### A Major Project report on

### OBJECT RECOGNIZATION SYSTEM USING IMAGE PROCESSING

### BACHELOR OF TECHNOLOGY

IN

## ELECTRONICS AND COMMUNICATION ENGINEERING



Submitted By

### NAMAJI VIVEK 19911A04F4

## Under the esteemed guidance of

### Dr. S . THULASI PRASAD

**Professor**

Department of Electronics and Communication Engineering

Vidya Jyothi Institute of Technology

(An Autonomous Institution)

(Accredited by NAAC & NBA, Approved by AICTE New Delhi & Permanently Affiliated to JNTUH)

Aziz Nagar Gate, C.B. Post, Hyderabad-500 075

2022-2023



[Accredited by NAAC & NBA, Approved by AICTE New Delhi & Permanently Affiliated to JNTUH]

(Aziz Nagar, C.B. Post, Hyderabad -500075)

**Department of Electronics and Communication Engineering**



**CERTIFICATE**

This is to certify that the project report entitled **OBJECT RECOGNIZATION SYSTEM USING IMAGE PROCESSING** is being submitted by **NAMAJI VIVEK (19911A04F4)** of B. Tech IV- II semester of Electronics & Communication Engineering is a record Bonafide work carried out by them. The results embodied in this report have not been submitted to any other University for the award of any degree.

**INTERNAL GUIDE**

**Dr. S. Thulasi Prasad**

**Professor**

**Department Of ECE**

**HEAD OF THE DEPARTMENT**

**Dr. M. Rajendra Prasad**

**Professor & HOD**

**Department of ECE**

**EXTERNAL EXAMINER**

**DECLARATION**

This is to certify that the work reported in the present project entitled **Object Recognization System Using Image Processing** is a record of work done by us in the Department of Electronics and Communication Engineering, Vidya Jyothi Institute of Technology, Jawaharlal Nehru Technological University, Hyderabad. The reports are based on the project work done entirely by us and not copied from any other source.

**PROJECT ASSOCIATES:**

**NAMAJI VIVEK (19911A04F4)**

**ACKNOWLEDGEMENT**

I would like to express our sincere gratitude to our guide **Dr. S. Thulasi Prasad,** Professor who has guided and supported us through every stage in the project.

I am really grateful to **Mr. S. Pradeep Kumar Reddy,** Project Coordinator for his time to time, much needed valuable guidance throughout my study.

I am really grateful to **Dr M. RAJENDRA PRASAD, Professor & HOD ECE Dept**, Vidya Jyothi Institute of Technology for his time to time, much needed valuable guidance throughout my study.

I express our hearted gratitude to **Dr. A. PADMAJA Principal** Vidya Jyothi Institute of Technology for giving us spontaneous encouragement for completing the project.

I thank **Dr. E. SAIBABA REDDY, Director** Vidya Jyothi Institute of Technology for encouraging us in the completion of our project.

It is our privilege to express our gratitude and indebtedness to **Dr. P. Rajeshwar Reddy** Secretary& Correspondent of Vidya Jyothi Institute of Technology for his moral support.

I express our heartful thanks to Staff of Electronics and Communication Department, Vidya Jyothi Institute of Technology for helping us in carrying out our project successfully.

### Project Associates:

**NAMAJI VIVEK 19911A04F4**

# ABSTRACT

Detecting and recognizing Yolo and CNN objects in unstructured as well as structured environments is one of the most challenging tasks in computer vision and artificial intelligence research. This paper introduces a new computer vision-based object detection method for technology and its applications. Each individual image pixel is classified as belonging either to an obstacle based on its appearance. The method uses a single lens webcam camera that performs in real-time, and also provides a binary image at high resolution. For training and detection of objects CNN(convolutional neural network) used for more accuracy. We can detect with in seconds using YOLO with RCNN. using this we can detect images, create images. Yolo gives better output and taking less speed than others.

An object recognition system using image processing is a computer vision technology that automatically identifies and classifies objects within digital images or video streams. This system involves several stages, including image acquisition, preprocessing, feature extraction, object detection, and classification. In the image acquisition stage, digital images or video frames are captured using a camera or other imaging devices. In the preprocessing stage, the acquired images are enhanced and normalized to ensure optimal quality. In the feature extraction stage, relevant features of the objects are extracted from the preprocessed images. In the object detection stage, the locations of the objects within the images are identified using techniques such as edge detection or template matching. Finally, in the classification stage, the recognized objects are classified into predefined categories based on their features. This system has numerous practical applications, including robotics, surveillance, and autonomous vehicles.

**INDEX**

|  |  |  |
| --- | --- | --- |
| **CHAPTER N0** | **TITLE** | **PAGE**  **NO** |
| 1 | INTRODUCTION   * 1. Introduction of the project   2. Problem statement   3. Objectives   4. Basics of image processing      1. Image      2. Image files      3. Image file formats   5. Introduction to Image processing      1. Image acquisition      2. Image enhancement      3. Image Restoration      4. Colour image processing      5. Segmentation      6. Image compression   6. Classification of Images      1. Binary image      2. Grey scale image      3. Colour image   7. Introduction to video streaming   8. Deep learning classification | 1  1  1  1  2  2  4  4  5  6  7  8  9  10  11  13  13  14  15  16  17 |
| 2 | Literature Review | 22 |
| 3 | Methodology   * 1. Proposed System   2. Proposed Block diagram   3. Working principle | 24  24  25  25 |
| 4 | Hardware Description   * 1. Raspberry Pi   2. Camera | 27  27  30 |
| 5 | Software Description   * 1. Convolution neural network   2. You Only Look Once   3. Introduction to computer vision | 31  31  33  37 |
| 6 | Results and Discussions | 58 |
| 7 | Conclusions and Future scope  7.1 Conclusion  7.2 Future scope  References | 60  60  60  61 |

**LIST OF FGURES**

|  |  |  |
| --- | --- | --- |
| **FIGURE NO** | **TITLE** | **PAGE NO** |
| 1.1 | Colour image to Gray scale Conversion Process | 2 |
| 1.2 | Gray Scale Image Pixel Value Analysis | 3 |
| 1.3 | BIT Transferred for Red, Green and Blue plane | 3 |
| 1.4 | Horizontal and Vertical Process | 4 |
| 1.5 | Basics steps of image Processing | 6 |
| 1.6 | Digital camera | 7 |
| 1.7 | Mobile based Camera | 7 |
| 1.8 | Image enhancement process for Gray Scale Image and Color Image using Histogram Bits | 8 |
| 1.9 | Noise image Enhancement | 9 |
| 1.10 | gray Scale image Color Image | 10 |
| 1.11 | Image Segment Process | 11 |
| 1.12 | Block diagram of image compression | 12 |
| 1.13 | Decompression Process for Image | 12 |
| 1.14 | Saturation Process of RGB SCALE Image | 16 |
| 3.1 | Hardware Block Diagram | 25 |
| 3.2 | Software Block Diagram | 25 |
| 4.1 | Raspberry pi | 27 |
| 4.2 | Raspberry pi 40 pin GPIO Header | 29 |
| 4.3 | Pi camera | 30 |
| 5.1 | Convolution neural network | 31 |
| 5.2 | Residual blocks | 34 |
| 5.3 | Bounding boxes | 35 |
| 5.4 | Intersection over union | 37 |
| 5.5 | Grey scale values to simple array numbers | 39 |
| 5.6 | RGB pixels | 40 |
| 6.1  6.2 | Result of object detection  Result of animals and vehicles detection | 58  59 |

**LIST OF TABLES**

|  |  |  |
| --- | --- | --- |
| **TABLE NO** | **TITLE** | **PAGE**  **NO** |
| 5.1 | Python Versions With Its Released Date | 47 |

**CHAPTER-1 INTRODUCTION**

## INTRODUCTION OF THE PROJECT

Object detection is the one of the best technique for the detection of the any object. For this lot of algorithms are used. Here we are using open-cv for any real time applications. Here, yolo and RCNN is used for the purpose of detection. Yolo recognize 49 images at a time. Popular algorithms used to perform object detection include convolutional neural networks (CNN, Convolutional Neural Networks) and YOLO (You Only Look Once). The CNN's are in the CNN family, while YOLO is part of the single-shot detector family.

The identification of objects in an image and this process would probably start with image processing techniques such as noise removal, followed by (low-level) feature extraction to locate lines, regions and possibly areas with certain textures. The clever bit is to interpret collections of these shapes as single objects, e.g. cars on a road, boxes on a conveyor belt or cancerous cells on a microscope slide. One reason this is an AI problem is that an object can appear very different when viewed from different angles or under different lighting. Another problem is deciding what features belong to what object and which are background or shadows etc. The human visual system performs these tasks mostly unconsciously but a computer requires skillful programming and lots of processing power to approach human performance. Manipulation of data in the form of an image through several possible techniques. An image is usually interpreted as a two-dimensional array of brightness values, and is most familiarly represented by such patterns as those of a photographic print, slide, television screen, or movie screen. An image can be processed optically or digitally with a computer.

## PROBLEM STATEMENT

Here for the object detection problem is with detection of object in real time is not possible with more objects.so, we need to avoid it. We are worrying with the object classification, for labelling the image, not detect all things. So, I m introducing new process.

## OBJECTIVES

* + - The main theme of the project is to detect the objects using yolo for fast operation with CNN training is very high.
    - The main objective of the experiment is detection of lot off objects at a time.
    - So, we can provide DNN, yolo.
    - Using this we can detect any type of object using this yolo.
    - It gives high speed, high accuracy, stronger and better.

## 

## 1.4 BASICS OF IMAGE PROCESSING

**1.4.1 IMAGE**

An image is a two-dimensional picture, which has a similar appearance to some subject usually a physical object or a person.

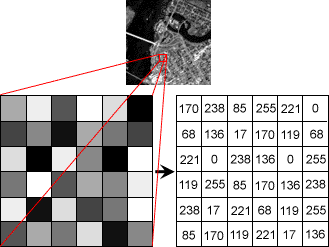
Image is a two-dimensional, such as a photograph, screen display, and as well as a three-dimensional, such as a statue. They may be captured by optical devices— such as cameras, mirrors, lenses, telescopes, microscopes, etc. and natural objects and phenomena, such as the human eye or water surfaces.

The word image is also used in the broader sense of any two-dimensional figure such as a map, a graph, a pie chart, or an abstract painting. In this wider sense, images can also be rendered manually, such as by drawing, painting, carving, rendered automatically by printing or computer graphics technology, or developed by a combination of methods, especially in a pseudo-photograph.



### Fig 1.1: Colour Image To Gray Scale Conversion Process

An image is a rectangular grid of pixels. It has a definite height and a definite width counted in pixels. Each pixel is square and has a fixed size on a given display. However different computer monitors may use different sized pixels. The pixels that constitute an image are ordered as a grid (columns and rows); each pixel consists of numbers representing magnitudes of brightness and color.



### Fig 1.2 : Gray Scale Image Pixel Value Analysis

Each pixel has a color. The color is a 32-bit integer. The first eight bits determine the redness of the pixel, the next eight bits the greenness, the next eight bits the blueness, and the remaining eight bits the transparency of the pixel.



**Fig 1.3: BIT Transferred for Red, Green and Blue plane (24bit=8bit red;8-bit green;8bit blue)**

## 1.4.2 IMAGE FILE SIZES

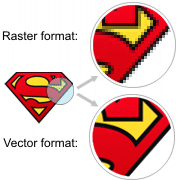
Image file size is expressed as the number of bytes that increases with the number of pixels composing an image, and the color depth of the pixels. The greater the number of rows and columns, the greater the image resolution, and the larger the file. Also, each pixel of an image increases in size when its color depth increases, an 8-bit pixel (1 byte) stores 256 colors, a 24-bit pixel (3 bytes) stores 16 million colors, the latter known as true color. Image compression uses algorithms to decrease the size of a file. High resolution cameras produce large image files, ranging from hundreds of kilobytes to megabytes, per the camera's resolution and the image-storage format capacity. High resolution digital cameras record 12 megapixel (1MP = 1,000,000 pixels / 1 million) images, or more, in true color. For example, an image recorded by a 12 MP camera; since each pixel uses 3 bytes to record true color, the uncompressed image would occupy 36,000,000 bytes of memory, a great amount of digital storage for one image, given that cameras must record and store many images to be practical. Faced with large file sizes, both within the camera and a storage disc, image file formats were developed to store such large images.

Image file sizes refer to the amount of digital storage space an image file occupies on a computer or storage device. The size of an image file depends on several factors such as the image resolution, color depth, and file format.

* + - * Resolution: The resolution of an image is the number of pixels or dots per inch (dpi) that it contains. A higher resolution image will generally have a larger file size than a lower resolution image.
      * Color depth: Color depth refers to the number of colors that an image can display. Images with higher color depth will have more detailed color information and therefore, a larger file size.
      * File format: Different file formats, such as JPEG, PNG, GIF, and BMP, have different compression methods and can result in varying file sizes. Some formats, like JPEG, use lossy compression to reduce file size, while others, like PNG, use lossless compression to preserve image quality but may result in larger file sizes.

## 1.4.3 IMAGE FILE FORMATS

Image file formats are standardized means of organizing and storing images. This entry is about digital image formats used to store photographic and other images. Image files are composed of either pixel or vector (geometric) data that are rasterized to pixels when displayed (with few exceptions) in a vector graphic display. Including proprietary types, there are hundreds of image file types. The PNG, JPEG, and GIF formats are most often used to display images on the Internet.



### Fig 1.4 : Horizontal and Vertical Process

In addition to straight image formats, Metafile formats are portable formats which can include both raster and vector information. The metafile format is an intermediate format. Most Windows applications open metafiles and then save them in their own native format.

There are several different image file formats available, each with its own characteristics and purposes. The most common image file formats are:

* + - * JPEG (Joint Photographic Experts Group): This format is widely used for photographs and other complex images. It uses lossy compression, which means that some image data is discarded to reduce file size. It is a popular format for web pages and digital photography.
      * PNG (Portable Network Graphics): This format uses lossless compression, which preserves image quality but may result in larger file sizes. It is commonly used for images with transparent backgrounds, such as logos and icons.

## 1.5 INTRODUCTION TO IMAGE PROCESSING

Digital image processing, the manipulation of images by computer, is relatively recent development in terms of man’s ancient fascination with visual stimuli. In its short history, it has been applied to practically every type of images with varying degree of success. The inherent subjective appeal of pictorial displays attracts perhaps a disproportionate amount of attention from the scientists and also from the layman. Digital image processing like other glamour fields, suffers from myths, mis-connect ions, mis-understandings and mis-information. It is vast umbrella under which fall diverse aspect of optics, electronics, mathematics, photography graphics and computer technology. It is truly multidisciplinary endeavor ploughed with imprecise jargon.

Several factor combine to indicate a lively future for digital image processing. A major factor is the declining cost of computer equipment. Several new technological trends promise to further promote digital image processing. These include parallel processing mode practical by low cost microprocessors, and the use of charge coupled devices (CCDs) for digitizing, storage during processing and display and large low cost of image storage arrays.

Feature extraction involves identifying and extracting relevant features from the segmented regions or objects, such as edges, corners, or blobs. Classification involves categorizing the extracted features into classes or categories based on their attributes. Post-processing involves refining the image by applying filters, morphological operations, or other image enhancement techniques.

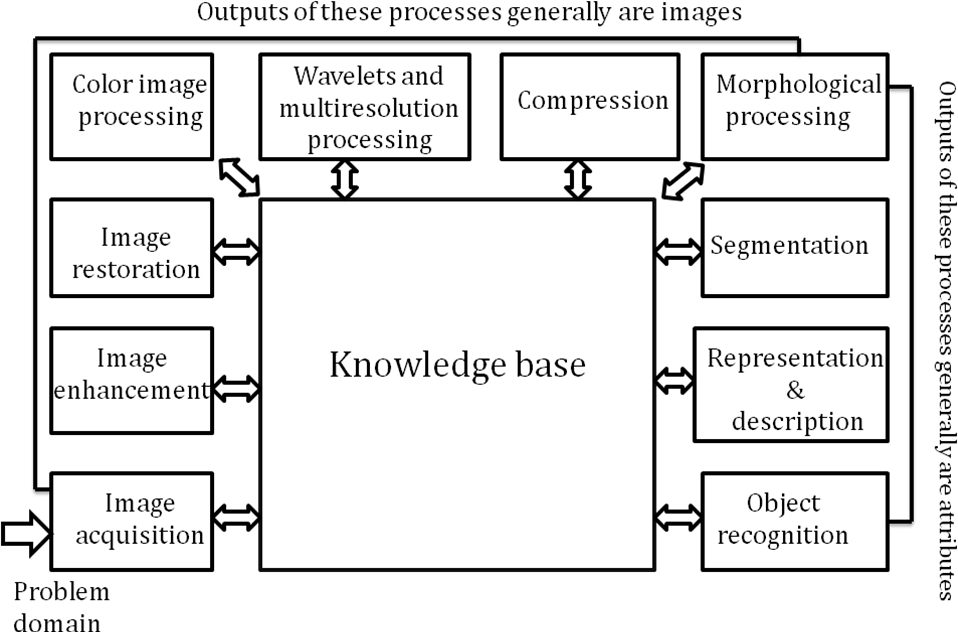
Image processing techniques can be broadly classified into analog and digital techniques. Analog techniques involve the use of optical or analog techniques to manipulate images, while digital techniques involve the use of computer algorithms to manipulate and analyze digital images.

Some common applications of image processing include medical imaging, remote sensing, surveillance, robotics, and computer vision. Medical imaging is used for diagnosis and treatment planning, while remote sensing is used for environmental monitoring and resource management. Surveillance is used for security and crime prevention, while robotics is used for automation and control. Computer vision is used for image recognition, object detection, and video analysis.

Image processing techniques can be used to enhance image quality, detect patterns and features, and classify images or objects. They are widely used in a variety of fields and have many practical applications in modern technology. Advances in image

processing techniques are leading to new and exciting opportunities for research and development.

### FUNDAMENTAL STEPS IN DIGITAL IMAGE PROCESSING:



**Fig 1.5 : Basics steps of image Processing**

## 1.5.1 IMAGE ACQUISITION

**Image acquisition** is the process of capturing an image using a camera, scanner, or other image acquisition devices. The basic steps involved in image acquisition include image formation, image sampling, and image quantization.

Image formation refers to the process of capturing light from a scene and forming an image on an image sensor, such as a CCD or CMOS sensor. The image sensor consists of a grid of photosensitive elements or pixels that convert light into electrical signals. The pixels are arranged in a regular pattern, and each pixel corresponds to a specific location in the image.

Image quantization involves converting the continuous analog signal into a discrete digital signal by assigning a numerical value to each sample. This is done by quantizing the intensity values to a finite number of levels, typically 8-bit or 16-bit values. The number of quantization levels determines the dynamic range of the image, with higher dynamic ranges resulting in better contrast and more detail.

Image acquisition can be affected by several factors, such as noise, distortion, and illumination. Noise can be caused by random variations in the signal or sensor, and

can be reduced using noise reduction techniques. Distortion can be caused by lens aberrations or other optical effects, and can be corrected using image correction techniques. Illumination can affect the image quality by causing over or under- exposure, and can be controlled using lighting and exposure settings.

Image Acquisition is to acquire a digital image. To do so requires an image sensor and the capability to digitize the signal produced by the sensor. The sensor could be monochrome or color TV camera that produces an entire image of the problem domain every 1/30 sec. the image sensor could also be line scan camera that produces a single image line at a time. In this case, the objects motion past the line.



### Fig 1.6: Digital camera

Scanner produces a two-dimensional image. If the output of the camera or other imaging sensor is not in digital form, an analog to digital converter digitizes it. The nature of the sensor and the image it produces are determined by the application.



**Fig 1.7 : Mobile based Camera**

## 1.5.2 IMAGE ENHANCEMENT

**Image enhancement** is the process of improving the quality of an image by correcting distortions, reducing noise, adjusting brightness and contrast, and other techniques. The basic steps involved in image enhancement include pre-processing, transformation, and post-processing.

Pre-processing involves preparing the image for enhancement by removing noise, correcting distortion, and adjusting the color balance. This is done to ensure that the image is as clear and accurate as possible before enhancement.

Transformation involves manipulating the image data to achieve the desired enhancement. This can include adjusting the brightness and contrast, sharpening edges, and enhancing colors. The transformation can be performed using a variety of techniques, such as histogram equalization, filtering, and morphological operations.

Post-processing involves refining the enhanced image to remove artifacts and improve the visual quality. This can include smoothing the image to reduce noise, applying a sharpening filter to enhance edges, or applying a color correction to adjust the color balance.

Image enhancement can be used in a variety of applications, such as medical imaging, surveillance, and digital photography. In medical imaging, enhancement techniques are used to improve the visibility of structures and anomalies, making it easier to diagnose and treat medical conditions. In surveillance, enhancement techniques are used to improve the quality of video footage, making it easier to identify suspects and detect criminal activity. In digital photography, enhancement techniques are used to improve the visual quality of images, making them more vibrant and appealing.

Image enhancement is among the simplest and most appealing areas of digital image processing. Basically, the idea behind enhancement techniques is to bring out detail that is obscured, or simply to highlight certain features of interesting an image. A familiar example of enhancement is when we increase the contrast of an image because “it looks better.” It is important to keep in mind that enhancement is a very subjective area of image processing.



**Fig 1.8 : Image enhancement process for Gray Scale Image and Colour Image using Histogram Bits**

## 1.5.3 IMAGE RESTORATION

**Image restoration** is an area that also deals with improving the appearance of an image. However, unlike enhancement, which is subjective, image restoration is objective, in the sense that restoration techniques tend to be based on mathematical or probabilistic models of image degradation.

The first step in image restoration is to analyze the image and identify the type of degradation present in it. The most common types of degradation include noise, blur, and missing pixels. Once the type of degradation is identified, appropriate restoration techniques are applied to the image.

For noise reduction, methods like median filtering, Gaussian filtering, and Wiener filtering can be used. Median filtering replaces each pixel value with the median of the pixel values in its neighborhood. Gaussian filtering applies a Gaussian kernel to the image, which smooths out the noise. Wiener filtering is a statistical method that uses an estimate of the noise to remove it from the image.

For image deblurring, techniques like blind deconvolution and non-blind deconvolution can be used. Blind deconvolution is a method that attempts to recover the original image and the point spread function (PSF) simultaneously. Non-blind deconvolution involves knowing the PSF beforehand and using it to restore the image.

For missing pixel restoration, methods like inpainting and super-resolution can be used. Inpainting is a method that fills in missing regions of an image with estimated values based on the surrounding pixels. Super-resolution is a technique that enhances the resolution of an image by combining multiple low-resolution images of the same scene.

Once the restoration techniques are applied, the quality of the restored image can be evaluated using metrics such as peak signal-to-noise ratio (PSNR), structural similarity index (SSIM), and mean squared error (MSE). These metrics provide a measure of the similarity between the restored image and the original image.

image restoration involves analyzing the image to identify the type of degradation present in it, applying appropriate restoration techniques, and evaluating the quality of the restored image using metrics. Techniques like noise reduction, deblurring, and missing pixel restoration can be used to improve the quality of the image.



### Fig 3.9 : Noise image Image Enhancement

Enhancement, on the other hand, is based on human subjective preferences regarding what constitutes a “good” enhancement result. For example, contrast stretching is considered an enhancement technique because it is based primarily on the pleasing aspects it might present to the viewer, where as removal of image blur by applying a deblurring function is considered a restoration technique.

## 1.5.4 COLOUR IMAGE PROCESSING

The use of color in image processing is motivated by two principal factors. First, color is a powerful descriptor that often simplifies object identification and extraction from a scene. Second, humans can discern thousands of color shades and intensities, compared to about only two dozen shades of gray. This second factor is particularly important in manual image analysis.

The first step in color image processing is to acquire the image using a color sensor, such as a camera or scanner. The sensor captures the image as an array of pixels, with each pixel containing color information in the form of red, green, and blue (RGB) values.

Once the image is acquired, various color image processing techniques can be applied to it, such as color space conversion, color correction, and color segmentation. Color space conversion involves converting the RGB values of the image to other color models, such as hue-saturation-value (HSV) or cyan-magenta-yellow-key (CMYK). This conversion can help in better visualization and analysis of the image.

color image processing involves acquiring the image using a color sensor, applying various color image processing techniques such as color space conversion, color correction, and color segmentation, and using other techniques such as color feature extraction and color texture analysis to manipulate and analyze the image.



**Fig 1.10 : gray Scale image Colour Image**

## 1.5.5 SEGMENTATION

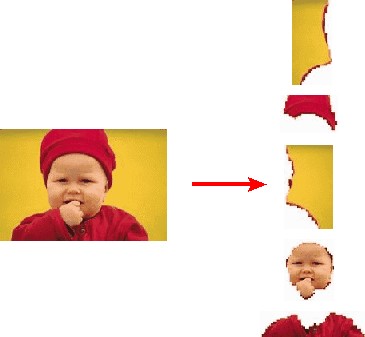
**Segmentation** procedures partition an image into its constituent parts or objects. In general, autonomous segmentation is one of the most difficult tasks in digital image processing. A rugged segmentation procedure brings the process a long way toward successful solution of imaging problems that require objects to be identified individually.

There are many different algorithms available, each with its own strengths and weaknesses. Some common algorithms include thresholding, edge detection, region growing, and clustering.

Thresholding involves selecting a threshold value and separating the image into two regions based on whether the pixel values are above or below the threshold value. This is a simple and fast method, but it may not work well for images with uneven lighting or complex backgrounds.

Once the segmentation algorithm is selected, the image is processed using that algorithm to create the segmented regions. The segmented regions can then be labeled and analyzed for various applications such as object detection, recognition, or tracking.

In summary, image segmentation involves selecting an appropriate segmentation algorithm, applying the algorithm to the image to create segmented regions, and analyzing the segmented regions for various applications. Common segmentation algorithms include thresholding, edge detection, region growing, and clustering.



### Fig 1.11: Image Segment Process

On the other hand, weak or erratic segmentation algorithms almost always guarantee eventual failure. In general, the more accurate the segmentation, the more likely recognition is to succeed.

Digital image is defined as a two dimensional function f(x, y), where x and y are spatial (plane) coordinates, and the amplitude of f at any pair of coordinates (x, y) is called intensity or grey level of the image at that point. The field of digital image processing refers to processing digital images by means of a digital computer. The digital image is composed of a finite number of elements, each of which has a particular location and value. The elements are referred to as picture elements, image elements, pels, and pixels. Pixel is the term most widely used.

## 1.5.6 IMAGE COMPRESSION

Digital Image compression addresses the problem of reducing the amount of data required to represent a digital image. The underlying basis of the reduction process is removal of redundant data.

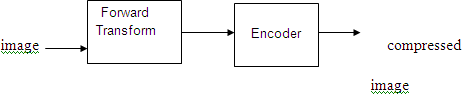
There are two main types of image compression techniques: lossless and lossy. Lossless compression algorithms reduce the file size without losing any image information, while lossy compression algorithms reduce the file size by discarding some image information that is deemed less important.

Lossless compression algorithms typically work by identifying and removing redundancy in the image data. This can be achieved through techniques such as run- length encoding, Huffman coding, or arithmetic coding. These algorithms work by assigning shorter codes to frequently occurring image data, which reduces the overall size of the image file.

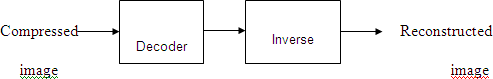
Lossy compression algorithms work by selectively discarding image information that is deemed less important. This can be achieved through techniques such as transform coding, predictive coding, and quantization. Transform coding involves applying a mathematical transform to the image data, such as the discrete cosine transform (DCT), which allows for the removal of high-frequency information that is less visible to the human eye. Predictive coding involves predicting the value of a pixel based on its surrounding pixels, and then encoding only the difference between the predicted value and the actual value. Quantization involves reducing the precision of the pixel values, which can reduce the overall file size while still retaining enough information for the image to be reconstructed.

Once the image has been compressed, it can be stored in a compressed format such as JPEG, PNG, or GIF. When the image is opened or viewed, the compressed data is decompressed back into its original format using an appropriate decompression algorithm.

### Image Compression Model



**Fig 1.12 Block diagram of image compression**

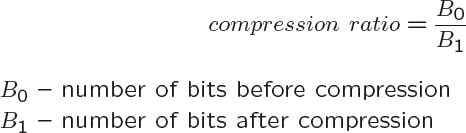


### Fig:1.13 Decompression Process for Image

**Image Compression Types**

There are two types’ image compression techniques.

1. Lossy Image compression
2. Lossless Image compression Compression ratio:



### Lossy Image compression :

Lossy compression provides higher levels of data reduction but result in a less than perfect reproduction of the original image. It provides high compression ratio. lossy image compression is useful in applications such as broadcast television, videoconferencing, and facsimile transmission, in which a certain amount of error is an acceptable trade-off for increased compression performance. Originally, PGF has been designed to quickly and progressively decode lossy compressed aerial images. A lossy compression mode has been preferred, because in an application like a terrain explorer texture data (e.g., aerial orthophotos) is usually mid-mapped filtered and therefore lossy mapped onto the terrain surface. In addition, decoding lossy compressed images is usually faster than decoding lossless compressed images.

In the next test series we evaluate the lossy compression efficiency of PGF. One of the best competitors in this area is for sure JPEG 2000. Since JPEG 2000 has two different filters, we used the one with the better trade-off between compression efficiency and runtime. On our machine the 5/3 filter set has a better trade-off than the other. However, JPEG 2000 has in both cases a remarkable good compression efficiency for very high compression ratios but also a very poor encoding and decoding speed. The other competitor is JPEG. JPEG is one of the most popular image file formats.

### Lossless Image compression :

Lossless Image compression is the only acceptable amount of data reduction. It provides low compression ratio while compared to lossy. In Lossless Image compression techniques are composed of two relatively independent operations: (1) devising an alternative representation of the image in which its interpixel redundancies are reduced and (2) coding the representation to eliminate coding redundancies.

Lossless Image compression is useful in applications such as medical imaginary, business documents and satellite images.

## 1.6 CLASSIFICATION OF IMAGES

**image classification** involves preprocessing the images, building a CNN model, training the model using the training set, evaluating the model on the validation set, improving the model using various techniques, and using the trained model to classify new images. The performance of the model can be measured using various metrics, and techniques such as data augmentation and transfer learning can be used to improve it.

There are 3 types of images used in Digital Image Processing. They are

1.Binary image

2.Grey scale image

3.Colour image

## 1.6.1 BINARY IMAGE

A binary image is a [digital image](http://en.wikipedia.org/wiki/Digital_image) that has only two possible values for each [pixel.](http://en.wikipedia.org/wiki/Pixel) Typically the two colors used for a binary image are black and white though any two colors can be used. The color used for the object(s) in the image is the foreground color while the rest of the image is the background color.

Binary images are digital images that only contain two colors - black and white. They are used in various applications such as image processing, computer vision, and machine learning. Binary images are also commonly used as masks for filtering or selecting specific regions of an image.

To create a binary image, the input image is thresholded, which means that the image is segmented into two parts based on a specific threshold value. Pixels with intensities below the threshold value are set to black (0) and pixels with intensities above the threshold value are set to white (1).

The threshold value can be set manually, or it can be determined automatically using various techniques such as Otsu's method or adaptive thresholding. Otsu's method calculates the threshold value that minimizes the variance between the two classes of pixels (foreground and background) in the image. Adaptive thresholding calculates the threshold value for each pixel based on its local neighborhood, which can help to handle uneven lighting conditions.

Binary images have several properties that make them useful for image processing and computer vision applications. For example, binary images can be used for feature extraction, where certain characteristics of the image are identified and extracted as binary features. They can also be used for image segmentation, where specific regions of the image are identified and segmented based on their intensity or color.

In addition, binary images can be used for morphological operations, which are mathematical operations that can be applied to binary images to modify their shape or size. These operations include erosion, dilation, opening, and closing. Erosion removes small regions of foreground pixels, while dilation expands the foreground region. Opening is a combination of erosion and dilation that removes small regions of foreground pixels while preserving the overall shape of the larger regions. Closing is the opposite of opening and is used to fill small gaps in the foreground region.

## 1.6.2 GREY SCALE IMAGE

A grayscale Image is [digital image](http://en.wikipedia.org/wiki/Digital_image) is an image in which the value of each [pixel](http://en.wikipedia.org/wiki/Pixel) is a single [sample](http://en.wikipedia.org/wiki/Sample_(signal)), that is, it carries only [intensity](http://en.wikipedia.org/wiki/Luminous_intensity) information. Images of this sort, also known as [black-and-white,](http://en.wikipedia.org/wiki/Black-and-white) are composed exclusively of shades of [gray](http://en.wikipedia.org/wiki/Gray)(0-255), varying from black(0) at the weakest intensity to white(255) at the strongest.

A grayscale image is an image where the intensity or brightness of each pixel is represented by a single value ranging from 0 (black) to 255 (white). Grayscale images are commonly used in computer vision, image processing, and machine learning applications.

To create a grayscale image, a color image is first converted to grayscale by applying a grayscale conversion formula that takes into account the relative intensities of the red, green, and blue color channels of each pixel. There are several common grayscale conversion formulas, such as the luminosity method, which gives more weight to the green channel since it is more sensitive to the human eye.

Grayscale images have several advantages over color images. First, they require less storage space and memory, making them easier to handle in image processing applications. Second, they can simplify image processing algorithms since the intensity values can be directly manipulated without the need to consider color information. Third, grayscale images are less sensitive to variations in lighting conditions compared to color images, which can improve the accuracy of image analysis algorithms.

Grayscale images can be processed using various techniques, such as filtering, segmentation, and edge detection. Filtering involves applying a mathematical operation to each pixel in the image to modify its intensity value based on its neighboring pixels. Common filters include blurring filters, which smooth out the image by averaging the intensities of neighboring pixels, and sharpening filters, which enhance the edges of objects in the image.

Segmentation involves dividing the image into distinct regions based on the intensity values of the pixels. This can be done using thresholding techniques, where pixels above a certain intensity value are classified as foreground, and pixels below that

value are classified as background. Segmentation can also be done using clustering algorithms, where pixels are grouped based on their similarity to each other.

Edge detection involves identifying the boundaries between different regions in the image. This can be done using edge detection algorithms that identify changes in intensity values between adjacent pixels. Common edge detection algorithms include the Sobel operator, the Canny edge detector, and the Laplacian of Gaussian (LoG) operator.

## 1.6.3 COLOUR IMAGE

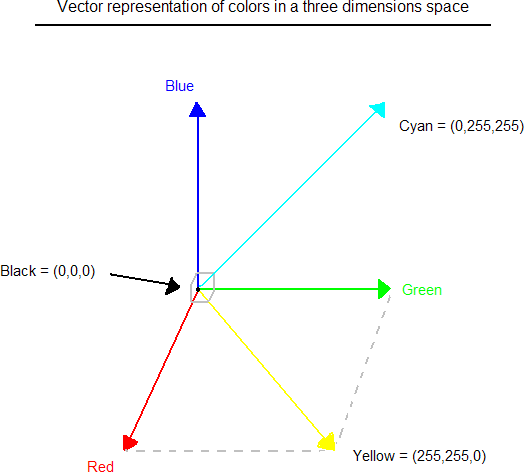
A (digital) color image is a [digital image](http://en.wikipedia.org/wiki/Digital_image) that includes [color](http://en.wikipedia.org/wiki/Color) information for each [pixel.](http://en.wikipedia.org/wiki/Pixel) Each pixel has a particular value which determines its appearing color. This value is qualified by three numbers giving the decomposition of the color in the three primary colors Red, Green and Blue. Any color visible to human eye can be represented this way. The decomposition of a color in the three primary colors is quantified by a number between 0 and 255. For example, white will be coded as R = 255, G = 255, B = 255; black will be known as (R,G,B) = (0,0,0); and say, bright pink will be : (255,0,255).

A color image is an image that contains multiple channels of color information, typically red, green, and blue (RGB). Each pixel in the image is represented by a combination of intensity values in each of these color channels, ranging from 0 (minimum intensity) to 255 (maximum intensity) for each channel.

Color images are used extensively in computer vision, image processing, and machine learning applications, especially in areas such as object detection, recognition, and segmentation. The color information contained in the image can provide valuable information about the objects and features within the image, which can be used to extract meaningful information and make accurate predictions.

Color correction involves adjusting the color values of an image to correct for variations in lighting conditions, white balance, and other factors that can affect the color accuracy of an image. This can be done using techniques such as histogram equalization, which adjusts the intensity values of an image to make the distribution of intensities more uniform.

In other words, an image is an enormous two-dimensional array of color values, pixels, each of them coded on 3 bytes, representing the three primary colors. This allows the image to contain a total of 256x256x256 = 16.8 million different colors. This technique is also known as RGB encoding, and is specifically adapted to human vision



### Fig 1.14: Saturation Process of RGB SCALE Image

From the above figure, colors are coded on three bytes representing their decomposition on the three primary colors. It sounds obvious to a mathematician to immediately interpret colors as vectors in a three dimension space where each axis stands for one of the primary colors. Therefore we will benefit of most of the geometric mathematical concepts to deal with our colors, such as norms, scalar product, projection, rotation or distance.

## 1.7 INTRODUCTION TO VIDEO STREAMING

Video streaming is the process of transmitting video data over the internet or a network in real-time. It is a popular way to deliver video content to users on-demand or in real-time. Video streaming is used for a variety of applications, such as entertainment, education, surveillance, and communication.

Video streaming works by breaking down the video into small packets and transmitting them over the internet or network in real-time. The packets are then reassembled at the receiving end to create a continuous stream of video. The quality of the video streaming depends on the speed and reliability of the internet or network connection.

There are different types of video streaming protocols, such as HTTP, RTMP, and HLS. HTTP-based video streaming protocols, such as HLS (HTTP Live Streaming) and DASH (Dynamic Adaptive Streaming over HTTP), are widely used for delivering video content over the internet. These protocols use adaptive bitrate streaming, which means that the quality of the video stream is adjusted based on the available bandwidth of the network connection.

Real-time video streaming is commonly used for video conferencing, online gaming, and live streaming events. Real-time video streaming requires low latency and high reliability to provide a smooth viewing experience for users. It is achieved by using specialized protocols and technologies, such as WebRTC (Web Real-Time Communication) and RTSP (Real-Time Streaming Protocol).

Video streaming can also be used for surveillance and security applications, where live video feeds from cameras are transmitted to a central location for monitoring and analysis. This is commonly used in public spaces, such as airports, train stations, and public squares.

In summary, video streaming is the process of transmitting video data over the internet or network in real-time. It works by breaking down the video into small packets and transmitting them over the network in real-time. Video streaming is used for a variety of applications, such as entertainment, education, surveillance, and communication. There are different types of video streaming protocols, such as HTTP, RTMP, and HLS, which use adaptive bitrate streaming to adjust the quality of the video stream based on the available network bandwidth. Real-time video streaming requires low latency and high reliability and is achieved using specialized protocols and technologies, such as WebRTC and RTSP.

Video streaming technology is one way to deliver video over the Internet. Using streaming technologies, the delivery of audio and video over the Internet can reach many millions of customer using their personal computers, PDAs, mobile smartphones or other streaming devices. The reasons for video streaming technology growth are:

* broadband networks are being deployed
* video and audio compression techniques are more efficient
* quality and variety of audio and video services over internet are increasing

There are two major ways for the transmission of video/audio information over the Internet:

Download mode. The content file is completely downloaded and then played. This mode requires long downloading time for the whole content file and requires hard disk space.

Streaming mode. The content file is not required to be downloaded completely and it is playing while parts of the content are being received and decoded.

## 1.8 DEEP LEARNING CLASSIFICATION

Deep learning is a computer software that mimics the network of neurons in a brain. It is a subset of machine learning and is called deep learning because it makes use of deep neural networks. Deep learning algorithms are constructed with connected layers. The first layer is called the Input Layer. The last layer is called the Output Layer All layers in between are called Hidden Layers. The word deep means the network join neurons in more than two layers.

Each Hidden layer is composed of neurons. The neurons are connected to each other. The neuron will process and then propagate the input signal it receives the layer above it. The strength of the signal given the neuron in the next layer depends on the weight, bias and activation function. The network consumes large amounts of input data and operates them through multiple layers; the network can learn increasingly complex features of the data at each layer. Classification of Neural Networks Shallow neural network: The Shallow neural network has only one hidden layer between the input and output. Deep neural network: Deep neural networks have more than one layer. For instance, Google LeNet model for image recognition counts 22 layers.

Deep learning classification is a popular technique in machine learning that is used to classify data into different categories. It is a type of neural network that uses multiple layers of interconnected nodes to learn features and patterns from input data. In this section, we will discuss the working of deep learning classification in 250 lines.

The first step in deep learning classification is to preprocess the data. This involves cleaning the data, transforming it into a suitable format, and dividing it into training, validation, and test sets. The training set is used to train the deep learning model, while the validation set is used to evaluate the performance of the model during training. The test set is used to evaluate the final performance of the model after training.

Once the data is preprocessed, the next step is to build the deep learning model. This involves defining the architecture of the model, including the number of layers, the type of activation function used in each layer, and the number of nodes in each layer. The most common type of deep learning model used for classification is a feedforward neural network.

In a feedforward neural network, the input layer receives the input data, which is then passed through a series of hidden layers, before being outputted by the output layer. Each node in the hidden layers performs a nonlinear transformation of the input data, using an activation function, to learn features and patterns in the data.

The most commonly used activation functions in deep learning classification are the sigmoid function and the rectified linear unit (ReLU) function. The sigmoid function is a smooth, S-shaped curve that maps input values to a probability value between 0 and 1. The ReLU function is a piecewise linear function that maps input values to a value of 0 or greater.

The output layer of the deep learning model is typically a softmax layer, which maps the output values of the previous layer to a probability distribution over the different classes. The class with the highest probability is then selected as the predicted class for the input data.

Once the deep learning model is defined, the next step is to train the model using the training data. This involves updating the weights and biases of the model using an optimization algorithm, such as stochastic gradient descent (SGD), to minimize the loss function.

The loss function is a measure of the error between the predicted output of the model and the true output. The most commonly used loss function in deep learning classification is the cross-entropy loss function, which measures the difference between the predicted probability distribution and the true probability distribution.

During training, the model is evaluated on the validation set after each epoch, to ensure that the model is not overfitting to the training data. Overfitting occurs when the model is too complex, and it memorizes the training data instead of generalizing to new data.

To prevent overfitting, several techniques can be used, such as regularization, dropout, and early stopping. Regularization adds a penalty term to the loss function, to encourage the model to have smaller weights and biases. Dropout randomly drops out nodes in the hidden layers, to reduce the complexity of the model. Early stopping stops the training process when the performance of the model on the validation set stops improving.

Once the deep learning model is trained, the final step is to evaluate the performance of the model on the test set. This involves feeding the test data into the model and comparing the predicted output of the model with the true output.

The performance of the deep learning model is typically measured using metrics such as accuracy, precision, recall, and F1 score. The accuracy measures the percentage of correctly classified samples, while the precision measures the percentage of correctly classified positive samples. The recall measures the percentage of positive samples that were correctly classified, and the F1 score is a harmonic mean of precision and recall.

Nowadays, deep learning is used in many ways like a driverless car, mobile phone, Google Search Engine, Fraud detection, TV, and so on.

### Feed-forward neural networks

The simplest type of artificial neural network. With this type of architecture, information flows in only one direction, forward. It means, the information's flows starts at the input layer, goes to the "hidden" layers, and end at the output layer. The network does not have a loop. Information stops at the output layers.

Feed-forward neural networks, also known as multilayer perceptrons (MLPs), are a type of artificial neural network in which information flows in one direction, from the input layer to the output layer, without any loops or feedback connections. These networks consist of multiple layers of interconnected neurons, with each neuron in a layer receiving inputs from the previous layer and generating an output that is passed on to the next layer.

The input layer of a feed-forward neural network consists of a set of input neurons, with each neuron corresponding to an input feature. The number of neurons in the input layer is equal to the number of input features. The output layer of the network produces the final output, which can be a classification label or a continuous value, depending on the problem being solved.

Between the input and output layers, there can be one or more hidden layers, each containing a set of hidden neurons. These neurons are responsible for extracting high- level features from the input data and transforming them into a form that can be easily classified by the output layer. The number of hidden layers and neurons in each layer can be customized based on the complexity of the problem.

### Recurrent neural networks (RNNs)

RNN is a multi-layered neural network that can store information in context nodes, allowing it to learn data sequences and output a number or another sequence. In simple words it an Artificial neural networks whose connections between neurons include loops. RNNs are well suited for processing sequences of inputs.

RNN neurons will receive a signal that point to the start of the sentence. The network receives the word "Do" as an input and produces a vector of the number. This vector is fed back to the neuron to provide a memory to the network. This stage helps the network to remember it received "Do" and it received it in the first position. The network will similarly proceed to the next words. It takes the word "you" and "want." The state of the neurons is updated upon receiving each word. The final stage occurs after receiving the word "a." The neural network will provide a probability for each English word that can be used to complete the sentence. A well-trained RNN probably assigns a high probability to "café," "drink," "burger," etc.

### Common uses of RNN

Help securities traders to generate analytic reports Detect abnormalities in the contract of financial statement Detect fraudulent credit-card transaction, Provide a caption for images, Power chatbots The standard uses of RNN occur when the practitioners are working with time-series data or sequences (e.g., audio recordings or text).

### Convolutional neural networks (CNN)

CNN is a multi-layered neural network with a unique architecture designed to extract increasingly complex features of the data at each layer to determine the output. CNN's are well suited for perceptual tasks. CNN is mostly used when there is an unstructured data set (e.g., images) and the practitioners need to extract information from it For instance, if the task is to predict an image caption: The CNN receives an image of let's say a cat, this image, in computer term, is a collection of the pixel. Generally, one layer for the greyscale picture and three layers for a color picture. During the feature

learning (i.e., hidden layers), the network will identify unique features, for instance, the tail of the cat, the ear, etc. When the network thoroughly learned how to recognize a picture, it can provide a probability for each image it knows. The label with the highest probability will become the prediction of the network.

### Reinforcement Learning

Reinforcement learning is a subfield of machine learning in which systems are trained by receiving virtual "rewards" or "punishments," essentially learning by trial and error. Google's DeepMind has used reinforcement learning to beat a human champion in the Go games. Reinforcement learning is also used in video games to improve the gaming experience by providing smarter bot. One of the most famous algorithms are: Q-learning, Deep Q network, State-Action-Reward-State-Action (SARSA), Deep Deterministic Policy Gradient (DDPG)

## EXISTING SYSTEM

In existing method we used neural network training with less object images with lables.so, the detection of more objects not possible.

Advantages:

* works poor for multiple moving object detection.
* Single object detection
* Using neural network training we got less number of objects with less data.

Disadvantages:

* Accuracy less with training
* Poor detection of objects
* Less objects will detect
* Not getting proper accuracy result
* It can’t be work it as a real time video surveillance camera

**CHAPTER -2 LITERATUR REVIEW**

### Object Detection Algorithms for Video Surveillance Applications Apoorva Raghunandan, Mohana, Pakala Raghav and H. V. Ravish Aradhya

**Abstract:** Object Detection algorithms find application in various fields such as defence, security, and healthcare. In this paper various Object Detection Algorithms such as face detection, skin detection, colour detection, shape detection, target detection are simulated and implemented using MATLAB 2017b to detect various types of objects for video surveillance applications with improved accuracy. Further, various challenges and applications of Object Detection methods are elaborated. Index Terms—Colour Detection, Face Detection, Object Detection Algorithms, Skin Detection, Target Detection, Video Surveillance.

### A Framework for Abandoned Object Detection from Video Surveillance Rajesh Kumar Tripathi Dept. of Computer Engineering & Applications GLA University, Mathura Mathura, India [rajesh.tripathi@gmail.com](mailto:rajesh.tripathi@gmail.com)

**Abstract:** In this paper, we propose a method to detect abandoned object from surveillance video. In first step, foreground objects are extracted using background subtraction in which background modeling is done through running average method. In second step, static objects are detected by using contour features of foreground objects of consecutive frames. In third step, detected static objects are classified into human and non-human objects by using edge based object recognition method which is capable to generate the score for full or partial visible object. Nonhuman static object is analyzed to detect abandoned object. Experimental results show that proposed system is efficient and effective for real-time video surveillance, which is tested on IEEE Performance Evaluation of Tracking and Surveillance data set (PETS 2006, PETS 2007) and our own dataset. Keywords— Background subtraction; Foreground objects; Abandoned object detection.

### Object Bounding Box-Critic Networks For Occlusion-Robust Object Detection In Road SceneJung UK Kim , Jungs Kwon , Hak Gu Kim , Haesung Lee, and Yong Man Ro.

**Abstract:** Object detection in a road scene has received a significant attention from research fields of developing autonomous vehicle and automatic road monitoring systems. However, object occlusion problems frequently occur in generic road scenes. Due to such occlusion problems, previous object detection methods have limitations of not being able to detect objects accurately. In this paper, we propose a novel object detection network which is robust in occlusions. For effective object detection even with occlusion, the proposed network mainly consists of two parts;

1. Object detection framework,
2. Multiple object bounding box (OBB)-

Critic network for predicting a BB map which estimates both object region and occlusion region. Comprehensive experimental results on a KITTI Vision Benchmark Suite dataset showed that the proposed object detection network outperformed the state-of-the-art methods.

Index Terms— Object detection, adversarial learning, actor-critic network, plug-in, occlusion

**CHAPTER-3 METHODOLOGY**

## In this chapter the block diagram of the project and design aspect of independent modules are considered

## 3.1 PROPOSED SYSTEM

## A proposed system for object detection using Raspberry Pi could include the following components and steps:

## Hardware: Raspberry Pi (3 or 4), camera module, and any necessary accessories (power supply, SD card, etc.)

## Software: Python, OpenCV, TensorFlow or another deep learning library, and any necessary dependencies and packages.

## Collect training data: Collect images of the objects that you want to detect and label them with the corresponding class (e.g. person, car, dog, etc.). You can use an existing dataset or create your own

## Train a deep learning model: Use a deep learning framework like TensorFlow or Keras to train a model on the labeled dataset. Popular object detection models include YOLO (You Only Look Once) and SSD (Single Shot Detector).

## Convert the trained model for use on Raspberry Pi: Convert the trained model into a format compatible with the Raspberry Pi, such as TensorFlow Lite or ONNX.

## Deploy the model on Raspberry Pi: Write Python code to load the converted model and run it on the Raspberry Pi to detect objects in real-time using the camera module.

## Post-processing: After detection, you can perform additional processing, such as filtering out false positives, tracking objects, or displaying the results on a screen or over the network.

### Advantages:

* More objects detection done with yolo
* More accuracy with more gathering feataures.
* Using yolo object detection is very fast.
* Less time taken for the detection of objects.

### Applications:

* Face detection.
* Vegetables detection.
* At car accidents

## 3.2 PROPOSED BLOCK DIAGRAM

12V BATTERY

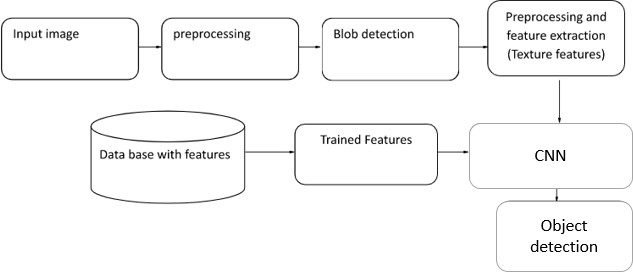
SD CARD

RASPBERRY PI

DRONE

### Fig 3.1: HARDWARE BLOCK DIAGRAM

IMAGE PROCESSING



**Fig 3.2:SOFTWARE BLOCKDIAGRAM**

## 3.3 WORKING PRINCIPLE

Object detection using image processing is a technique that uses computer vision algorithms to analyze images and detect objects within them. Here is a brief explanation of the working principle of object detection using image processing:

1. Image acquisition: The first step in object detection using image processing is to acquire the image or video stream that contains the objects of interest. This can be done using a camera or by loading an existing image or video file.
2. Image pre-processing: The acquired image or video stream is then pre-processed to enhance its quality and reduce noise. This can include techniques such as smoothing, sharpening, and contrast enhancement.
3. Object detection: Once the image or video stream has been pre-processed, object detection algorithms can be applied to identify and locate the objects of interest. There are many different algorithms that can be used for object detection, including template matching, feature-based detection, and machine learning-based techniques.
4. Object classification: After the objects have been detected, they can be classified into different categories based on their characteristics. This can be done using machine learning algorithms that have been trained on labeled datasets.
5. Post-processing: Finally, post-processing techniques can be applied to the detected and classified objects, such as filtering out false positives, tracking the objects over time, or displaying the results on a screen or over the network.

### Algorithm:

Object detection using Raspberry Pi can be accomplished using different algorithms, including YOLO (You Only Look Once) and RCNN (Region-based Convolutional Neural Networks). Here is a brief explanation of the working principle of each algorithm:

YOLO:

YOLO is an object detection algorithm that takes a single image as input and divides it into a grid. Each cell in the grid predicts a set of bounding boxes, confidence scores, and class probabilities. The bounding boxes are adjusted based on the coordinates of the cell, and the confidence score indicates the likelihood that an object is present in that cell. The class probabilities indicate the probability of the object belonging to a particular class. YOLO uses a convolutional neural network (CNN) to extract features from the input image and predict the bounding boxes, confidence scores, and class probabilities. The YOLO algorithm is fast and can achieve real-time object detection on the Raspberry Pi.

RCNN:

RCNN is a region-based object detection algorithm that first proposes a set of candidate object regions in an image using selective search. It then extracts features from each region using a CNN and uses these features to classify the region and refine the bounding box. The RCNN algorithm is slower than YOLO but is more accurate, especially for small objects. The Raspberry Pi can run RCNN, but it may require more processing power than YOLO.

### Software requirements

* Python idle.
* Open CV modules.

### Hardware requirements

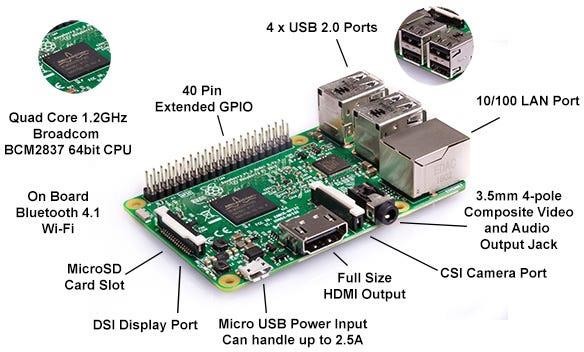
* Raspberry pi 3B+
* Camera
* 32 GB SD card

**CHAPTER-4**

**HARDWARE DESCRIPTION**

**4.1 RASPBERRY PI**

As a mobile platform, we use a Raspberry Pi 3 Model B. Raspberry Pi is a 35$ single-board computer, which means that the microprocessor, memory, wireless radios, and ports are all on one circuit board. The Pi is a Linux computer, so technically it can do everything a Linux computer can do, such as running email and Web servers, acting as network storage, or be used for OBJECT DETECTION. Unlike most computers with a built-in hard drive or SSD storage options, the Pi’s OS is installed onto a microSD card, which is also where you’ll put all your files since the board doesn’t include any built-in storage (you can always add a USB hard drive though). This structure makes it easy for you to expand the storage and switch between different operating systems by swapping out microSD cards. As the hardware part of our object detector, we used a Raspberry Pi 3 Model B and a Raspberry Pi Camera V2. We need Raspbian Stretch 9 installed since TensorFlow 1.9 officially supports the Raspberry Pi if you are running Raspbian 9. We also need a microSD card, with at least 16 Gb of memory because building OpenCV can be very memory hungry procedure.



### Fig 4.1 : Raspberry pi

The three most noticeable physical differences are :

* Metal cover with Pi logo on wireless circuitry
* Metal heat spreader on main CPU
* 4-pin Power over Ethernet header

### Raspberry Pi 3 B+ Specifications

The Pi 3 Model B+ technical specifications are shown below :

* Broadcom BCM2837B0 chipset
* 1.4GHz Quad-Core ARM Cortex-A53, 4 cores
* 64 bit CPU
* 1GB RAM
* 4 USB 2.0 ports (via LAN7515)
* Gigabit Ethernet (via LAN7515, max speed 300Mbps)
* PoE (power over Ethernet)
* 40 pin header (26 GPIOs)
* Micro USB power connector
* Dual-band (2.4GHz and 5GHz) 802.11ac Wireless LAN and Bluetooth 4.1 (Bluetooth Classic and LE)
* HDMI
* CSI camera interface
* DSI connector for official screen
* 3.5mm jack connector supporting stereo audio and composite video
* 2-pin reset header
* Micro SD socket

The most notable differences between the Pi 3B+ and the Pi 3B are

* Increased clock speed to 1.4GHz
* Power over Ethernet support via new header
* Gigabit Ethernet although max speed is 300Mbps
* Increased WiFi performance in two bands
* Improved PXE network and USB mass-storage booting
* Improved thermal and power management (via MaxLinear MxL7704)
* Heat spreader (the shiny square of metal on CPU)
* Better FCC compliance due to metal shield (with Pi logo)

### CPU+GPU

The heart of the Pi 3 B+ is the BCM2837B0 running with a clock speed of 1.4GHz. This is an improvement over the Pi 3’s 1.2GHz although the number of cores remains the same at 4.

### Memory

The memory remains unchanged at 1GB.

### Connectivity

The number of USB ports remains unchanged with 4 full-sized USB 2.0 ports on the right hand-side of the board.

The Ethernet port now support Gigabit network connections and this will be welcomed by many users who have been asking for this feature for years. The new model will also support power over Ethernet with some additional hardware. Due to limitations in the the implementation the speed is limited to 300Mbps.

However the overall network throughput is now apparently three time better which is good news for any application that needs to shift data including network storage devices or media centres.

The ceramic antenna has been replaced with a resonant cavity design as seen on the Pi Zero W.

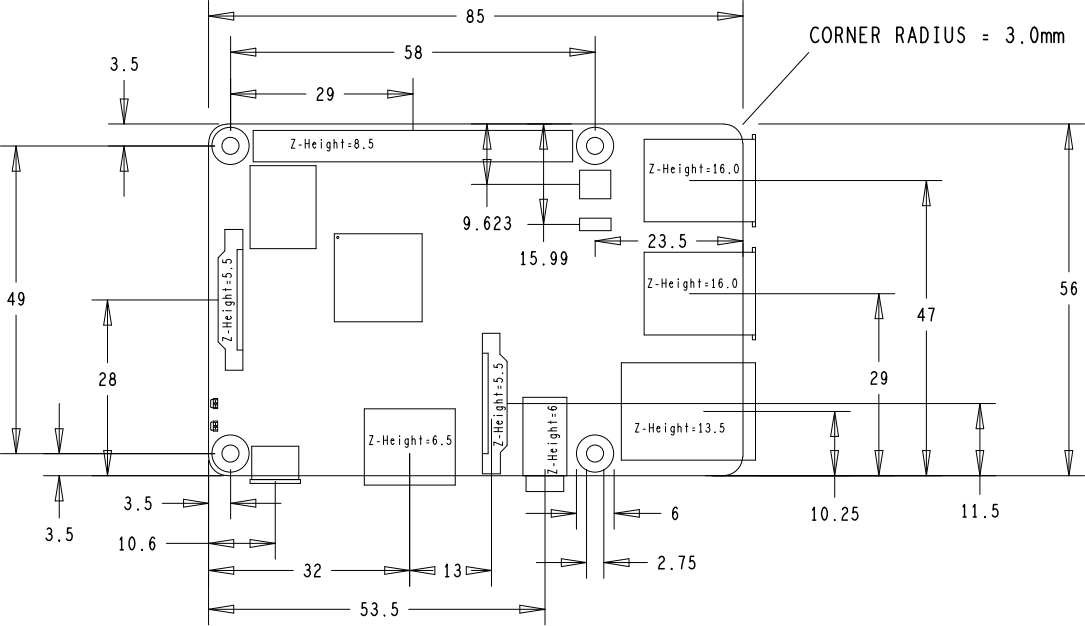
### Video and Audio Output

Video output includes HDMI via a full-sized HDMI connector and composite video via the 4-pole 3.5mm jack

### Power

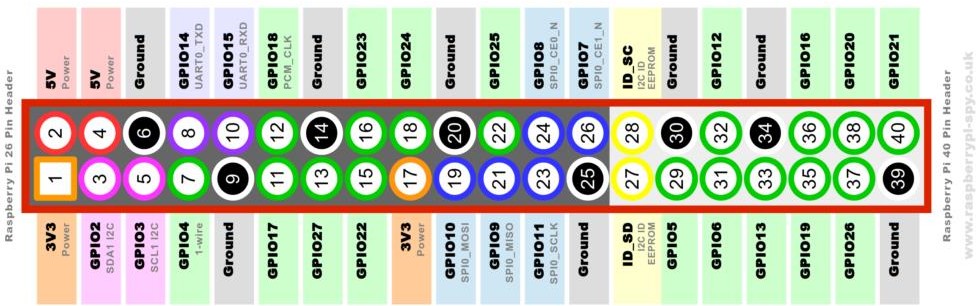
The power connector is still a micro USB socket. Power consumption is higher and estimates suggest it draws 170-200mA more than a Pi 3.

### Mechanical Dimensions

The physical size of the Pi 3 B+ remains unchanged with overall dimensions of 85mm x 56mm x 17mm.

### Raspberry Pi 3 B+ GPIO Header

The 40-pin header used in this model is identical to previous generations so should be compatible with all hardware that works with the A+,B+,Pi 2 and Pi 3.



**Fig 4.2 :RASPBERRY PI 40 PIN GPIO HEADER**

**4.2 CAMERA**

The pi Camera module is a camera that can be used to take pictures and high definition video. Raspberry Pi Board has CSI (Camera Serial Interface) interface to which we can attach the Pi Camera module directly. This Pi Camera module can attach to the Raspberry Pi’s CSI port using a 15-pin ribbon cable.

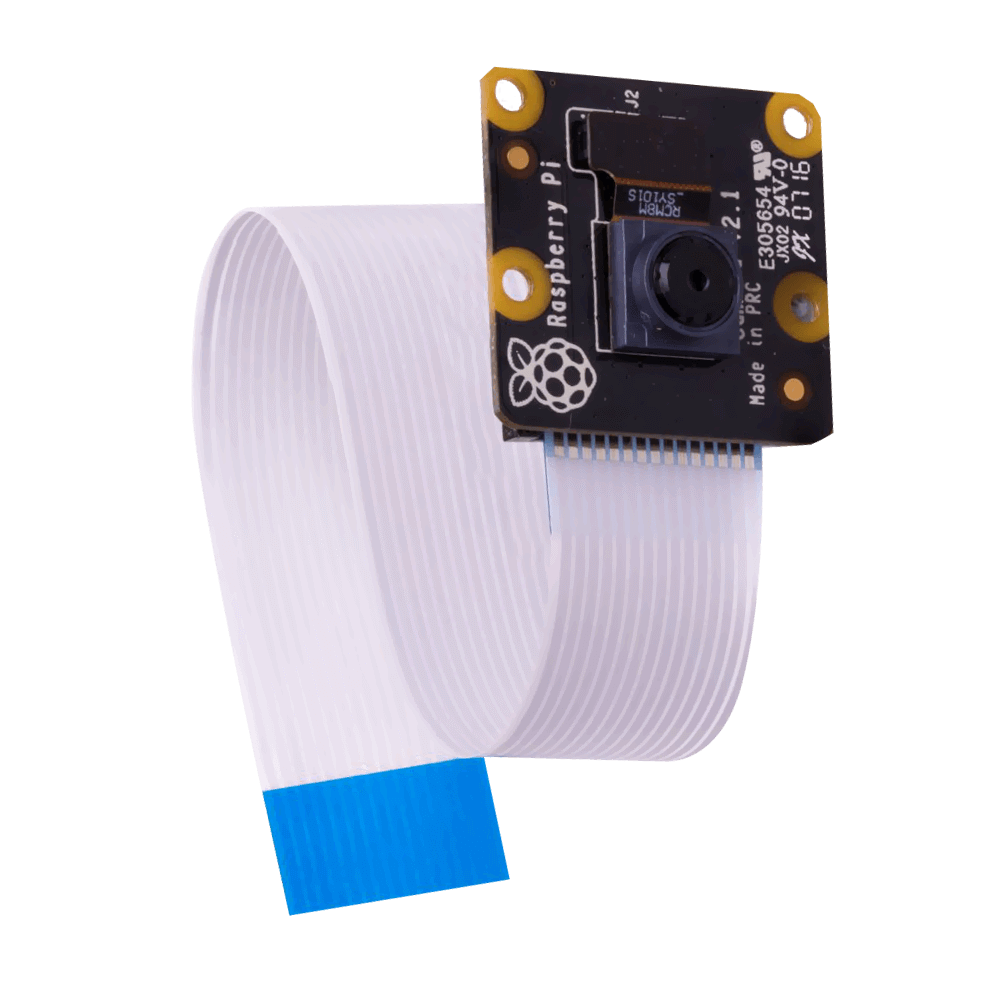
This 5 megapixels sensor with OV5647 camera module is capable of 1080p video and still images that connect directly to your Raspberry Pi. This is the plug-and-play-compatible latest version of the Raspbian operating system, making it perfect for time-lapse photography, recording video, motion detection and security applications. Connect the included ribbon cable to the CSI (Camera Serial Interface) port on your Raspberry Pi, and you are good to go!

The board itself is tiny, at around 25mm x 23mm x 9mm and weighing in at just over 3g, making it perfect for mobile or other applications where size and weight are important. The sensor has a native resolution of 5 megapixel, and has a fixed focus lens on board. In terms of still images, the camera is capable of 2592 x 1944 pixel static images, and also supports 1080p30, 720p60 and 640x480p90 video.

# Features of Pi Camera

Here, we have used Pi camera v1.3. Its features are listed below,

* Resolution – 5 MP
* HD Video recording –     1080p @30fps, 720p @60fps, 960p @45fps and so on.
* It Can capture wide, still (motionless) images of a resolution 2592x1944 pixels
* CSI Interface enabled.



**Fig 4.3 pi camera**

**CHAPTER-5**

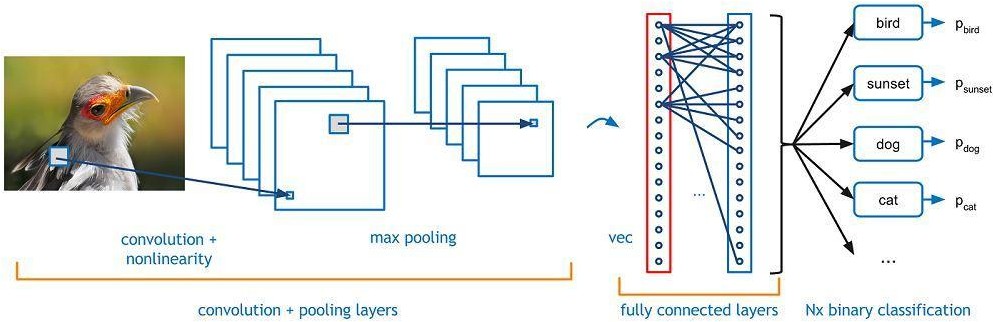
**SOFTWARE DESCRIPTION**

### Algorithm:

* convolutional neural network. It is two stage of CNN. Using this we can detect object easily with grid lines for the classification.
* You Only Look Once:This is an algorithm that detects and recognizes various objects in a picture (in real-time). Object detection in YOLO is done as a regression problem and provides the class probabilities of the detected images.

## 5.1 CONVOLUTION NEURAL NETWORK

Convolutional neural networks. Sounds like a weird combination of biology and math with a little CS sprinkled in, but these networks have been some of the most influential innovations in the field of computer vision. 2012 was the first year that neural nets grew to prominence as Alex Krizhevsky used them to win that year’s ImageNet competition (basically, the annual Olympics of computer vision), dropping the classification error record from 26% to 15%, an astounding improvement at the time. Ever since then, a host of companies have been using deep learning at the core of their services. Facebook uses neural nets for their automatic tagging algorithms, Google for their photo search, Amazon for their product recommendations, Pinterest for their home feed personalization, and Instagram for their search infrastructure.



### Fig 5.1 : Convolution neural network

Image classification is the task of taking an input image and outputting a class (a cat, dog, etc) or a probability of classes that best describes the image. For humans, this task of recognition is one of the first skills we learn from the moment we are born and is one that comes naturally and effortlessly as adults. Without even thinking twice, we’re able to quickly and seamlessly identify the environment we are in as well as the objects that surround us. When we see an image or just when we look at the world around us, most of the time we are able to immediately characterize the scene and give each object a label, all without even consciously noticing. These skills of being able to quickly recognize patterns, generalize from prior knowledge, and adapt to different image environments are ones that we do not share with our fellow machines.

### Inputs and Outputs

When a computer sees an image (takes an image as input), it will see an array of pixel values. Depending on the resolution and size of the image, it will see a 32 x 32 x 3 array of numbers (The 3 refers to RGB values). Just to drive home the point, let's say we have a color image in JPG form and its size is 480 x 480. The representative array will be 480 x 480 x 3. Each of these numbers is given a value from 0 to 255 which describes the pixel intensity at that point. These numbers, while meaningless to us when we perform image classification, are the only inputs available to the computer. The idea is that you give the computer this array of numbers and it will output numbers that describe the probability of the image being a certain class (.80 for cat, .15 for dog, .05 for bird, etc).

### What We Want the Computer to Do

Now that we know the problem as well as the inputs and outputs, let’s think about how to approach this. What we want the computer to do is to be able to differentiate between all the images it’s given and figure out the unique features that make a dog a dog or that make a cat a cat. This is the process that goes on in our minds subconsciously as well. When we look at a picture of a dog, we can classify it as such if the picture has identifiable features such as paws or 4 legs. In a similar way, the computer is able perform image classification by looking for low level features such as edges and curves, and then building up to more abstract concepts through a series of convolutional layers. This is a general overview of what a CNN does. Let’s get into the specifics.

### Biological Connection

But first, a little background. When you first heard of the term convolutional neural networks, you may have thought of something related to neuroscience or biology, and you would be right. Sort of. CNNs do take a biological inspiration from the visual cortex. The visual cortex has small regions of cells that are sensitive to specific regions of the visual field. This idea was expanded upon by a fascinating experiment by Hubel and Wiesel in 1962 (Video) where they showed that some individual neuronal cells in the brain responded (or fired) only in the presence of edges of a certain orientation. For example, some neurons fired when exposed to vertical edges and some when shown horizontal or diagonal edges. Hubel and Wiesel found out that all of these neurons were organized in a columnar architecture and that together, they were able to produce visual perception. This idea of specialized components inside of a system having specific tasks (the neuronal cells in the visual cortex looking for specific characteristics) is one that machines use as well, and is the basis behind CNNs.

### Structure

Back to the specifics. A more detailed overview of what CNNs do would be that you take the image, pass it through a series of convolutional, nonlinear, pooling (downsampling), and fully connected layers, and get an output. As we said earlier, the output can be a single class or a probability of classes that best describes the image. Now, the hard part is understanding what each of these layers do. So let’s get into the most important one.

### First Layer – Math Part

The first layer in a CNN is always a Convolutional Layer. First thing to make sure you remember is what the input to this conv (I’ll be using that abbreviation a lot) layer is. Like we mentioned before, the input is a 32 x 32 x 3 array of pixel values. Now, the best way to explain a conv layer is to imagine a flashlight that is shining over the top left of the image. Let’s say that the light this flashlight shines covers a 5 x 5 area. And now, let’s imagine this flashlight sliding across all the areas of the input image. In machine learning terms, this flashlight is called a filter(or sometimes referred to as a neuron or a kernel) and the region that it is shining over is called the receptive field. Now this filter is also an array of numbers (the numbers are called weights or parameters). A very important note is that the depth of this filter has to be the same as the depth of the input (this makes sure that the math works out), so the dimensions of this filter is 5 x 5 x 3. Now, let’s take the first position the filter is in for example. It would be the top left corner. As the filter is sliding, or convolving, around the input image, it is multiplying the values in the filter with the original pixel values of the image (aka computing element wise multiplications). These multiplications are all summed up (mathematically speaking, this would be 75 multiplications in total). So now you have a single number. Remember, this number is just representative of when the filter is at the top left of the image. Now, we repeat this process for every location on the input volume. (Next step would be moving the filter to the right by 1 unit, then right again by 1, and so on). Every unique location on the input volume produces a number. After sliding the filter over all the locations, you will find out that what you’re left with is a 28 x 28 x 1 array of numbers, which we call an activation map or feature map. The reason you get a 28 x 28 array is that there are 784 different locations that a 5 x 5 filter can fit on a 32 x 32 input image. These 784 numbers are mapped to a 28 x 28 array.

### First Layer – High Level Perspective

However, let’s talk about what this convolution is actually doing from a high level. Each of these filters can be thought of as feature identifiers. When I say features, I’m talking about things like straight edges, simple colors, and curves. Think about the simplest characteristics that all images have in common with each other. Let’s say our first filter is 7 x 7 x 3 and is going to be a curve detector. (In this section, let’s ignore the fact that the filter is 3 units deep and only consider the top depth slice of the filter and the image, for simplicity.)As a curve detector, the filter will have a pixel structure in which there will be higher numerical values along the area that is a shape of a curve (Remember, these filters that we’re talking about as just numbers!).

## 5.2 YOU ONLY LOOK ONCE

YOLO is an abbreviation for the term ‘You Only Look Once’. This is an algorithm that detects and recognizes various objects in a picture (in real-time). Object detection in YOLO is done as a regression problem and provides the class probabilities of the detected images.

YOLO algorithm employs convolutional neural networks (CNN) to detect objects in real-time. As the name suggests, the algorithm requires only a single forward propagation through a neural network to detect objects.

This means that prediction in the entire image is done in a single algorithm run. The CNN is used to predict various class probabilities and bounding boxes simultaneously.

The YOLO algorithm consists of various variants. Some of the common ones include tiny YOLO and YOLOv3.

Why the YOLO algorithm is important

YOLO algorithm is important because of the following reasons:

* **Speed:** This algorithm improves the speed of detection because it can predict objects in real-time.
* **High accuracy:** YOLO is a predictive technique that provides accurate results with minimal background errors.
* **Learning capabilities:** The algorithm has excellent learning capabilities that enable it to learn the representations of objects and apply them in object detection.

How the YOLO algorithm works

YOLO algorithm works using the following three techniques:

* Residual blocks
* Bounding box regression
* Intersection Over Union (IOU)

1. **Residual blocks**

First, the image is divided into various grids. Each grid has a dimension of S x S. The following image shows how an input image is divided into grids.



**Fig 5.2 residual blocks**

[Image Source](https://www.guidetomlandai.com/assets/img/computer_vision/grid.png)

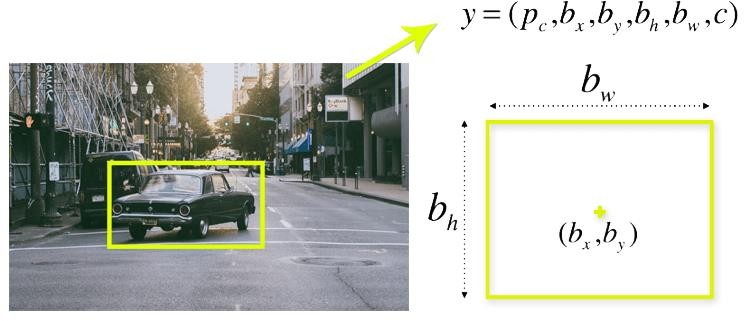
In the image above, there are many grid cells of equal dimension. Every grid cell will detect objects that appear within them. For example, if an object center appears within a certain grid cell, then this cell will be responsible for detecting it.

1. **Bounding box regression**

A bounding box is an outline that highlights an object in an image. Every bounding box in the image consists of the following attributes:

* Width (bw)
* Height (bh)
* Class (for example, person, car, traffic light, etc.)- This is represented by the letter c.
* Bounding box center (bx,by)

The following image shows an example of a bounding box. The bounding box has been represented by a yellow outline.



**Fig 5.3 Bounding boxes**

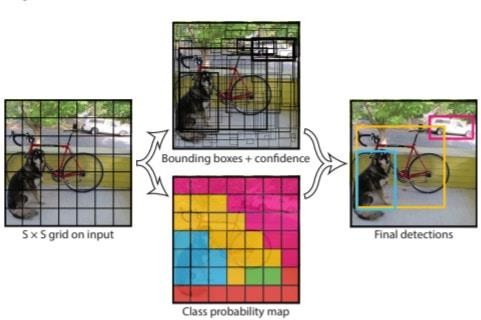
[Image Source](https://appsilondatascience.com/assets/uploads/2018/08/bbox-1.png)

YOLO uses a single bounding box regression to predict the height, width, center, and class of objects. In the image above, represents the probability of an object appearing in the bounding box.

1. **Intersection over union (IOU)**

Combination of the three techniques

The following image shows how the three techniques are applied to produce the final detection results.



**Fig 5.4 Intersection over union**

[Image Source](https://www.guidetomlandai.com/assets/img/computer_vision/YOLO.PNG)

First, the image is divided into grid cells. Each grid cell forecasts B bounding boxes and provides their confidence scores. The cells predict the class probabilities to establish the class of each object.

For example, we can notice at least three classes of objects: a car, a dog, and a bicycle. All the predictions are made simultaneously using a single convolutional neural network.

Intersection over union ensures that the predicted bounding boxes are equal to the real boxes of the objects. This phenomenon eliminates unnecessary bounding boxes that do not meet the characteristics of the objects (like height and width). The final detection will consist of unique bounding boxes that fit the objects perfectly.

For example, the car is surrounded by the pink bounding box while the bicycle is surrounded by the yellow bounding box. The dog has been highlighted using the blue bounding box.

Applications of YOLO

YOLO algorithm can be applied in the following fields:

* **Autonomous driving:** YOLO algorithm can be used in autonomous cars to detect objects around cars such as vehicles, people, and parking signals. Object detection in autonomous cars is done to avoid collision since no human driver is controlling the car.
* **Wildlife:** This algorithm is used to detect various types of animals in forests. This type of detection is used by wildlife rangers and journalists to identify animals in videos (both recorded and real-time) and images. Some of the animals that can be detected include giraffes, elephants, and bears.
* **Security:** YOLO can also be used in security systems to enforce security in an area. Let’s assume that people have been restricted from passing through a certain area for security reasons. If someone passes through the restricted area, the YOLO algorithm will detect him/her, which will require the security personnel to take further action.

## 5.3 INTRODUCTION TO COMPUTER VISION

Using software to parse the world’s visual content is as big of a revolution in computing as mobile was 10 years ago, and will provide a major edge for developers and businesses to build amazing products.

Computer Vision is the process of using machines to understand and analyze imagery (both photos and videos). While these types of algorithms have been around in various forms since the 1960’s, recent advances in [Machine Learning](https://blog.algorithmia.com/introduction-to-machine-learning/), as well as leaps forward in data storage, computing capabilities, and cheap high-quality input devices, have driven major improvements in how well our software can explore this kind of content.

### What is Computer Vision?

Computer vision is a field of artificial intelligence and computer science that focuses on enabling machines to interpret and understand the visual world. It involves the use of algorithms and models to analyze, process, and understand images and videos in order to extract information and make decisions.

The goal of computer vision is to enable machines to perform tasks that normally require human visual perception, such as recognizing objects, detecting and tracking motion, interpreting facial expressions, and identifying patterns and anomalies in visual data.

Computer vision algorithms and models can be trained on large datasets of labeled images and videos using techniques such as deep learning and neural networks. These models learn to recognize patterns and features in visual data, allowing them to make predictions and decisions based on new data.

Applications of computer vision can be found in a wide range of industries and fields, including healthcare, automotive, security, entertainment, and more. Some examples of computer vision applications include:

Computer vision involves a range of techniques and technologies, including image processing, machine learning, deep learning, and neural networks. These techniques are used to develop algorithms and models that can analyze and understand visual data.

In order to develop effective computer vision models, it is important to have high- quality and diverse datasets for training and validation. These datasets must be carefully labeled and annotated to ensure that the models learn the correct features and patterns.

While computer vision has made significant progress in recent years, there are still many challenges that must be addressed, such as handling variability in lighting conditions and perspectives, dealing with occlusions and clutter, and improving the interpretability and explainability of the models.

Overall, computer vision is a rapidly evolving field with tremendous potential for impact in a wide range of applications and industries. As algorithms and models continue to improve and new datasets become available, we can expect to see continued advancements in this exciting field.

Computer Vision is the broad parent name for any computations involving visual content – that means images, videos, icons, and anything else with pixels involved. But within this parent idea, there are a few specific tasks that are core building blocks:

* In object classification, you train a model on a dataset of specific objects, and the model classifies new objects as belonging to one or more of your training categories.
* For object identification, your model will recognize a specific instance of an object – for example, parsing two faces in an image and tagging one as Tom Cruise and one as Katie Holmes.

A classical application of computer vision is handwriting recognition for digitizing handwritten content (we’ll explore more use cases below). Outside of just recognition, other methods of analysis include:

* Video motion analysis uses computer vision to estimate the velocity of objects in a video, or the camera itself.
* In image segmentation, algorithms partition images into multiple sets of views.
* Scene reconstruction creates a 3D model of a scene inputted through images or video (check out [Selva](https://www.selva3d.com/)).
* In image restoration, noise such as blurring is removed from photos using Machine Learning based filters.

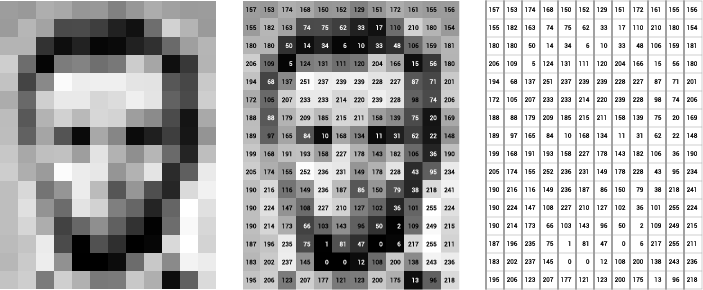
Any other application that involves understanding pixels through software can safely be labeled as computer vision.

### How Computer Vision Works

One of the major open questions in both Neuroscience and Machine Learning is: how exactly do our brains work, and how can we approximate that with our own algorithms? The reality is that there are very few working and comprehensive theories of brain computation; so despite the fact that Neural Nets are supposed to “mimic the way the brain works,” nobody is quite sure if that’s actually true. Jeff Hawkins has an [entire book on this topic called On Intelligence.](https://www.amazon.com/Intelligence-Understanding-Creation-Intelligent-Machines/dp/0805078533)

The same paradox holds true for computer vision – since we’re not decided on how the brain and eyes process images, it’s difficult to say how well the algorithms used in production approximate our own internal mental processes. For example, [studies have shown](https://www.technologyreview.com/s/508376/in-a-frogs-eye/) that some functions that we thought happen in the brain of frogs actually take place in the eyes. We’re a far cry from amphibians, but similar uncertainty exists in human cognition.

Machines interpret images very simply: as a series of pixels, each with their own set of color values. Consider the simplified image below, and how grayscale values are converted into a simple array of numbers:



**Fig 5.5 Grey scale values to simple array numbers**

Think of an image as a giant grid of different squares, or pixels (this image is a very simplified version of what looks like either Abraham Lincoln or a Dementor). Each pixel in an image can be represented by a number, usually from 0 – 255. The series of numbers on the right is what software sees when you input an image. For our image, there are 12 columns and 16 rows, which means there are 192 input values for this image.

When we start to add in color, things get more complicated. Computers usually read color as a series of 3 values – red, green, and blue (RGB) – on that same 0 – 255 scale. Now, each pixel actually has 3 values for the computer to store in addition to its position. If we were to colorize President Lincoln (or Harry Potter’s worst fear), that would lead to 12 x 16 x 3 values, or 576 numbers.

### https://blog.algorithmia.com/wp-content/uploads/2018/03/word-image-13.png

**Fig 5.6 RGB pixels**

For some perspective on how computationally expensive this is, consider this tree:

* Each color value is stored in 8 bits.
* bits x 3 colors per pixel = 24 bits per pixel.
* A normal sized 1024 x 768 image x 24 bits per pixel = almost 19M bits, or about 2.36 megabytes.

That’s a lot of memory to require for one image, and a lot of pixels for an algorithm to iterate over. But to train a model with meaningful accuracy – especially when you’re talking about [Deep Learning](https://blog.algorithmia.com/introduction-to-deep-learning/) – you’d usually need tens of thousands of images, and the more the merrier. Even if you were to use [Transfer Learning](https://en.wikipedia.org/wiki/Transfer_learning) to use the insights of an already trained model, you’d still need a few thousand images to train yours on.

With the sheer amount of computing power and storage required just to train deep learning models for computer vision, it’s not hard to understand why advances in those two fields have driven Machine Learning forward to such a degree.

### Business Use Cases for Computer Vision

Computer vision is one of the areas in Machine Learning where core concepts are already being integrated into major products that we use every day. [Google is using](https://research.googleblog.com/2017/05/updating-google-maps-with-deep-learning.html)

[maps](https://research.googleblog.com/2017/05/updating-google-maps-with-deep-learning.html) to leverage their image data and identify street names, businesses, and office buildings. Facebook is using computer vision to identify people in photos, and do a number of things with that information.

Computer vision has revolutionized various industries by enabling machines to interpret visual information like humans. Here are some of the business use cases for computer vision:

Retail: Computer vision is widely used in the retail industry for various applications like product recognition, inventory management, and customer experience. With product recognition, computer vision algorithms can identify products on the shelves, which enables retailers to track inventory levels and optimize shelf space. Computer vision can also analyze customer behavior to improve their experience.

Healthcare: Computer vision is used in healthcare for a variety of applications, including medical imaging, diagnosis, and patient monitoring. Computer vision algorithms can identify anomalies in medical images, enabling healthcare professionals to diagnose diseases at an early stage. Computer vision can also be used for patient monitoring, detecting falls or changes in patient behavior.

Agriculture: Computer vision is increasingly used in agriculture for crop monitoring and yield prediction. By analyzing aerial images, computer vision algorithms can identify crop patterns and predict yields. This helps farmers to make informed decisions and optimize their crop yields.

Automotive: Computer vision is widely used in the automotive industry for autonomous driving, driver assistance, and safety systems. Computer vision algorithms can recognize objects and pedestrians on the road, enabling cars to avoid collisions and navigate through traffic.

Security: Computer vision is widely used in security systems for facial recognition, object detection, and motion detection. Computer vision algorithms can identify individuals, track their movements, and alert security personnel in case of suspicious behavior.

Manufacturing: Computer vision is used in manufacturing for quality control, inspection, and assembly. By analyzing images of products, computer vision algorithms can detect defects and anomalies, ensuring that products meet quality standards.

Finance: Computer vision is used in the finance industry for fraud detection, identity verification, and compliance. By analyzing facial biometrics and signatures, computer vision algorithms can verify the identity of individuals and detect fraud.

In summary, computer vision has various business use cases across industries like retail, healthcare, agriculture, automotive, security, manufacturing, and finance. It is used for applications like product recognition, medical imaging, crop monitoring, autonomous driving, quality control, fraud detection, and more. By enabling machines to interpret visual information like humans, computer vision is transforming the way businesses operate and make decisions.

But it’s not just tech companies that are leverage Machine Learning for image applications. Ford, the American car manufacturer that has been around [literally since](https://en.wikipedia.org/wiki/Ford_Motor_Company) [the early 1900’s,](https://en.wikipedia.org/wiki/Ford_Motor_Company) is [investing heavily in autonomous vehicles (AVs)](https://media.ford.com/content/fordmedia/fna/us/en/news/2016/08/16/ford-targets-fully-autonomous-vehicle-for-ride-sharing-in-2021.html). Much of the underlying technology in AVs relies on analyzing the multiple video feeds coming into the car and using computer vision to analyze and pick a path of action. Another major area where computer vision can help is in the medical field. Much of diagnosis is image processing, like reading x-rays, MRI scans, and other types of diagnostics. [Google has been working with medical research teams](https://research.google.com/teams/brain/healthcare/) to explore how deep learning can help medical workflows, and have made significant progress in terms of accuracy. To paraphrase from their research page:

“Collaborating closely with doctors and international healthcare systems, we developed a state-of-the-art computer vision system for reading retinal fundus images for diabetic retinopathy and determined our algorithm’s performance is on par with

U.S. board-certified ophthalmologists. We’ve recently published some of our research in the [Journal of the American Medical Association](https://research.google.com/pubs/archive/45732.pdf) and summarized the highlights in a [blog post](https://research.googleblog.com/2016/11/deep-learning-for-detection-of-diabetic.html). ”But aside from the groundbreaking stuff, it’s getting much easier to integrate computer vision into your own applications. A number of high-quality third party providers like Clarifai offer [a simple API for tagging and understanding images](https://www.clarifai.com/), while Kairos [provides functionality around facial recognition.](https://www.kairos.com/) We’ll dive into the open-source packages available for use below.

### Computer Vision on Algorithm

Algorithm makes it easy to deploy computer vision applications as scalable microservices. Our marketplace has a few algorithms to help get the job done:

* [SalNet](https://algorithmia.com/algorithms/deeplearning/SalNet) automatically identifies the most important parts of an image
* [Nudity Detection](https://algorithmia.com/algorithms/sfw/NudityDetectioni2v) detects nudity in pictures
* [Emotion Recognition](https://algorithmia.com/algorithms/deeplearning/EmotionRecognitionCNNMBP) parses emotions exhibited in images
* [Deep Style](https://demos.algorithmia.com/deep-style/) transfers next-level filters onto your image
* [Face Recognition](https://algorithmia.com/algorithms/cv/FaceRecognition)…recognizes faces.
* [Image Memorability](https://algorithmia.com/algorithms/deeplearning/LargescaleImageMemorability) judges how memorable an image is.

A typical workflow for your product might involve passing images from a security camera into Emotion Recognition and raising a flag if any aggressive emotions are exhibited, or using Nudity Detection to block inappropriate profile pictures on your web application.

For a more detailed exploration of how you can use the Algorithmia platform to implement complex and useful computer vision tasks,

### Computer Vision Resources Packages and Frameworks

[OpenCV](https://opencv.org/) – “OpenCV was designed for computational efficiency and with a strong focus on real-time applications. Adopted all around the world, OpenCV has more than 47 thousand people of user community and estimated number of downloads exceeding 14 million. Usage ranges from interactive art, to mines inspection, stitching maps on the web or through advanced robotics.”

[SimpleCV](http://simplecv.org/) – “SimpleCV is an open source framework for building computer vision applications. With it, you get access to several high-powered computer vision libraries such as OpenCV – without having to first learn about bit depths, file formats, color spaces, buffer management, eigenvalues, or matrix versus bitmap storage.”

[Mahotas](http://mahotas.readthedocs.io/en/latest/) – “Mahotas is a computer vision and image processing library for Python. It includes many algorithms implemented in C++ for speed while operating in numpy arrays and with a very clean Python interface. Mahotas currently has over 100 functions for image processing and computer vision and it keeps growing.”

NumPy, which stands for Numerical Python, is a library consisting of multidimensional array objects and a collection of routines for processing those arrays. Using NumPy, mathematical and logical operations on arrays can be performed. This tutorial explains the basics of NumPy such as its architecture and environment. It also discusses the various array functions, types of indexing, etc. An introduction to Matplotlib is also provided. All this is explained with the help of examples for better understanding.

Audience

This tutorial has been prepared for those who want to learn about the basics and various functions of NumPy. It is specifically useful for algorithm developers. After completing this tutorial, you will find yourself at a moderate level of expertise from where you can take yourself to higher levels of expertise.

Prerequisites

You should have a basic understanding of computer programming terminologies.

A basic understanding of Python and any of the programming languages is a plus.

NumPy is a Python package. It stands for 'Numerical Python'. It is a library consisting of multidimensional array objects and a collection of routines for processing of array.

**Numeric**, the ancestor of NumPy, was developed by Jim Hugunin. Another package Numarray was also developed, having some additional functionalities. In 2005, Travis Oliphant created NumPy package by incorporating the features of Numarray into Numeric package. There are many contributors to this open source project.

Operations using NumPy

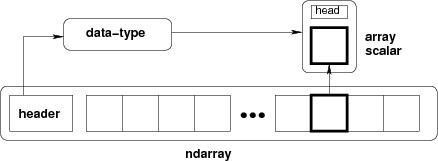
Using NumPy, a developer can perform the following operations −

* Mathematical and logical operations on arrays.
* Fourier transforms and routines for shape manipulation.
* Operations related to linear algebra. NumPy has in-built functions for linear algebra and random number generation.

NumPy – A Replacement for MatLab

NumPy is often used along with packages like **SciPy** (Scientific Python) and **Mat−plotlib** (plotting library). This combination is widely used as a replacement for MatLab, a popular platform for technical computing. However, Python alternative to MatLab is now seen as a more modern and complete programming language. The most important object defined in NumPy is an N-dimensional arraytype called **ndarray**. It describes the collection of items of the same type. Items in the collection can be accessed using a zero-based index. Every item in an ndarray takes the same size of block in the memory. Each element in ndarray is an object of data-type object (called **dtype**).Any item extracted from ndarray object (by slicing) is represented by a Python object of one of array scalar types. The following diagram shows a relationship between ndarray, data type object (dtype) and array scalar type

−



An instance of ndarray class can be constructed by different array creation routines described later in the tutorial. The basic ndarray is created using an array function in NumPy as follows −

numpy.array

It creates an ndarray from any object exposing array interface, or from any method that returns an array.

Language used:

### What is Python

**Python** is an object-oriented, high level language, interpreted, dynamic and multipurpose programming language. Python is easy to learn yet powerful and versatile scripting language which makes it attractive for Application Development. Python's syntax and dynamic typing with its interpreted nature, make it an ideal language for scripting and rapid application development in many areas. Python supports multiple programming pattern, including object oriented programming, imperative and functional programming or procedural styles. Python is not intended to work on special area such as web programming. That is why it is known as multipurpose because it can be used with web, enterprise, 3D CAD etc. We don't need to use data types to declare variable because it is dynamically typed so we can write a=10 to declare an integer value in a variable. Python makes the development and debugging fast because there is no compilation step included in python development and edit-test-debug cycle is very fast.

Python is a popular high-level programming language that is known for its simplicity, readability, and versatility. It was created by Guido van Rossum and first released in 1991. Since then, it has become one of the most widely used programming languages in the world, with applications in web development, data analysis, artificial intelligence, scientific computing, and many other areas.

Python is a dynamically typed language, which means that variables do not need to be declared with a specific type. Instead, the interpreter automatically determines the type of a variable based on its value. Python also uses whitespace indentation to define code blocks, rather than using curly braces or keywords like "begin" and "end".

One of the key features of Python is its extensive standard library, which includes modules for a wide range of tasks, such as file I/O, network communication, regular expressions, and more. There are also many third-party libraries and frameworks available for Python, such as NumPy, Pandas, TensorFlow, Django, Flask, and many others.

Python is often used for scripting and automation tasks, as well as for data analysis and visualization. It is also widely used in machine learning and artificial intelligence applications, due to its ease of use and the availability of powerful libraries like TensorFlow, PyTorch, and Keras.

In addition to its versatility, Python is also known for its strong community and ecosystem. There are many online resources available for learning and using Python, including documentation, tutorials, and forums. Python is also frequently taught in schools and universities as an introductory programming language.

Overall, Python is a popular and powerful programming language that is used in a wide range of applications and industries. Its simplicity, readability, and versatility make it a great choice for both beginners and experienced programmers alike.

### Python Features

1. **Easy to Use:**

Python is easy to very easy to use and high level language. Thus it is programmer- friendly language.

### Expressive Language:

Python language is more expressive. The sense of expressive is the code is easily understandable.

### Interpreted Language:

Python is an interpreted language i.e. interpreter executes the code line by line at a time. This makes debugging easy and thus suitable for beginners.

### Cross-platform language:

Python can run equally on different platforms such as Windows, Linux, Unix , Macintosh etc. Thus, Python is a portable language.

### Free and Open Source:

Python language is freely available(www.python.org).The source-code is also available. Therefore it is open source.

### Object-Oriented language:

Python supports object oriented language. Concept of classes and objects comes into existence.

### Extensible:

It implies that other languages such as C/C++ can be used to compile the code and thus it can be used further in your python code.

### Large Standard Library:

Python has a large and broad library.

### GUI Programming:

Graphical user interfaces can be developed using Python.

### Python History

* Python laid its foundation in the late 1980s.
* The implementation of Python was started in the December 1989 by **Guido Van Rossum** at CWI in Netherland.
* ABC programming language is said to be the predecessor of Python language which was capable of Exception Handling and interfacing with Amoeba Operating System.
* Python is influenced by programming languages like:
  + - ABC language.
    - Modula-3

### Python Version

Python programming language is being updated regularly with new features and support. There are a lot of up dation in python versions, started from 1994 to current date. Python is a popular programming language used for various applications such as web development, data analysis, machine learning, and more. Python has several versions, with each version having its own unique features, improvements, and bug fixes.

Python's development is managed by the Python Software Foundation, which releases new versions of the language periodically. These releases are labeled with version numbers, which consist of three numbers separated by dots, such as "3.10.0". The first number represents the major version, the second number represents the minor version, and the third number represents the patch version.

Major versions introduce significant changes to the language, such as syntax changes or the introduction of new features. Minor versions typically include bug fixes and improvements, while patch versions fix critical issues and security vulnerabilities.

Python has two major versions that are currently in use: Python 2.x and Python 3.x. Python 2.x was released in 2000 and has been widely used for many years. However, Python 2.x is no longer being actively developed and is considered a legacy version. Python 3.x was released in 2008 and has since become the default version for most applications. One of the key differences between Python 2.x and Python 3.x is the way they handle strings. Python 2.x treats strings as a sequence of bytes, while Python 3.x treats them as a sequence of Unicode characters. This change can affect how code written for Python 2.x works in Python 3.x.

A list of python versions with its released date is given below.

|  |  |
| --- | --- |
| **Python Version** | **Released Date** |
| Python 1.0 | January 1994 |
| Python 1.5 | December 31, 1997 |
| Python 1.6 | September 5, 2000 |
| Python 2.0 | October 16, 2000 |
| Python 2.1 | April 17, 2001 |
| Python 2.2 | December 21, 2001 |
| Python 2.3 | July 29, 2003 |
| Python 2.4 | November 30, 2004 |
| Python 2.5 | September 19, 2006 |
| Python 2.6 | October 1, 2008 |
| Python 2.7 | July 3, 2010 |
| Python 3.0 | December 3, 2008 |
| Python 3.1 | June 27, 2009 |
| Python 3.2 | February 20, 2011 |
| Python 3.3 | September 29, 2012 |

**Table 5.1 Python versions and its release dates**

**Python Applications**

Python as a whole can be used in any sphere of development.

Let us see what are the major regions where Python proves to be handy.

### Console Based Application

Python can be used to develop console based applications. For example: **IPython**.

### Audio or Video based Applications

Python proves handy in multimedia section. Some of real applications are: Tim Player, cplay etc.

### 3D CAD Applications

Fandango is a real application which provides full features of CAD.

### Web Applications

Python can also be used to develop web based application. Some important developments are: Python Wiki Engines, Pocoo, Python Blog Software etc.

### Enterprise Applications

Python can be used to create applications which can be used within an Enterprise or an Organization. Some real time applications are: Open Erp, Tryton, Picalo etc.

### Applications for Images

Using Python several application can be developed for image. Applications developed are: VPython, Gogh, img Seek etc.

There are several such applications which can be developed using Python

### Python Example

Python code is simple and easy to run. Here is a simple Python code that will print "Welcome to Python".

A simple python example is given below.

1. >>> a="Welcome To Python"
2. >>> print a
3. Welcome To Python 4. >>>

### Explanation:

* + Here we are using IDLE to write the Python code. Detail explanation to run code is given in Execute Python section.
  + A variable is defined named "a" which holds "Welcome To Python".
  + "print" statement is used to print the content. Therefore "print a" statement will print the content of the variable. Therefore, the output "Welcome To Python" is produced.

Python 3.4 Example

In python 3.4 version, you need to add parenthesis () in a string code to print it.

1. >>> a=("Welcome To Python Example")
2. >>> print a
3. Welcome To Python Example 4. >>>

### 7. How to execute python

There are three different ways of working in Python:

1. Interactive Mode:

You can enter python in the command prompt and start working with Python.



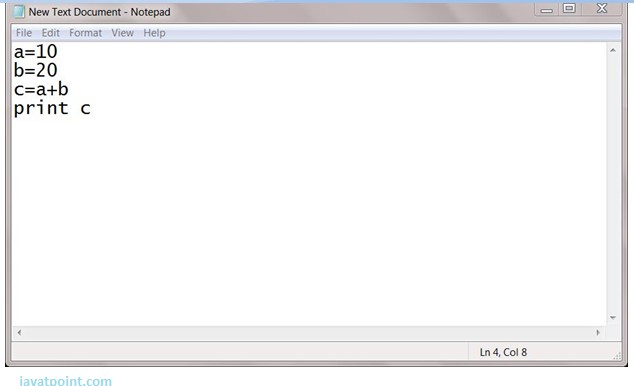
Press Enter key and the Command Prompt will appear like:

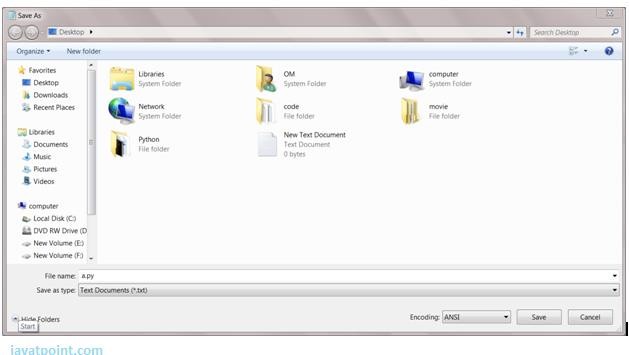
### write python program

### execute python in interactive modeNow you can execute your python commands

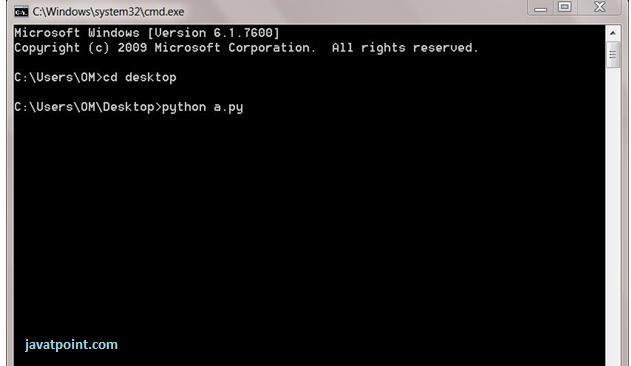
1. Script Mode:

Using Script Mode , you can write your Python code in a separate file using any editor of your Operating System.



Save it by .py extension.

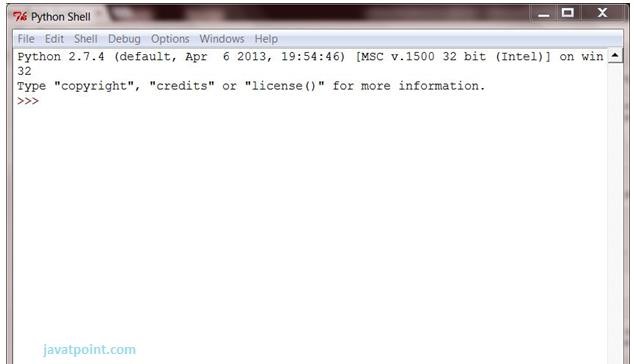
Now open Command prompt and execute it by :



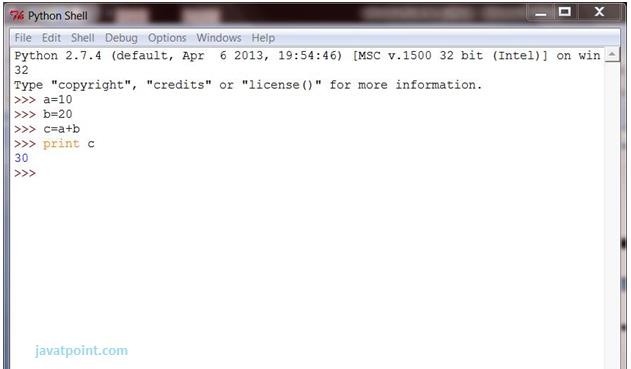
1. Using IDE: (Integrated Development Environment)

You can execute your Python code using a Graphical User Interface (GUI). All you need to do is:

Click on Start button -> All Programs -> Python -> IDLE(Python GUI)

You can use both Interactive as well as Script mode in IDE.

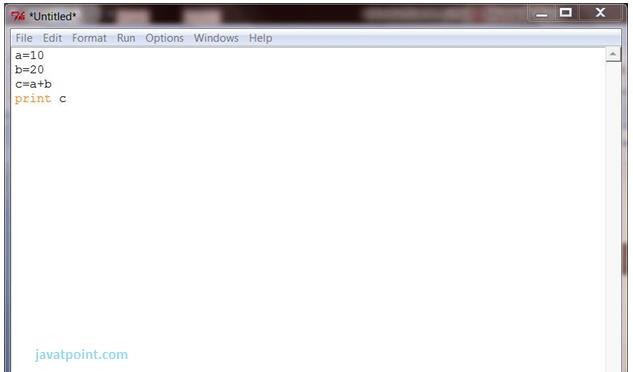
### Using Interactive mode:

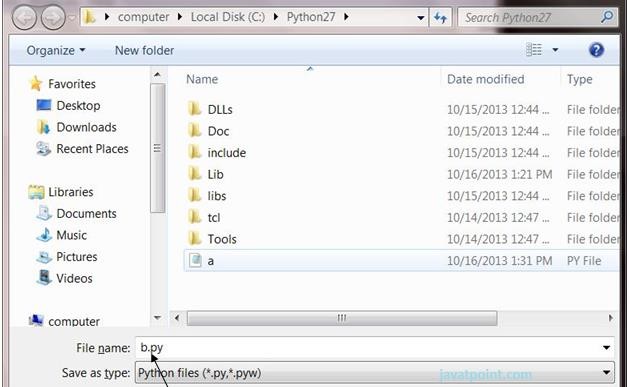
Execute your Python code on the Python prompt and it will display result simultaneously.

### Using Script Mode:

* 1. Click on Start button -> All Programs -> Python -> IDLE(Python GUI)
  2. Python Shell will be opened. Now click on File -> New Window.

A new Editor will be opened . Write your Python code here.

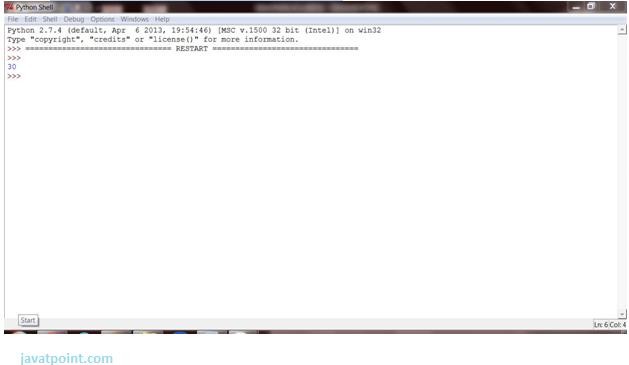


Click on file -> save as

Run then code by clicking on Run in the Menu bar.

Run -> Run Module

Result will be displayed on a new Python shell as:



### Python Variables

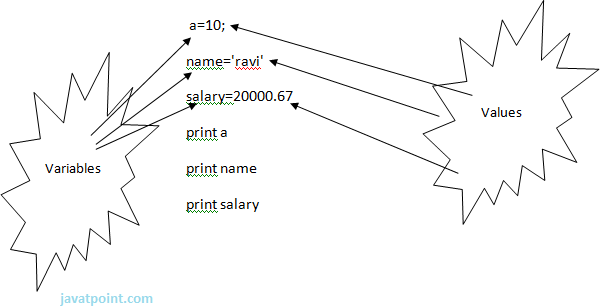
Variable is a name of the memory location where data is stored. Once a variable is stored that means a space is allocated in memory.

Assigning values to Variable:

We need not to declare explicitly variable in Python. When we assign any value to the variable that variable is declared automatically.

The assignment is done using the equal (=) operator.

### Eg:



**Output:**

1. >>>

2. 10

1. ravi

4. 20000.67

5. >>>

Multiple Assignment:

Multiple assignment can be done in Python at a time.

There are two ways to assign values in Python:

**1.Assigning single value to multiple variables:**

**Eg:**

**1**. x=y=z=50

**2**.print x

**3**.print y

**4**.print z

### Output:

1. >>>

2. 50

3. 50

4. 50

5. >>>

**2.Assigning multiple values to multiple variables:**

**Eg:**

1. a,b,c=5,10,15

1. print a
2. print b
3. print c

### Output:

1. >>>

2. 5

3. 10

4. 15

5. >>>

The values will be assigned in the order in which variables appears.

Basic Fundamentals:

This section contains the basic fundamentals of Python like :

**i.Tokens and their types.**

**ii.Comments**

**a)Tokens:**

* + - Tokens can be defined as a punctuator mark, reserved words and each individual word in a statement.
    - Token is the smallest unit inside the given program.

There are following tokens in Python:

* + - Keywords.
    - Identifiers.
    - Literals.
    - Operators.

Tuples:

* + - Tuple is another form of collection where different type of data can be stored.
    - It is similar to list where data is separated by commas. Only the difference is that list uses square bracket and tuple uses parenthesis.
    - Tuples are enclosed in parenthesis and cannot be changed.

### Eg:

1. >>> tuple=('rahul',100,60.4,'deepak')

2. >>> tuple1=('sanjay',10)

3. >>> tuple

4. ('rahul', 100, 60.4, 'deepak')

5. >>> tuple[2:]

6. (60.4, 'deepak')

7. >>> tuple1[0]

1. 'sanjay'
2. >>> tuple+tuple1

10. ('rahul', 100, 60.4, 'deepak', 'sanjay', 10)

11. >>>

* + Diction ved by square bracket([]).**Eg:**

1. >>> dictionary={'name':'charlie','id':100,'dept':'it'}
2. >>> dictionary

3. {'dept': 'it', 'name': 'charlie', 'id': 100}

1. >>> dictionary.keys()
2. ['dept', 'name', 'id']

6. >>dictionary.values()

1. 7. ['it', 'charlie', 100]

8. >>>

**CHAPTER-6**

**RESULTS AND DISCUSSIONS**

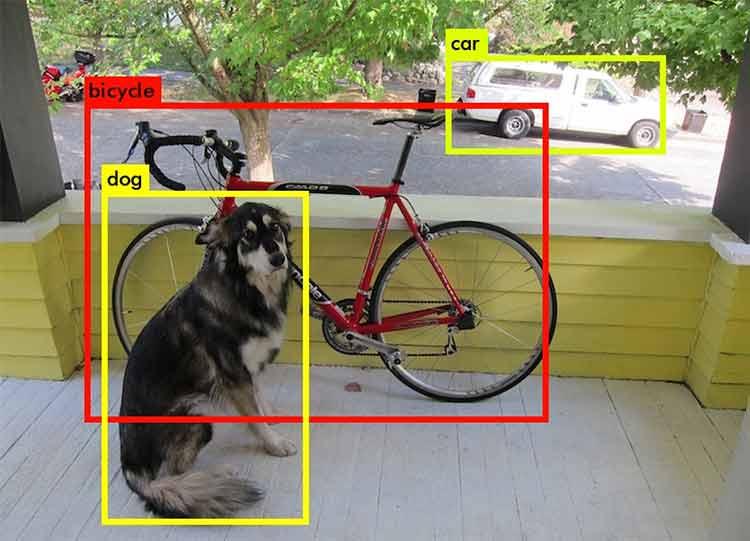
Object detection using Raspberry pi will depend on specific goals of the project, the hardware and software used, and quality of the dataset and model. Typically, an object detection using raspberry pi involves training a deep learning model on a dataset of images to detect specific objects or classes. The trained model is then deployed on the Raspberry pi to perform real time object detection.

Here we are using algorithms like YOLO and RCNN for object detection using raspberry pi. Though the algorithms show accurate results and speed of detection is too fast. Due to the limited computing power of raspberry pi can result in slower inference times and lower accuracy of detection system.

Hence the object detection using a Raspberry pi can achieve decent results, but the performance may not be as good as on more powerful devices such as desktop computers or servers.



**Fig 6.1 Result of object detection**



**Fig 6.2 Result of animals and vehicles detection**

**CHAPTER-7**

**CONCLUSIONS AND FUTURE SCOPE**

## 7.1 CONCLUSION

In the proposed system we have detected the object from the video on raspberry pi 3B+ using two models RCNN and YOLO v3. We introduce YOLO, a unified model for object detection. Our model is simple to construct and can be trained directly on full images. Unlike classifier-based approaches, YOLO is trained on a loss function that directly corresponds to detection performance and the entire model is trained jointly. Fast YOLO is the fastest general-purpose object detector in the literature and YOLO pushes the state-of-the-art in real-time object detection. YOLO also generalizes well to new domains making it ideal for applications that rely on fast, robust object detection.

## 7.2 FUTURE SCOPE

* + - For the next generation with fully real time based automatic detection we can use augmentation process for the more detection. These can be improved for multimedia and real time applications.
    - In future scope we can add gps module for location tracking using IOT.

ary:

* + Dictionary is a collection which works on a key-value pair.
  + It works like an associated array where no two keys can be same.
  + Dictionaries are enclosed by curly braces ({}) and values can be retrie

**REFERENCES**

1. Mohapatra, Hitesh. (2015). HCR (English) using neural network. International journal of advance research and innovative ideas in education, 1(4), 379385.
2. Mohapatra, H., & Rath, A. K. (2019). Detection and avoidance of water loss through municipality taps in India by using smart taps and ICT. IET Wireless sensor systems, 9(6), 447-457.
3. Mohapatra, H., & Rath, A. K. (2019). Fault tolerance in WSN through PE- LEACH protocol. IET wireless sensor systems, 9(6), 358-365.
4. Mohapatra, H., Debnath, S., & Rath, A. K. (2019). Energy management in wireless sensor network through EB-LEACH (No. 1192). Easy Chair.
5. Nirgude, V., Mahapatra, H., & Shivarkar, S. (2017). Face recognition system using principal component analysis & linear discriminant analysis method simultaneously with 3d morphable model and neural network BPNN method. Global journal of advanced engineering technologies and sciences, 4(1), 1-6.