BLIND ASSISTANCE SYSTEM

Project Report Submitted

In Partial Fulfillment of the Requirements

For the Degree Of

BACHELOR OF ENGINEERING

IN

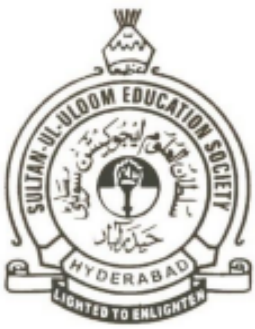
COMPUTER SCIENCE AND ENGINEERING

Submitted By

Ayesha Sultana (1604-17-733-062)

Mehnaz Fatima (1604-17-733-070)

Syed Iftaqar Hussain (1604-17-733-318)



COMPUTER SCIENCE AND ENGINEERING DEPARTMENT
MUFFAKHAM JAH COLLEGE OF ENGINEERING & TECHNOLOGY (Affiliated to Osmania University)
Mount Pleasant, 8-2-249, Road No. 3, Banjara Hills, Hyderabad-34

2021

Date: 4 / 06 / 2021

CERTIFICATE

This is to certify that the project dissertation titled "**BLIND ASSISTANCE SYSTEM**" being submitted by

1. Ayesha Sultana (1604-17-733-062)
2. Mehnaz Fatima (1604-17-733-070)
3. Syed Iftaqar Hussain (1604-17-733-318)

In Partial Fulfilments of the requirements for the award of the degree of BACHELOR OF ENGINEERING IN COMPUTER SCIENCE AND ENGINEERING in MUFFAKHAM JAH COLLEGE OF ENGINEERING AND TECHNOLOGY, Hyderabad for the academic year 2020-21 is the bonafide work carried out by them. The results embodied in this report have not been submitted to any other University or Institute for the award of any degree or diploma.

Signatures:

Internal Project Guide Head CSE (Dr. A.A. Moiz Qyser)

Associate Professor CSE (Mrs Krishna Keerthi)

External Examiner

DECLARATION

This is to certify that the work reported in the major project entitled "**BLIND ASSISTANCE SYSTEM**" is a record of the bonafide work done by us in the Department of Computer Science and Engineering, Muffakham Jah College of Engineering and Technology, Osmania University. The results embodied in this report are
Based on the project work done entirely by us and not copied from any other source.

1. Ayesha Sultana (1604-17-733-062)
2. Mehnaz Fatima (1604-17-733-070)
3. Syed Iftaqar Hussain (1604-17-733-318)

ACKNOWLEDGEMENT

Our hearts are filled with gratitude to the Almighty for empowering us with courage, wisdom and strength to complete this project successfully. We give him all the glory, honour and praise.

We thank our Parents for having sacrificed a lot in their lives to impart the best education to us and make us promising professionals for tomorrow.

We would like to express our sincere gratitude and indebtedness to our project supervisor Mrs **Maniza Hijab** for her valuable suggestions and interest throughout the course of this project.

We are happy to express our profound sense of gratitude and indebtedness to **Proof Dr. Ahmed Abdul Moiz Qyser,** Head of the Computer Science and Engineering Department, for his valuable and intellectual suggestions apart from educating guidance, constant encouragement right throughout our work and making us successful.

With a great sense of pleasure and privilege, we extend our gratitude to **Proof Dr. Syed Shabbier Ahmed,** Associated Head of Computer Science and Engineering Department, project in-charge, who offered valuable suggestions was a pre-requisite to carry out support in every step.

We are pleased to acknowledge our indebtedness to all those who devoted themselves directly or indirectly to make this project work a total success.

Ayesha Sultana

Mehnaz Fatima

Syed Iftaqar Hussain

List of Figures

List of Tables

Abstract

The machine learning model project helps blind and visually impaired people to detection and recognition the office tools around them, which they see through a small camera. This technique helps providing job opportunities for the blind, especially office work through a voice message sent to an earphone placed on the blind ear to help him/her find various items easily and independently. This saves time and efforts.

Our aim is to create an intelligent system, imitating the human eye, which transfers different scenes and images to the brain. The brain in turn analyses the images or scenes, and based on previously stored information, the surrounding objects are identified. For this purpose, we use a small device that performs similar to the human brain, called smart phone; it is a small device that analyses the images and scenes with the help of the camera, which moves the images to the small device. Then, the process of analysis begins through long complex algorithms known as the neural network algorithms. This network analyses the images to parts in order to compare them with the most important characteristics of the objects in the images related to the database, through which the images are compared. When ensuring that the characteristics match the mathematical equations programmed in the language of the Python, the objects in the image are detected. Finally, the sound of each tool in the database is called, and a message is sent to tell the blind about the tools in front of him/ her.

Content

Appendix

A

CHAPTER ONE

1

Introduction

1.1 Overview

1.2 Project Motivation

1.3 Project Aims

1.4 Project idea and Importance

1.5 Literature Review

1.6 List of Abbreviation

1.7 Estimated Cost

1.8 Scheduling Table

1.1 Overview

The project of Blind assistance aims to promote a wide challenge in computer vision such as recognition of objects of the surrounding objects practiced by the blind daily. The camera placed on blind person's glasses, MS COCO is large-scale object detection, segmentation, are employed to provide the necessary information about the surrounding environment. A dataset of objects gathered from daily scenes is created to apply the required recognition. Objects detection is used to find objects in the real world from an image of the world such as bicycles, chairs, doors, or tables that are common in the scenes of a blind. Based on their locations, and the camera is used to detect any objects. The proposed method for the blind aims at expanding possibilities to people with vision loss to achieve their full potential. The main object of the project is to design and implement real-time object recognition using blind glass.

1.2 List of Abbreviation

Table (1.1): List of Abbreviation

Abbreviation	Full Meaning
GPS	Global Positioning System
TF	Tensor flow
HOG	Histogram of Oriented Gradients
SURF	Speeded Up Robust Features
MS COCO	Common Objects in Context
API	Application Programming Interface
PASCAL	pattern analysis statically modelling and computational learning
VPU	Visual processing Unit

1.3 Project Motivation

Percentage of persons with disabilities in Palestinian society. Especially those with visual disabilities (blind) which it is at estimated [0.6 %] is not simple. From here the idea of our project begins where it aims. The project of Blind assistance aims to promote a wide challenge in computer vision such as **recognition of objects** of the surrounding objects practiced by the blind daily.

1.4 Project Aims

The aim of this project is to propose a system, designed for visually impaired individuals to assist them with getting around. Unlike most commercially available assistive devices, this system should provide-

- Directions to locations.

- Give voice alerts to the users of obstacles in their path.
- Calculates distance to provide warnings whether he or she is close or far away from the object.

1.5 Project idea and Importance

This project is mainly aimed at helping people who are blind and who suffer from a total lack of vision.

Due to the development of technology, we must be tapped to help blind people.

Due to a large number of blind people in Palestine.

The next future and the future of technology is to serve people and help them in life.

1.6 Literature Review

1- Real-Time Objects Recognition Approach for Assisting Blind People.

[Jamal S. Zraqou Wissam M. Alkhadour and Mohammad Z. Siam, Multimedia Systems Department, Electrical Engineering Department, Isra University, Amman-Jordan Accepted 30 Jan 2017, Available online 31 Jan 2017, Vol.7, No.1]

Blind assistance is promoting a wide challenge in computer vision such as navigation and path finding in this paper, two cameras placed on blind person's glasses, GPS free service, and ultra-sonic sensor are employed to provide. The necessary information about the surrounding environment. A dataset of objects gathered from daily scenes is created to apply the required recognition. Objects detection is used to find objects in the real world from an image of the world such as faces, bicycles, chairs, doors, or tables that are common in the scenes of a blind.

The two cameras are necessary to generate the depth by creating the disparity map of the scene, GPS service is used to create groups of objects based on their locations, and the sensor is used to detect any obstacle at a medium to long distance.

The descriptor of the Speeded-Up Robust Features method is optimized to perform the recognition. The proposed method for the blind aims at expanding possibilities to people with vision loss to achieve their full potential. The experimental results reveal the performance of the proposed work in about real-time system

2-Vocal vision for visually impaired

[The International Journal Of ISSN: 2319 – 1813 Isbn: 2319 – 1805 Engineering And Science (Ijes)-01-07||2013|| Shrilekha Banger, Preetam Narkhede, Rajashree Parajape.]

This project is a vision substitute system designed to assist blind people with autonomous navigation. Its working concept is based on ‘image to sound’ conversion. The vision sensor captures the image in front of a blind user. This image is then fed to MATLAB for processing.

Process intuit processes the captured image and enhances the significant vision data. This processed image is then compared with the database kept in the microcontroller. The processed information is then presented as a structured form of the acoustic signal and it is conveyed to the blind user using a set of earphones time system relation from the interested objects evaluated to determine the colour of the object. The colour output is informed to the blind user through headphones.

3-Object Detection Combining Recognition and Segmentation

[Fudan University, Shanghai, PRC, yfshen@fudan.edu.cn University of Pennsylvania,3330WalnutStreet, Philadelphia, PA19104 Liming Wang1, Jianbo Shi2, Gang Song2, and I-fan Shen.]

We develop an object detection method combining top-down recognition with bottom-up image segmentation. There are two main steps in this method: a hypothesis generation step and a verification step. In the top-down hypothesis generation step, we design an improved Shape Context feature, which is more robust to object deformation and background clutter. The improved Shape Context is used to generate a set of hypotheses of object locations and figure-ground masks, which have high recall and low precision rates. In the verification step, we first compute a set of feasible segmentations that are consistent with top-down object hypotheses, then we propose a False Positive Pruning(FPP) procedure to prune out false positives. We exploit the fact that false-positive regions typically do not align with any feasible image segmentation. Experiments show that this simple framework is capable of achieving both high recall and high precision with only a few positive training examples and that this method can be generalized to many object classes.

4 - Microsoft COCO Common Objects in Context

[Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, Piotr Dollár(Submitted on 1 May 2014 (v1), last revised 21 Feb 2015 (this version, v3))]

We present a new dataset to advance the state-of-the-art in object recognition by placing the question of object recognition in the context of the broader question of scene understanding. This is achieved by gathering images of complex everyday scenes containing common objects in their natural context. Objects are labelled using per-instance

segmentations to aid in precise object localization. Our dataset contains photos of 91 object types that would be easily recognizable by a 4-year-old. With a total of 2.5 million labelled instances in 328k images, the creation of our dataset drew upon extensive crowd worker involvement via novel user interfaces for category detection, instance spotting, and instance segmentation. We present a detailed statistical analysis of the dataset in comparison to PASCAL, Image Net, and SUN. Finally, we provide baseline performance analysis for bounding box and segmentation detection results using a Deformable Parts Model.

1.7 Project Cost

Table (1.2): Project Cost

Component	Cost(in $)
Mini Camera	20 $
Ear Phone	10$
Rechargeable Battery	30$
Other Requirements	10$

1.8 Scheduling Table

Table (1.3): shows the activities that are done in the project attachment a and the time of each one.

Activities Weeks	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
System Definition	■	■	■	■													
System Analysis					■	■	■	■									
System Design									■	■	■	■	■				
Presentation Preparing														■	■	■	■
Documentation				■	■	■	■	■	■	■	■	■	■	■			

CHAPTER TWO

2

Human Eye: Anatomy & Physiology

2.1 Introduction

2.2 Human Eye Anatomy

2.3 Visual Processing

2.4 Visual Impairment and blindness

2.4.1 Causes of Visual Impairment and Blindness

2.1 Introduction

The human eye is the organ that gives us the sense of sight, allowing us to observe and learn more about the surrounding world than we do with any of the others from sense. We use our eyes in almost every activity we perform, whether reading, working, watching television, writing a letter, driving a car, and in countless other ways. Most people probably would agree that sight is the sense they value more than all the rest.

2.2 Human Eye Anatomy

The human eye is very nearly spherical, with a diameter of approximately 24 millimetres (nearly 1 inch), or slightly smaller than a Ping-Pong ball. It consists of three concentric layers, each with its characteristic appearance, structure, and functions [1].

From outermost to innermost, the three layers are the fibrous tunic, which protects the eyeball; and the retina which detects light and initiates neural messages bound for the brain [2].

The eye is made up of three coats, enclosing three transparent structures. The outermost layer, known as the fibrous tunic, is composed of the cornea and sclera. The middle layer known as the vascular tunic or uvea consists of the choroid, ciliary body, and iris. The innermost is the retina, which gets its circulation from the vessels of the choroid as well as the retinal vessels, which can be seen in an ophthalmoscope [3].

Within these coats are the aqueous humour, the vitreous body, and the flexible lens. The aqueous humour is a clear fluid that is contained in two areas: the anterior chamber between the cornea and the iris, and the posterior chamber between the iris and the lens. The lens is suspended to the ciliary body by the suspensory ligament (Zonule of Zinn), made up of fine transparent fibres the vitreous body is a clear jelly that is much larger than the aqueous humour present behind the lens, and the rest is bordered by the sclera, zonula, and lens. They are connected via the pupil [4]. Figure (2.1) illustrates the main components of the human eye [5].

Conjunctiva: is a thin protective covering of epithelial cells. It protects the cornea against damage by friction (tears from the tear glands help this process by lubricating the surface of the conjunctiva).

Cornea: is the transparent, curved front of the eye that helps to converge the light rays which enter the eye.

The sclera: is an opaque, fibrous, protective outer structure. It is soft connective tissue and the spherical shape of the eye is maintained by the pressure of the liquid inside. It provides an attachment, a the surface for eye muscle

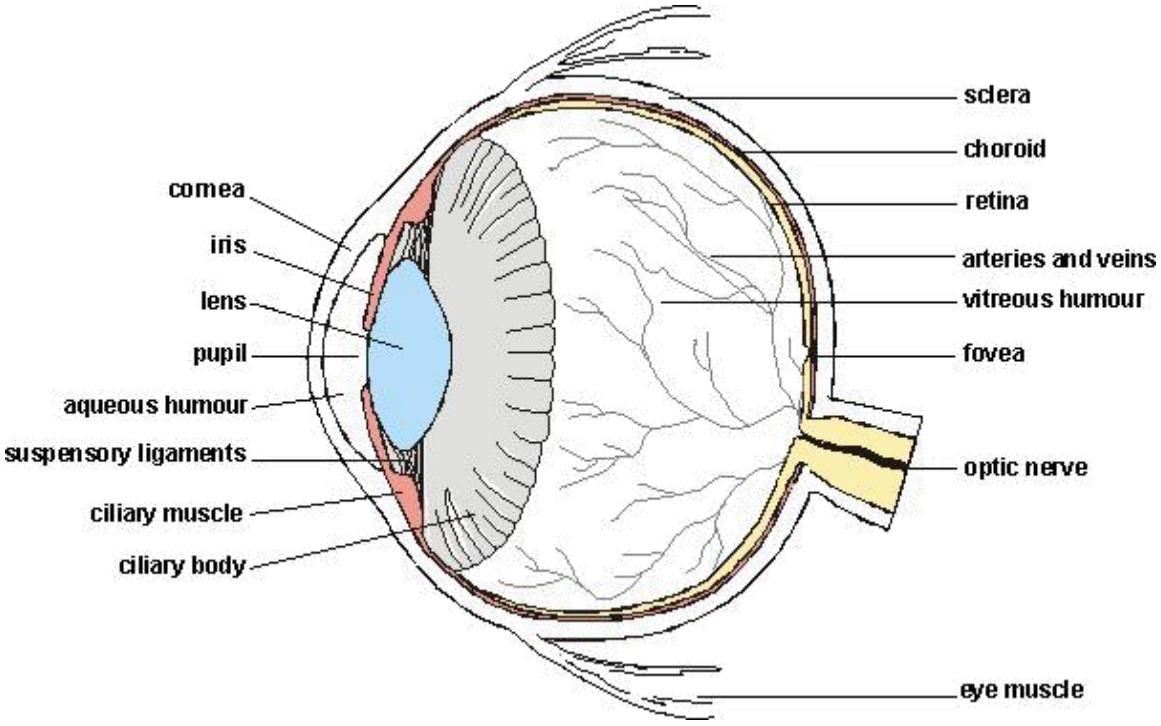


Figure (2.1): Components of eye [5].

Choroid: has a network of blood vessels to supply nutrients to the cells and remove waste products. It is pigmented that makes the retina appear black, thus preventing reflection of light within the eyeball.

Ciliary body: has suspensory ligaments that hold the lens in place. It secrets the aqueous humour and contains ciliary muscles that enable the lens to change shape, during accommodation (focusing on near and distant objects).

Iris: is a pigmented muscular structure consisting of an inner ring circular muscle and an outer layer of radial muscle.

Pupil: is a hole in the middle of the iris where light is allowed to continue its passage. In bright light, it is constricted and in dim light is dilated.

Lens: is a transparent, flexible, curved structure. Its function is to focus incoming light rays onto the retina using its refractive properties.

Retina: is a layer of sensory neurons, the key structure being photoreceptors (rod and cone cells) which respond to light. Contains relay neurons and sensory neurons that pass impulses along the optic nerve to the part of the brain that controls vision.

Fovea (yellow spot): a part of the retina that is directly opposite the pupil and contains only cone cells. It is responsible for good visual acuity (good resolution).

A blind despite where the bundle of sensory fibres from the optic nerve Despite in, principal no light-sensitive receptors.

Vitreous Humour: is a transparent, jelly-like mass located behind the lens. It acts as a 'suspension' for the lens so that the delicate lens is not damaged. It helps to maintain the shape of the posterior chamber of the eyeball.

Aqueous Humour: helps to maintain the shape of the anterior chamber of the eyeball.

2.3 Visual Processing

The ability to see depends on how well these parts work together. Light rays bounce off all objects. If a person Is looking at a particular object, such as a tree, light is reflected off the tree to the person's eye and enters the eye through the cornea (the clear, transparent portion of the coating that surrounds the eyeball) [6].

Next, light rays pass through an opening in the iris (colour part of the eye), called the pupil. The iris controls the amount of light entering the eye by dilating or constricting the pupil. In bright light, for example, the pupil shrinks to the size of a pinhead to prevent too much light from entering. In the dim light, the pupil enlarges to allow more light to enter the eye [7].

The light then reaches the crystalline lens. The lens focuses light rays onto the retina by bending (refracting) them. The cornea does most of the refraction and the crystalline lens fine-tunes the focus. In a healthy eye, the lens can change its shape (accommodate) to provide clear vision at various distances. If an object is close, the ciliary muscle of the eye contract, and the lens becomes rounder. To see a distant object, the same muscle relaxes and the lens flattens [8].

Behind the lens and in front of the retina is a chamber called the vitreous body, which contains a clear, gelatinous fluid called the vitreous humour? Light rays pass through the vitreous before reaching the retina. The retina lines the back two-thirds of the eye and is responsible for the wide field of vision that most people experience. For clear vision, light rays must focus directly on the retina. When light focuses in front or behind the retina, the result is blurry vision [9].

The retina contains millions of specialized photoreceptor cells called rods and cones that convert light rays into electrical signals that are transmitted to the brain through the optic nerve. Rods and cones provide the ability to see in dim light and see in colour, respectively [10].

The macula, located in the canter of the retina, is where most of the cone cells are located. The fovea, a small depression in the canter of the macula, has the highest concentration of cone cells. The macula is responsible for central vision, seeing colour, and distinguishing fine detail. The outer portion (peripheral retina) is the primary location of rod cells and allows for night vision and seeing movement and object to the side (i.e., peripheral vision) [11].

The optic nerve, located behind the retina, transmits signals from the photoreceptor cells to the brain. Each eye transmits signals of a slightly different image are inverted. Once they reach the brain a corrected and combined into one image. This complex process of analysing data transmitted through the optic nerve is called visual processing [12].

2.4 Visual Impairment and Blindness

The World Health Organization (WHO) defines **Visual impairment** decrease or severe reduction in vision that cannot be corrected with standard glasses or contact lenses and reduce an individual's ability to function at a specific or all task [13].

Blindness is severe sight loss, where a person is unable to see clearly how many fingers are being held up at a distance of 3m (9.8 feet) or less, even when they are wearing glasses or contact lenses. However, someone who is blind may still have some degree of vision [13].

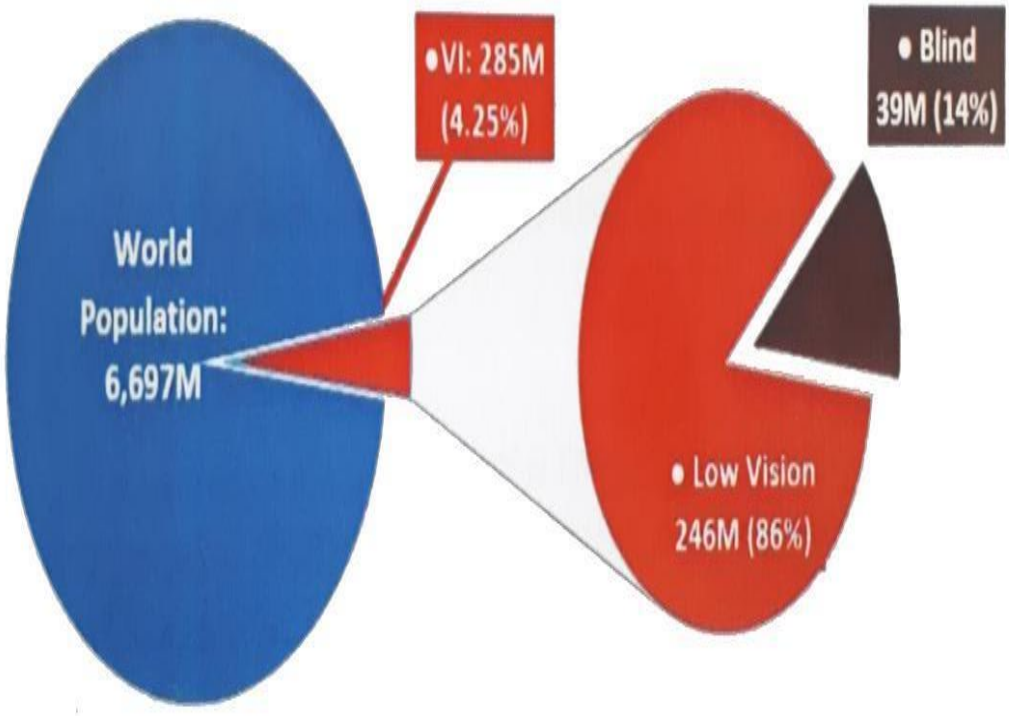


Figure (2.2): Global estimate of visual impairment [13].

According to WHO criteria, a global estimate predicts that there are 285 million people with visual impairment approximately are blind.

Most people (87%) who are visually impaired live in developing countries. In developing countries, cataracts (a cloudy area that forms in the lens of the eye) are responsible for most cases of blindness (48%). visual impairment usually affects older people. Globally, women are

more at risk than men. With the right treatment, about 85% of visual impairment cases are avoidable, and approximately 75% of all blindness can be treated or prevented [13].

2.4.1 Causes of Visual Impairment and Blindness

Despite in, principal the progress made in surgical techniques in many countries during the last ten years, cataract (47.9%) remains the leading cause of visual impairment in all areas of the world, except for developed countries.

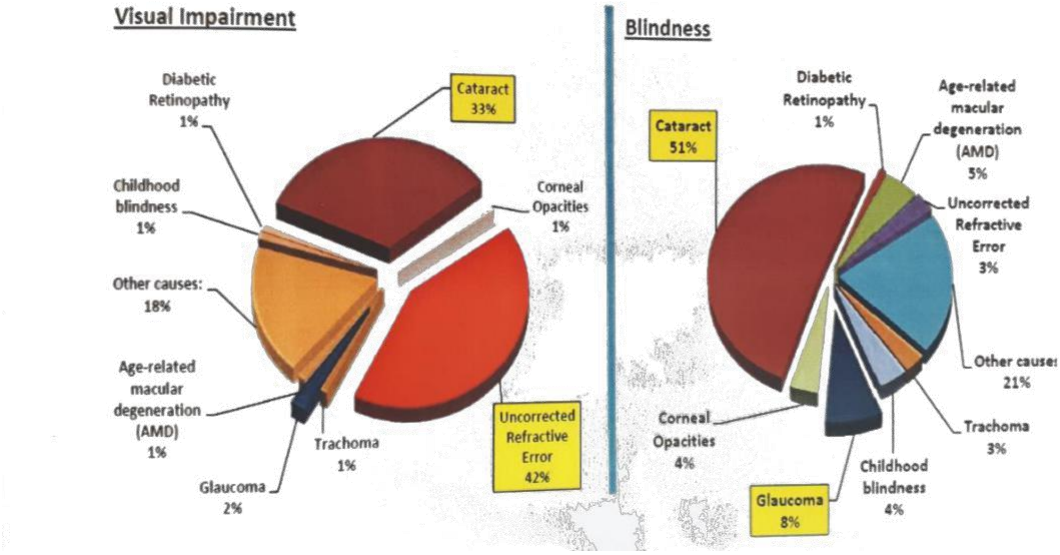


Figure (2.3): Causes of Visual Impairment and Blindness [13]

Other main causes of visual impairment on 2010 are glaucoma (2%), age-related macular degeneration (AMD) (1%), corneal opacities (1%), diabetic retinopathy (4.8%), childhood blindness (1%), trachoma (1%), and onchocerciasis (0.8%). The causes of avoidable visual impairment worldwide are all the above except for AMD. In the least-developed countries, and in particular Sub-Saharan Africa, the causes of avoidable blindness are primarily, cataract (51%), glaucoma (8%), corneal opacities (4%), trachoma (3%), childhood blindness (4%) and onchocerciasis (1%) [13].

Looking at the global distribution of the avoidable blindness based on the population in each of the WHO regions, we see the following: South Asian 28%, Western Pacific 26%, African 16.6%, Eastern Mediterranean 10%, the American 9.6%, and European 9.6% [13].

In addition to uncorrected refractive errors, these six diseases or groups of diseases which have effective known strategies for their elimination, make up the targets of the WHO Global

Initiative to Eliminate Avoidable Blindness, “VISION 2020: The Right to Sight”, which aims to

Eliminate these causes as a public health problem by the year 2020. Cataract, onchocerciasis, and trachoma are the principle diseases for which world strategies and programmers have been developed. For glaucoma, diabetic retinopathy, uncorrected refractive errors, and childhood blindness (except for exophthalmia), the development of screening and management strategies for use at the primary care level is ongoing at WHO [13]

CHAPTER THREE

3

Computer vision and Image Processing

3.1 Human vision

3.2 Computer vision

3.2.1 Human Vision VS Computer Vision

3.2.2 Main goal of computer vision

3.2.3 Advantages and Disadvantages of computer vision

3.2.4 Applications of Computer Vision

3.3 Levels of Computer vision

3.4 Fundamental steps in digital image processing

3.4.1 Image Acquisition

3.4.2 Enhancement Image Processing

3.4.3 Restoration Image Processing

3.4.4 Colour Image Processing

3.4.5 Wavelets and multi resolution processing

3.4.6 Compression Image Processing

3.4.7 Morphological Image Processing

3.4.8 Segmentation Image Processing

3.4.9 Representation & description
3.4.10 Object Recognition

3.1 human vision

Vision is the process of discovering what is present in the world and where it is by looking the human visual system can be regarded as consisting of two parts. The eyes act as image receptors that capture light and convert it into signals which are then transmitted to image processing canters in the brain. These canters process the signals received from the eyes and build an internal "picture" of the scene being viewed. Processing by the brain consists partly of simple image processing and partly of higher functions that build and manipulate an internal model of the outside world. Although the division of function between the eyes and the brain is not clear-cut, it is useful to consider each of the components separately [24].

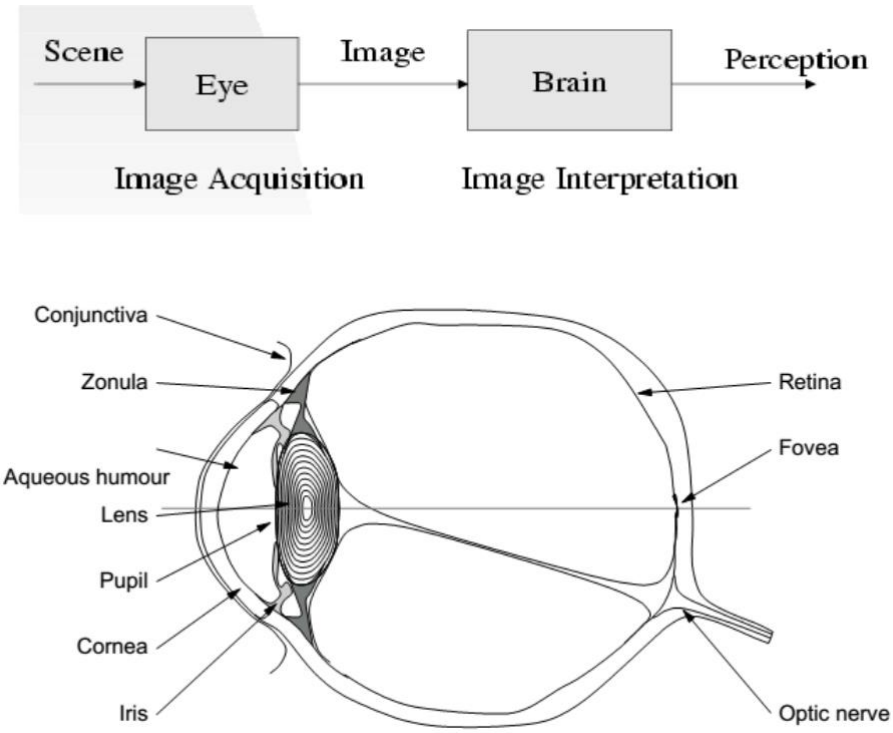


Figure (3.1): A cross-section of the right human eye, viewed from above [24].

3.2 Computer vision

Computer vision is an interdisciplinary field that deals with how computers can be made for gaining a high-level understanding from digital images or videos. From the perspective of engineering, it seeks to automate tasks that the human visual system can do [14] [15] [16].

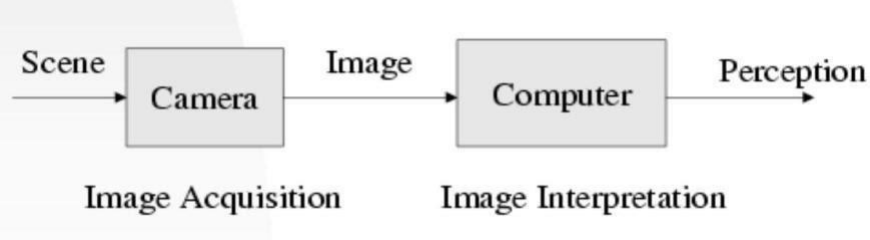


Figure (3.2): Computer vision similar to those of humans [24].

Computer vision tasks include methods for acquiring, processing, analysing, and understanding digital images, and extraction of high-dimensional data from the real world to produce numerical or symbolic information, e.g., in the forms of decisions[17][18][19][20].

Understanding in this context means the transformation of visual images (the input of the retina) into descriptions of the world that can interface with other thought processes and elicit appropriate action. This image understanding can be seen as the disentangling of symbolic information from image data using models constructed with the aid of geometry, physics, statistics, and learning theory [21]

3.2.1 Human Vision VS Computer Vision

In object classification, it was known that the human brain tactics visible know-how in a semantic area traditionally, i.e. extracting the semantically significant elements equivalent to line segments, shape, boundaries, etc. In any case, by late data handling methods, these sorts of components can't be recognized by PCs heartily so that in PC vision it's still hard to prepare visual data as people do. PCs need to prepare visual data in information space framed by the vigorously distinguishable yet less important components, for example, hues, surfaces, and so on. In this way, the handling philosophy in PCs is entirely not the same as that in individuals [29].

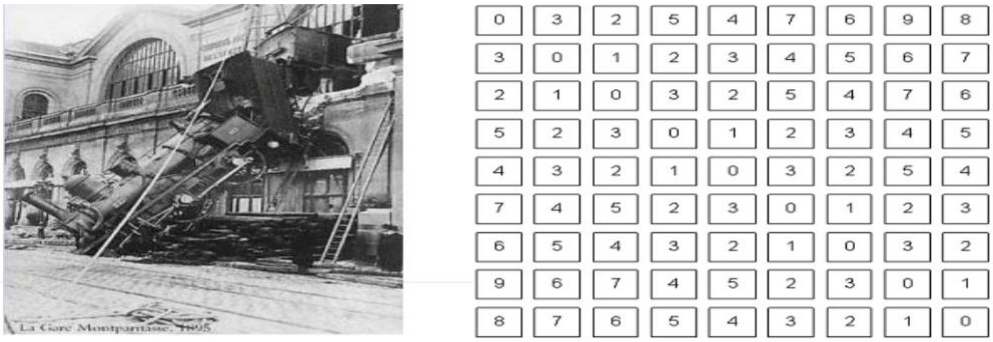


Figure (3.3): What we see what the computer sees [31].

Computer vision is an interdisciplinary field that deals with how computers can be made for gaining a high-level understanding from digital images or videos. From the perspective of engineering, it seeks to automate tasks that the human visual system can do [14] [14] [16]. "Computer vision is concerned with the automatic extraction, analysis, and understanding of useful information from a single image or a sequence of images. It involves the development of a theoretical and algorithmic basis to achieve automatic visual understanding."[22] As a scientific discipline, computer vision is concerned with the theory behind artificial systems that extract information from images. The image data can take many forms, such as video sequences, views from multiple cameras, or multi-dimensional data from a medical scanner. [23] As a technological discipline, computer vision seeks to apply its theories and models for the construction of computer vision systems.

3.2.2 Main goal of computer vision:

Computer Vision has a dual goal. From the biological science point of view, computer vision aims to come up with computational models of the human visual system. From the engineering point of view, computer vision aims to build autonomous systems which could perform some of the tasks which the human visual system can perform (and even surpass it in many cases). Many vision tasks are related to the extraction of 3D and temporal information from time-varying 2D data such as obtained by one or more television cameras, and more generally the understanding of such dynamic scenes. Of course, the two goals are intimately related. The properties and characteristics of the human visual system often give inspiration to engineers who are designing computer vision systems. Conversely, computer vision algorithms can offer insights into how the human visual system works. In this paper, we shall adopt the engineering point of view [28].

3.2.3 Advantages and Disadvantages of computer vision [29]

Advantages of computer vision:

Price discount: - Time is saved on people and devices, therefore misguided merchandise is eliminated. Recovery and motion recovery. Without exaggeration, image matching is part of the base for vision [24].

Optical flow is a kind of image observation of motion, but it is not true motion. Since it only measures the optical changes in images, an aperture problem is unavoidable. But based on optical flows, camera motion or object motion could be estimated [34].

2. Middle-level Vision:

There are two major aspects in middle-level vision: (1) inferring the geometry and (2) inferring the motion. These two aspects are not independent but highly related. A simple question is “can we estimate geometry based on just one image?” The answer is obvious. We need at least two images. They could be taken from two cameras or come from the motion of the scene [34].

Some fundamental parts of geometric vision include multi-view geometry, stereo, and structure from motion (SFM), which fulfil the step from 2D to 3D by inferring 3D scene information from 2D images. Based on that, geometric modelling is to construct 3D models for 6 objects and scenes, such that 3D reconstruction and image-based rendering could be made possible [29].

Another task of middle-level vision is to answer the question “how the object moves”. Firstly, we should know which areas in the images belong to the object, which is the task of image segmentation. Image segmentation has been a challenging fundamental problem in computer vision for decades. Segmentation could be based on spatial similarities and continuities. However, uncertainty cannot be overcome for a static image. When considering motion continuities, we hope the uncertainty of segmentation could be alleviated. On top of that is visual tracking and visual motion capturing, which estimate 2D and 3D motions, including deformable motions and articulated motions [34]?

3. High-level Vision:

High-level vision is to infer the semantics, for example, object recognition and scene understanding. A challenging question in many decades is that how to achieve invariant recognition, i.e., recognize 3D objects from different view directions. There have been two approaches for recognition: model-based recognition and learning-based recognition. It is noticed that there was a spiral development of these two approaches in history even higher-level vision is image understanding and video understanding. We are interested in answering questions like “Is there a car in the image? Or is this video a drama or an action? Or is the person in the video jumping? Based on the answers to these questions, we should be able to fulfil different tasks in intelligent human-computer interaction, intelligent robots, smart environment, and environment-based multimedia [34].

3.4 Fundamental steps in digital image processing

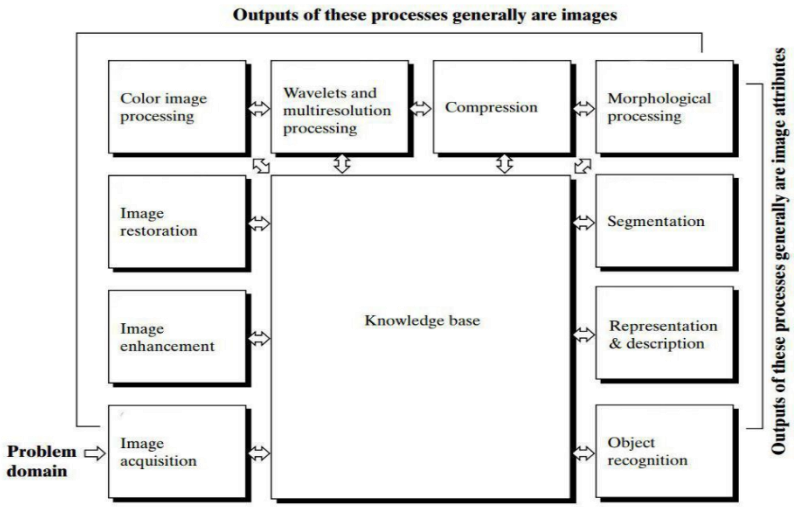


Figure (3.4): Fundamental steps in digital image processing [40].

3.4.1 Image Acquisition

Before any video or image processing can commence an image must be captured by a camera and converted into a manageable entity. This is the process known as acquisition. The image acquisition process consists of two steps; energy reflected from the object of interest, an optical system that focuses the energy, and finally a sensor that measures the amount of energy. In Fig. 3.1 the three steps are shown for the case of an ordinary camera with the sun as the energy source.

Energy to capture an image a camera requires some sort of measurable energy. The energy of interest in this context is light or more generally electromagnetic waves. An electromagnetic (EM) wave can be described as a massless entity, a photon, whose electric and magnetic fields vary sinusoidal, hence the name wave. The photon belongs to the group of fundamental particles and can be described in three different ways [35].

$$\lambda = c/f, \quad E = h \cdot f \quad \Rightarrow \quad E = h \cdot c\,\lambda \qquad \text{.....(3.1)}$$

A photon can be described by its energy E, which is measured in electron volts [eV]

A photon can be described by its frequency f, which is measured in Hertz [Hz]. A frequency is the number of cycles or wave-tops in one second

A photon can be described by its wavelength λ, which is measured in meters [m].

A wavelength is the distance between two wave-tops the three different notations are connected through the speed of light c and Planck's constant h.

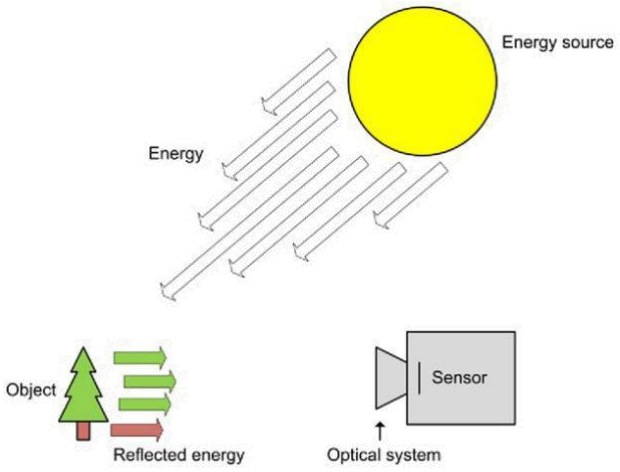


Figure (3.5): Overview of the typical image acquisition process, with the sun as light source, a tree as an environment digital camera to capture the image [35].

The range from approximately 400–700 nm (nm = nanometre = 10−9) is denoted by the visual spectrum. The EM waves within this range are those your eye (and most cameras) can detect. This means that the light from the sun (or a lamp) in principle is the same as the signal used for transmitting TV, radio or for mobile phones, etc. The only difference, in this context, is the fact that the human eye can sense EM waves in this range and not the waves used e.g., radio. Or in other words, if our eyes were sensitive to EM waves with a frequency around 2 · 109 Hz, then your mobile phone would work as a flashlight, and big antennas would be perceived as "small suns". Evolution has (of course) not made the human eye sensitive to such frequencies but rather to the frequencies of the waves coming from the sun, hence visible light [35].

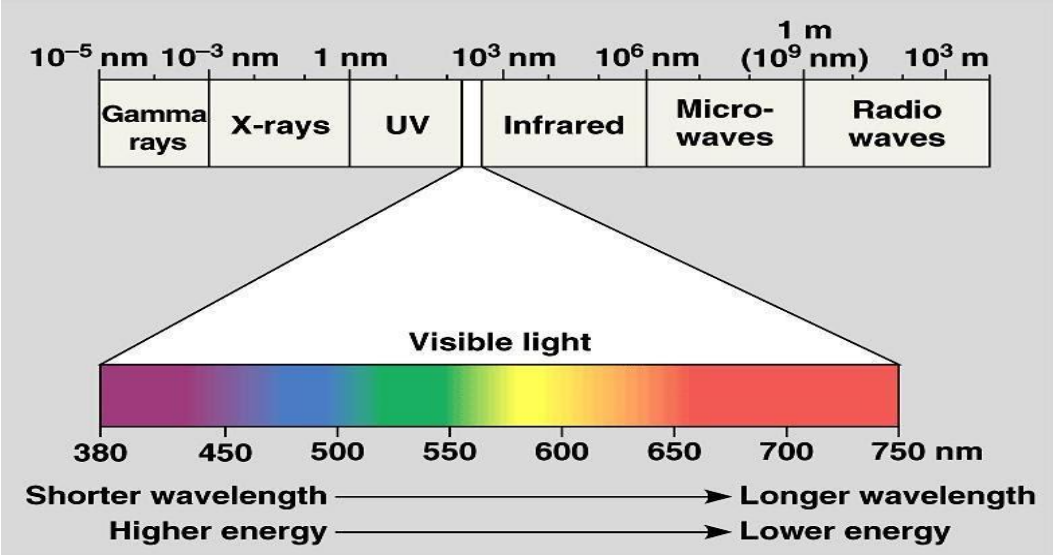


Figure (3.6): The visual spectrum [35].

3.4.2 Enhancement Image Processing

The principal objective of enhancement is to process an image so that the result is more suitable than the original image for a specific application. Regardless of the method used, however, image enhancement is one of the most interesting and visually appealing areas of image processing.

Image enhancement approaches fall into two broad categories: The term spatial domain refers to the image plane itself, and approaches in this category are based on the direct manipulation of pixels in an image. Frequency domain processing techniques are based on modifying the Fourier transform of an image

There is no general theory of image enhancement. When an image is processed for visual interpretation, the viewer is the ultimate judge of how well a particular method works. Visual evaluation of image quality is a highly subjective process, thus making the definition of a "good image" an elusive standard by which to compare algorithm performance. When the problem is one of processing. Images for machine perception, the evaluation task is somewhat easier [35].

"Good image" an elusive standard by which to compare algorithm performance. When the problem is one of processing. Images for machine perception, the evaluation task is somewhat easier [35].

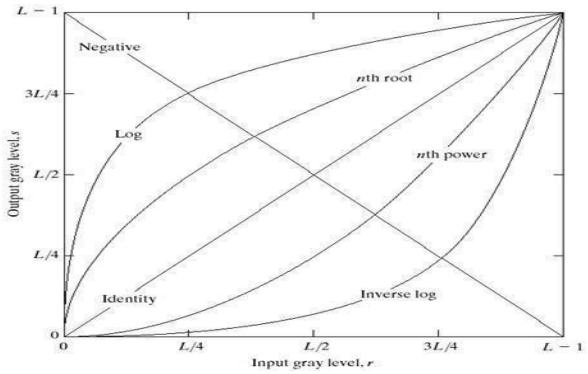


Figure (3.7): Some basic gray-level transformation functions used for image enhancement [35]

We begin the study of image enhancement techniques by discussing gray-level transformation functions. These are among the simplest of all image enhancement techniques. The values of pixels, before and after processing, will be denoted by *r* and *s*,

Respectively. As indicated in the previous section, these values are related by an expression of the form: s=T(r), where T is a transformation that maps a

Pixel value *r* into a pixel value *s*. Since we are dealing with digital quantities, values of the transformation function typically are stored in a one-dimensional array and the mappings from *r* to *s* are implemented via table lookups. For an 8-bit environment, a lookup table containing the values of *T* will have 256 entries. As an introduction to gray-level transformations, consider Fig. 3.3, which shows three basic types of functions used frequently for image enhancement: linear (negative and identity transformations), logarithmic (log and inverse-log transformations), and power-law (*n*th power and *n*th root transformations) [35].

3.4.3 Restoration Image Processing

Image Restoration techniques aim at modelling a degradation corrupting the image and inverting this degradation to correct the image so that it is as close as possible to the original.

Image restoration attempts to restore images that have been degraded.

Identify the degradation process and attempt to reverse it.

Similar to image enhancement, but more objective.

The sources of noise in digital images arise during image acquisition (digitization) and transmission.

Imaging sensors can be affected by ambient conditions.

Interference can be added to an image during transmission.
We can consider a noisy image to be modelled as follows:

g(x, y) f (x, y) (x, y)…….. (3.5)

where *f (x, y)* is the original image pixel, *η (x, y)* is the noise term and *g (x, y)* is the resulting noisy pixel If we can estimate the model of the noise in an image, this will help us to figure out how to restore the image [36] [37] [38].

The Noise Sources

Where *f (x, y)* is the original image pixel, *η (x, y)* is the noise term and *g (x, y)* is the resulting noisy pixel if we can estimate the model of the noise in an image, this will help us to figure out how to restore the image.

The principal sources of noise in digital images arise during image acquisition and/or transmission.

Filtering to Remove Noise

We can use spatial filters of different kinds to remove different kinds of noise
The arithmetic mean filter is a very simple one and is calculated as follows:

$f'(x,y)=1/mn\Sigma g(s,t)$…….……(3.6)

This is implemented as the simple smoothing filter blurs the image to remove noise.

3.4.4 Color Image Processing

Color Image Processing is divided into two major areas:

Full-colour processing

Images are acquired with a full-colour sensor, such as a colour TV camera or colour scanner.

Used in publishing, visualization, and the Internet

Pseudo colour processing

Assigning a colour to a particular monochrome intensity or range of intensities.

Visible light as a narrow band of frequencies in EM

A body that reflects light that is balanced in all visible wavelengths appears white

However, a body that favours reflectance in a limited range of the visible spectrum exhibits some shades of colour

Green objects reflect wavelengths in the 500 nm to 570 nm range while absorbing most of the energy at other wavelengths [40].

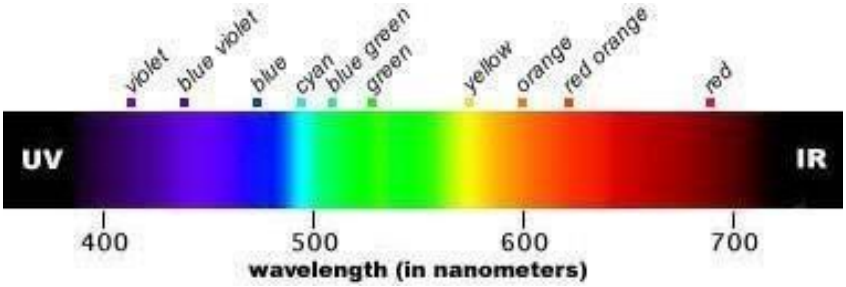


Figure (3.8): wavelength (in nanometres)[40].

3.4.5 Wavelets and multi resolution processing

Wavelets are mathematical functions that split up data into different frequency components, and then study each component with a resolution matched to its scale.

Wavelet transform decomposes a signal into a set of basic functions. These basis functions are called "wavelets "[41]. Use wavelet for:

-Good approximation properties.
-Efficient way to compress the smooth data except in localized region.

Easy to control wavelet properties.
(Example: Smoothness, better accuracy near sharp gradients).

Wavelets are a powerful statistical tool that can be used for a wide range of applications:

Signal processing. - Image processing.-Smoothing and image demonising. - Speech recognition.
The advantage of wavelet compression is that, in contrast to JPEG, the wavelet algorithm does not divide the image into blocks, but analyzes the whole image. Wavelet transform is applied sub-images, so it produces no blocking artifacts. Wavelets have the great advantage of being able to separate the fine details in a signal. Wavelet allows getting the best compression ratio while maintaining the quality of the images [41].

Image compression using wavelet transforms results in an improved compression ratio as well as image quality. Wavelet transform is the only method that provides both spatial and frequency

domain information. These properties of the wavelet transform greatly help in the identification and selection of significant and no significant coefficients. Wavelet transform techniques currently provide the most promising approach to high-quality image compression [41].

3.4.6 Compression Image Processing

In terms of storage, the capacity of a storage device can be effectively increased with methods that compress a body of data on its way to storage. device and decompresses it when it is retrieved. In terms of communications, the bandwidth of a digital communication link can be effectively increased by compressing data at the sending end and decompressing data at the receiving end. At any given time, the ability of the Internet to transfer data is fixed. Thus, if data can effectively be compressed wherever possible, significant improvements in data throughput can be achieved. Many files can become binned into one compressed document making sending easier [38].

Image Compression is the art and science of reducing the amount of data required to represent an image. A technique used to reduce the volume of information to be transmitted about an image.

The Flow of Image Compression Coding

What is the so-called image compression coding? Image compression coding is to store the image into bit-stream as compact as possible and to display the decoded image on the monitor as exact as possible. Now consider an encoder and a decoder as shown in Fig. 3.11 When the encoder receives the original image file, the image file will be converted into a series of binary data, which is called the bit-stream. The decoder then receives the encoded bit-stream and decodes it to form the decoded image. If the total data quantity of the bit-stream is less than the total data quantity of the original image, then this is called image The full compression flow is as shown **in Fig. 3.9**

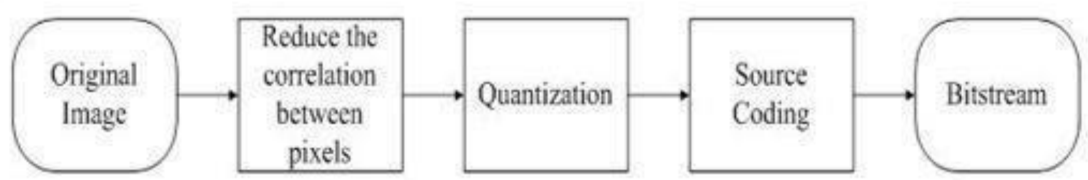


Figure (3.9): The general encoding flow of image compression [38].

The compression ratio is defined as follows:

Cr=n1/n2 ……………………………………………………… (3.6)

Where *n1* is the data rate of the original image and *n2* is that of the encoded bit-stream.

3.4.7 Morphological Image Processing

Binary images may contain numerous imperfections. In particular, the binary regions produced by simple thresholding are distorted by noise and texture. Morphological image processing pursues the goals of removing these imperfections by accounting for the form and structure of the image. These techniques can be extended to greyscale images [43].

Basic concepts

Morphology branch in biology that deals with the form and structure of animals and plants. [40]

Mathematical Morphology is 0,1 The main compacts A number forThe main extracting image components, that are useful in the representation and description of region shape, and The language of mathematical morphology is Set theory [40].

Table (3.1): Set Theory [40].

Subset	⊆
Union	∪
Intersection	∩
Disjoint / mutually exclusive	∩ =∅
Complement	≡{ | ∉ }
Difference	− ≡{ | ∈ , ∉ = ∩ }
Reflection	≡{ | =− ,∀ ∈ }
Translation	( ) ≡{ | = + ∀ ∈ }

In binary images, the set elements are members of the 2-D integer space, where each element (x, y) is a coordinate of a black (or white) pixel in the image. [40]

Morphological techniques probe an image with a small shape or template called a **structuring element**, the structuring element is positioned at all possible locations in the image and it is compared with the corresponding neighbourhoods of pixels. Some operations test

Whether the element "fits" within the neighbourhoods, while others test whether it "hits" or intersects the neighbourhoods:

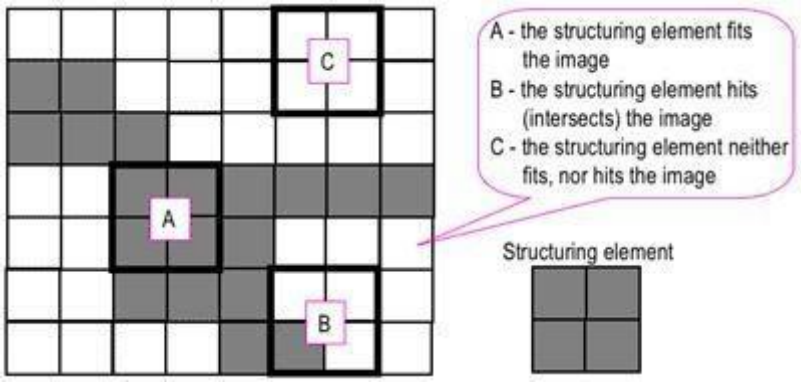


Figure (3.10): Probing of an image with a structuring element[40].

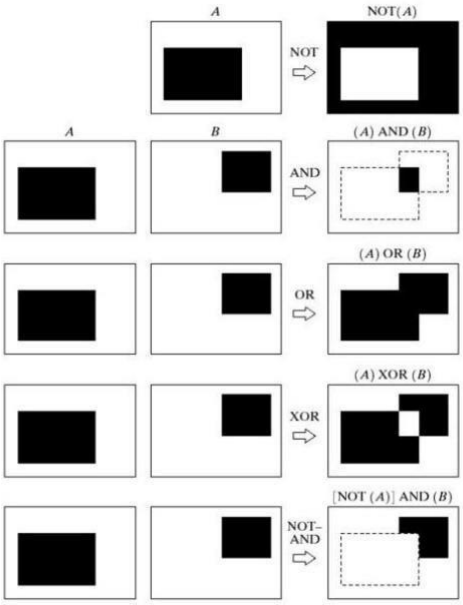


Figure (3.11): Some logic operations between binary images. Black represents binary 1s and white binary 0s in this example. [40]

A morphological operation on a binary image creates a new binary image in which the pixel has a non-zero value only if the test is successful at that location in the input image.

The **structuring element** is a small binary image, i.e. a small matrix of pixels, each with a value of zero or one: [43]

The matrix dimensions specify the *size* of the structuring element.
The pattern of ones and zeros specifies the *shape* of the structuring element.

An *origin* of the structuring element is usually one of its pixels, although generally, a main similar regions main number of a compacts the origin can be outside the structuring element.

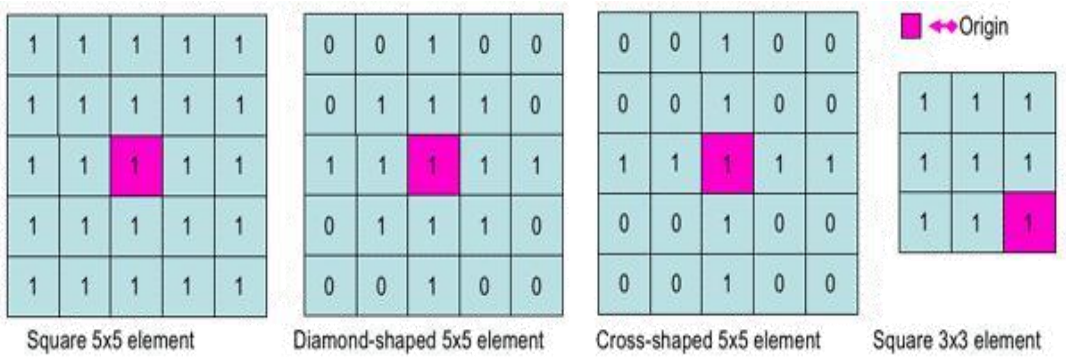


Figure (3.12): Examples of simple structuring elements. [43]

A common practice is to have odd dimensions of the structuring matrix and the origin defined as the canter of the matrix. Structuring elements play in morphological image processing the same role as convolution kernels in linear image filtering [43].

Morphological filtering

of a binary image is conducted by considering compound operations like opening and closing as filters. They may act as filters of shape. For example, opening with a disc structuring element smooth's corners from the inside, and closing with a disc smooth's corners from the outside. But also these operations can filter out from an image any details that are smaller in size than the structuring element, e.g. opening is filtering the binary image at a scale defined by the size of the structuring element. Only those portions of the image that fit the structuring element are passed by the filter; smaller structures are blocked and excluded from the output image. The size of the structuring element is most important to eliminate noisy details but not to damage objects of interest [40].

Boundary Extraction

First, erode A by B, then make set difference between A and the erosion The thickness of the contour depends on the size of constructing object –:

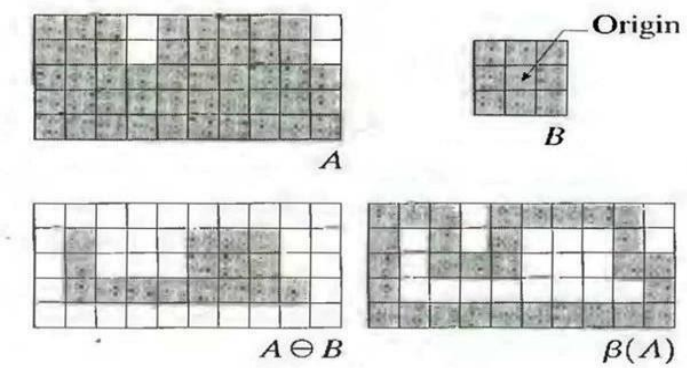


Figure (3.13): Boundary Extraction using logic Theory [40].

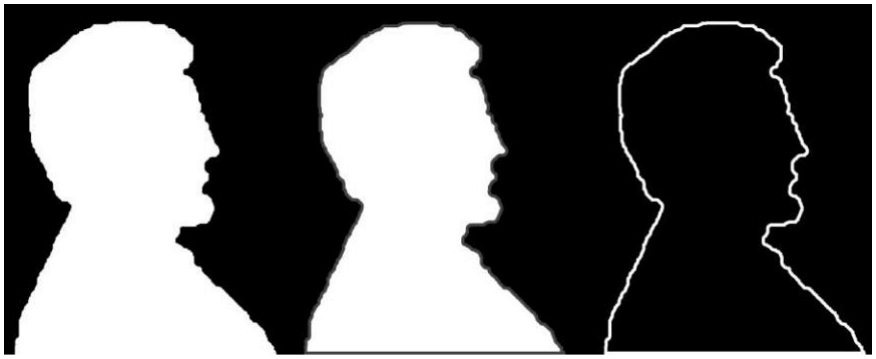


Figure (3.14): Example of Boundary Extraction [40].

3.4.8 Segmentation Image Processing

Image analysis:

Segmentation, i.e. subdivision of the image into its constituent parts or objects. Autonomous segmentation is one of the most difficult tasks in image processing, Segmentation algorithms are based on two basic properties of gray level values:

•**Discontinuity**: the image is partitioned based on abrupt changes in gray level. Main approach is edge detection.

•**Similarity**: partition an image into regions that are similar. Main approaches are thresholding, region growing, and region splitting and merging. [40]

Three basic types of discontinuities in digital images: Points, Lines, Edges.

−1	−1	−1	−1	−1	2	−1	2	−1	2	−1	−1
2	2	2	−1	2	−1	−1	2	−1	−1	2	−1
−1	−1	−1	2	−1	−1	−1	2	−1	−1	−1	2
Horizontal			+45°			Vertical			−45°		

Figure (3.15): The corresponding direction [40], Note: zero-sum masks

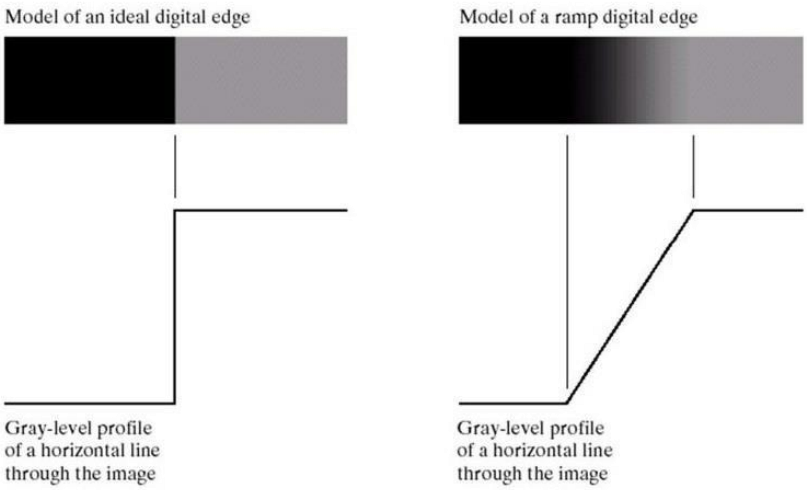


Figure (3.16): (a) Model of an ideal digital edge. (b)Model of ramp edge. The slope of the ramp is proportional to the degree of blurring in the edge [40].

Thresholding

Selecting features within a scene or image is an important prerequisite for most kinds of measurement or analysis of the scene. Traditionally, one simple way this selection has been accomplished is to define a range of brightness values in the original image, select the pixels within

This range as belonging to the foreground, and reject all of the other pixels to the background. Such an image is then usually displayed as a binary or two-level image, using black and white (or sometimes other colours) to distinguish the regions. There is no standard convention on whether the features of interest are white or black; the choice depends on the particular display hardware in use and the designer's preference; in the examples shown here, the features are black and the background is white, which matches most modern computer displays and printing that show black text on a white background[40].

3.4.9 Representation & description Image Processing

The segmentation techniques usually consider the pixel along a boundary and pixel contained in the region. And an approach to obtain the descriptor that is compact the data into representation.

The results of segmentation are a set of regions. Regions have then to be represented and described. Two main ways of representing a region, external characteristics (its boundary): focus on shape, internal characteristics (its internal pixels), focus on colour, textures. [32].

Chain Code

The chain code is used to represent a boundary by the length and the direction of straight-line segments. Typically, this representation is based on 4 or 8 connectivity of the segments. [32]

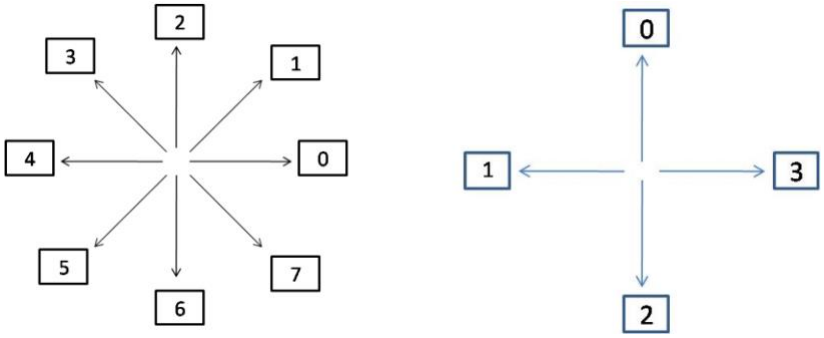


Figure (3.17): (a) 4-directional chain code (b) 8-directional chain code. **Merging Techniques**

Merging techniques based on average error or other criteria have been applied to the problem of polygonal approximation. The approach is to merge points along a boundary until the least square error line fit of the points merged so far exceeds a preset threshold.

Descriptor

In general, descriptors are some set of numbers that are produced to describe a given shape. The shape may not be entirely reconstructing able from the descriptors, but the descriptors for different shapes should be different enough that the shapes can be discriminated. [32].

Simple descriptors:

In general, descriptors are some set of numbers that are produced to describe a given shape. The shape may not be entirely reconstructing able from the descriptors, but the descriptors for different shapes should be different enough that the shapes can be discriminated. [32].

Simple descriptors:
Length

Number of pixels

A number of vertical and horizontal components + $\sqrt{2}$ times the number of diagonal components.

Diameter (length of the major axis).

Basic rectangle (formed by the major and the minor, axis encloses the boundary) and its eccentricity (major/minor axis).

1) Shape Numbers.

Order of a shape: the number of digits Shape numbers, the first difference of a chain-coded boundary depends on the starting point. The shape number of such a boundary, based on the 4-directional code is defined as the first difference of the smallest magnitude. For the desired shape order, we find the rectangle of order n whose eccentricity best approximates that of the basic rectangle and use this new rectangle to establish the grid size. [32]

2) Fourier Descriptors.

The Fourier descriptors are starting at an arbitrary point be treated as a complex number so that:
(x, y)
.Each coordinate pair can

$s(k) \quad x(k) jy(k)$..... (3.8)

Fourier descriptors are not insensitive to translation, but effects on the transform coefficients are known.

Typical Region Features:

Color

Mean RGB value.

Colour histograms in R, G, and colour histogram in (R, G, B).

Shape

The numbers of pixels.

Width and height attributes.

Boundary smoothness attributes.

Adjacent region labels.

3.4.10 Object Recognition

Common Image Descriptors for Detection

Descriptors encode local neighbouring window around key points

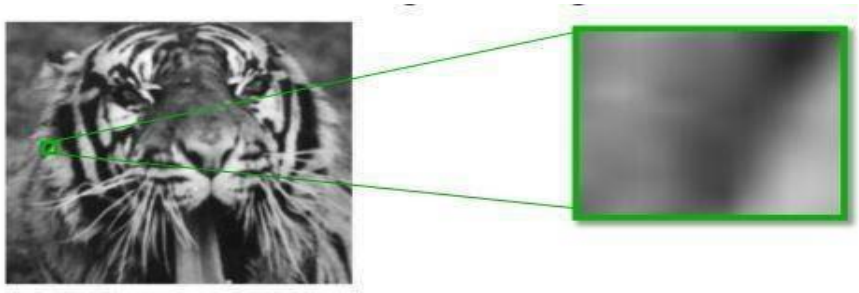


Figure (3.18): Example neighbouring window around key points [44].

- commonly descriptors in object detection try to capture gradient information [44].

– Human Perception is sensitive to gradient orientation – Invariant to changes in lighting and small deformations

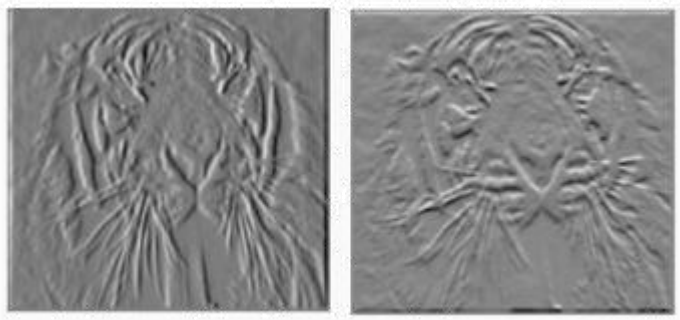


Figure (3.19): Capture gradient information [44].

Most common image descriptors currently used in object detection [44].

– Scale Invariant Feature Transform

– Histogram of Oriented Gradients

– many variants of these

Scale Invariant Feature Transform (SIFT) [44].

- Input an Image
- Extract Key points
  - Finds “corners”

– Determines scale and orientation of the key point • Compute Descriptor for each Key point

– Histogram of gradients in Gaussian window around key point

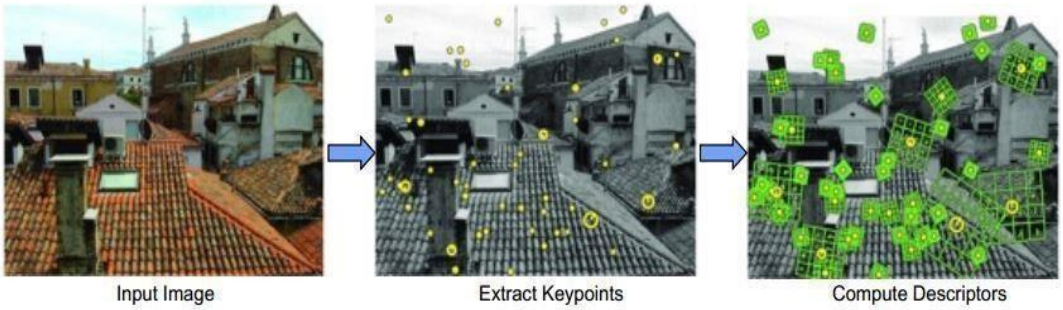


Figure (3.20): Scale Invariant Feature Transform [44].

• Compute the gradient for each pixel in local neighbouring window

– Typically 8 gradient directions

• 8x4x4=128 dimensional output vector normalized to 1

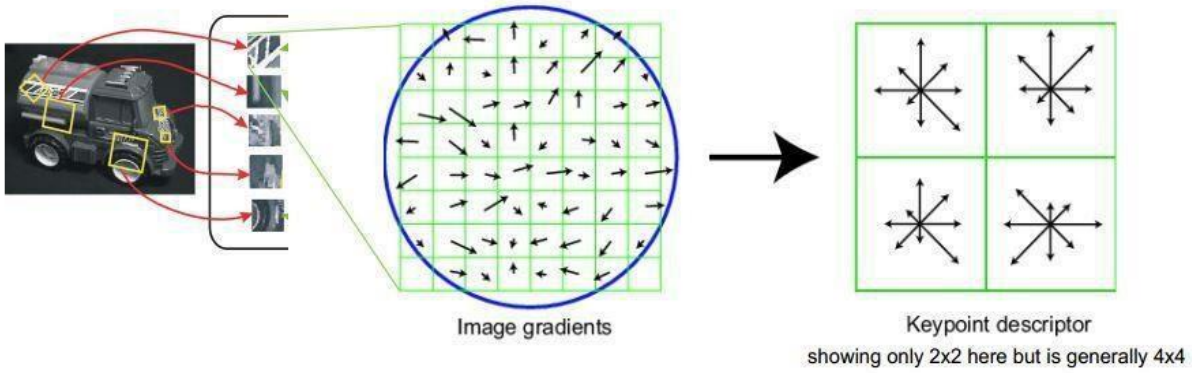


Figure (3.21): key point descriptor Scale Invariant Feature Transform [44].

• Match groups of key points several et al for the across images

– Invariant to scale and some changes in lighting and orientation • Great for finding the same instance of an object!

• not good at finding different instances of an object [44].

Histogram of Oriented Gradients (HOS) [44]

• Input an Image

• Normalize Gamma and Color

• Compute Gradients

• Accumulate weighted votes for gradient orientation over spatial bins • Normalize contrast within overlapping blocks of cells • Collect HOGs for all blocks over image

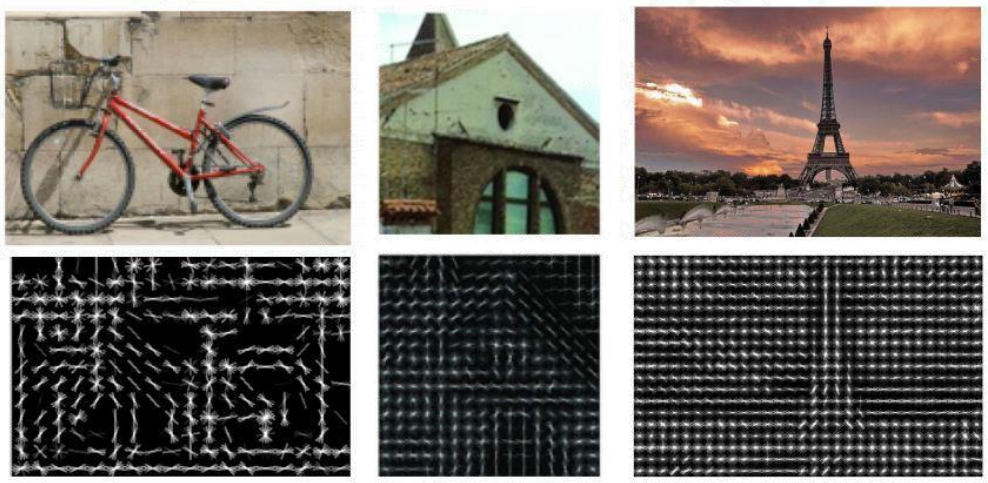


Figure (3.22): Example Histogram of Oriented Gradients [44].

Structure Model

Models an object as a number of smaller parts that are allowed to deviate slightly from average appearance. [44].

– Star model - coarse root and higher resolution part filters

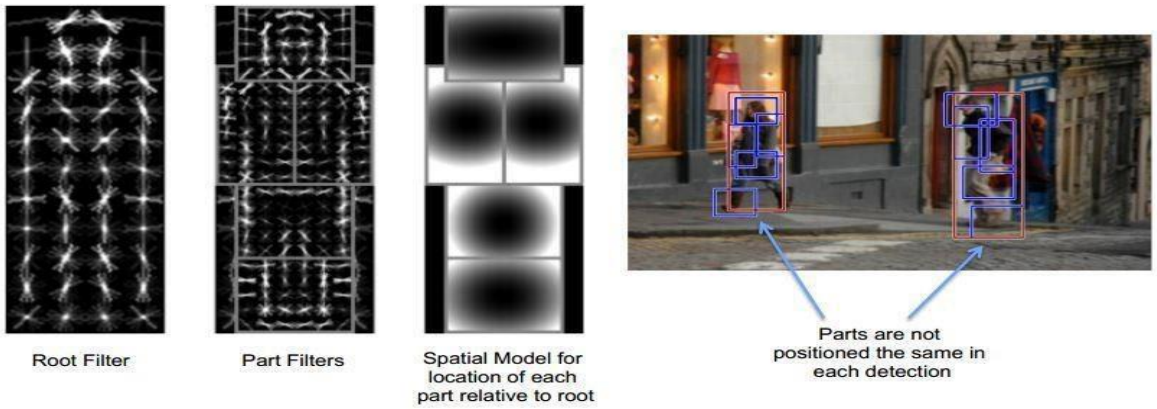


Figure (3.23): Part Based Model [44].

Voting Models

- Create weak detectors by using parts and voting for the objects canter location

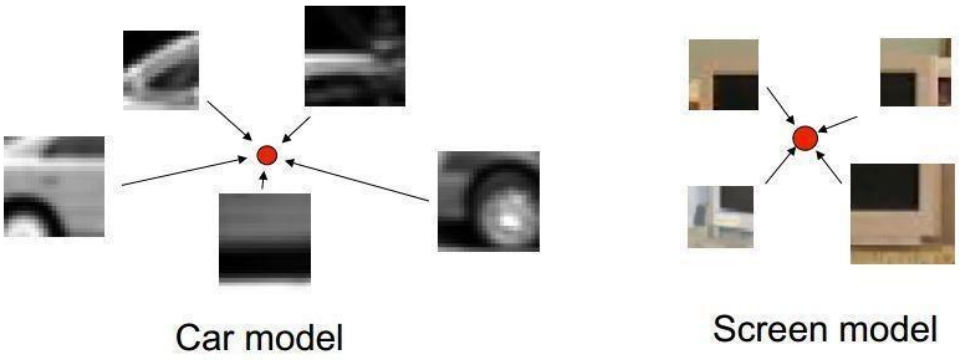


Figure (3.24): Voting Models [44].

Collecting Parts

First, we collect a set of part templates from a set of training objects.

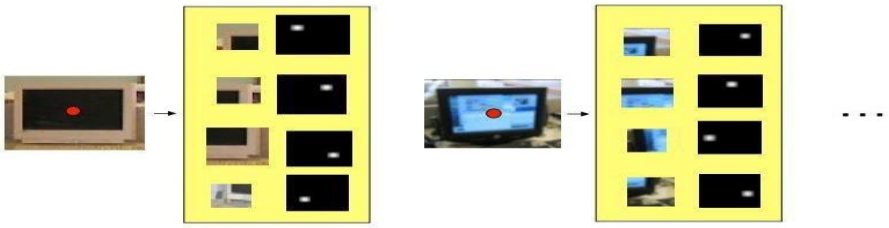


Figure (3.25): Example Collecting Parts [44].

Weak Part Detectors

-We now define a family of "weak detectors" as:

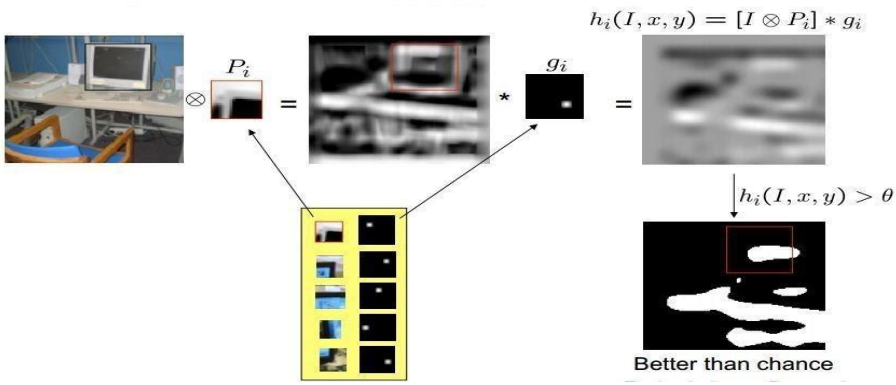


Figure (3.26): Weak Part Detectors [44].

-We can do a better job using filtered images

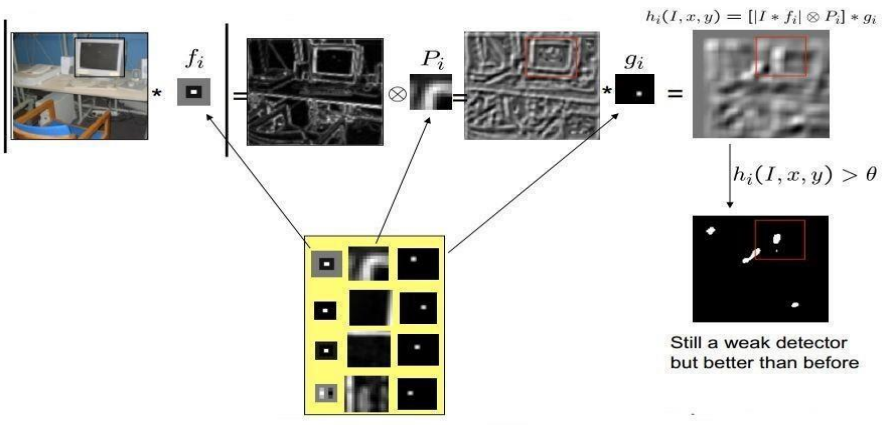


Figure (3.27): Weak Part Detectors using filtered images [44].

Voting Model

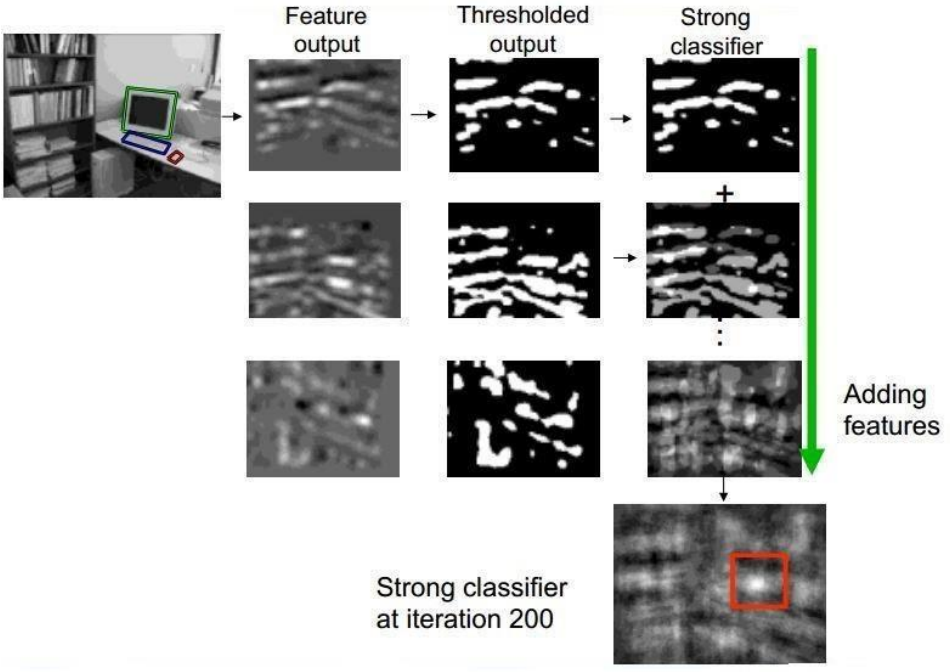


Figure (3.28): Example of Screen Detection [44]

Datasets for Object Classification Detection

- Caltech101
- Caltech256
- PASCAL
- ImageNET
- LabelMe

CHAPTER FOUR

4

System Design

4.1 Introduction

4.2 System Block Diagram

4.3 System Flow Chart

4.4 Special mini camera

4.5 Programming Language using Python

4.6 Raspberry pi 3

4.7 Dataset of image

4.8 Recognition

4.9 Power Supply

4.1 Introduction

One of the primary goals of computer vision is the understanding of visual scenes. Scene understanding involves numerous tasks including recognizing what objects are present, localizing the objects in 2D and 3D, determining the objects' and scene's attributes, characterizing relationships between objects, and providing a semantic description of the scene.

4.2 System Block Diagram

The main idea of the system block diagram is to work like the similar part eye in human of the person who is not blind and help blind people understand the object around his life by

Detection and recognition using camera and tell the result on earphone what's the camera can see.

This block can descript it.

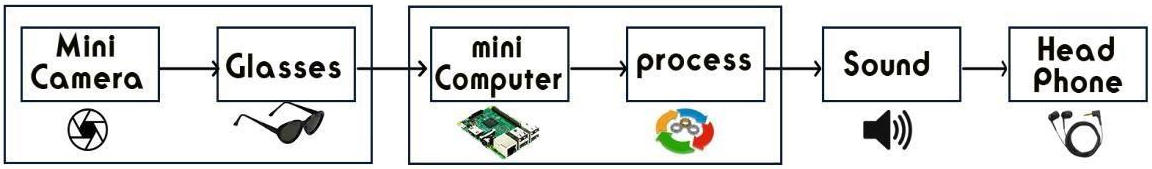
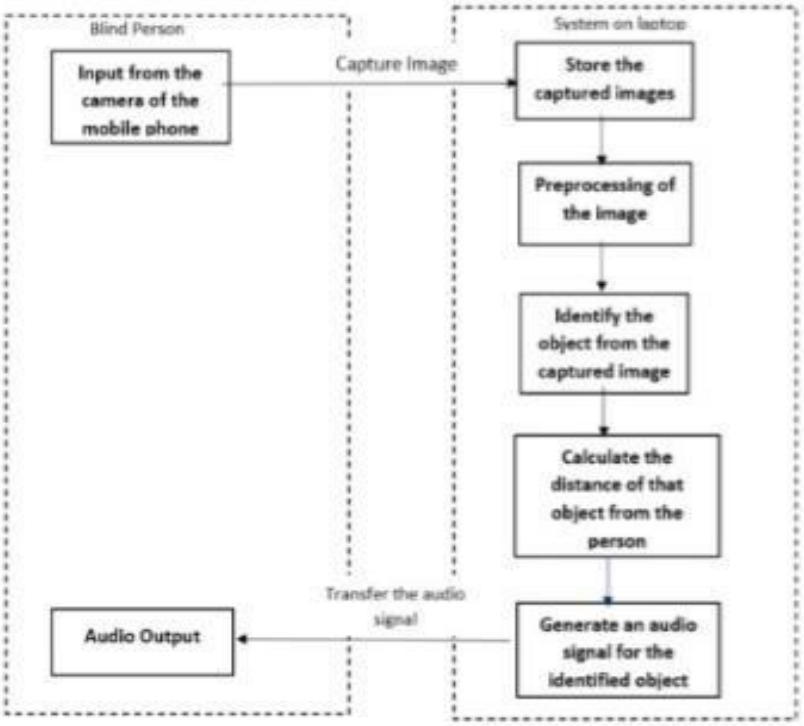


Figure (4.1): Block diagram of the project.



The camera is a box that controls the amount of light that reaches a light-sensitive surface inside (either film, a digital sensor, or another surface). The original cameras did not even have a glass lens, though today we can say that most cameras include: a light-tight box, a glass lens, and a surface that captures light.

The camera has come a long way from its humble beginnings, but it is still just a box that controls the amount of light that reaches a piece of film (or sensor). The camera has different types of body and size and shapes in this project we use a mini special camera that uses in surgical it very small and can but it easily on glasses and it is very good in low dark and low current it has 6 LEDs inside camera work in dark and high sensitivity and high pixel of an image to get high quality than is better for fast recognition and connect its USB cable to a computer.

in real-time camera as a Visual multimedia source that combines a sequence of images to form a moving picture. The video transmits a signal to a screen and processes the order in which the screen captures should be shown.

The main processing in my project to work as similar part of the brain to understand the object on what camera can see by matching into the database of images we use an app connected to web based processing and analysing when getting input sequence of the image by the camera then do method of image processing we talk it on chapter 3 and when finish the processing the minicomputer matching all sequence of the image with image net or database is stored in cloud storage and it detected using python programming language the result of output sound of an object by connecting earphone to tell

4.3 System Flow Chart

In this work, object recognition approach is presented by applying several steps: Object detection, creating a unique descriptor for each object, retrieving from the model database, and matching. A model database is a prerequisite stage required to be built to apply the matching process. It contains features for all common objects in the environments of the blind [58].

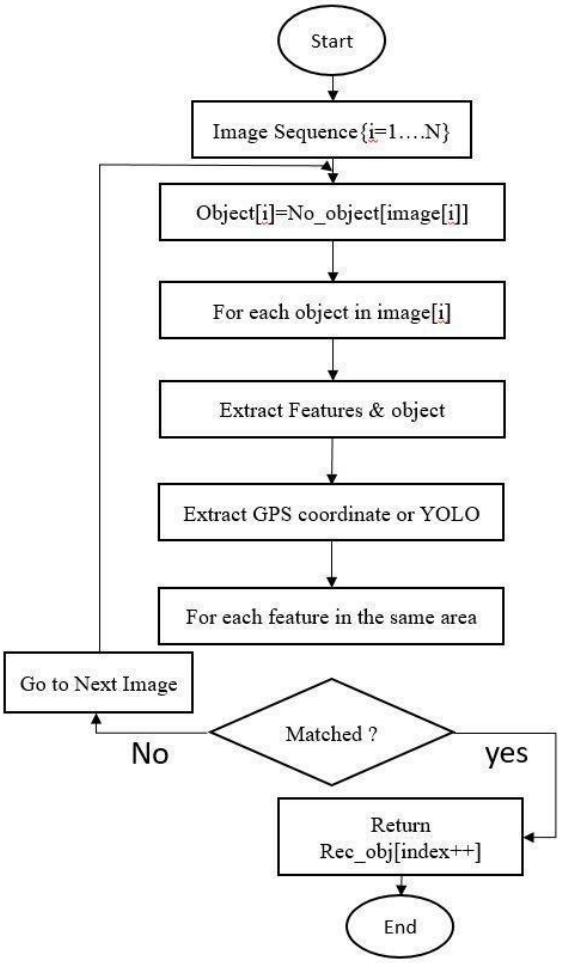


Figure (4.2): System Flow chart of the proposed method [58].

All objects that exist in a blind's environment are manually extracted and identified by the user to apply machine learning. The model database saves the features for each object, which is used later to apply the matching process. The extracted features are then saved in the database as shown in Table 4.1. To model's the use mismatches and computational time, GPS service is

Used to determine the location for each object. Hence the comparisons are only applied to the objects that exist in an involved area [58].

Table (4.1): Database structure for instances in the real world [58].

Object ID	Feature vector	GPS Coordinate
Chair	V-object1	(Latitude1,Longitude1)

Door	V-object1	(Latitude2,Longitude2)
….	…….	……
Object{N}	V-objectN	(LatitudeN,LongitudeN)

The database of models is created by applying the following: [58]

For each object in the input image: Do the next steps.

Extract the SURF features descriptor.

Get the GPS coordinate.

Identify the object by the user.

Save the extracted info.

Based on the models database, fast indexing is performed using the sign of the Laplacian for the underlying interest point and the GPS service to specify the involved areas. Typically, as performed in (Bay teal, 2006), the interest points are found at blob-type structures and the sign of Laplacian differentiates bright blobs on dark backgrounds from the reverse situation. This feature and the GPS area-based service are utilized to apply the proposed work at no extra computational cost. It should be noted that we only compare features if they have the same type of contrast and in same location. Therefore, this information allows faster matching and provides a slight increase in the performance as shown in the flow chart. [58]

4.4 Special Mini Camera

MISUMI is specialized in making customized designs and modifications of our products to meet your specific needs. Mishmi R&D team is equipped with a "Rapid Prototyping (RP) Machine" to custom-make your sample as quickly as 1 day (excluding shipping time). We are also equipped with PADS software & Printed Circuit Board Plotter, which are instrumental in accelerating the process of designing, engineering, producing, and testing our ongoing new products. In addition, a T.Q.M. programmer has been implemented to ensure the highest quality standard at all times. [56]

Figure (4.3): Special mini camera [57].

Properties of a mini camera:

Low dark current for low-light.
Conditions.

High sensitivity.

High performance.

Full HD image pixel.

Video cameras are used primarily in two modes. The first, characteristic of much early broadcasting, is live television, where the camera feeds real-time images directly to a screen for immediate observation. A few cameras still serve live television production, but most live connections are for security, military/tactical, and industrial operations where surreptitious or remote viewing is required. In the second mode, the images are recorded to a storage device for archiving or further processing and we use this camera in the figure is suitable for processing and progress scan. [56]

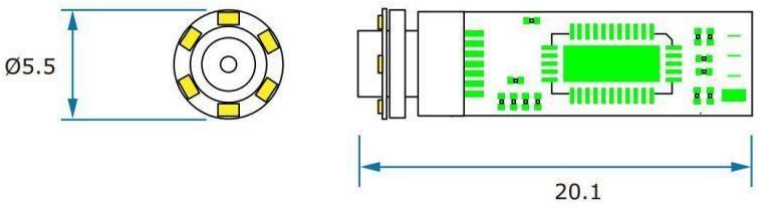


Figure (4.4): The size of a mini camera [App. A]

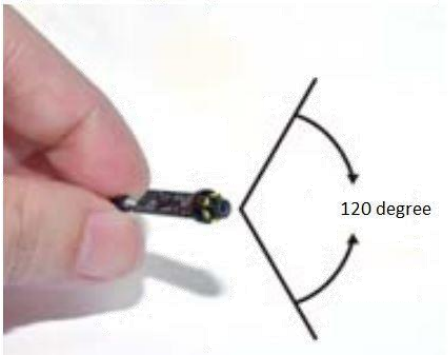


Figure (4.5): Angle of viewing camera [App. A]

Now we design the size of the camera to be comfortable and small size to place on glasses, the diameter of the lens is 5.5 mm and the angle can see around 120 degrees, to make space wider the eyes of the camera and the possibility of the person to recognize the more object.

4.5 Programming Language using Python

Being a very high-level language, Python reads like English, which takes a lot of syntax-learning stress off coding beginners. Python handles a lot of complexity for you, so it is very beginner-friendly in that it allows beginners to focus on learning programming concepts and not have to worry about too many details. [46]

As a dynamically typed language, Python is flexible. This means there are no hard rules on how to build features, and you'll have more flexibility in solving problems using different methods (though the Python philosophy encourages using the obvious way to solve things). Furthermore, Python is also more forgiving of errors, so you'll still be able to compile and run your program until you hit the problematic part. [46]

As you step into the programming world, you'll soon understand how vital support is, as the developer community is all about giving and receiving help. The larger a community, the more likely you'd get help and the more people will be building useful tools to ease the process of development. [46]

Stack Overflow is a programming Q&A site you will no doubt become intimate with as a coding beginner. Python has 85.9k followers, with over 500k Python questions. Python questions are also the 3rd most likely to be answered when compared to other popular programming languages. [46]

4.6 Dataset of image

The ImageNet dataset [47], which contains an unprecedented number of images, has recently enabled breakthroughs in both object classification and detection research [48], [50],

The community has also created datasets containing object attributes [51], scene attributes [52], key points [53], and 3D scene information [54] the goal of advancing the state-of-the-art in object recognition by placing the question of object recognition in the context of the broader question of scene understanding. [55]

The properties of the Microsoft Common Objects in Context (MS COCO) dataset in comparison to several other popular datasets. These include ImageNet [47], PASCAL VOC 2012 [48], and SUN [49]. Each of these datasets varies significantly in size, list of labelled categories, and types of images. ImageNet was created to capture a large number of object categories, many of which are fine-grained. SUN focuses on labelling scene types and the objects that commonly occur in them. Finally, PASCAL VOC's primary application is object detection in natural images. MS COCO is designed for the detection and segmentation of objects occurring in their natural context [55].

The Microsoft Common Objects in Context (MS COCO) dataset contains 91 common object categories with 82 of them having more than 5,000 labelled instances, Figure (4.7) in total the dataset has 2,500,000 labelled instances in 328,000 images. In contrast to the popular ImageNet dataset [1], COCO has fewer categories but more instances per category. This can aid in learning detailed object models capable of precise 2D localization. The dataset is also significantly larger in number of instances per category than the PASCAL VOC [48] and SUN

Datasets. Additionally, a critical distinction between our dataset and others is the number of labelled instances per image which may aid in learning contextual information [55].

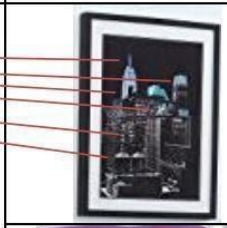
Figure (4.7): Samples of images in the MS COCO dataset [55].

4.7 Recognition

The results are presented based on our dataset. The dataset includes objects captured from images of real life. Initially, the objects are gathered and identified by ourselves. The dataset consists of 300 images of 25 objects. The tested images comprise 180 images that were taken from the right camera. The resolution of images is about 600 x 500 pixels. Objects recognition from the model database proceeds as follows:

The GPS coordinate is extracted, and it retrieves all objects that are associated with it. The images in the test set are compared to all objects in the database having the same location coordinates. The objects that have acknowledged features under the location term from the database are then chosen as recognized objects as shown in Table (4.4) the matching is applied by calculating the Euclidean distance between the descriptor vectors of the input object and all objects having the same location in the database. If the distance is closer than 0.8 times the distance of the second nearest neighbour, then a matching pair is considered to be detected. This threshold value was adapted based on the best result that has been achieved. The output is composed of concatenated strings for both English and Arabic languages. The API of Google cloud speech was used to convert text to audio. This tool was chosen because it supports over 80 languages. Hence, the proposed approach can be globally used based on the supported languages by Google cloud.

Table (4.4): The features of objects for the involved scene (GPS-based location) only are extracted and matched with the reference image.

Image of real scene	Only the identified objects in the matched area are extracted	Features descriptor in models database	GPS	
			Latitude	Longitude
		Features of paint	X_1	Y_1
		Features of couch	X_2	Y_2
		Features of flower vase	X_3	Y_3

CHAPTER FIVE

5

Object Detection & Recognition Using Tensor Flow

5.1 Introduction

5.2 Tensor Flow

5.3 Why Tensor Flow?

5.4 Neural Network

5.5 Object Detection with Tensor Flow

- **5.5.1 Computations are done in Two steps**
- **5.5.2 Convert labels to the TF Record format**

5.6 Detection Models

- **5.6.1 Single Shot Detector (SSD)**
- **5.6.2 RCNN**
- **5.6.3 Fast RCNN**

5.7 Recognition

- **5.7.1 Three Steps Recognition**

5.1 Introduction

Computer vision is an interdisciplinary field that deals with how computers can be made for gaining a high-level understanding from digital images or videos. From the perspective of engineering, it seeks to automate tasks that the human visual system can do. [61] Computer vision is concerned with the automatic extraction, analysis, and understanding of useful information from a single image or a sequence of images. It involves the development of a theoretical and algorithmic basis to achieve automatic visual understanding.

One of the primary goals of computer vision is the understanding of visual scenes. Scene understanding involves numerous tasks including recognizing what objects are present, localizing the objects in 2D and 3D, determining the objects' and scene's attributes, characterizing relationships between objects, and providing a semantic description of the scene. [61] [62].

5.2 Tensor Flow

Tensor Flow is an open-source software library for high-performance numerical computation. Its flexible architecture allows easy deployment of computation across a variety of platforms (CPUs, GPUs, TPUs), and from desktops to clusters of servers to mobile and edge devices. Originally developed by researchers and engineers from the Google Brain team within Google's AI organization, it comes with strong support for machine learning and deep learning and the flexible numerical computation core is used across many other scientific domains [63].

5.3 Why Tensor Flow

Python API

Portability: deploy computation to one or more CPUs or GPUs in a desktop, server, or mobile device with a single API.

Flexibility: from Raspberry Pi, Android, Windows, IOS, and Linux to server farms. Visualization.

Checkpoints (for managing experiments).

Auto-differentiation (no more taking derivatives by hand)

Large community (> 10,000 commits and > 3000 TF-related repos in 1 year). Awesome projects powerful using Tensor Flow [63].

5.4 Neural Network

Neural Network commonly referred to as "Neural Networks" has been motivated right from their inception by the recognition that the human brain computes in an entirely different way from the conventional digital computer. The brain is a highly complex, nonlinear, and parallel computer (information-processing system). It can organize its structural constituents, known as neurons, to perform certain computations (e.g., pattern recognition, perception, and motor control) many times faster than the fastest digital computer in existence today. Consider, for

example, human vision, which is an information-processing It is the function of the visual system to provide a representation of the environment around us and more importantly, to supply the information we need to interact with the environment. To be specific, the brain routinely accomplishes perceptual recognition tasks (e.g., recognizing) [65].

Neural Networks help us cluster and classify. You can think of them as a clustering and classification layer on top of the data you store and manage. They help to group unlabelled data according to similarities among the example inputs, and they classify data when they have a labelled dataset to train on. (Neural networks can also extract features that are fed to other algorithms for clustering and classification; so you can think of deep neural networks as components of larger machine-learning applications involving algorithms for reinforcement learning, classification, and regression.)

The Convolutional Neural Networks (CNNs), an important and powerful kind of learning architecture widely diffused especially for Computer Vision applications. They currently represent the state of a art algorithm for image classification tasks and constitute the main architecture used in Deep Learning [65.

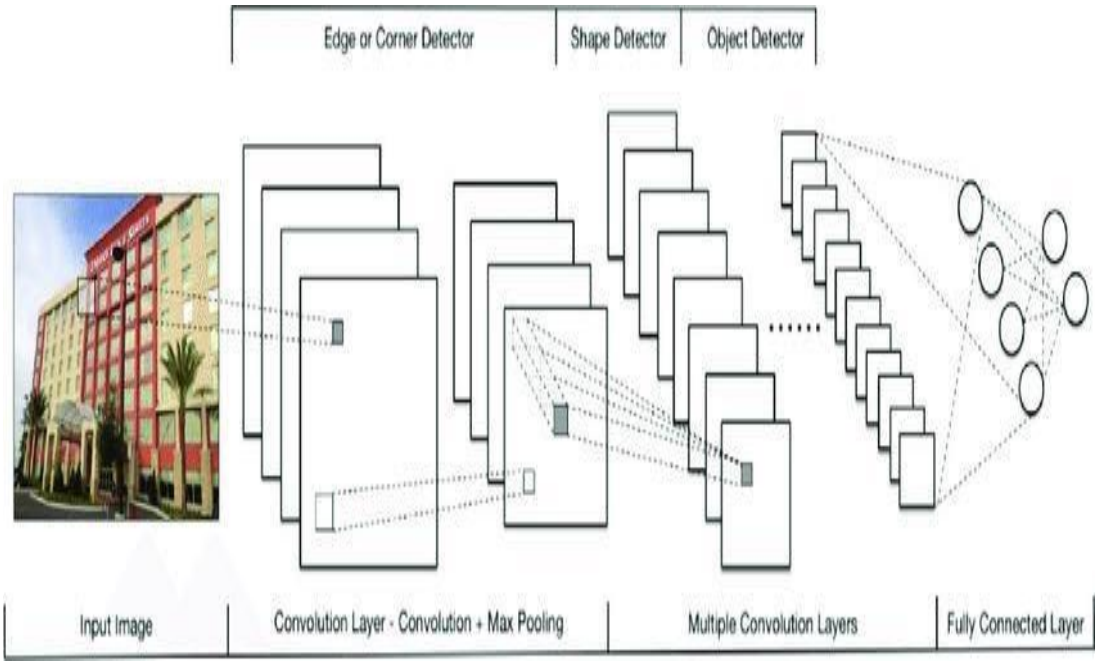


Figure (5.1): Convolutional Neural Networks (CNN) [65].

5.5 Object detection with Tensor Flow

5.5.1 Computations are done in two steps:

– **First:** Build the graph.

– **Second:** Execute the graph. Both steps can be done in many languages (python, C++) Best supported so far is python [64].

We will walk through all the steps for building a custom object classification model using

Tensor Flow's API**:**

Gathering a data set:

Some very large detection data sets, such as MS-COCO, exist already.

Creating bounding boxes:

Toan train our object detection model, for each image we will need the image's width, height, and each class with their respective xmin, xmax, ymin, and ymax bounding box. Simply put, our bounding box is the frame that captures exactly where our class is in the image.

Creating these labels can be a huge ordeal, but thankfully some programs help create bounding boxes. Labelling is an excellent open-source free software that makes the labelling process much easier. It will save individual XML labels for each image, which we will convert into a CSV table for training. The labels for all the images used in the pawn detector we are building are included in the Get Hub repository [64].

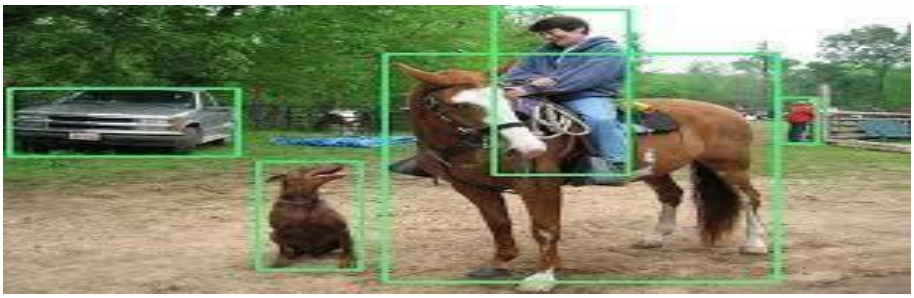


Figure (5.2): Train our object detection model [64].

Install the object detection API:

Before getting started, we have to clone and install the object detection API into our Get Hub repository. Installing the Object detection API is extremely simple; you just need to clone the Tensor Flow Models directory and add some things to your Python path.

5.5.2 Convert labels to The Tensor Flow Record format:

When training models with Tensor Flow using Tensor Flow Record, files help optimize your data feed. We can generate a Tensor Flow Record file using code adapted from this raccoon detector.

Choose a model:

There are models in the Tensor Flow API you can use depending on your needs. If you want a high-speed model that can work on detecting video feed at high fps, the single-shot detection (SSD) network works best. Some other object detection networks detect objects by sliding different sized boxes across the image and running the classifier many times on different sections of the image, this can be very resource consuming. As its name suggests, the SSD network determines all bounding box probabilities in one go; hence, it is a vastly faster model [64].

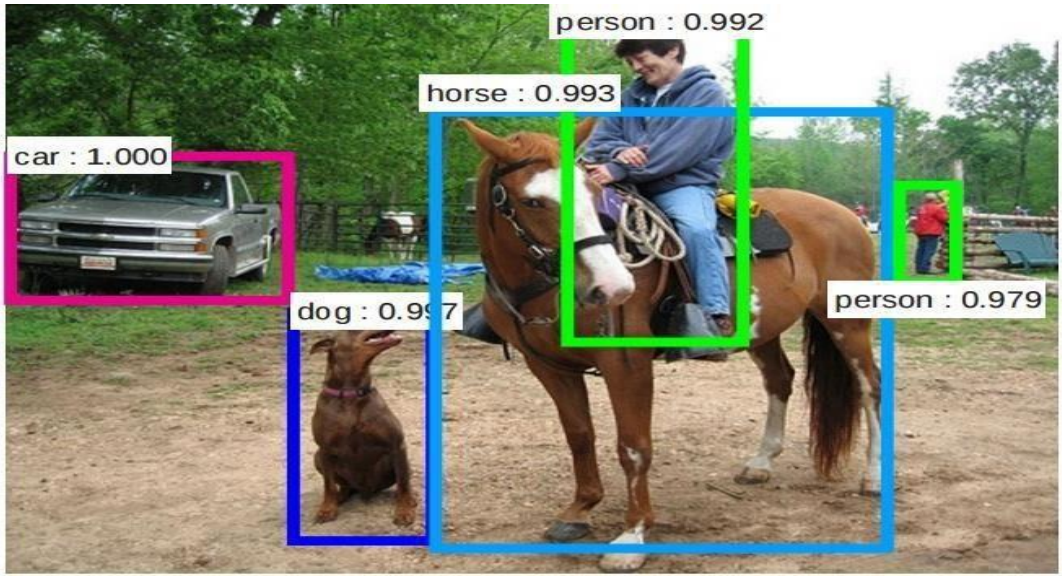


Figure (5.3): Object Detection [64].

5.6 Detection Models

5.6.1 Single Shot Detector SSD:

We present a method for detecting objects in images using a single deep neural network. Our approach, Named **SSD**.

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

Multi-Box: this is the name of a technique for bounding box regression.

Detector: The network is an object detector that also classifies those detected objects [65].

Detectors are convolutional filters, each detector outputs a single value. Discretizes the output space of bounding boxes into a set of default boxes over different aspect ratios and scales per feature map location. At prediction time, the network generates scores for the presence of each object category in each default box and produces adjustments to the box to better match the object shape. Additionally, the network combines predictions from multiple feature maps with different resolutions to naturally handle objects of various sizes. Our SSD model is simply relative to methods that require object proposals because it eliminates proposal generation and subsequent pixel or feature resampling stage and encapsulates all computation in a single network. This makes SSD easy to train and straightforward to integrate into systems that require a detection component [65].

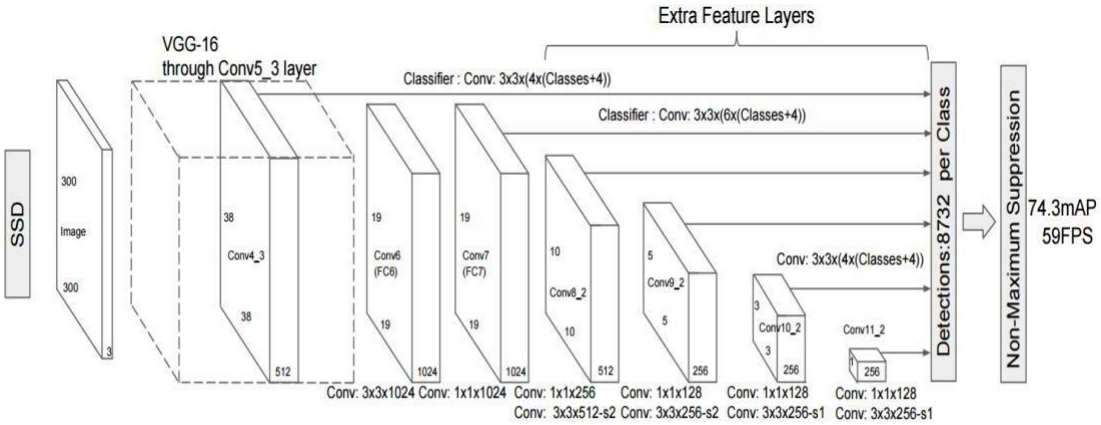


Figure (5.4): Single Shot Detector SSD [65].

5.6.2 RCNN (Region Proposal + CNN)

The Region-based Convolutional Network method (RCNN) achieves excellent object detection accuracy by using a deep ConvNet to classify object proposals. R-CNN [65].

Use selective search to come up with regional proposal First object detection method using CNN.

Training RCNN:

Step1: train your own CNN model for classification using the Image Net dataset.

Step2: focus on 20 classes + 1 background. Remove the last FC layer and replace it with a smaller layer and fine-tune the model using the PASCAL VOC dataset.

Step3: extract feature. Store all the features.

Step4: train SVM for each class: -Crop /Warp image.

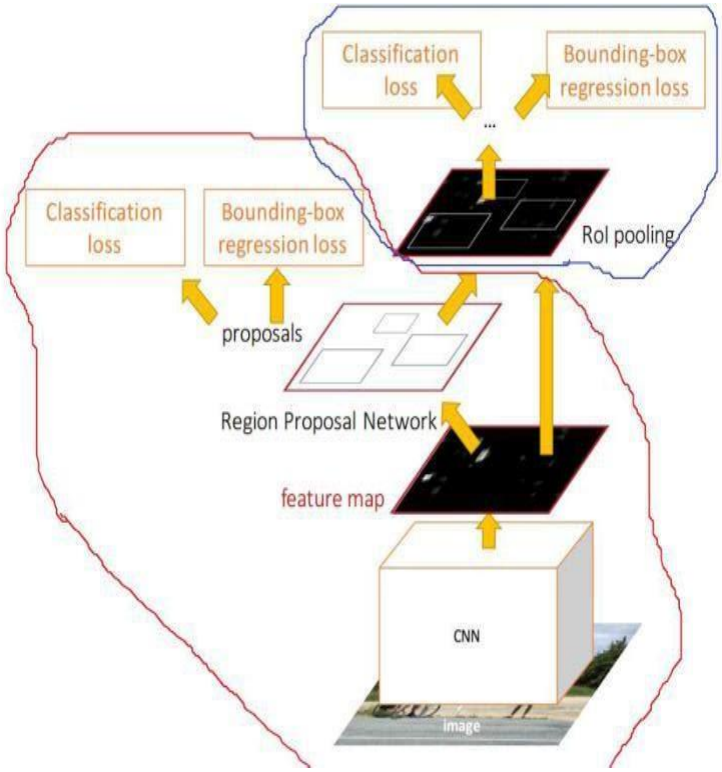


Figure (5.5): RCNN [65].

5.7 Object Recognition with Tensor Flow

A recognition algorithm (image classifier) takes an image as input and outputs what the image contains. In other words, the output is a class label (e.g. “cat”, “dog”, “table” etc.) [66].

5.7.1 Three Steps Recognition:

Step 1: Pre-processing

Often an input image is pre-processed to normalize contrast and brightness effects. A very common pre-processing step is to subtract the mean of image intensities and divide by the standard deviation. Sometimes, gamma correction produces slightly better results. While dealing with colour images, a colour space transformation (e.g. RGB to LAB colour space) may help get better results [66].

Step 2: Feature Extraction

The input image has too much extra information that is not necessary for classification. Therefore, the first step in image classification is to simplify the image by extracting the important information contained in the image and leaving out the rest. For example, if you want to find a shirt and coat buttons in images, you will notice a significant variation in RGB pixel values. However, by running an edge detector on an image we can simplify the image. You can still easily discern the circular shape of the buttons in these edge images and so we can conclude that edge detection retains the essential information while throwing away non-essential information. The step is called feature extraction. In traditional computer vision approaches designing these features is crucial to the performance of the algorithm.

Turns out we can do much better than simple edge detection and find features that are much more reliable. In our example of the shirt and coat buttons, a good feature detector will not only capture the circular shape of the buttons but also information about how buttons are different from other circular objects like car tires [66].

Step 3: Learning Algorithm for Classification

In the previous section, we learned how to convert an image to a feature vector. In this section, we will learn how a classification algorithm takes this feature vector as input and outputs a class label (e.g. cat or background).

Before a classification algorithm can do its magic, we need to train it by showing thousands of examples of cats and backgrounds. Different learning algorithms learn differently, but the general principle is that learning algorithms treat feature vectors as points in higher dimensional space, and try to find planes/surfaces that partition the higher dimensional space in such a way that all examples belonging to the same class are on one side of the plane/surface [66].

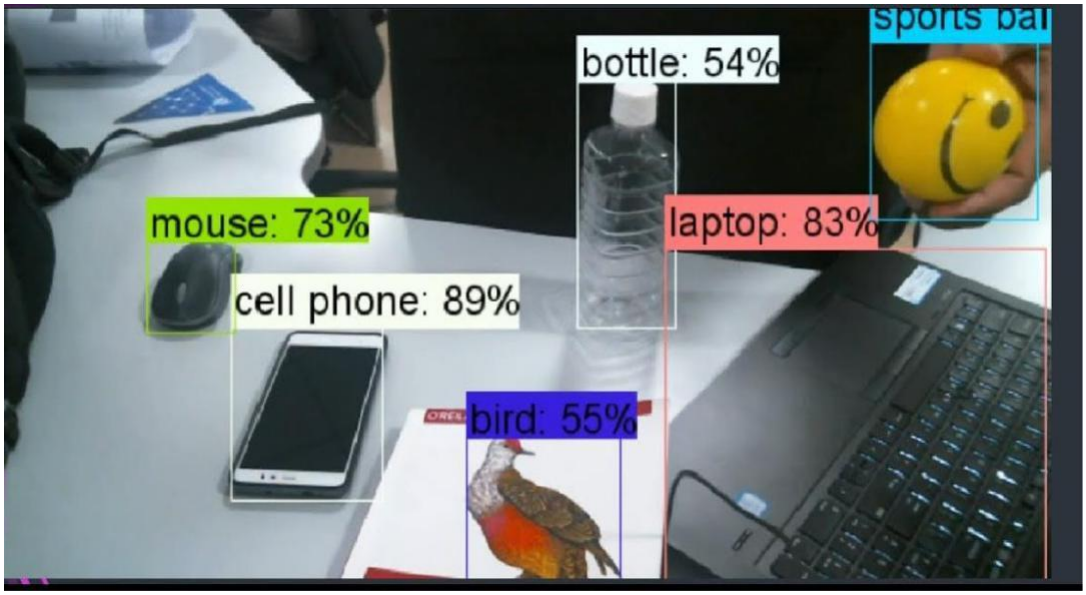


Figure (5.7): Result of Objects Recognition [66].

CHAPTER SIX

6

SIMULATION & RESULTS

6.1 Simulation

6.1.1 Connecting the Camera

6.1.2 Camera Setup and Configuration

6.1.3 Understanding Training process

6.2 Results

6.3 challenges

6.4 conclusion & future work

6.1 Simulation:

6.1.1 Connecting the Camera

The system starts by connecting the OpenCV camera module with the computer/smartphone through a cable, the cable connects between the fast camera Serial Interface bus and the system-on-chip processor, the camera is connected to the computer.

6.1.2 Camera Setup and Configuration

This stage Operating System has already installed all dependence packages on computer/cloud, Programming codes after the system is ready on the side with the correct programs, all required programs that will do the job was written in python.

Here, we will employ a dataset that includes objects captured from images of real life, the dataset consists of more than 300 images of 90 objects. The resolution of images is about 600 x 500 pixels.

6.1.3 Understanding Training process:

Deep neural networks are nothing but mathematical models of intelligence which to a certain extent mimic human brains, Deep learning recognizes objects in images by using three or more layers of artificial neural networks in which each layer is responsible for extracting one or more features of the image [69].

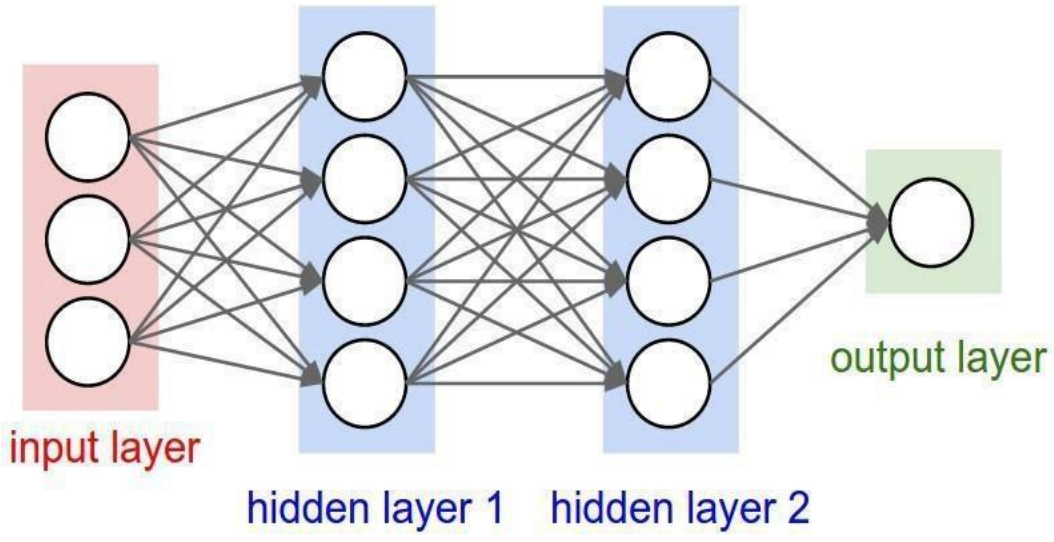


Figure (6.1): Layers in Neural Networks [69].

A neural network is a computational model that is analogous to the arrangement of neurons in the human brain. Each neuron takes an input, performs an operation, and then sends output to one or more adjacent neurons.

Train a Neural Network

Training a Neural Network is very similar to training a little child. You show the child a ball and tell her that it is a “ball”. When you do that many times with different kinds of balls, the child figures out that it is the shape of the ball that makes it a ball and not the colour, texture, or size. You then show the child an egg and ask, “What is this?” She responds “Ball.” You correct them that it is not a ball, but an egg. When this process is repeated several times, the child can tell the difference between a ball and an egg [68] [69].

To train a Neural Network, you show it several thousand examples of the classes (e.g. table, cup, other) you want it to learn. This kind of training is called **Supervised Learning** because you are providing the Neural Network an image of a class and explicitly telling it that it is an image from that class [68] [69].

To train a Neural Network, we need three things:

1-**Training data:** Thousands of images of each class and the expected output.

2- **Cost function:** We need to know if the current setting is better than the previous knob setting. A cost function sums up the errors made by the neural network over all images in the training set.

3- **How to update the knob settings**: Finally, we need a way to update the knob settings based on the error we observe overall training images [68] [69].

Steps Category Labelling in Image:

The first task in annotating our dataset is determining which object categories are present in each image.

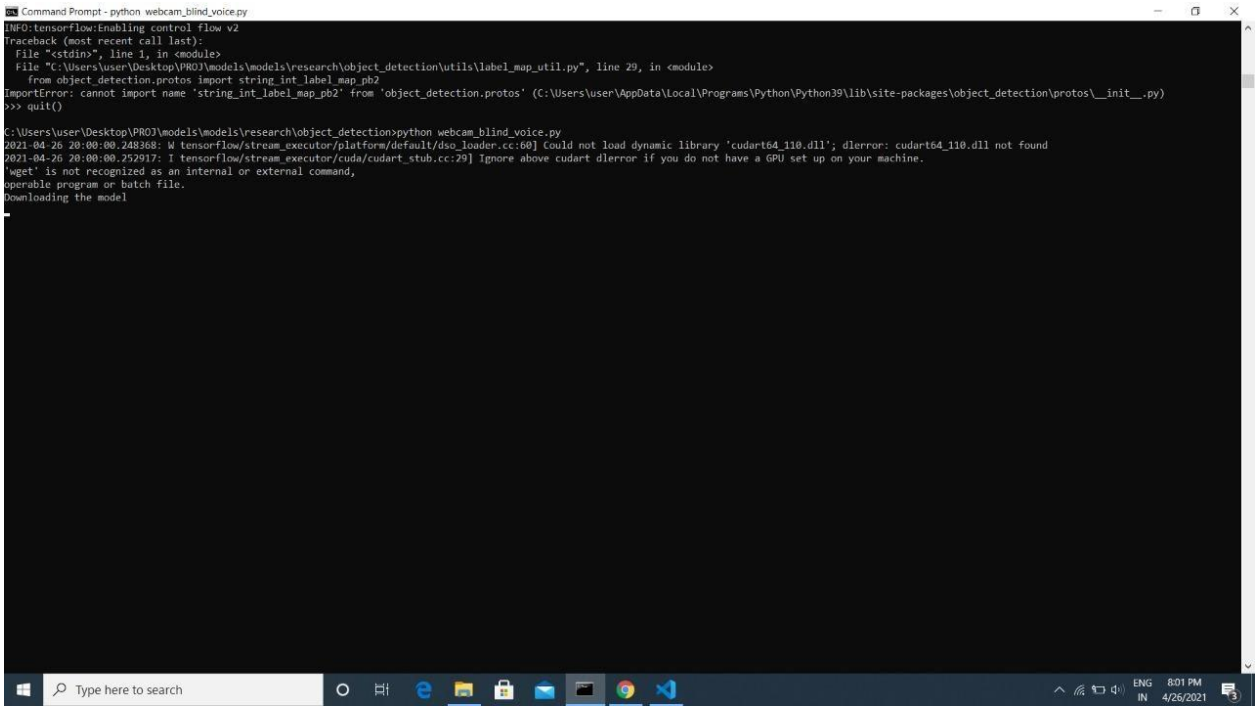
In the next stage, all instances of the object categories in an image were labelled. The final stage is the laborious task of segmenting each object instance, this stage for image segmentation.

Finally, PASCAL VOC’s primary application is object detection in natural images. MS COCO is designed for the detection and segmentation of objects occurring in their natural context.

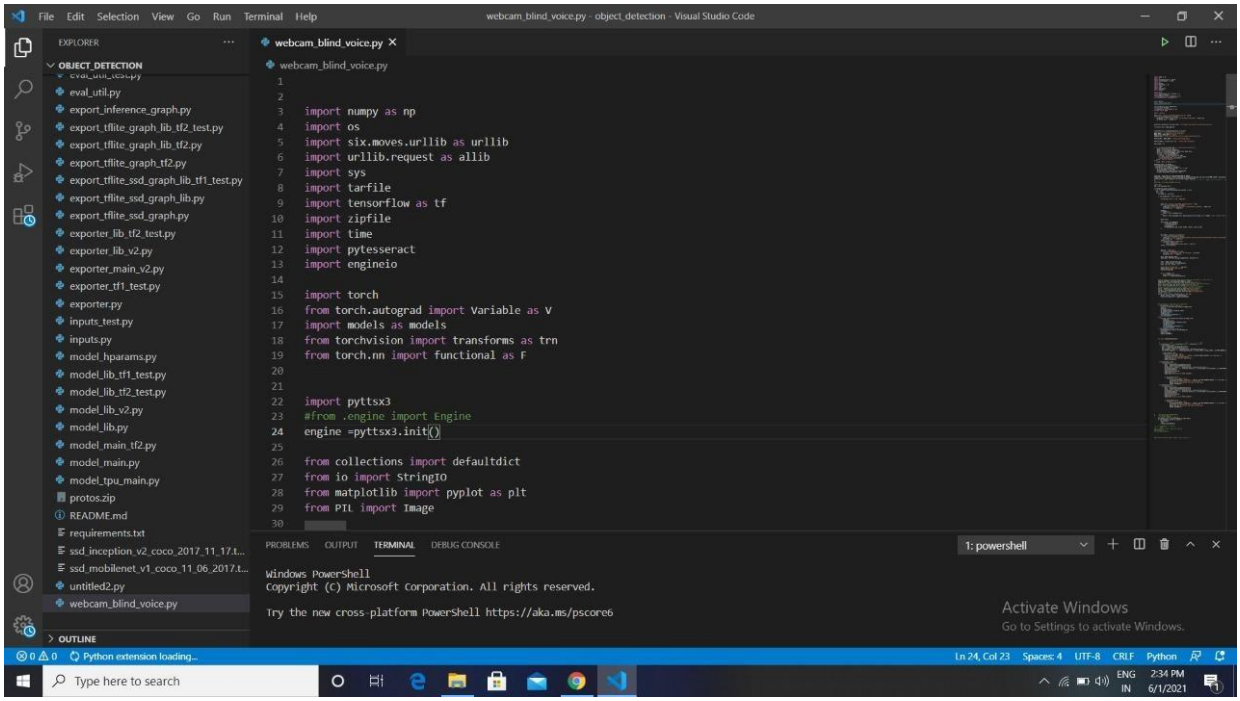
Time Testing

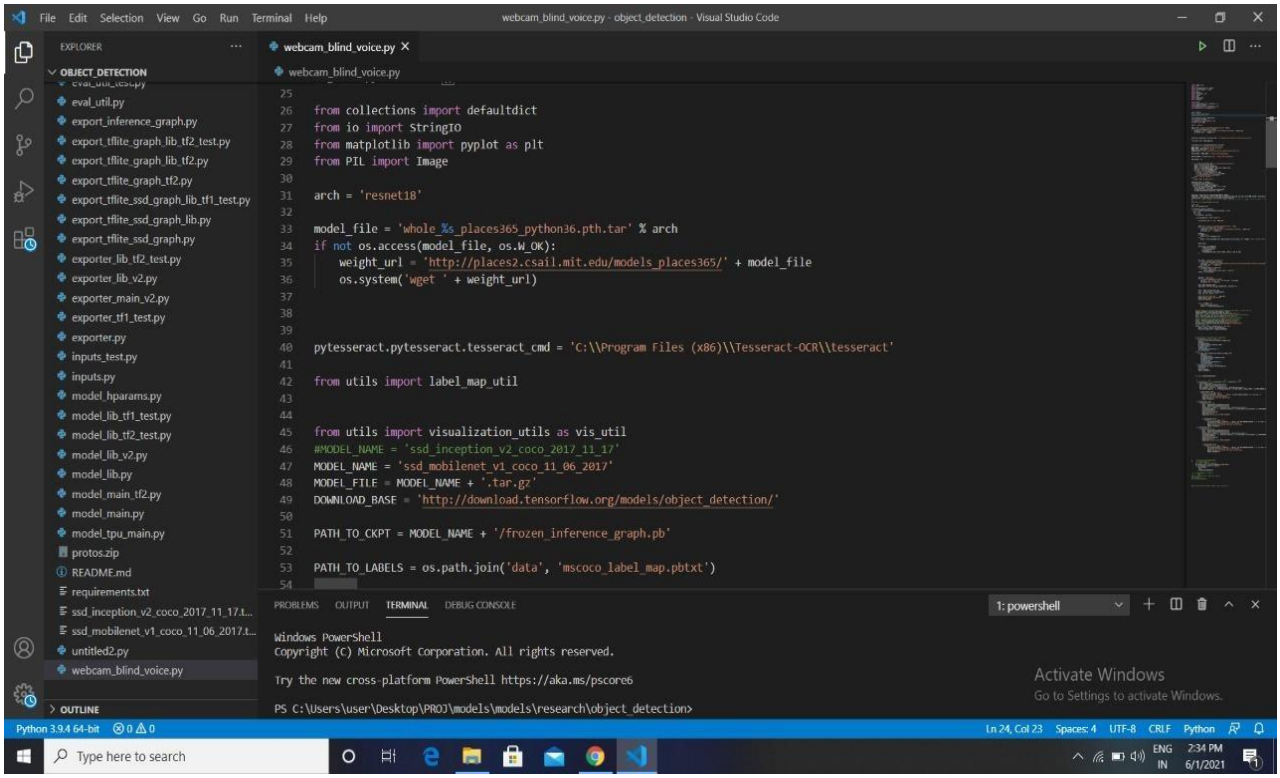
Our graduation project can process and match each training or recognition image in about two seconds on a computer, when using a smartphone its takes 0.5second to 1 second but is enough and good for blind people to know the object

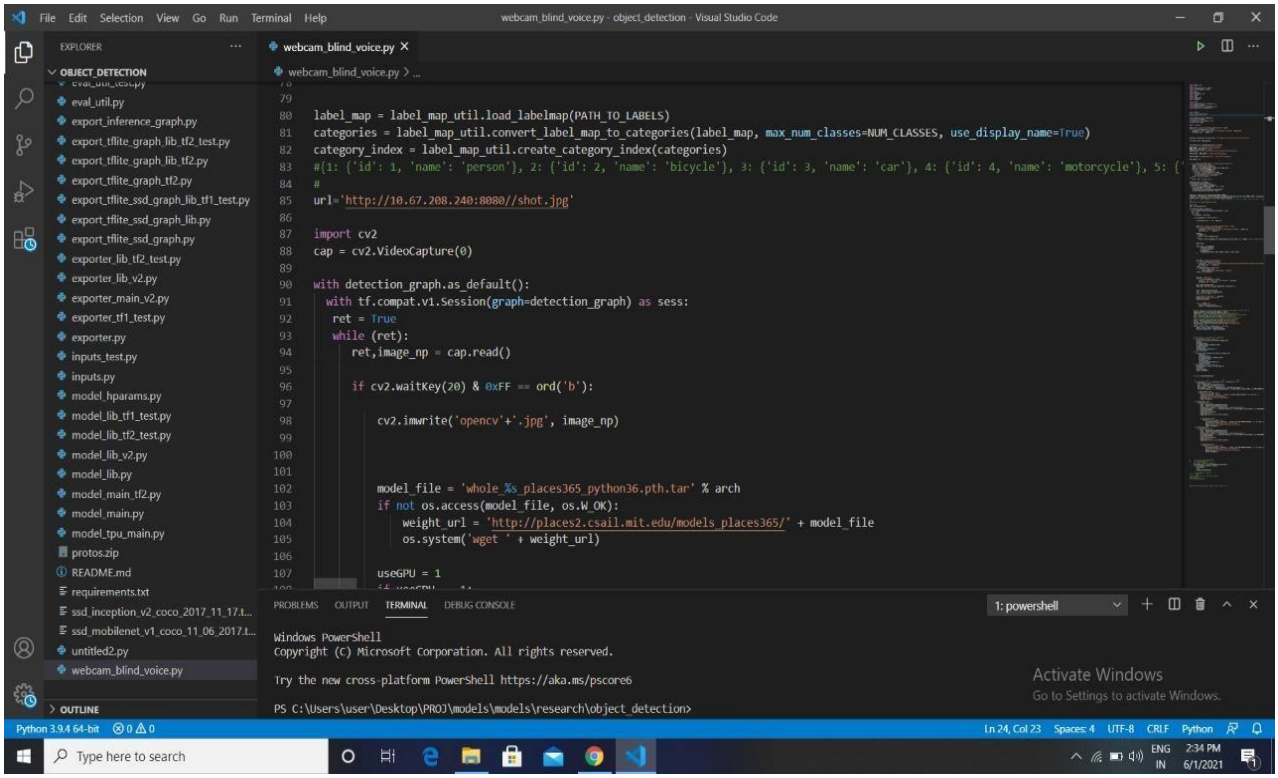
Run Model:

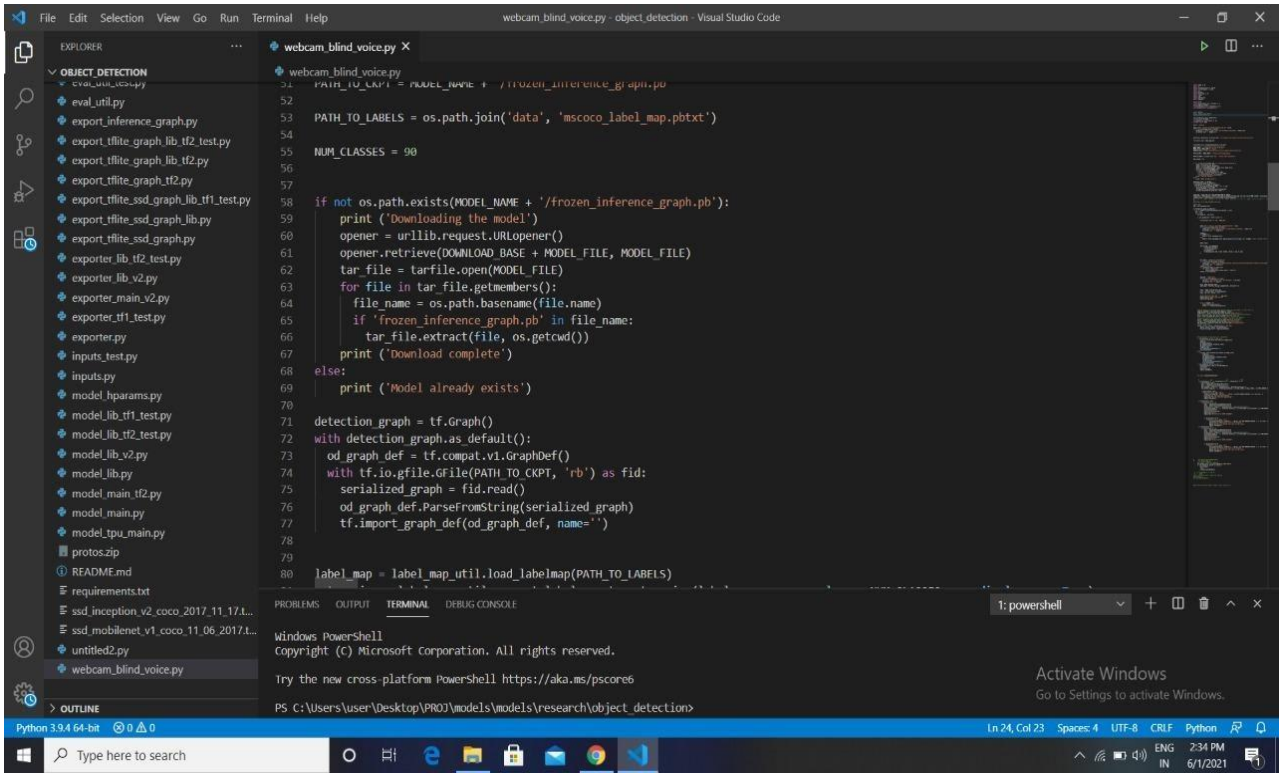


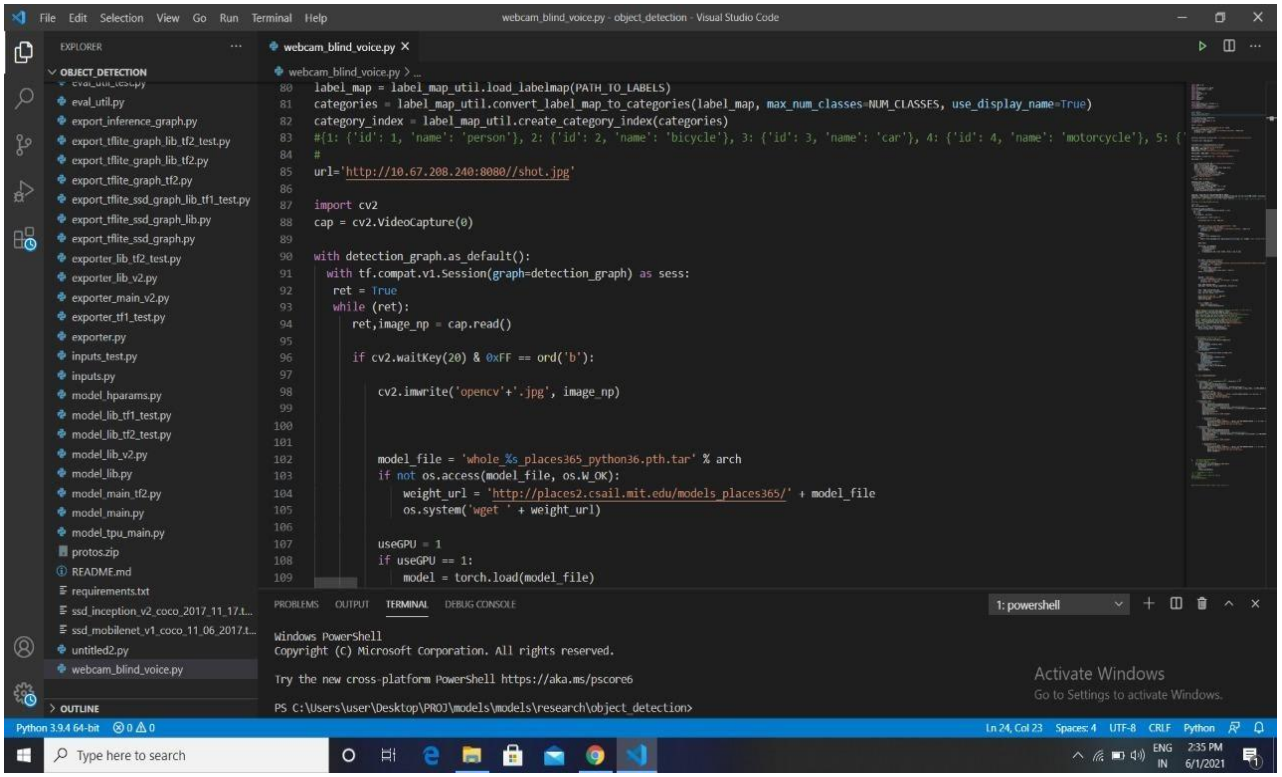
Some Code Screen Short:

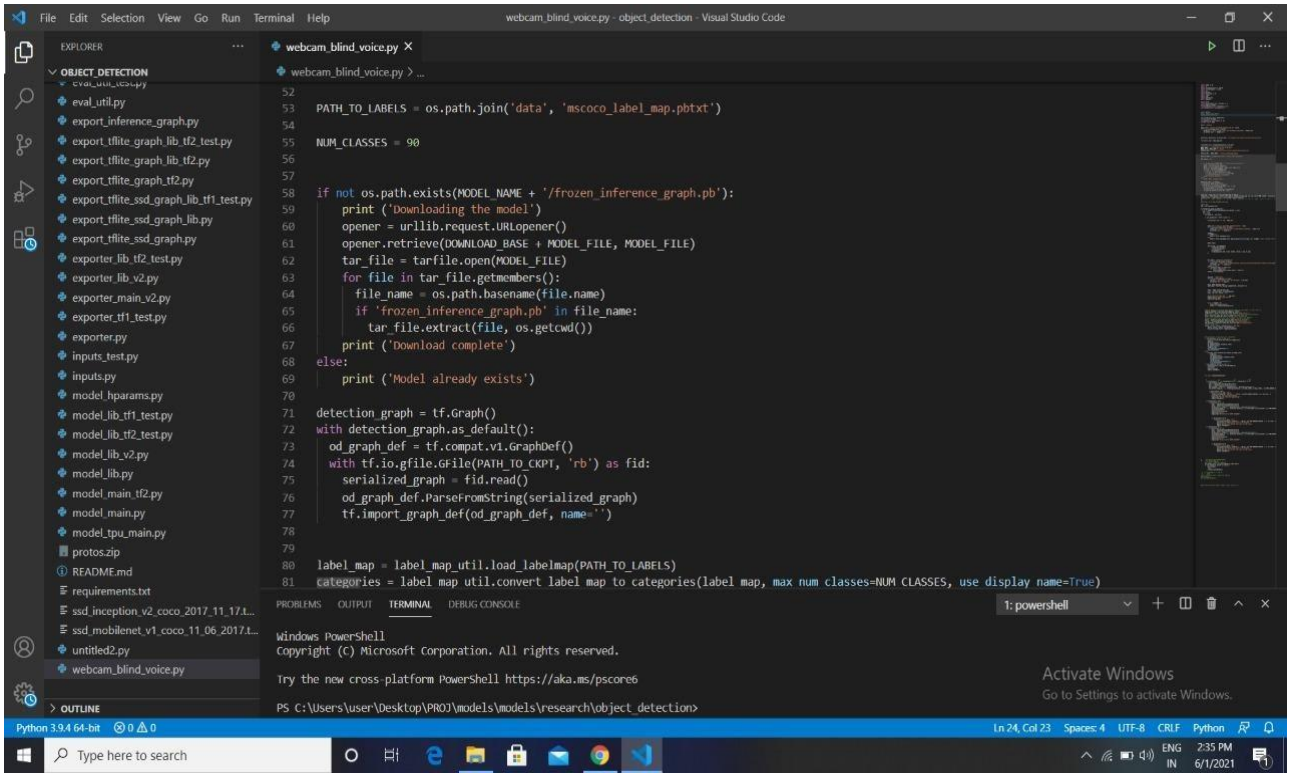



File Edit Selection View Go Run Terminal Help
webcam_blind_voice.py - object_detection - Visual Studio Code
EXPLORER
OBJECT_DETECTION
webcam_blind_voice.py
eval_util.py
export_inference_graph.py
export_tflite_graph_lib_tf2_test.py
export_tflite_graph_lib_tf2.py
export_tflite_graph_tf2.py
export_tflite_ssd_graph_lib_tf1_test.py
export_tflite_ssd_graph_lib.py
export_tflite_ssd_graph.py
exporter_lib_tf2_test.py
exporter_lib_v2.py
exporter_main_v2.py
exporter_tf1_test.py
exporter.py
inputs_test.py
inputs.py
model_hparams.py
model_lib_tf1_test.py
model_lib_tf2_test.py
model_lib_v2.py
model_lib.py
model_main_tf2.py
model_main.py
model_tpu_main.py
protos.zip
README.md
requirements.txt
ssd_inception_v2_coco_2017_11_17.t...
ssd_mobilenet_v1_coco_11_06_2017.t...
untitled2.py
webcam_blind_voice.py
OUTLINE
from collections import defaultdict
from io import StringIO
from matplotlib import pyplot as plt
from PIL import Image
arch = 'resnet18'
model_file = 'whole_%s_places365_python36.pth.tar' % arch
if not os.access(model_file, os.W_OK):
weight_url = 'http://places2.csail.mit.edu/models_places365/' + model_file
os.system('wget ' + weight_url)
pytesseract.pytesseract.tesseract_cmd = 'C:\\Program Files (x86)\\Tesseract-OCR\\tesseract'
from utils import label_map_util
from utils import visualization_utils as vis_util
#MODEL_NAME = 'ssd_inception_v2_coco_2017_11_17'
MODEL_NAME = 'ssd_mobilenet_v1_coco_11_06_2017'
MODEL_FILE = MODEL_NAME + '.tar.gz'
DOWNLOAD_BASE = 'http://download.tensorflow.org/models/object_detection/'
PATH_TO_CKPT = MODEL_NAME + '/frozen_inference_graph.pb'
PATH_TO_LABELS = os.path.join('data', 'mscoco_label_map.pbtxt')
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
1: powershell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.
Try the new cross-platform PowerShell https://aka.ms/pscore6
PS C:\Users\user\Desktop\PROJ\models\models\research\object_detection>
Activate Windows
Go to Settings to activate Windows.
Python 3.9.4 64-bit
Ln 24, Col 23 Spaces: 4 UTF-8 CRLF Python
Type here to search
ENG IN
2:34 PM 6/1/2021


File Edit Selection View Go Run Terminal Help
webcam_blind_voice.py - object_detection - Visual Studio Code
EXPLORER
OBJECT_DETECTION
webcam_blind_voice.py
eval_util.py
export_inference_graph.py
export_tflite_graph_lib_tf2_test.py
export_tflite_graph_lib_tf2.py
export_tflite_graph_tf2.py
export_tflite_ssd_graph_lib_tf1_test.py
export_tflite_ssd_graph_lib.py
export_tflite_ssd_graph.py
exporter_lib_tf2_test.py
exporter_lib_v2.py
exporter_main_v2.py
exporter_tf1_test.py
exporter.py
inputs_test.py
inputs.py
model_hparams.py
model_lib_tf1_test.py
model_lib_tf2_test.py
model_lib_v2.py
model_lib.py
model_main_tf2.py
model_main.py
model_tpu_main.py
protos.zip
README.md
requirements.txt
ssd_inception_v2_coco_2017_11_17.t...
ssd_mobilenet_v1_coco_11_06_2017.t...
untitled2.py
webcam_blind_voice.py
OUTLINE
label_map = label_map_util.load_labelmap(PATH_TO_LABELS)
categories = label_map_util.convert_label_map_to_categories(label_map, max_num_classes=NUM_CLASSES, use_display_name=True)
category_index = label_map_util.create_category_index(categories)
#{1: {'id': 1, 'name': 'person'}, 2: {'id': 2, 'name': 'bicycle'}, 3: {'id': 3, 'name': 'car'}, 4: {'id': 4, 'name': 'motorcycle'}, 5: {
#
url='http://10.67.208.240:8080//shot.jpg'
import cv2
cap = cv2.VideoCapture(0)
with detection_graph.as_default():
with tf.compat.v1.Session(graph=detection_graph) as sess:
ret = True
while (ret):
ret,image_np = cap.read()
if cv2.waitKey(20) & 0xFF == ord('b'):
cv2.imwrite('opencv'+'.jpg', image_np)
model_file = 'whole_%s_places365_python36.pth.tar' % arch
if not os.access(model_file, os.W_OK):
weight_url = 'http://places2.csail.mit.edu/models_places365/' + model_file
os.system('wget ' + weight_url)
useGPU = 1
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
1: powershell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.
Try the new cross-platform PowerShell https://aka.ms/pscore6
PS C:\Users\user\Desktop\PROJ\models\models\research\object_detection>
Activate Windows
Go to Settings to activate Windows.
Python 3.9.4 64-bit
Ln 24, Col 23 Spaces: 4 UTF-8 CRLF Python
Type here to search
ENG IN
2:34 PM 6/1/2021


File Edit Selection View Go Run Terminal Help
webcam_blind_voice.py - object_detection - Visual Studio Code
EXPLORER
OBJECT_DETECTION
webcam_blind_voice.py
PATH_TO_LABELS = os.path.join('data', 'mscoco_label_map.pbtxt')
NUM_CLASSES = 90
if not os.path.exists(MODEL_NAME + '/frozen_inference_graph.pb'):
print ('Downloading the model')
opener = urllib.request.URLopener()
opener.retrieve(DOWNLOAD_BASE + MODEL_FILE, MODEL_FILE)
tar_file = tarfile.open(MODEL_FILE)
for file in tar_file.getmembers():
file_name = os.path.basename(file.name)
if 'frozen_inference_graph.pb' in file_name:
tar_file.extract(file, os.getcwd())
print ('Download complete')
else:
print ('Model already exists')
detection_graph = tf.Graph()
with detection_graph.as_default():
od_graph_def = tf.compat.v1.GraphDef()
with tf.io.gfile.GFile(PATH_TO_CKPT, 'rb') as fid:
serialized_graph = fid.read()
od_graph_def.ParseFromString(serialized_graph)
tf.import_graph_def(od_graph_def, name='')
label_map = label_map_util.load_labelmap(PATH_TO_LABELS)
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
1: powershell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.
Try the new cross-platform PowerShell https://aka.ms/pscore6
PS C:\Users\user\Desktop\PROJ\models\models\research\object_detection>
Activate Windows
Go to Settings to activate Windows.
Python 3.9.4 64-bit
Ln 24, Col 23 Spaces: 4 UTF-8 CRLF Python
Type here to search
ENG IN 2:34 PM 6/1/2021


File Edit Selection View Go Run Terminal Help
webcam_blind_voice.py - object_detection - Visual Studio Code
EXPLORER
OBJECT_DETECTION
webcam_blind_voice.py
label_map = label_map_util.load_labelmap(PATH_TO_LABELS)
categories = label_map_util.convert_label_map_to_categories(label_map, max_num_classes=NUM_CLASSES, use_display_name=True)
category_index = label_map_util.create_category_index(categories)
#{1: {'id': 1, 'name': 'person'}, 2: {'id': 2, 'name': 'bicycle'}, 3: {'id': 3, 'name': 'car'}, 4: {'id': 4, 'name': 'motorcycle'}, 5: {
#
url='http://10.67.208.240:8080//shot.jpg'
import cv2
cap = cv2.VideoCapture(0)
with detection_graph.as_default():
with tf.compat.v1.Session(graph=detection_graph) as sess:
ret = True
while (ret):
ret,image_np = cap.read()
if cv2.waitKey(20) & 0xFF == ord('b'):
cv2.imwrite('opencv'+'.jpg', image_np)
model_file = 'whole_%s_places365_python36.pth.tar' % arch
if not os.access(model_file, os.W_OK):
weight_url = 'http://places2.csail.mit.edu/models_places365/' + model_file
os.system('wget ' + weight_url)
useGPU = 1
if useGPU == 1:
model = torch.load(model_file)
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
1: powershell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.
Try the new cross-platform PowerShell https://aka.ms/pscore6
PS C:\Users\user\Desktop\PROJ\models\models\research\object_detection>
Activate Windows
Go to Settings to activate Windows.
Python 3.9.4 64-bit
Ln 24, Col 23 Spaces: 4 UTF-8 CRLF Python
Type here to search
ENG IN 2:35 PM 6/1/2021



time system

Results:

The view detection and recognition approaches have proven to work well in practice, after testing the project inside the office in front of the blind person. It was detection and recognition correctly and in short time not exceeding two second.

Table 6.1: Test of Objects

Object Name	Number of Tries	Detection Ratio	Pass	Failure	Percentage Error
person	50	91	49	1	2%
backpack	50	92	48	2	4%
bottle	50	94	48	2	4%
cup	50	91	49	1	2%
banana	50	93	49	1	2%
apple	50	93	48	2	5%
spoon	50	94	49	1	2%
bowl	50	91	47	3	6%
chair	20	91	47	3	6%
laptop	25	92	48	2	4%
tv	20	93	47	3	7%
mouse	50	92	49	1	2%
keyboard	50	92	49	1	2%

cell phone	50	91	49	1	2%
book	50	94	48	2	4%
clock	50	93	47	3	7%
scissors	50	92	49	1	2%
remote	50	93	48	2	4%
toothbrush	50	92	49	1	2%

In this table, we show you 20 object of the total 90, where we calculated the error rate in the object selection and recognition it. A range of errors was set between 2% and 7%. This value is based on the object and the accuracy of the camera, And the project is providing more features for an accurate of object detection like the Possibility to detect different object of the same type, detection from different angle, detection multiple object together.

Outputs:

Figure (6.2): Test of result detection and recognition by camera of project.

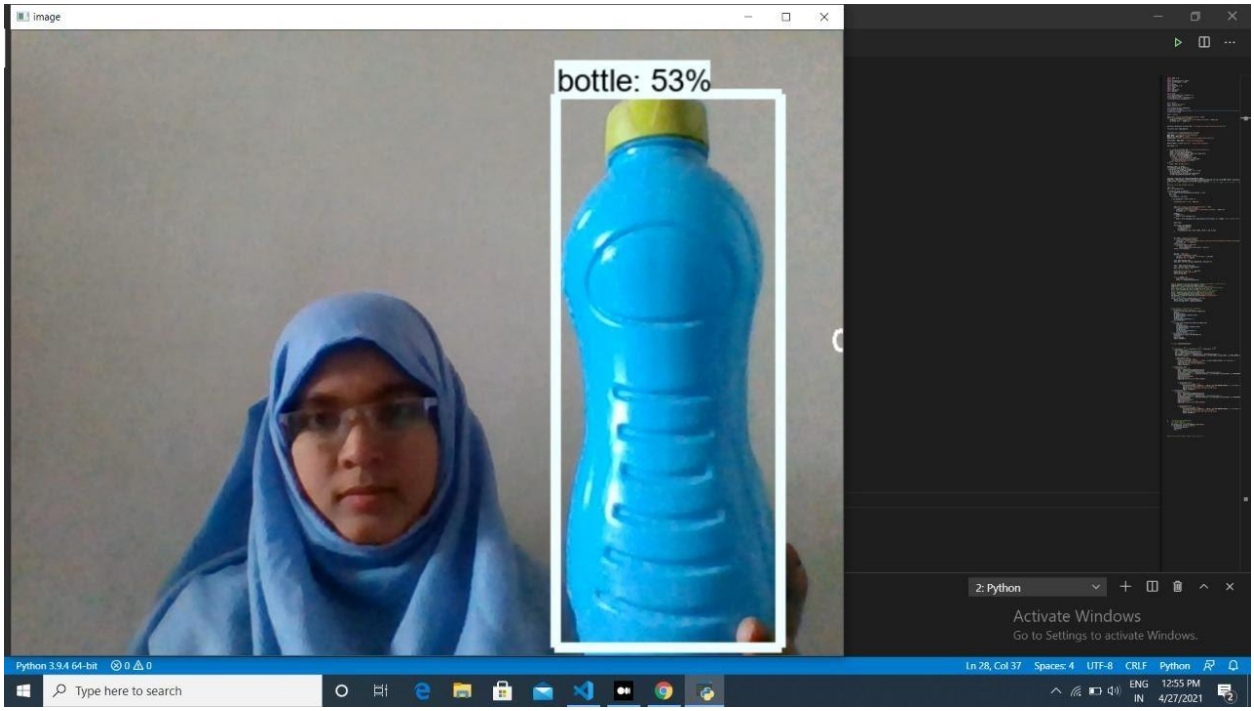


Figure (6.2): Test of result detection and recognition by camera of project.

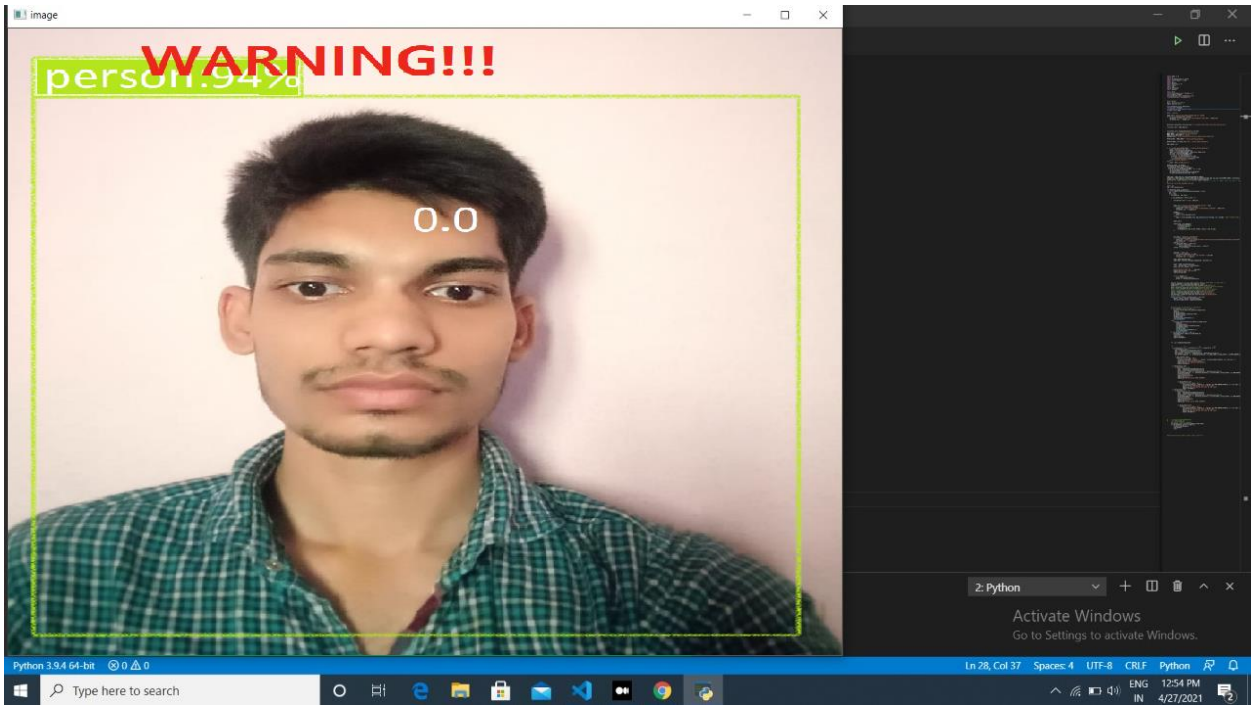


Figure (6.2): Test of result detection and recognition by camera of project.

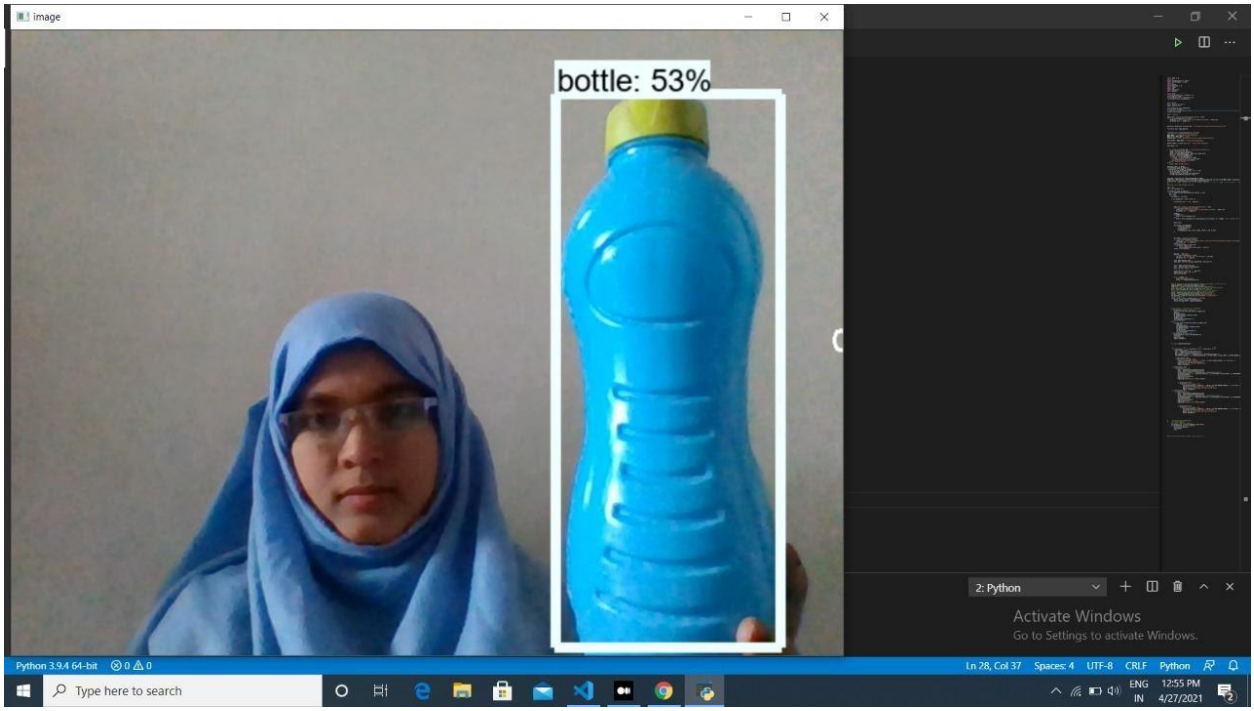


Figure (6.2): Test of result detection and recognition by camera of project.

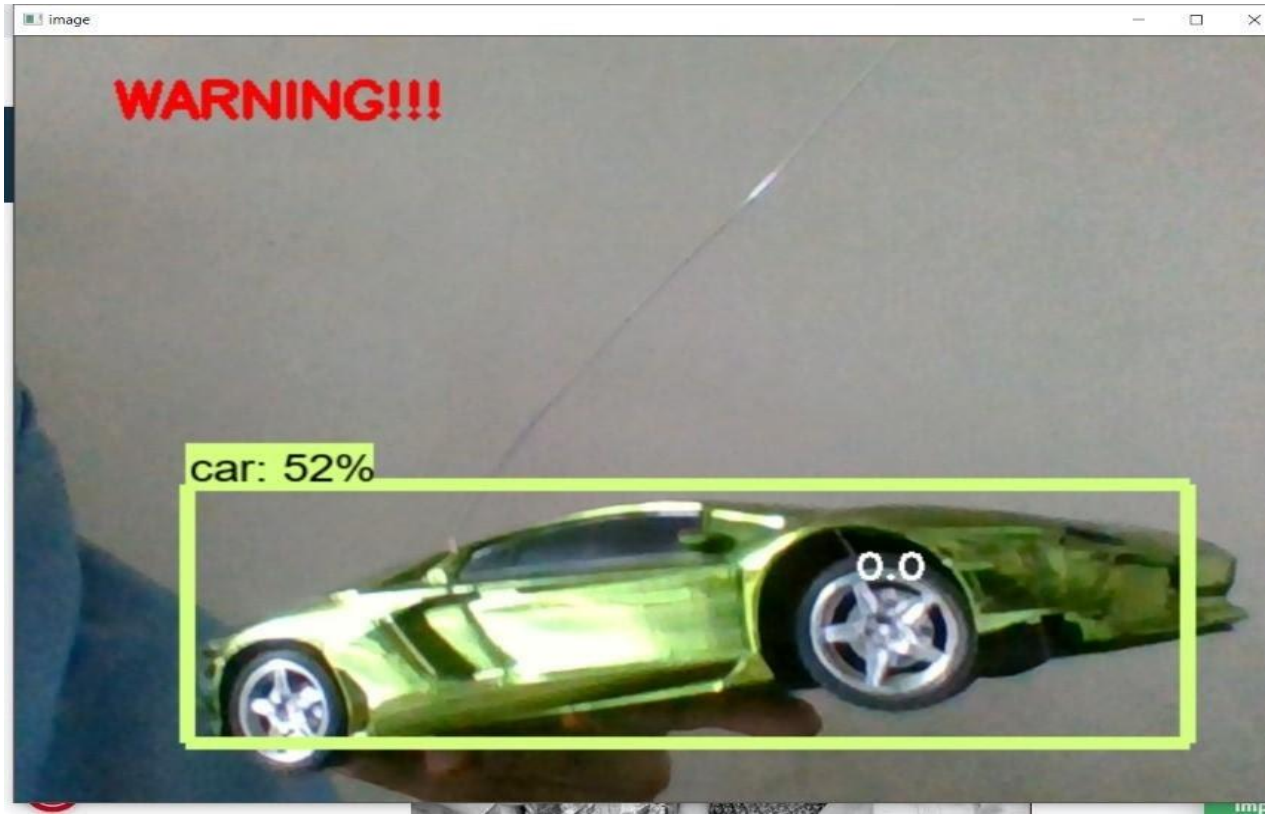


Figure (6.2): Test of result detection and recognition by camera of project.

6.3 Challenges

While building the system, there are some challenge was faced, such as: Not all the required cloud subscription for the project are available

Some problems in dealing and Understand Python programming language also with some project development like training machine learning model.

6.4 Conclusion & Future work

We designed and implemented a smart glass for blind people using special mini camera.

Objects detection is used to find objects in the real world from an image of the world that are common in the scenes of a blind. Based on their locations, and the camera is used to detect any objects.

We expect further improvements in the future as we develop new feature types including colour, distance and other features.

We also recommend using this component Movidius Neural Compute Stick (NCS) is a deep learning USB drive. The NCS is powered by the low-power high-performance Movidius Visual Processing Unit (VPU). Run multiple devices on the same platform to scale performance.

[REFERENCES]

Real Time Object Detection and Recognition for Blind People

References

Y.-J. liu, Z.-Q. Wang , L.-P. Song and G.-G Mu , An anatomically accurate eye model with a shell-structure lens , Optik 116, 241 (2005).

M. A. Rama , M. V. Perez, C. Bao , M. T. Flores-Arias and C. Gomez-Reino , Gradientindex crystalline lens model : A new method for determining the paraxial properties by axial and field rays , Opt. commun. 249,595 (2005).

P. Mouroulis , Visual Instrumentation (McGraw Hill , New York , 1999).

A. Valberg , Light Vision Color (Wiley, Chichester , 2005). 36-5 M. Juttner, Physiological Optics , in T. G. Brown (Ed.) , the Optics Encyclopedia (Willey-VCH , Berlin , 2004),Vol . 4, P. 2511.

Paper of "The Structure of the eye ", www.BiologyMad.com

D. A. Atchison , A. Joblin and G. Smith , Influence of the Stiles-Crawford effect apodization on spatial visual performance , JOSA A 15 , 2545 (1998).

A. Popiolek-Masajada and H. Kasprazak, Model of the optical system of the human eye during Accommodation , Ophthal. Physiol. Opt. 22, 201 (2002).

G. Wyszecki and W. S. Stiles , Color Science (Wiley Interscience , New York , 2000) .

S.G. de Groot and J. W. Gebhard , Pupil size as determined by adapting Luminance , JOSA 42, 249 (1952).

H. Goersch , Handbuch fur Augenoptik (Maurer Verlag , Geislingen , 1992).

W. Lotmar ,Theoretical eye model with asperics , JOSA 61 , 1522 (1971).

E. R. Villegas , L. Carretero and A. Fimia , Le Grand eye for the study of ocular chromatic aberration , Ophthal. PHysiol. Opt. 16, 528 (1996).

World Health Organization, www.who.com .

Dana H. Ballard; Christopher M. Brown (1982). Computer Vision. Prentice Hall. ISBN 0-13-165316-4.

bHuang, T. (1996-11-19). Vandoni, Carlo, E, ed. Computer Vision : Evolution And Promise (PDF). 19th CERN School of Computing. Geneva: CERN. pp. 21–25. ISBN 978-9290830955. doi:10.5170/CERN-1996-008.21.

bMilanSonka; Vaclav Hlavac; Roger Boyle (2008). Image Processing, Analysis, and Machine Vision. Thomson. ISBN 0-495-08252-X.

ReinhardKlette (2014). Concise Computer Vision. Springer. ISBN 978-1-4471-6320-6.

Linda G. Shapiro; George C. Stockman (2001). Computer Vision. Prentice Hall. ISBN 0-13-030796.

Tim Morris (2004). Computer Vision and Image Processing. Palgrave Macmillan. ISBN 0-333-99451-5.

Bernd Jähne; Horst Haußecker (2000). Computer Vision and Applications, A Guide for Students and Practitioners. Academic Press. ISBN 0-13-085198-1.

David A. Forsyth; Jean Ponce (2003). Computer Vision, A Modern Approach. Prentice Hall. ISBN 0-13-085198-1.

http://www.bmva.org/visionoverview The British Machine Vision Association and Society for Pattern Recognition Retrieved February 20, 2017

Murphy, Mike. "Star Trek's "tricorder" medical scanner just got closer to becoming a reality".

The Human EyeStructure and Function , Clyde W. Oyster The University of Alabamaat Birmingham.

Y. Alimonies (ed.), Special Issue on Purposive and Quantitative Active Vision, CVGIP B: Image Understanding, Vol. 56(1992).

D. Marr, ``Vision: A Computational Investigation into the Human Representation and Processing of Visual Information'', Freeman, San Francisco (1982).

L. Roberts, ``Machine perception of 3D solids", Chapter 9 in J. T. Tippet, et al. (ends), Optical and Electro Optical Information Processing, MIT Press, pp. 159-197 (1965).
Computer Vision: Evolution and Promise, T. S. Huang University of Illinois at Urbana-Champaign, Urbana , IL 61801, U. S. A

IMPORTANCE OF COMPUTER VISION FOR HUMAN LIFE Amrita Parashar. Research Scholar, Amity University Madhya Pradesh.
Dictionary of Computer Vision and Image Processing, Robert Fisher, Ken Dawson-Howe, Andrew Fitzgibbon, Craig Robertson, Emmanuelle Trucco, Wiley, 2005.

R. M. Haralick and L. G. Shapiro, "Glossary of Computer Vision Terms," Pattern Recognition 24:69-93, 1991.
R. M. Haralick, "Glossary and index to Remotely Sensed Image Pattern Recognition Concepts," Pattern Recognition 5:391-403, 1973

The slides are from several sources through James Hays (Brown); SrinivasaNarasimhan (CMU); Silvio Savarese (U. of Michigan); Bill Freeman and Antonio Torralba (MIT), including their own slides.

An Introduction to Computer Vision ,Ying Wu ,Electrical Engineering & Computer Science ,Northwestern University ,Evanston, IL 60208 ,yingwu@ece.northwestern.edu [35]. Library of Congress Cataloging-in-Pubblication Data Gonzalez, Rafael C. Digital Image Processing / Richard E. Wood. p. cm. Includes bibliographical references. ISBN 0-201-18075-81. Digital Imaging. 2. Digital Techniques. I. Title.

T. A. Iinuma and T. Nagai. "Image restoration in radioisotopic imaging systems." In: Phys.

Med. Biol. 12.4 (Oct. 1967), 501–510. DOI: 10.1088/0031-9155/12/4/005 (cit. on p. 1.2).

H. C. Andrews and B. R. Hunt. Digital image restoration. NJ: Prentice-Hall, 1977 (cit. on p.

1.2).

R. H. T. Bates and M. J. McDonnell. Image restoration and reconstruction. New York: Oxford, 1986 (cit. on p. 1.2).

Wei-Yi Wei E-mail: s9361121@nchu.edu.tw Graduate Institute of Communication Engineering National Taiwan University, Taipei, Taiwan, ROC.

Digital Image Processing, 3rd edition by Gonzalez and Woods.

WAVELET TRANSFORM IN IMAGE COMPRESSION Presented By E . JEEVITHA 16MMAT05 M.Phil Mathematics.

Image Segmentation, Representation and Description Wei-De Chang mail:tmac579969@hotmail.com Graduate Institute of Communication Engineering National Taiwan University, Taipei Taiwan, ROC.

MAJOR EYE DISEASES & TREATMENT.

https://www.thespruce.com/what-is-a-camera-2688050

http://www.bestprogramminglanguagefor.me/why-learn-python

http://www.misumi.com.tw/CONTACT_custom.asp

MD-T21106L-camera data sheet

N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor segmentation and support inference from RGBD images," in ECCV,2012

P. Dollar, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection: An evaluation of the state of the art," PAMI, vol. 34, 2012.

https://www.thespruce.com/what-is-a-camera-2688050

http://www.bestprogramminglanguagefor.me/why-learn-python

J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale

Hierarchical Image Database," in CVPR,2009.

Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL visual object classes (VOC) challenge,"IJCV, vol. 88, no. 2, pp. 303–338, Jun. 2010.

J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba, "SUN database: Large-scale scene recognition from abbey to zoo," in CVPR, 2010.

R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in CVPR, 2014.

P. Sermanet, D. Eigen, S. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, “OverFeat:

Integrated recognition, localization and detection using convolutional networks,” in ICLR, April 2014.

G. Patterson and J. Hays, “SUN attribute database: Discovering, annotating, and recognizing scene attributes,” in CVPR, 2012.

L. Bourdev and J. Malik, “Poselets: Body part detectors trained using 3D human pose annotations,” in ICCV, 2009.

N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, “Indoor segmentation and support inference from RGBD images,” in ECCV,2012.

Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, Piotr Dollár(Submitted on 1 May 2014 (v1), last revised 21 Feb 2015 (this version, v3))

http://www.misumi.com.tw/CONTACT_custom.asp

[57] MD-T21106L-camera data sheet

[58] Jamal S. Zraqou, Wissam M. Alkhadour and Mohammad Z. Siam, Isra University, Amman-Jordan ,Accepted 30 Jan 2017, Available online 31 Jan 2017, Vol.7, No.1 (Feb 2017)

[59] https://www.raspberrypi.org

J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “ImageNet: A Large-Scale

Hierarchical Image Database,” in *CVPR* 2009.

M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, “The PASCAL visual object classes (VOC) challenge,” *IJCV*, vol. 88, no. 2, pp. 303–338, Jun. 2010.

J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba, “SUN database: Large-scale scene recognition from abbey to zoo,” in *CVPR*, 2010.

P. Dollar, C. Wojek, B. Schiele, and P. Perona, “Pedestrian detec ´

tion: An evaluation of the state of the art,” *PAMI*, vol. 34, 2012.

R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In CVPR, 2014.

Kunihiko Fukushima. Neocognitron: A hierarchical neural network capable of visual pattern recognition. Neural networks, 1(2):119–130, 1988.

Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, Alexander C. Berg *(Submitted on 8 Dec 2015 (v1), last revised 29 Dec 2016 (this version, v5)).*

Kunihiko Fukushima. Neocognitron: A hierarchical neural network capable of visual pattern recognition. Neural networks, 1(2):119–130, 1988.

Tensorflow Tutorial 2: image classifier using convolutional neural network.