that a convolutional network modelled after Inception is employed in the MultiBox research. The 1x1 convolutions you can see below aid in dimensionality reduction since they reduce the number of dimensions while maintaining the same "width" and "height".

Diagram

Description automatically generated

Additionally, MultiBox loss function combined two essential elements that found their way into SSD:

**Confidence Loss:** This gauges the network's trust in the computed bounding box's objectivity. To compute this loss, categorical cross-entropy is employed.

**Location Loss:** It is a metric used to express how far the network's projected bounding boxes are from the training set's actual ones. Here, L2-Norm is applied.

The equation for the loss, which expresses how far off our prediction "landed," is as follows. I won't go too mathematical here; read the paper if you're inquisitive and want a more exact notation:

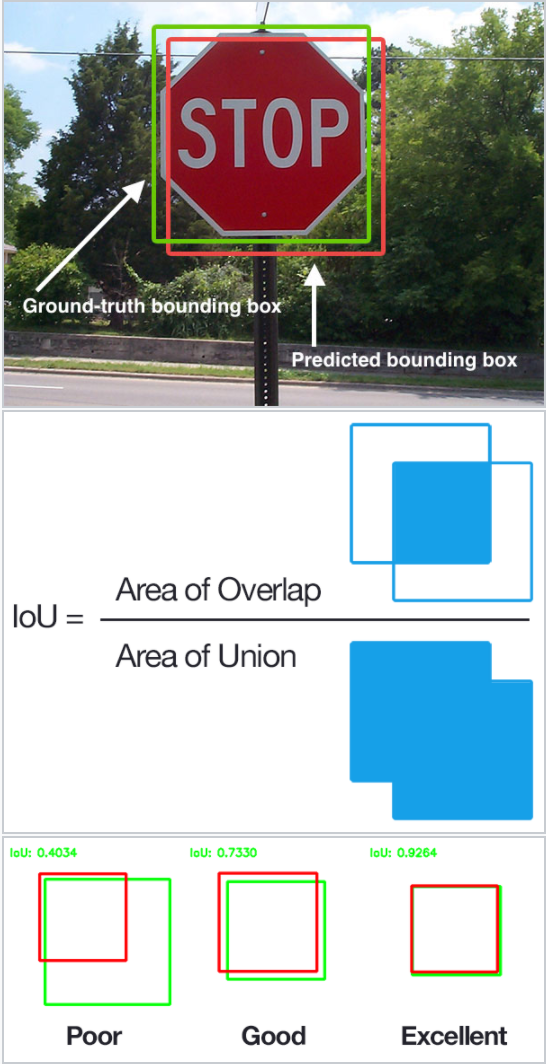
**Multibox\_Loss = Confidence\_Loss + Alpha \* Location\_Loss**

We can balance the contribution of the location loss with the aid of the alpha term. As is customary in deep learning, the objective is to identify the parameter values that minimise the loss function and increase the accuracy of our predictions.

## **MultiBox Priors and IoU:**

Contrary to what I previously indicated, the rationale underlying the production of bounding boxes is considerably more intricate. But do not worry; it is still attainable.

Priors, also known as anchors in Faster-R-CNN parlance, are pre-computed, fixed-size bounding boxes that are constructed by the researchers in MultiBox and closely resemble the distribution of the initial ground truth boxes. In actuality, such priors are chosen so that their Intersection over Union ratio (also known as IoU and occasionally as the Jaccard index) is greater than 0.5. As you can see from the figure below, an IoU of 0.5 still falls short of ideal, but it does give the bounding box regression technique a firm starting point. This is a much better approach than having the predictions begin with random coordinates! MultiBox attempts to regress towards the true bounding boxes by starting with the priors as predictions.



The resulting architecture, which has 11 priors per feature map cell (8x8, 6x6, 4x4, 3x3, and 2x2) and just one on the 1x1 feature map, yields a total of 1420 priors per image and allows for robust coverage of input images at various scales, allowing for the detection of objects of different sizes.

MultiBox only keeps the top K predictions that have the fewest location (LOC) and confidence (CONF) losses at the end.

# TRAINING & RUNNING SSD

## **Datasets**

You will require training and test datasets with assigned class labels (just one per bounding box) and ground truth bounding boxes. The COCO and VOC datasets from Pascal are fantastic places to start.

Graphical user interface, website

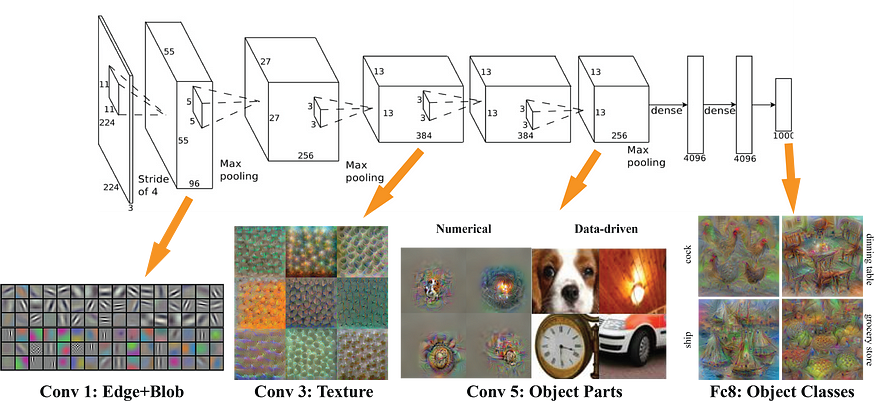
Description automatically generated

## **Default Bounding Boxes**

To ensure that the majority of objects could be caught, it is advised to define a diversified collection of default bounding boxes, of different scales and aspect ratios. Each feature map cell in the SSD article includes about 6 bounding boxes.

## **Feature Maps**

Running MultiBox on multiple feature maps increases the likelihood that any object, big or small, will eventually be detected, localised, and correctly classified. Features maps, or the results of the convolutional blocks, are a representation of the dominant features of the image at different scales. How the network "sees" a particular image across its feature maps is depicted in the graphic below:



**Hard Negative Mining**

We risk having an excessive number of negative examples in our training set since the majority of the bounding boxes will have low IoU during training and will consequently be interpreted as negative training examples. Therefore, it is advised to maintain a ratio of negative to positive examples of about 3:1 rather than using only negative predictions. You must maintain negative samples since the network needs to understand what makes an inaccurate detection and be explicitly told what does not.

Diagram

Description automatically generated

## **Non-Maximum Suppression (NMS)**

Given the large number of boxes produced during a forward pass of SSD at inference time, it is crucial to prune the majority of the bounding box using a method known as non-maximum suppression. Only the top N predictions are kept, and boxes with a confidence loss threshold less than ct (for example, 0.01) and IoU less than lt (for example, 0.45) are kept. This makes sure that the network only keeps the most likely predictions and discards the noisier ones.

Graphical user interface, website

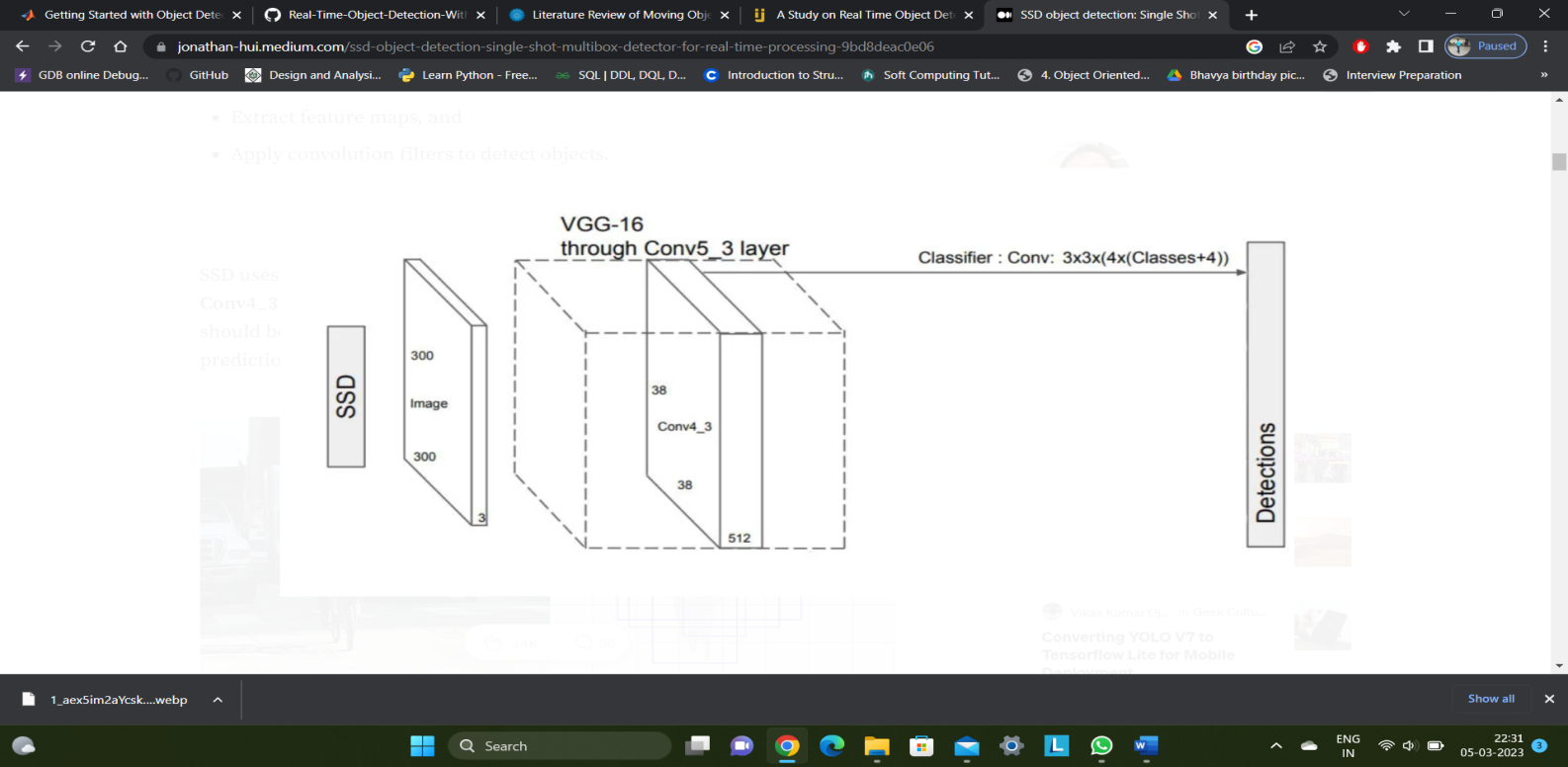
Description automatically generated

**Working**

Real-time object detection is a feature of SSD. Faster R-CNN uses boundary boxes that are generated by a region proposal network to classify things. The entire process moves at a speed of 7 frames per second, yet it is thought to be state-of-the-art in precision. Below the requirements for real-time processing. By doing away with the necessity for the regional proposal network, SSD accelerates the process. SSD implements a few improvements, such as multi-scale features and default boxes, to make up for the loss in accuracy. These enhancements enable SSD to match the Faster R-accuracy CNN’s while employing images of lesser quality, significantly increasing speed. The following comparison shows that it surpasses the accuracy of the Faster R-CNN and even achieves real-time processing performance.

The SSD object detection composes of 2 parts:

* Extract feature maps.
* Apply convolution filters to detect objects.



SSD uses VGG16 to extract feature maps. Then it detects objects using the Conv4\_3 layer. For illustration, we draw the Conv4\_3 to be 8 × 8 spatially (it should be 38 × 38). For each cell (also called location), it makes 4 object predictions

Graphical user interface, application

Description automatically generated

Each prediction composes of a boundary box and 21 scores for each class (one extra class for no object), and we pick the highest score as the class for the bounded object. Conv4\_3 makes a total of 38 × 38 × 4 predictions: four predictions per cell regardless of the depth of the feature maps. As expected, many predictions contain no object. SSD reserves a class “0” to indicate it has no objects.

Graphical user interface, website

Description automatically generated

**Convolution predictors for object detection:**

SSD does not use a delegated region proposal network. Instead, it resolves to a very simple method. It computes both the location and class scores using small convolution filters. After extracting the feature maps, SSD applies 3 × 3 convolution filters for each cell to make predictions. (These filters compute the results just like the regular CNN filters.) Each filter outputs 25 channels: 21 scores for each class plus one boundary box (detail on the boundary box later).

Graphical user interface

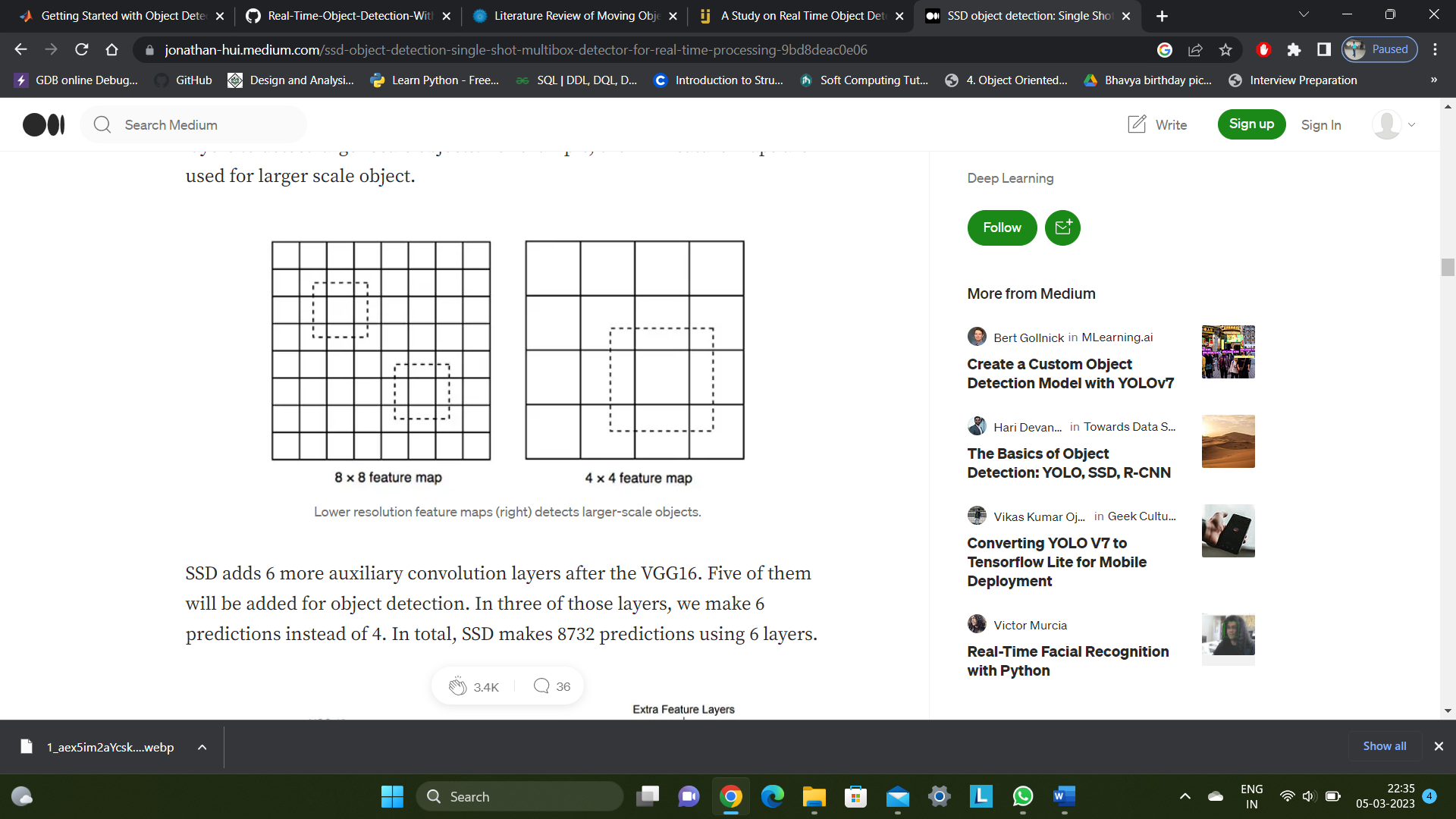
Description automatically generated

For example, in Conv4\_3, we apply four 3 × 3 filters to map 512 input channels to 25 output channels.



**Multi-scale feature maps for detection:**

At first, we describe how SSD detects objects from a single layer. Actually, it uses multiple layers (multi-scale feature maps) to detect objects independently. As CNN reduces the spatial dimension gradually, the resolution of the feature maps also decrease. SSD uses lower resolution layers to detect larger scale objects. For example, the 4× 4 feature maps are used for larger scale object.



SSD adds 6 more auxiliary convolution layers after the VGG16. Five of them will be added for object detection. In three of those layers, we make 6 predictions instead of 4. In total, SSD makes 8732 predictions using 6 layers.

**Choosing default boundary boxes:**

**Graphical user interface, text

Description automatically generated**

Default boundary boxes are chosen manually. SSD defines a scale value for each feature map layer. Starting from the left, Conv4\_3 detects objects at the smallest scale 0.2 (or 0.1 sometimes), and then increases linearly to the rightmost layer at a scale of 0.9. Combining the scale value with the target aspect ratios, we compute the width and the height of the default boxes. For layers making 6 predictions, SSD starts with 5 target aspect ratios: 1, 2, 3, ½, and 1/3. Then the width and the height of the default boxes are calculated as:

Graphical user interface, text

Description automatically generated

And aspect ratio = 1.

**Matching Strategy:**

SSD predictions are classified as positive matches or negative matches. SSD only uses positive matches in calculating the localization cost (the mismatch of the boundary box). If the corresponding default boundary box (not the predicted boundary box) has an IoU greater than 0.5 with the ground truth, the match is positive. Otherwise, it is negative. (IoU, the intersection over the union, is the ratio between the intersected areas over the joined area for two regions.)

## **RESULTS ANALYSIS AND VALIDATION**

* Open the command prompt na dthen enter the command as below :

**python real\_time\_object\_detection.py --prototxt MobileNetSSD\_deploy.prototxt.txt --model MobileNetSSD\_deploy.caffemodel**



1. **CONCLUSION AND FUTURE WORK**

One of the newest and most fascinating areas of deep learning is object recognition. Face detection is a popular use of object detection, found in almost all smartphone cameras. On multiple deep learning models capable of performing real-time object detection and recognition. By studying the performance of these algorithms on standard datasets, YOLOv3, Tiny-YOLOv3 and Faster R-CNN have been identified as the most suitable and efficient deep learning models to perform detection and recognition of real-time objects on scale engineering vehicles. It is concluded that Faster R-CNN performs on par with SSD and FCN models in terms of speed, while exhibiting better accuracy compared to these models.

As In the modern world, object detection and recognition can be regarded as one of the most difficult, complicated, and crucial tasks in the field of computer vision. As far as we know, this project was created with the fundamental goal of capturing real-time objects in images, videos, or web cams.

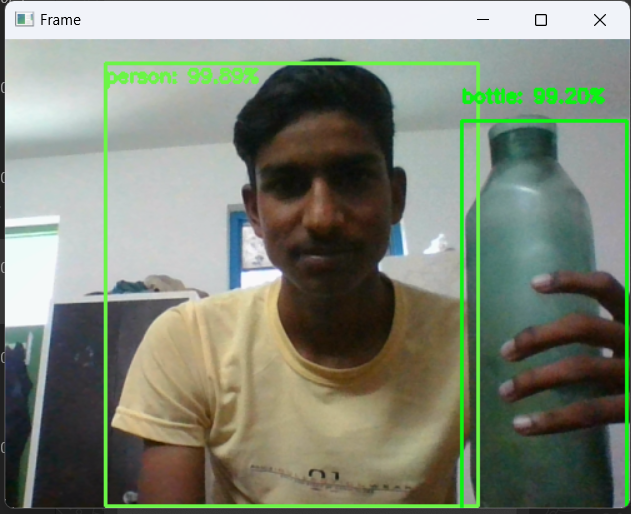
• Future innovations can be concentrated by putting the project on a system with GPU for quicker outcomes and greater accuracy. • It can be improved and innovated in the future by anyone without worrying about complexity.

• For instance, MS COCO does small object detection in several face detection tasks and applications. For better small object localization across partial barriers. In order to improve the network architecture, we will make some changes.

• To achieve accurate and efficient recognition of small objects, thereby reducing reliance on data networks.

Thus, it can be said in the end that in order to improve accuracy and performance, preprocessing techniques like edge detection and increasing image augmentation and contrast should be used.

1. **OUTPUT**



1. **REFERENCES**
2. S. Murugan, K. S. Devi (2018), A. Sivaranjani, and P. Srinivasan, “A study on various methods used for video summarization and moving object detection for video surveillance applications,” Multimer. Tools Appl., vol. 77, no. 18, pp. 23273– 23290.
3. G. Flitton, T. P. Breckon, and N. Megherbi, “A comparison of 3D interest point descriptors with application to airport baggage object detection in complex CT imagery,” Pattern Recognition., vol. 46, no. 9, pp. 2420–2436, 2013.
4. Bezak, P. (2016, September). Building recognition system based on deep learning. In 2016 Third International Conference on Artificial Intelligence and Pattern Recognition (AIPR) (pp. 1-5). IEEE.
5. Jung, H., Lee, S., Park, S., Kim, B., Kim, J., Lee, I., & Ahn, C. (2015, January). Development of deep learning-based facial expression recognition system. In 2015 21st Korea-Japan Joint Workshop on Frontiers of Computer Vision (FCV) (pp. 1-4). IEEE.
6. Tenguria, R., Parkhedkar, S., Modak, N., Madan, R., & Tondwalkar, A. (2017, April). Design framework for general purpose object recognition on a robotic platform. In 2017 International Conference on Communication and Signal Processing (ICCSP) (pp. 2157-2160). IEEE.
7. Etemad, E., & GAO, Q. (2017, September). Object localization by optimizing convolutional neural network detection score using generic edge features. In 2017 IEEE International Conference on Image Processing (ICIP) (pp. 675-679). IEEE.
8. Mazumdar, M., Sarasvathi, V., & Kumar, A. (2017, August). Object recognition in videos by sequential frame extraction using convolutional neural networks and fully connected neural networks. In 2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS) (pp. 1485-1488). IEEE.
9. Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 779-788).
10. Ren, S., He, K., Girshick, R., & Sun, J. (2016). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In Advances in Neural Information Processing Systems (NIPS) (pp. 91-99).
11. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. (2016). SSD: Single Shot MultiBox Detector. In European Conference on Computer Vision (ECCV) (pp. 21-37).
12. Redmon, J., & Farhadi, A. (2018). YOLOv3: An Incremental Improvement. arXiv preprint arXiv:1804.02767.
13. Liu, W., Rabinovich, A., & Berg, A. C. (2015). ParseNet: Looking Wider to See Better. arXiv preprint arXiv:1506.04579.
14. Dai, J., Li, Y., He, K., & Sun, J. (2016). R-FCN: Object Detection via Region-based Fully Convolutional Networks. In Advances in Neural Information Processing Systems (NIPS) (pp. 379-387).
15. Lin, T. Y., Dollár, P., Girshick, R., He, K., Hariharan, B., & Belongie, S. (2017). Feature Pyramid Networks for Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 2117-2125).
16. Redmon, J., & Farhadi, A. (2017). YOLO9000: Better, Faster, Stronger. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 7263-7271).
17. Li, Z., Peng, C., Yu, G., Zhang, X., Deng, Y., & Sun, J. (2017). DetNet: A Backbone network for Object Detection. arXiv preprint arXiv:1804.06215.
18. Huang, J., Rathod, V., Sun, C., Zhu, M., Korattikara, A., Fathi, A.,... & Murphy, K. (2017). Speed/accuracy trade-offs for modern convolutional object detectors. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 3296-3297).
19. Lin, T. Y., Goyal, P., Girshick, R., He, K., & Dollár, P. (2017). Focal Loss for Dense Object Detection. In Proceedings of the IEEE International Conference on Computer Vision (ICCV) (pp. 2999-3007).
20. Law, H., & Deng, J. (2018). CornerNet: Detecting Objects as Paired Keypoints. In Proceedings of the European Conference on Computer Vision (ECCV) (pp. 734-750).
21. Zhou, Y., Zhu, Y., Bai, S., & Bai, X. (2019). Objects

**LINKS:**

www.mindwise-groningen.nl/deep-learning-the-beautiful-mind/

https://jwcn-eurasipjournals.springeropen.com/articles/10.1186/s13638-020-01826-x

https://www.ijert.org/a-study-on-real-time-object-detection-using-deep-learning#:~:text=Object%20detection%20empowers%20recognizing%20instance,much%20like%20human%20vision%20works.

http://www.diva-portal.org/smash/get/diva2:1414033/FULLTEXT02.pdf

https://www.ijraset.com/research-paper/moving-object-detection-using-ml

https://youtu.be/ATw1Dy4p1GU

https://youtu.be/hMFx1TXjAJc

https://youtu.be/1P6K9pVKgVo