AN INTERNSHIP REPORT

Submitted by

PRAKHAR SHITALKUMAR PATEL

(180280116083)

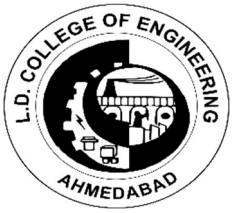
In Partial Fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

in

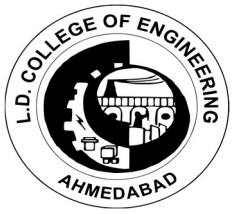
Information Technology

L. D. College of Engineering College, Ahmedabad



Gujarat Technological University, Ahmedabad

April, 2022



L. D. College of Engineering College,
Ahmedabad

CERTIFICATE

This is to certify that the Internship at **AXISRAY** has been carried out by **Prakhar Patel (180280116083)**, under my guidance in partial fulfillment of for the degree of Bachelor of Engineering in Information Technology,8th Semester of Gujarat Technological University, Ahmedabad during the academic year 2021-22.

Prof. Swati J. Patel
Internal guide

Dr. Hiteishi Diwanji
Head of Department

Axisray

January 04, 2022

To,

Prakhar Patel

Subject – **Internship offer letter**

Dear, Prakhar Patel

Axisray is pleased to offer a 3 months internship opportunity as a "**Python Intern**". This position is located in Ahmedabad, Gujarat.

Your internship will start on January 10, 2022 and will end on approximately April 11, 2022. By accepting this internship employment offer, you acknowledge that you must adhere to the company's confidentiality policy.

During the internship, you would be assigned tasks and projects that improve your understandings of the of the concepts you learned in college also based on your personal and professional skill set, and therefore you would be expected to put your best efforts in executing the tasks given to you.

Congratulations and we look forward to working with you.

Sincerely,

Axisray

405, Sigma Legacy, Above Hyundai Showroom, IIM Road, Panjrapol Cross Road, Ambawadi, Ahmedabad-380015 Gujarat, INDIA Contact@axisray.in

AXISRAY

Date: 12/04/2022

TO WHOM SO EVER IT MAY CONCERN

This is to certify that **Prakhar Patel**, a student of **L.D. College of Engineering** has successfully completed his internship in the field of **Python with Artificial Intelligence** from **10-01-2022** to **11-04-2022** (Total number of Weeks: **12**) under the guidance of **Mr. Asit Pandit.**

His internship activities include understanding a programming language and tools & technology. Also working on the different module of project and implementing the same best of knowledge.

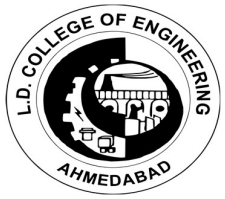
During the period of his internship program with us, he had been exposed to different processes and was found diligent, hardworking and inquisitive.

We wish him every success in his life and career.

For AXISRAY

AXISRAY AHMEDABAD

405, Sigma Legacy, Above Hyundai Showroom, Nr. Panjrapol Cross Road, IIM Road, Ambawadi, Ahmedabad-380015, Gujarat, INDIA
contact@axisray.in, wwwaxisary.in



L. D. College of Engineering,
Ahmedabad

DECLARATION

I hereby declare that the work presented in this report entitled **"Internship Report"** in partial fulfillment of the requirements for the award of the Degree of Bachelor of Engineering in Instrumentation & Control Engineering, submitted to the Information Technology Department, L. D. College of Engineering, Ahmedabad is an authentic record of my own work carried out during a period from 10th January,2022 to 11th April, 2022 under the supervision of **Prof. Swati J. Patel,** Associate Professor, Department of Information Technology Department, L. D. College of Engineering, Ahmedabad. The matter presented in this internship report has not been submitted by me for the award of any other degree of this institute or any other institute.

Name of the Student	Sign of Student
Prakhar Patel (180280116083)	P. S. Patel

ACKNOWLEDGEMENT

The success and final out comes of this internship required a lot of guidance and assistance from many people and I am extremely privileged to have got this all along the completion of my internship. All that I have done is only due to such supervision and assistance and I would not forget to thank them.

I would like to take the opportunity to thank and express my deep sense of gratitude corporate mentor **Mr. Ashit Pandit**. I am greatly indebted to them for providing their valuable guidance at all stage of the study, their advice, constructive suggestions, positivity and supportive attitude and continuous encouragement, without it would have not been possible to complete the internship. I would also like to thank **Mr. Rushit Patel**. Who in spite of busy schedule has co-operated with us continuously and indeed, his valuable contribution and guidance have been certainly indispensable for my internship work.

I heartily thank my faculty mentor of Information Technology department **Mrs. Swati Patel**. I am thankful to and fortunate enough to get constant encouragement, support and guidance from all teaching staffs of Information Technology which helped me in successfully completing my internship program. I am thankful to **Mr. Harshal Trivedi** for giving me the opportunity to work with **AXISRAY pvt ltd**. I owe my wholehearted thanks and appreciation to the entire staff of the company for their cooperation and assistance during the course of my work.

I am also extremely great full to my department staff members and friends who helped me in successful completion of this internship.

ABSTRACT

Industrial training is an important phase of a student life. A well planned, properly executed and evaluated industrial training helps a lot in developing a professional attitude. It develops an awareness of industrial approach to problem solving, based on a broad understanding of process and mode of operation of organization.

The aim and motivation of this industrial training is to receive discipline, skills, teamwork and technical knowledge through a proper training environment, which will help me, as a student in the field of Information Technology, to develop a responsiveness of the self-disciplinary nature of problems.

During a period of thirteen weeks of training at **AXISRAY Pvt. Ltd.**, I was assigned as a trainee engineer in order to execute and coordinate the company's projects in the field of Computer Vision. Throughout this industrial training, I have worked with software's such as VS Code, PyCharm, Jupyter Notebook hat required for executing python code of my daily task along with final project. I am also able to implement what I have learnt for the past years as an Information Technology student in **L. D. College of Engineering, Ahmedabad.**

A part of this, I have created a system which help to detect illegal parked cars on parking spots and determine whether car is parked properly or not which can improve the efficiency of law enforcement for vehicle parking management. This proposed project provides a concept with small scale implementation to enhance the performance of detecting cars from videos, based on an existing deep learning approach and popular model named YOLO. Also, we provide a method to estimate vehicle parking locations. The results indicate that the proposed platform can successfully identify cars and estimate their parking locations to support the management of urban parking infrastructure. This platform is applicable for videos of security patrolling, traffic-surveillance and so forth.

List of Figures

List of Abbreviations

IT	Information Technology
AI	Artificial Intelligence
NLP	Natural Language Processing
CNN	Convolutional Neural Networks
RNN	Recurrent Neural Network
SVM	Support Vector Machine
IoT	Internet Of Things
HTTP	Hyper Text Transfer Protocol
CoAP	Constrained Application Protocol
MQTT	MQ Telemetry Transport
IDE	Integrated Development Environment
CV	Computer Vision
OCR	Optical Character Recognition
HOG	Histogram of Oriented Gradients
ROI	Region Of Interest
IEEE	Institute of Electrical and Electronics Engineers
CVF	Computer Vision Forum

Table of Content

1. OVERVIEW OF THE COMPANY

1.1 COMPANY DETAILS

"AXISRAY" delivers best, cost-effective & innovative solutions to multiple & diverse business verticals such as Healthcare, SMART-HOME, E-commerce, Voice-Solutions Construction & logistics and much more.

Axisray works with a vision and our vision pitches innovation and believes that diverse & like-minded people bring the best. Axisray is a next generation service provider. AXISRAY as a Solution Provider have expertise in Artificial Intelligence, Bigdata, IoT, Cloud Computing and other. They empower & consult startup with our success pedigree and approaches to make sure that best-results are achieved.

Fig. 1.1

The most prominent focus is on data acquisition and control systems is to build & supply state-of-the-art integrated solutions using off-the-shelf available astounding technologies which provides a flexible approach to developing total, modular and cost-effective solutions.

It has achieved a unique identity as one of the leading solution providers of in IT industry Company is headed under the profound guidance of their mentor **Mr. Harshal Trivedi**, who possesses a rich experience of 15 years in the areas of Amazon Web Services (AWS) and Artificial Intelligence (AI) who wants to simplify the complexes and usage of AI and computer vision and on a mission of providing automation in Computer Vision to deliver value to everyone in an organization via a No-Code AI Computer Vision platform.

They are backed by state-of-the-art of plethora of latest technologies and have some of the foremost worldwide technical experts with us. They assist us in providing effective control and solution for Automation in all industries.

[1] http://axisray.in/

1.2 COMPANY'S VISION AND MISSION

Vision: AXISRAY is on a mission to promote automation & amend Innovation with a well-orchestrated technical stack to provide best-cost optimized Cloud services to niche startups globally. We have taken a pledged to deliver our every ounce in order to reach best possible threshold.

Mission: With our years of experience in delivering the best outcomes for global enterprises, we expertly steer our potential clients through innovation & optimized solutions. We accelerate business verticals with agile vision at global scale and deliver unprecedented milestones of results. AXISRAY's drives continuous & up-to-date swift improvement through building and passing on expertise and vision from our innovation system to the entire community.

Achievement: One of our product solutions on Tablet Press instrumentation using the LabVIEW RT platform has been highly appreciated and awarded for best innovative application solutions by National Instruments.

1.3 CLIENTS

"AXISRAY Pvt. Ltd." has clients all over the world such as the Indorama Ventures (ISRO) – Thailand, Kodestarters – USA, Radiant Labs, The Fair Food Company – Ahmedabad, TATA – Simply Smart Home – Canada and many forth.

1.4 DIFFERENT SERVICES PROVIDING BY AXISRAY

Cloud Services: AXISRAY assists organizations in cloud adoption operational activities, they help them become more agile and responsive to the changing market inscape, thus supporting them in the right decision-making and maximizing their Return on Investment (ROI) with our on-demand delivery of computing services, tools and applications.

Big Data and Business Intelligence: With AIXRAY Solutions, Harness the power of big data analytics in your current operational activities to decisively use data to uncover hidden patterns, correlations, and solve key business challenges. Our big data analytics experts understand that today's growth-hungry enterprises need a highly proactive approach and they at AXISRAY deliver every ounce of ours in achieving it.

Artificial Intelligence: They at AIXRAY are a team of entrepreneurs, engineers, machine learning experts and more, working together to advance the state of the art in AI. AXISRAY Solutions integrates Deep learning algorithms with Machine Learning frameworks and incorporating feeds and insights with our Custom Computer Vision Solutions. They regularly partner with other experts, finding ways to apply our advances to real-world challenges mentioning Natural Language Processing (NLP) in algorithms like Reinforcement Learning, Neural Networks, CNN, RNN, SVM, Regression, Classification algorithms, and more others.

Internet of Things: AXISRAY provides IoT Solutions for data collection, processing, visualization, device management and also enables device connectivity via industry standard IoT protocols - MQTT, CoAP and HTTP and supports both cloud and on-premises deployments. AXISRAY combines both scalability, fault-tolerance and performance so you will never lose your data.

Enterprise Solution Development: AXIRAY's offerings to Enterprise involves cutting-edge solutions that set a new benchmark and adapts to the changing digital world. We enable you to build future-ready products that suit your industry within the least possible time. Our Solutions Empowers Business Growth, Improve Employee Productivity, and Manage the Operations Effectively**.**

Chatbot and Voice Solutions: AXISRAY Conversational Interface Solutions allows chatbots to converse with users. Our Voice solutions are easy to use and enables you to create comprehensive custom chats, using advance workflow builder and conversational AI tools.

Video Streaming Solutions: AXISRAY facilitates Next-Gen video streaming and analytics solution when it comes to Request on-demand broadcasting solutions or integrating with established OTT channels for the Media industry. With assurance of Data compliances.

2. TOOLS AND TECHNOLOGIES LEARNED

There are different types of tools and technologies available in digital IT field such as Software Engineering, Cyber Security, Data Science, Computer Vision, Machine Learning, Artificial Intelligence, Deep Learning, Networking and so forth. But in this internship, we majorly worked on Python Programming language which helped to dive in to field of Computer Vision.

1. **Python:** Python is a high-level, general-purpose programming language. Its design philosophy emphasizes code readability with the use of significant indentation. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small- and large-scale projects.

Fig 2.1 Python.

2. **Pycharm:** PyCharm is an integrated development environment (IDE) used in computer programming, specifically for the Python programming language. It is developed by the Czech company JetBrains (formerly known as IntelliJ). It provides code analysis, a graphical debugger, an integrated unit tester, integration with version control systems, and supports web development with Django as well as data science with Anaconda.

Fig 2.2 PyCharm.

3. **VS Code:** Visual Studio Code is a streamlined code editor with support for development operations like debugging, task running, and version control. It aims to provide just the tools a developer needs for a quick code-build-debug cycle and leaves more complex workflows to fuller featured IDEs.

Fig 2.3 VS Code.

4. **Computer Vision (CV):** Computer vision is a field of artificial intelligence (AI) that enables computers and systems to derive meaningful information from digital images, videos and other visual inputs — and take actions or make recommendations based on that information.

5. **YOLO Algorithm:** YOLO is an abbreviation for the term 'You Only Look Once'. This is an algorithm that detects and recognizes various objects in a picture (in real-time). Object detection in YOLO is done as a regression problem and provides the class probabilities of the detected images. YOLO algorithm employs convolutional neural networks (CNN) to detect objects in real-time. As the name suggests, the algorithm requires only a single forward propagation through a neural network to detect objects. This means that prediction in the entire image is done in a single algorithm run. The CNN is used to predict various class probabilities and bounding boxes simultaneously.

   Features: -

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

6. **Anaconda:** Anaconda is open-source software that contains jupyter, spyder and so on that are used for large data processing, data analytics, and heavy scientific computing. Anaconda works for R and python programming languages. Spyder (sub-application of Anaconda) is used for python. Opencv for python will work in spyder. Package versions are managed by the package management system conda.

3. Computer Vision

3.1 Introduction

Computer Vision, often abbreviated as CV, is a field of artificial intelligence (AI) that enables computers and systems to derive meaningful information from digital images, videos and other visual inputs and take actions or make recommendations based on that information. It is a multidisciplinary field that could broadly be called a subfield of artificial intelligence and machine learning.

3.2 Applications of Computer Vision

- **Defect Detection:** we can detect defects such as cracks in metals, paint defects, bad prints etc. in sizes smaller than 0.05mm. Much better than the human eye! These vision cameras need an algorithm that is the "intelligent brain" which is able to differentiate what is a defect and what is not.
- **Intrusion Detection:** Modern monitoring of specified areas uses computer vision, specifically deep learning algorithms, to automatically detect intrusion events. AI vision-based event detection and recognition are applied using the real-time video stream of common CCTV surveillance cameras.
- **Self-Driving cars:** Computer vision is used to detect and classify objects (e.g., road signs or traffic lights), create 3D maps or motion estimation, and played a key role in making autonomous vehicles a reality.
- **Reading Text and barcodes:** As most products have barcodes on their packaging, a computer vision technique called OCR can be successfully applied to automatically detect, verify, convert and translate barcodes into readable text.
- **Plant Diseases Detection:** Computer vision is also widely used for the automated detection of plant diseases, especially crucial at an early stage of plant growth. Deep learning-based algorithms use image data to identify the diseases, estimate their severity and impact on yield predictions.
- **Cancer Detection:** mage recognition allows doctors to identify anomalies and changes by comparing cancerous and non-cancerous cells in images. Automated detection allows for a faster cancer diagnosis using data from magnetic resonance imaging (MRI) scans. Computer vision is already successfully applied in breast and skin cancer as well as tumor detection.

[4] https://www.v7labs.com/blog/computer-vision-applications

3.3 Fundamental Operations

3.3.1 Image Classification: Image Classification (often referred to as Image Recognition) is the task of associating one (single-label classification) or more (multi-label classification) labels to a given image.

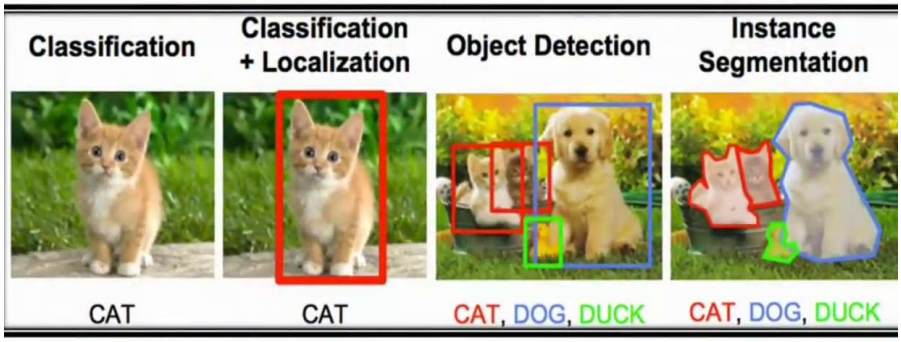


Fig 3.1 Image Classification.

Image Classification is a solid task to benchmark modern architectures and methodologies in the domain of computer vision.

Single Label Classification: Single-label classification is the most common classification task in supervised Image Classification. As the name suggests, a single label or annotation is present for each image in single-label classification.

Multi-Label Classification: Multi-label classification is a classification task where each image can contain more than one label, and some images can contain all the labels simultaneously.

3.3.2 Image Processing: Image processing is the process of transforming an image into a digital form and performing certain operations to get some useful information from it. The image processing system usually treats all images as 2D signals when applying certain predetermined signal processing methods.

Color Transformations: Color transformation, or color space conversion, is the transformation of the representation of a color from one color space to another. This calculation is required whenever data is exchanged inside a color-managed chain and carried out by a Color Matching Module.

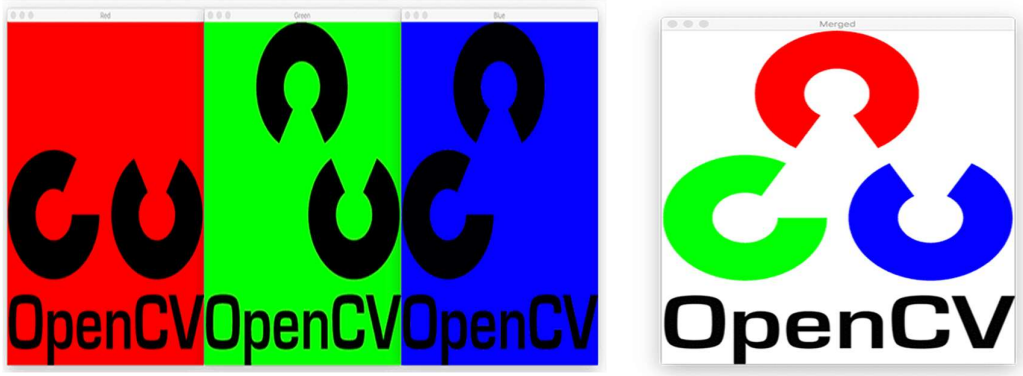


Fig 3.2 Color Transformations.

3.3.3 Image filtering: Image filtering is changing the pixel value of a specific image to blur, sharpen, emboss, or make edges more clear. It changes the appearance of the original image. Nowadays many image filters are ruling social media applications. Most people, using Instagram, maybe knowing about the image filters used in it. Image filtering is application-specific, as sometimes we need to blur the image and sometimes, we wish to have sharpness in the image.

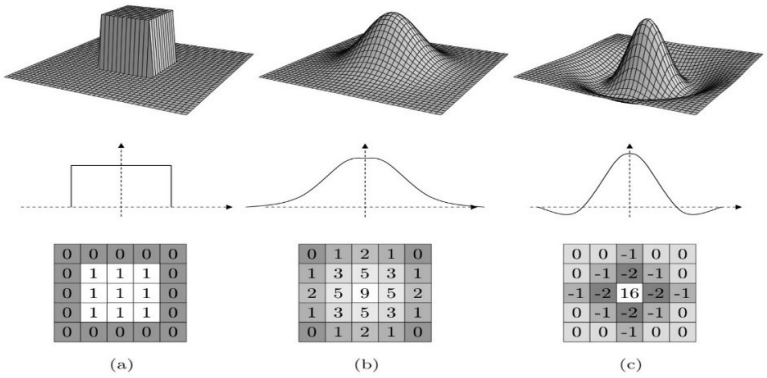


Fig 3.3 Image Filtering.

1. Gaussian Filtering: In a Gaussian blur, the pixels nearest the center of the kernel are given more weight than those far away from the center. This averaging is done on a channel-by-channel basis, and the average channel values become the new value for the pixel in the filtered image. It is done with the function, **cv2.GaussianBlur().** Gaussian filtering is highly effective in removing Gaussian noise from the image.

Fig 3.4 Gaussian Filter.

2.Median Filtering: The median filter considers each pixel in the image in turn and looks at its nearby neighbors to decide whether or not it is representative of its surroundings. Instead of simply replacing the pixel value with the *mean* of neighboring pixel values, it replaces it with the *median* of those values. The median is calculated by first sorting all the pixel values from the surrounding neighborhood into numerical order and then replacing the pixel being considered with the middle pixel value. the function **cv2.medianBlur()** computes the median of all the pixels under the kernel window and the central pixel is replaced with this median value.

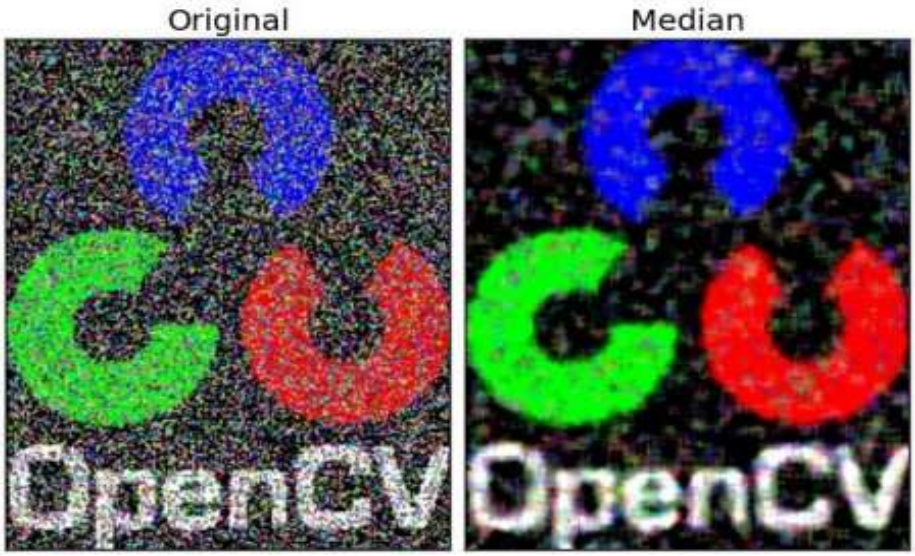


Fig 3.5 Median Filter.

3.Bilateral Filtering: The Bilateral Filter operation applies a bilateral image to a filter. You can perform this operation on an image using the **medianBlur()** method of the **imgproc** class. Following is the syntax of this method. A bilateral filter is a non-linear, edge-preserving, and noise-reducing smoothing filter for images. It replaces the intensity of each pixel with a weighted average of intensity values from nearby pixels. This weight can be based on a Gaussian distribution.

4.Averaging: Average (or mean) filtering is a method of 'smoothing' images by reducing the amount of intensity variation between neighboring pixels. The average filter works by moving through the image pixel by pixel, replacing each value with the average value of neighboring pixels, including itself.

3.3.4 Histogram Processing:

Histogram equalization is a method to process images in order to adjust the contrast of an image by modifying the intensity distribution of the histogram. The objective of this technique is to give a linear trend to the cumulative probability function associated to the image. It accomplishes this by effectively spreading out the most frequent intensity values, i.e. stretching out the intensity range of the image. This method usually increases the global contrast of images when its usable data is represented by close contrast values. This allows for areas of lower local contrast to gain a higher contrast.

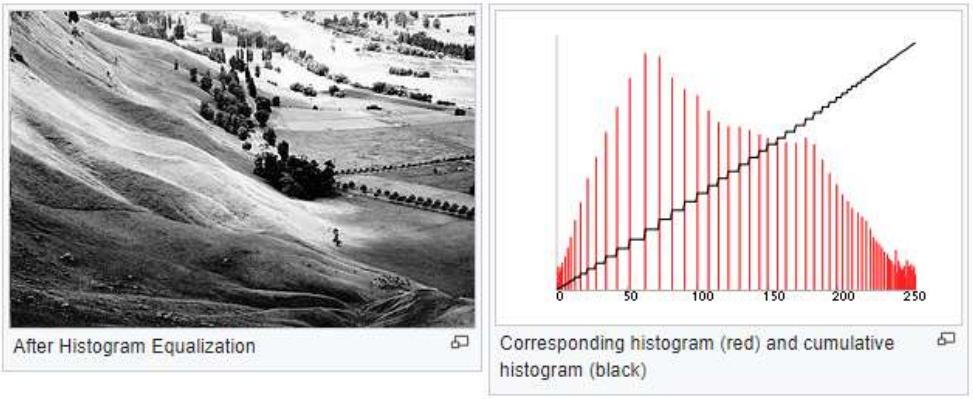


Fig 3.6 Pixel Distribution.

3.3.5 Bit-wise Operation on Images:

This includes the bitwise AND, OR, NOT, and XOR operations. Bitwise operations are performed on an image when we need to extract only the required parts of the image. Consider a situation, in which we need to extract an irregular-shaped object from an image and paste it on another image. That's exactly when we use bitwise operations on the image to separate the

foreground from the background. Bitwise operations are used to extract specific regions of interest from images by using masks. Masks can be created by performing thresholding on images.

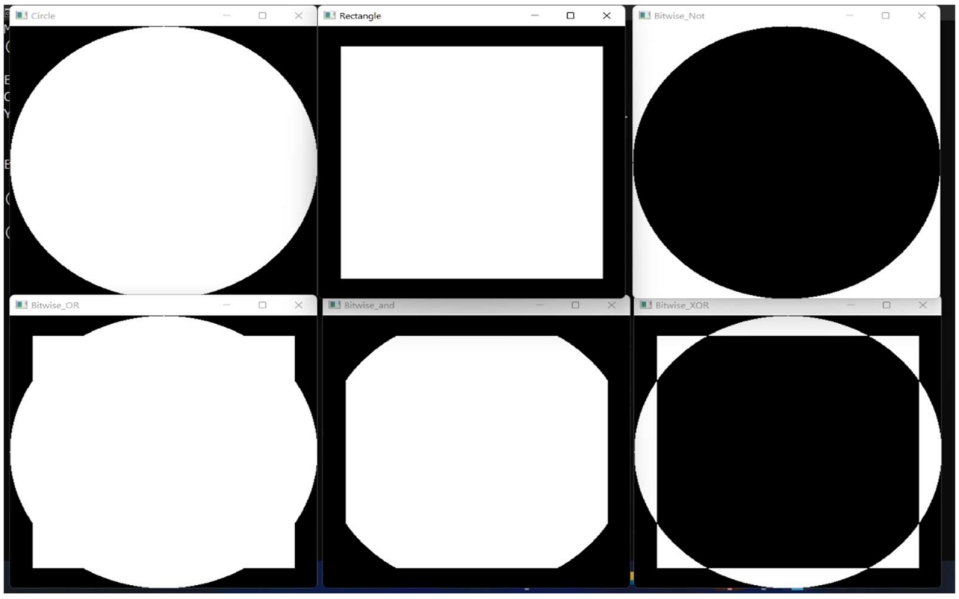


Fig 3.7 Bit-wise Operations.

3.3.6 Contours Detection: Contours are defined as the line joining all the points along the boundary of an image that are having the same intensity. Contours come handy in shape analysis, finding the size of the object of interest, and object detection. OpenCV has findContour() function that helps in extracting the contours from the image. It works best on binary images, so we should first apply thresholding techniques, Sobel edges, etc. It is often the first step for many interesting applications, such as image-foreground extraction, simple-image segmentation, detection and recognition.

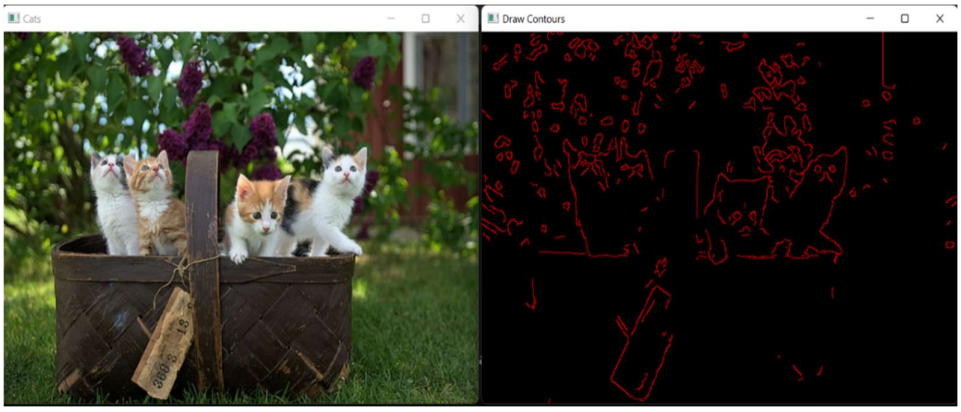


Fig 3.8 Contour Detection.

3.3.7 Edge Detection: Edge detection is a basic but important task in Computer Vision. It is usually done to reduce the amount of data yet preserving the structural properties of the objects in the image. The popular tasks that use edge detection are lane detection and converting an image into a sketch. The purpose of detecting sharp changes in image brightness is to capture important events and changes in properties of the world. It can be shown that under rather general assumptions for an image formation model, discontinuities in image brightness are likely to correspond to:

- discontinuities in depth
- discontinuities in surface orientation,
- changes in material properties and
- variations in scene illumination.

Sobel Edge Detection: The sobel is one of the most commonly used edge detectors. It is based on convolving the image with a small, separable, and integer valued filter in horizontal and vertical direction and is therefore relatively inexpensive in terms of computations. The Sobel edge enhancement filter has the advantage of providing differentiating (which gives the edge response) and smoothing (which reduces noise) concurrently.

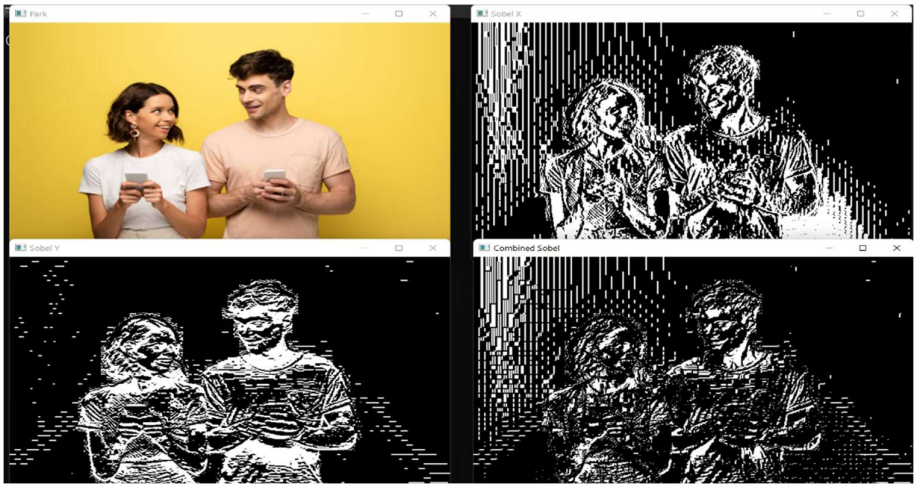


Fig 3.9 Sobel Edge Detection.

Laplacian Edge Detection: Laplacian is somewhat different from the methods we have discussed so far. Unlike the Sobel and Prewitt's edge detectors, the Laplacian edge detector uses only one kernel. It calculates second order derivatives in a single pass. Because these masks are approximating a second derivative measurement on the image, they are very sensitive to noise.

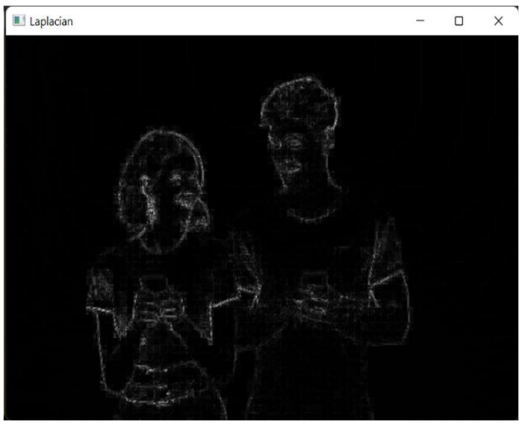


Fig 3.10 Laplacian Edge Detection.

Canny Edge Detection Operator: Canny edge detector is probably the most commonly used and most effective method, it can have it's own tutorial, because it's much more complex edge detecting method then the ones described above.

1. Smooth the image with a Gaussian filter to reduce noise.
2. Compute gradient of using any of the gradient operators Sobel or Prewitt.
3. Extract edge points: non-maximum suppression.
4. Linking and thresholding: Hysteresis.

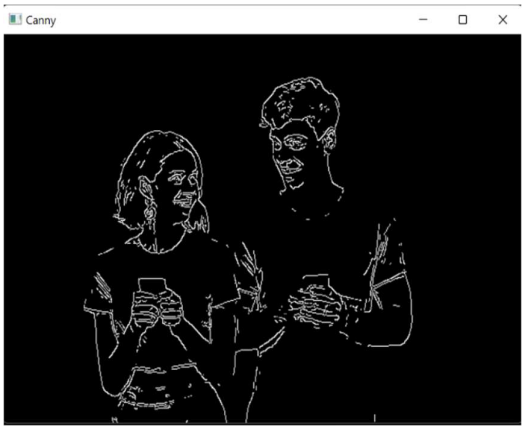


Fig 3.11 Canny Edge Detection.

3.3.8 Image Segmentation: Image segmentation is a method in which a digital image is broken down into various subgroups called Image segments which helps in reducing the complexity of the image to make further processing or analysis of the image simpler. Segmentation in easy words is assigning labels to pixels. All picture elements or pixels belonging to the same category have a common label assigned to them.

Approaches in Image Segmentation

1. **Similarity approach:** This approach is based on detecting similarity between image pixels to form a segment, based on a threshold. ML algorithms like clustering are based on this type of approach to segment an image.

2. **Discontinuity approach:** This approach relies on the discontinuity of pixel intensity values of the image. Line, Point, and Edge Detection techniques use this type of approach for obtaining intermediate segmentation results which can be later processed to obtain the final segmented image.

Fig 3.12 Image Segmentation.

3.3.9 Face Detection: Face detection is a problem in computer vision of locating and localizing one or more faces in a photograph**.** *Face detection* is the crucial part of face recognition determining the number of faces on the picture or video without remembering or storing details. It may define some demographic data like age or gender, but it cannot recognize individuals.

Different Techniques:

- **Haar Cascades Classifier:** The first machine learning-based cascading classifier to fulfill the requirement of fast implementation on low-power CPUs, such as cameras and phones.
- **Histogram of Oriented Gradients (HOG):** Histogram of Oriented Gradient is used as features followed by linear Support Vector Machine classifier.
- **Multi-task Cascaded Convolutional Networks (MTCNN):** Deep learning-based face detection technique.

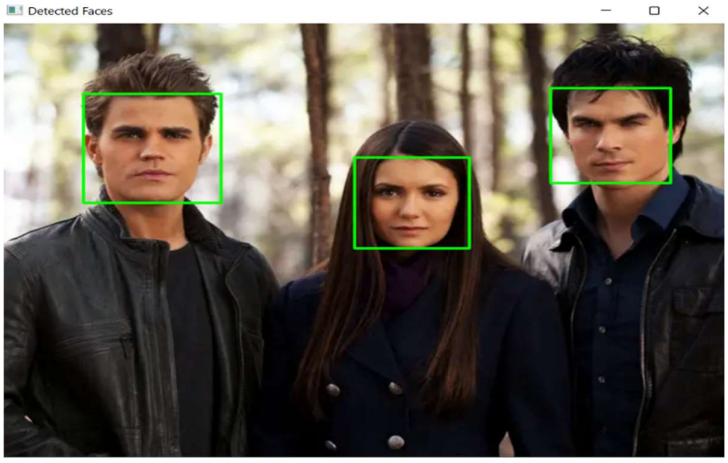


Fig 3.13 Face Detection.

3.3.10 Morphological Operations: *Morphology* is a broad set of image processing operations that process images based on shapes. Morphological operations apply a structuring element to an input image, creating an output image of the same size. In a morphological operation, the value of each pixel in the output image is based on a comparison of the corresponding pixel in the input image with its neighbors.

Terminologies in Morphological Operations:

1. **Structuring Element:** It is a matrix or a small-sized template that is used to traverse an image. The structuring element is positioned at all possible locations in the image, and it is compared with the connected pixels. It can be of any shape.
2. **Fit**: When all the pixels in the structuring element cover the pixels of the object, we call it Fit.
3. **Hit**: When at least one of the pixels in the structuring element cover the pixels of the object, we call it Hit.
4. **Miss**: When no pixel in the structuring element covers the pixels of the object, we call it miss.

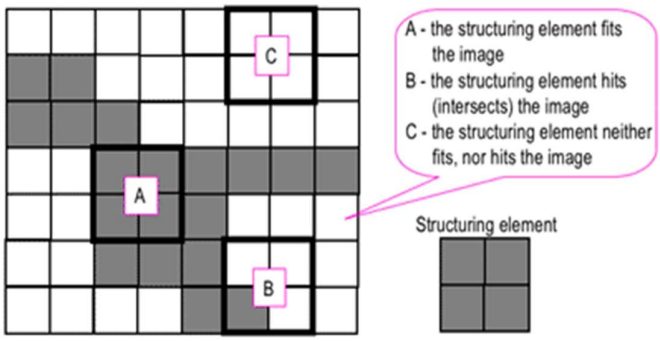


Fig 3.14 Structuring Element.

There are mainly two morphological operations:

1. Erosion: Erosion shrinks the image pixels, or erosion removes pixels on object boundaries.

2. Dilation: Dilation expands the image pixels, or it adds pixels on object boundaries.

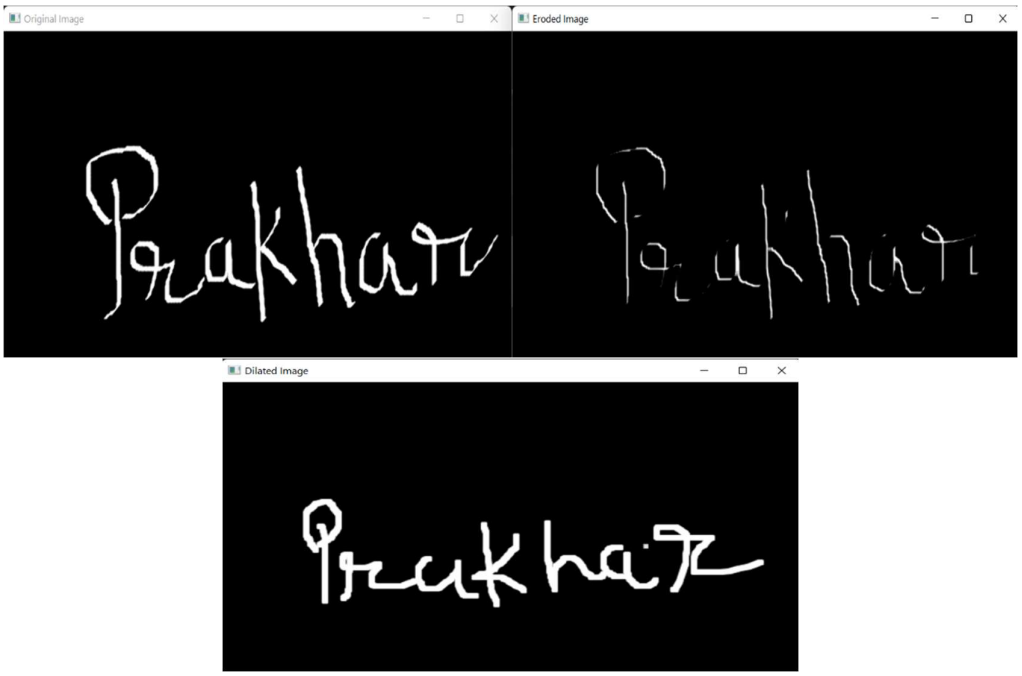


Fig 3.15 Erosion & Dilation.

3. **Opening**: Opening is just another name of erosion followed by dilation. **It is** useful in removing noise

4. **Closing:** Closing is reverse of Opening, Dilation followed by Erosion. It is useful in closing small holes inside the foreground objects, or small black points on the object.

Fig 3.16 Opening & Closing.

3.3.11 Color Spaces: Color spaces are different types of color modes, used in image processing and signals and system for various purposes. Some of the common color spaces are:

1. RGB (Red, Green, Blue)
2. BGR (Blue, Green, Red)
3. Gray-Scale
4. HSV (Hue, Saturation, Value)
5. CMY (cyan, magenta, yellow)
6. CMYK (cyan, magenta, yellow, black)

We can convert images from one color spaces to another color spaces for better efficiency for major operation for better output.

4. OBJECT DETECTION

4.1 INTRODUCTION[5]

Object Detection is a computer technology related to computer vision, image processing, and deep learning that deals with detecting instances of objects in images and videos. Problem statement of Object Detection can be stated as: to determine the locations of the objects and the classes to which it belongs to. Object detection has been determined the numerous applications in computer vision such as object tracking, retrieval, video surveillance, image captioning, Image segmentation, Medical Imagine and several greater number other applications as well.

Object detection can be categorized as follows:

1. **Single-Stage Object Detection:** This particular family of detectors removes the ROI (Region of Interest) extraction process. In fact, it directly classifies and regresses the boxes. Derives from it that this family of detectors is used when inference and detection speed play an important factor, with a reduction in terms of accuracy.
2. **Two-Stage Object Detection:** As the name states, it divides the object detection task into two stages: the ROI (Region of Interest) is extracted from the image, classified and then ROIs are regressed. The most famous architectures include: R-CNN, Fast-RCNN, Faster-RCNN, Mask-RCNN.

4.2 Object Detection Using Pre-Trained Model YOLO

Simply put, a pre-trained model is a model created by someone else to solve a similar problem. Instead of building a model from scratch to solve a similar problem, you use the model trained on other problem as a starting point.

For example, if we want to build a self-learning car. we can spend years to build a decent image recognition algorithm from scratch or we can take inception model from Google which was built on ImageNet data to identify images in those pictures.

There are plethora of pre-trained deep-learning based object detection models are available with their unique and effective functionalities. But in this internship, we worked on one of the most popular models named **YOLO**.

4.2.1 YOLO

YOLO - **You Only Look Once** is an algorithm proposed by Redmond et. al in a research article published at the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) as a conference paper, winning OpenCV People's Choice Award. Compared to the approach taken by object detection algorithms before YOLO, which repurpose classifiers to perform detection, YOLO proposes the use of an end-to-end neural network that makes predictions of bounding boxes and class probabilities all at once.

Yolo Architecture:

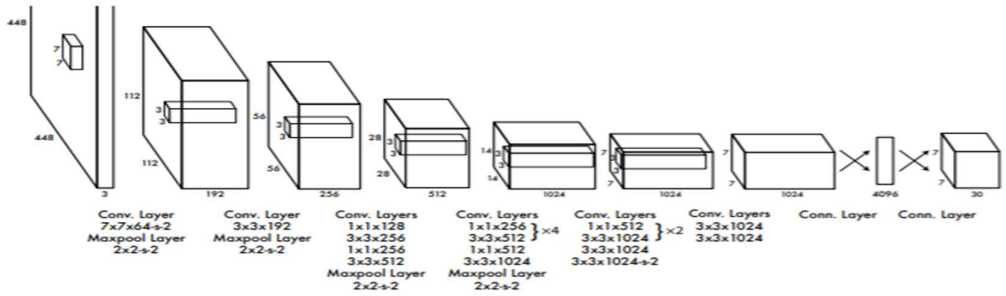


Fig 4.1 Yolo Architecture.

YOLO is a convolution neural network. It consists of a total of 24 convolutional layers and followed by 2 fully connected layers. Each layer has its own importance and the layers are separated by their functionality.

How Yolo Works

YOLO algorithm uses a completely different approach. The algorithm applies a single neural network to the entire full image. Then this network divides that image into regions which provides the bounding boxes and also predicts probabilities for each region. These generated bounding boxes are weighted by the predicted probabilities.

Applications of YOLO

1. Autonomous Driving.
2. Security.
3. Object Detection.

Car detection Using YOLO

Using YOLO, we have achieved our primary task of our final internship project which was detecting car from videos and images to move on to further phases of our project.

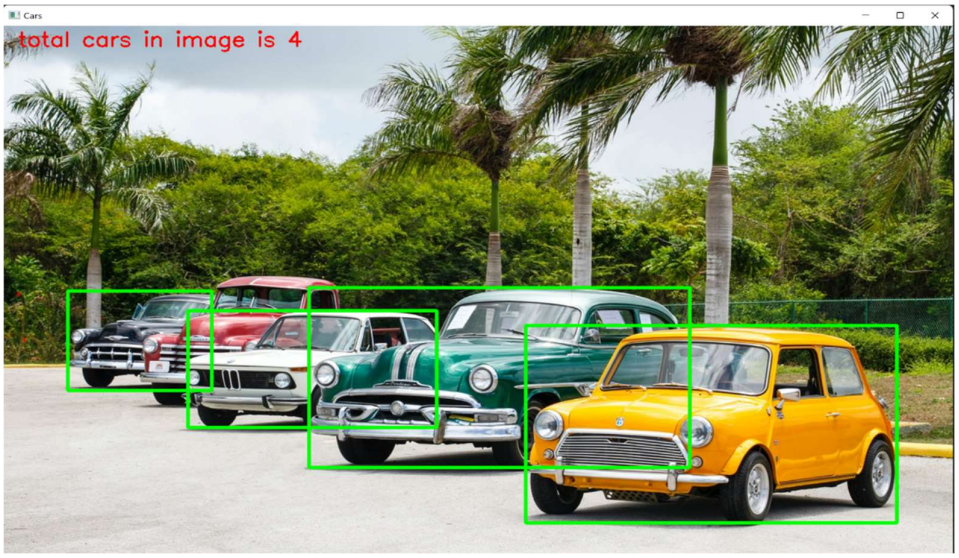


Fig 4.2 Car Detection.

5. FINAL INTERNSHIP PROJECT DETAILS

5.1 ABSTRACT

Recent advances in the technologies into all aspects of modern life caused demand for advanced security and monitoring systems. Video surveillance system is used for home security, border security, ATM security, effective traffic monitoring, toll collection, etc. Automated systems are designed for numerous applications for monitoring and security, but the task of detecting illegally parked vehicle at no parking area has been left largely to the human.

This project proposes simple and small-scale approach of identifying whether the car is parked in the parking spot properly or not and used image processing approach for detection and identification of illegally parked vehicles.

The main lug panel is similar to the one above, with the distinction that there's no main breaker. Sometimes these types of panels are used as sub-panels. The line wires with this kind of panel run to a lug, which is a type of electrical connector.

5.2 PURPOSE OF THE PROJECT

The purpose of our project "Illegal Parking Detection" is to find about the various vehicles which are parked in a random manner anywhere around the parking space. As all of them have observed that a lot of vehicles are parked randomly at railway stations and other public spaces, we decided to find a solution to this messy problem.

5.3 OBJECTIVE OF THE PROJECT

The objective of our project is to validate whether a vehicle has been parked in the correct place. It is a user-friendly system for all users. The system is designed to handle all the primary information about parking spaces in detail. The user will be able to see if he has parked haphazardly or in a proper manner.

5.4 SCOPE OF THE PROJECT

Illegal parking detection provides with the key feature which decides whether a person has parked his vehicle in the right spot. By using this detection method, he can easily determine whether he has parked in a correct spot or not.

5.5 CODING STANDARDS

They are a series of procedures that can be defined for a particular programming language specifying a programming style, the methods, & different procedures. These procedures can be for various aspects of the program written in that language. They can be considered essential attributes of software development.

A coding standard makes sure that all the developers working on the project are following certain specified guidelines. The code can be easily understood and proper consistency is maintained.

5.5.1 File Naming Convention.

1. File names use proper letters only.
2. File names should have its first letter small cap.
3. All code and text files should be completely free of TAB characters.
4. All indentation should be done with space.

5.5.2 Creating the source code

1. We first require adding the purpose in all the files at the beginning in the comment.
2. Class files should be coded correctly and objects also should be declared correctly.

5.5.3 General Coding Standard

1. It is acceptable to break up long statements into multiple lines.
2. Do not put multiple statements on one line.
3. Every coding block should be intended.
4. All the variables and constant declaration should be at the top of the source code in the particular form or module or class module.
5. Deep nesting structure should be avoided.
6. Do Proper organization of file and affiliated folders.

5.5.4 Variable standard

1. Meaningful and understandable variables name helps anyone to understand the reason of using it.
2. Local variables should be named using camel case lettering starting with small letter (e.g., localData) whereas Global variables names should start with a capital letter (e.g., GlobalData). Constant names should be formed using capital letters only (e.g., CONSDATA).
3. Variables are formed from letters A-Z, a-z, digits 0-9 and underscore_.
4. It is better to avoid the use of digits in variable names.
5. Variable names consist of a data type used in it and they are case sensitives.

5.5.5 Programming Conventions

We have listed below some general conventions to be followed in programming.

- Statements: We have returned only one statement per line.
- Spacing: We have typed a space before and after all operators such as (+, *, 1, <, =) and the assignment symbols (=).
- Indenting: Improve the readability of script by using tabs to indent the body of statements such as these: DO, FOR, IF.

Also use a tab to indent statements continued onto a new line. The coding standard is well defined and standard style of coding. With the help of coding standards any person can find the exact meaning of code and understand the working of the system. A coding standard includes the way the variables are to be named, the code that is to be laid out, the comments that are to be described etc.

General Coding Standards

The coding standard we have used is Camel case that includes proper naming and programming conventions. It includes the first letter in the identifier as a small letter and first letter of each subsequent concatenated word as capitalized.

File Naming Conventions

The files created in the project are named in such a way that the first letter is capitalized. The files in the same module are grouped together in a folder with

appropriate name. This makes it easier for the developer to navigate to the appropriate file. Thus, it is a good coding technique that enables faster access to the files.

Comment Standards

The other technique used for programming is comments. The few lines of comments make the code more elegant. Sometimes it makes the code easier to understand and works as a help for the programmers.

5.5.6 Try to formalize Exception Handling

'Exception' refers to problems, issues, or uncommon events that occur when code is run and disrupt the normal flow of execution. This either pauses or terminates program execution, which is a scenario that must be avoided. However, when they do occur, use the following techniques to minimize damage to overall execution in terms of both time and dev effort:

1. Keep the code in a try-catch block when you think code might throw any kind of exception.
2. Ensure that auto recovery has been activated and can be used.
3. Consider that it might be an issue of software/network slowness. Wait a few seconds for the required elements to show up.

5.5.7 Advantages of Implementing Coding Standards

1. Enhanced Efficiency.
2. Risk of project failure is reduced.
3. Minimal Complexity.
4. Easy to maintain.
5. Bug Rectification.
6. Cost-Effective.

6. PROJECT IMPLEMENTATION AND TESTING

6.1 TESTING PLAN

A test plan is the cornerstone of a successful testing implementation. The testing plan represents the overall approach to the test. In many ways, the test plan serves as a summary of the test activities that will be performed. It shows how the tests will be organized, and outlines all of the tester's needs that must be met in order to properly carry out the test. The goal of test planning is to establish the list of tasks that, if performed, will identify all of the requirements that have not been met in the software. There are many standards that can be used for developing test plans. Early in the deployment planning phase, the testing effort, and identifies the methodology that your team will use to conduct tests.

It also identifies the hardware, software, and tools required for testing and the features and functions that will be tested. A well-rounded test plan notes any risk factors that jeopardize testing and includes a testing schedule. So, I can say that Test Planning details the activities, dependencies and effort required to conducting the system test.

6.2 TESTING STRATEGY

1. The test strategy is a formal description of how a software product will be tested. A test strategy is developed for all levels of testing, as required. The test team analyzes the requirements, writes the test strategy and reviews the plan with the project team.
2. The test plan may include test cases, conditions, and the test environment, a list of related tasks, pass/fail criteria and risk assessment.
3. The purpose of the testing strategy is to define the overall context for the entire testing process. The process is different depending on the specific characteristics of your solution. In many respects, this is the most important part of the testing process, since all future testing decisions will be made within the context of the strategy.
4. As a programmer, we have to just do a unit testing which is a part of White Box testing. Other type of the testing in each phase of the software is done by testing department.
5. Unit Testing begins at the vortex of the spiral and concentrates on each unit (i.e., component) of the software as implemented in source code.

6.3 IMPLEMENTATION

In this strikingly knowledgeable internship, we have built two major projects.

1. Illegal Parking Detection. (Primary)
2. Mask Detection. (Secondary)

1. Illegal Parking Detection

```
import os
import random
import time
import cv2
import imutils
import numpy as np

UPLOAD_INPUT_CAMERA = '/adminResources/input/Car4.mp4'
UPLOAD_OUTPUT_CAMERA = '/adminResources/output/'

name = UPLOAD_INPUT_CAMERA.split("/")[2].replace("mp4", "")
cameraOutputFileName = name + '_detected{}.webm'.format(random.randrange(0, 9))
default_confidence = 0.5
default_threshold = 0.3

inputVideo = UPLOAD_INPUT_CAMERA
outputVideo = UPLOAD_OUTPUT_CAMERA + cameraOutputFileName

# load the COCO class labels our model was trained on
labelsPath = 'adminResources/models/coco.names'
LABELS = open(labelsPath).read().strip().split("\n")
print(LABELS)

# initialize a list of colors to represent each possible class label
np.random.seed(42)
COLORS = np.random.randint(0, 255, size=(len(LABELS), 3), dtype="uint8")

# derive the paths to the model weights and model configuration
weightsPath = 'D:\illegal_car_parking\illegal\\adminResources\models\yolov4.weights'
configPath = 'D:\illegal_car_parking\illegal\\adminResources\models\yolov4.cfg'
# load our model object detector trained on COCO dataset (80 classes)
# and determine only the *output* layer names that we need from model
print("[INFO] loading model from disk...")
net = cv2.dnn.readNetFromDarknet(configPath, weightsPath)
ln = net.getLayerNames()
ln = [ln[i - 1] for i in net.getUnconnectedOutLayers()]
data_list = []
file = inputVideo.split(os.sep)[-1].split(".")[0]
parking_coordinate = "D:\illegal_car_parking\illegal\adminResources\parking-coordinate"
# f = open("{}{}{}.txt".format(parking_coordinate, os.sep, file), "r")
f=open('D:\illegal_car_parking\illegal\\adminResources\parking-coordinate\\test1.txt','r')
for lines in f:
    lines = eval(lines)
    data_list.append(lines)


def check_point_intersection(x1, y1, x2, y2, x, y):
    """
        :param x1: Roi top left x Coordinate
        :param y1: Roi top left y Coordinate
        :param x2: Roi bottom right x Coordinate
        :param y2: Roi bottom right x Coordinate
        :param x: Object center point x Coordinate
```

```
        :param y: Object center point y Coordinate
        :return: If the center point lies within ROI. Returns True or False
    """

    if x > x1 and x < x2 and y > y1 and y < y2:
        return True
    else:
        return False


# initialize the video stream, pointer to output video file, and
# frame dimensions
vs = cv2.VideoCapture('D:\illigal_car_parking\illigal\\adminResources\input\\test1.mp4')

frame_width = int(vs.get(3))
frame_height = int(vs.get(4))
size = (frame_width, frame_height)
result = cv2.VideoWriter('output.avi',
                         cv2.VideoWriter_fourcc(*'MJPG'),
                         10, size)
# frame_width = int(vs.get(3))
# print(frame_width)
# frame_height = int(vs.get(4))
fps = 10  # int(vs.get(5))
writer = None
(W, H) = (None, None)
count = 0
frame_rate = 5

# try to determine the total number of frames in the video file
try:
    prop = cv2.cv.CV_CAP_PROP_FRAME_COUNT if imutils.is_cv2() \
        else cv2.CAP_PROP_FRAME_COUNT
    total = int(vs.get(prop))
    print(total)
    print("[INFO] {} total frames in video".format(total))

# an error occurred while trying to determine the total
# number of frames in the video file
except:
    print("[INFO] could not determine # of frames in video")
    print("[INFO] no approx. completion time can be provided")
    total = -1
# loop over frames from the video file stream
while True:
    # read the next frame from the file
    grabbed, frame = vs.read()
    # if the frame was not grabbed, then we have reached the end
    # of the stream
    if not grabbed:
        break
    # if the frame dimensions are empty, grab them
    if W is None or H is None:
```

```
    (H, W) = frame.shape[:2]

# construct a blob from the input frame and then perform a forward
# pass of the Model object detector, giving us our bounding boxes
# and associated probabilities
blob = cv2.dnn.blobFromImage(frame, 1 / 255.0, (416, 416), swapRB=True,
                             crop=False)
net.setInput(blob)
start = time.time()
layerOutputs = net.forward(ln)
end = time.time()

# initialize our lists of detected bounding boxes, confidences,
# and class IDs, respectively
boxes = []
confidences = []
classIDs = []
for i in data_list:
    cv2.rectangle(frame, (i[0], i[1]), (i[2], i[3]), (0, 255, 0), 2)

# loop over each of the layer outputs
for output in layerOutputs:
    # loop over each of the detections
    for detection in output:
        # extract the class ID and confidence (i.e., probability)
        # of the current object detection
        scores = detection[5:]
        classID = np.argmax(scores)
        confidence = scores[classID]

        # filter out weak predictions by ensuring the detected
        # probability is greater than the minimum probability
        if confidence > default_confidence:
            # scale the bounding box coordinates back relative to
            # the size of the image, keeping in mind that Model
            # actually returns the center (x, y)-coordinates of
            # the bounding box followed by the boxes' width and
            # height
            box = detection[0:4] * np.array([W, H, W, H])
            (centerX, centerY, width, height) = box.astype("int")

            # use the center (x, y)-coordinates to derive the top
            # and and left corner of the bounding box
            x = int(centerX - (width / 2))
            y = int(centerY - (height / 2))

            # update our list of bounding box coordinates,
            # confidences, and class IDs
            boxes.append([x, y, int(width), int(height)])
            confidences.append(float(confidence))
            classIDs.append(classID)

# apply non-maxima suppression to suppress weak, overlapping
```

```
        # bounding boxes
        idxs = cv2.dnn.NMSBoxes(boxes, confidences, default_confidence,
                                default_threshold)

        # ensure at least one detection exists
        if len(idxs) > 0:
            # loop over the indexes we are keeping
            for i in idxs.flatten():
                # extract the bounding box coordinates
                (x, y) = (boxes[i][0], boxes[i][1])
                (w, h) = (boxes[i][2], boxes[i][3])
                center_x = (x + w // 2)
                center_y = (y + h // 2)

                # draw a bounding box rectangle and label on the frame
                color = [int(c) for c in COLORS[classIDs[i]]]
                if LABELS[classIDs[i]] == "car":
                    cv2.circle(frame, (center_x, center_y), 1, (255, 0, 0), 2)
                    cv2.rectangle(frame, (x, y), (x + w, y + h), color, 2)
                    text = "{}".format(LABELS[classIDs[i]])
                    cv2.putText(frame, text, (x, y - 5), cv2.FONT_HERSHEY_SIMPLEX,
                                0.5, color, 2)
                    for i in data_list:
                        if check_point_intersection(i[0], i[1], i[2], i[3],
                                                    center_x, center_y):
                            cv2.rectangle(frame, (i[0], i[1]), (i[2], i[3]),
                                          (0, 0, 255), 2)

        # check if the video writer is None
        if writer is None:
            # initialize our video writer
            fourcc = cv2.VideoWriter_fourcc(*"VP80")
            writer = cv2.VideoWriter(outputVideo, fourcc, fps,
                                     (frame.shape[1], frame.shape[0]), True)
            # some information on processing single frame
            if total > 0:
                elap = (end - start)
                print("[INFO] single frame took {:.4f} seconds".format(elap))
                print("[INFO] estimated total time to finish: {:.4f}".format(
                    elap * total))

        # write the output frame to disk
        writer.write(frame)
        cv2.imshow('frame', frame)
        result.write(frame)
        count += frame_rate
        vs.set(1, count)
        if cv2.waitKey(1) & 0xFF == ord('q'):
            break

    # release the file pointers
    print("[INFO] cleaning up...")
    # writer.release()


    vs.release()
    cv2.destroyAllWindows()
```

2. **Mask Detection**

```
import cv2
from mtcnn import MTCNN
import torch
import torchvision.transforms as transforms
import argparse
from PIL import Image

class MaskDetector:

    def __init__(self, modelPath):
        self.modelPath = modelPath
        self.setup = True

    def maskSetup(self):
        if self.setup:
            checkpoint = torch.load(self.modelPath, map_location='cpu')
            model = checkpoint['model']
            model.load_state_dict(checkpoint['state_dict'])
            for parameter in model.parameters():
                parameter.requires_grad = False

            self.model = model.eval()

            self.faceDetector = MTCNN()

            self.transforms = transforms.Compose([
                transforms.Resize((224, 224)),
                transforms.ToTensor(),
                transforms.Normalize((0.5, 0.5, 0.5), (0.5, 0.5, 0.5))
            ])

            self.setup = False

    def maskProcess(self, frame):
        image = frame.copy()
        displayImg = frame.copy()
        image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
        results = self.faceDetector.detect_faces(image)
        faces = []
        font_scale = 1
        thickness = 2
        font = cv2.FONT_HERSHEY_SIMPLEX

        for i in results:
            faces.append(i['box'])

        for (x, y, w, h) in faces:
            x, y = max(0, x), max(0, y)
            cv2.rectangle(frame, (x, y), (x + w, y + h), (255, 0, 0), 2)
            cropped_img = frame[y:y + h, x:x + w, :]
            temp_image = Image.fromarray(cropped_img, mode="RGB")
            temp_image = self.transforms(temp_image)
            image = temp_image.unsqueeze(0)
```

```
            result = self.model(image)
            _, maximum = torch.max(result.data, 1)
            prediction = maximum.item()

            if prediction == 0:
                cv2.putText(displayImg, "Masked", (x, y - 10), font, font_scale, (0, 255, 0),
thickness)
                cv2.rectangle(displayImg, (x, y), (x + w, y + h), (0, 255, 0), 2)
            elif prediction == 1:
                cv2.putText(displayImg, "No Mask", (x, y - 10), font, font_scale, (0, 0, 255),
thickness)
                cv2.rectangle(displayImg, (x, y), (x + w, y + h), (0, 0, 255), 2)

        self.outputFrame = displayImg.copy()

    def maskDisplay(self):
        return self.outputFrame

if __name__ == "__main__":

    def resize(image, width):
        newShape = None
        (h, w) = image.shape[:2]
        r = width / float(w)
        dim = (width, int(h * r))
        resized = cv2.resize(image, dim, interpolation=cv2.INTER_AREA)
        return resized


    parser = argparse.ArgumentParser()
    parser.add_argument('source', nargs='?', default="live",
                        help="Enter 'video' if you want to run the detector on a video.")
    args = parser.parse_args()

    if args.source.lower() == 'video':
        videoMode = True
        videoPath = input('Enter video path, (example: /media/video/hi.mp4) : ')
        if videoPath[0] == "":
            videoPath = videoPath[1:-1]
        writer = None
        opname = "output.avi"
    else:
        videoMode = False
        videoPath = 0
        writer = "Not required"
        opname = None

    cap = cv2.VideoCapture(videoPath)
    fno = 0
    modelPath = "models/custom_model.pth"
    detectorObj = MaskDetector(modelPath)
```

```
while True:

    ret, frame = cap.read()
    if not ret:
        break

    currentImg = frame.copy()
    currentImg = resize(currentImg, width=480)
    imageShape = currentImg.shape
    fno += 1

    if fno % 1 == 0 or fno == 1:
        detectorObj.maskSetup()
        detectorObj.maskProcess(currentImg)
        outputFrame = detectorObj.maskDisplay()

        if writer is None:
            print("Writing the output file to: ", opname)
            fourcc = cv2.VideoWriter_fourcc(*"MJPG")
            writer = cv2.VideoWriter(opname, fourcc, 30,
                                     (outputFrame.shape[1], outputFrame.shape[0]), True)

        cv2.imshow('Mask Detection Dashboard', outputFrame)

    if videoMode:
        writer.write(outputFrame)

    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

# print("Full time taken {} minutes".format((fulltock-fulltick)/60))

# When everything done, release the capture
cap.release()
cv2.destroyAllWindows()
```

7 PROJECT OUTPUTS / SCREENSHOTS

7.1 ILLEGAL PARKING DETECTION

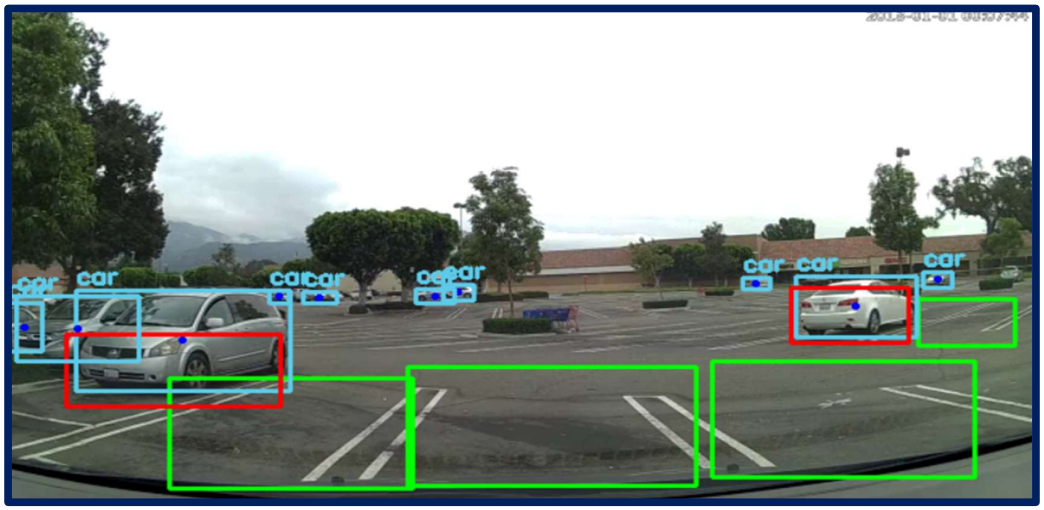


Fig.7.1 Illegal Parking Detection Output 1.

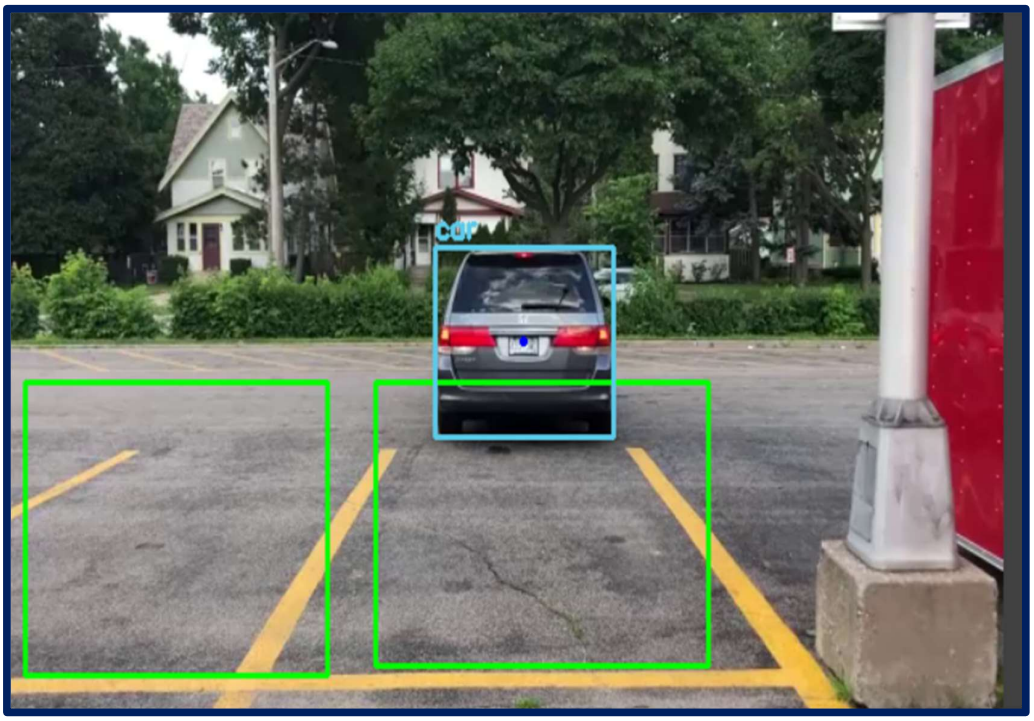


Fig.7.2 Illegal Parking Detection Output 2.

7.2 MASK DETECTION

Fig 7.3 Mask Detected.

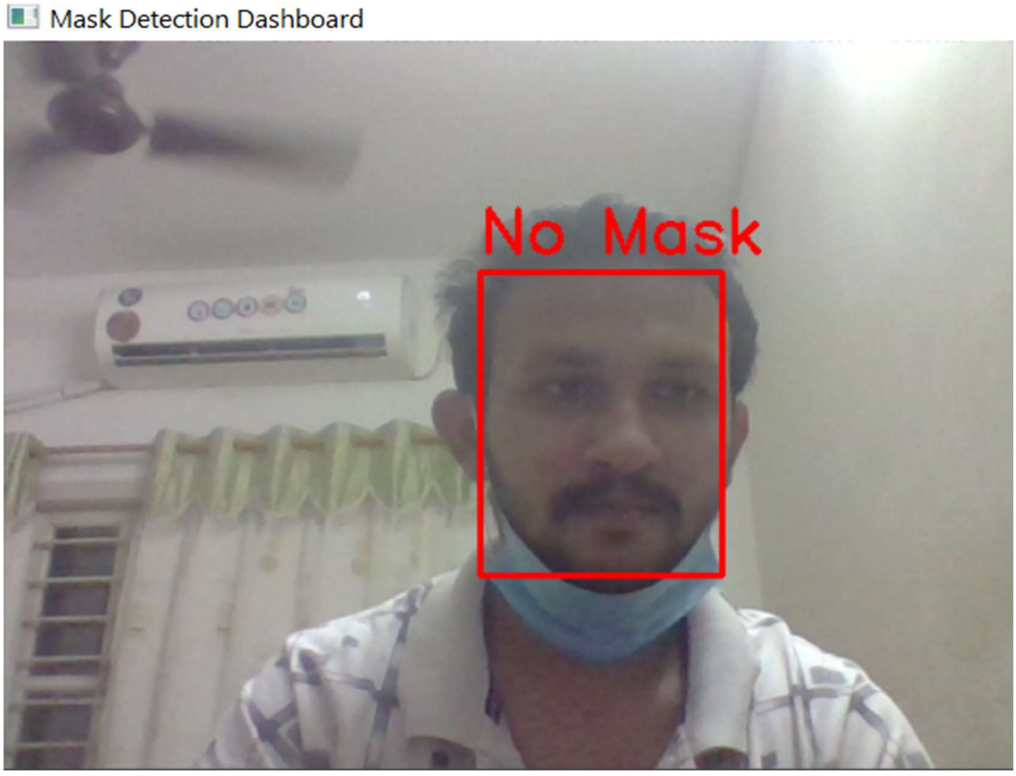


Fig 7.4 No Mask Detected

8 CONCLUSION AND FUTURE PLANS

8.1 CONSLUSION

My industrial project has been one of the most interesting, productive and instructive experience in my life. Through this journey of building of this project, I have gained new insight and more comprehensive understanding about the real industrial working condition and practice, it also improved teamwork communication and functional skills. All these valuable experiences and knowledge's that I have gained were not only acquired through the direct involvement in task but also through other aspects of training such as: work observation, interaction with collogues, supervisors and other people related to the field. I am sure that industrial project has achieved its primary objectives. As result of this project completion, I am more confident to build my future career. I would be able to acquire the above mention skills which would help me to boost my professional career as a Computer Engineer in any firm.

8.2 FUTURE PLANS

With the existing constraints, the developed system was what we planned initially. The primary aim of this project has been met. All the objectives that were set out have been completed and giving positive results in the ends. Future changes in the model/application will make this project more reliable and will become interactive in future.

Talking about my future goals, after I am done with internship, I hope to take everything I learned with me and apply it to my career in the future. I would most probably try to apply for my post-graduate studies in Canada which will enhance knowledge of my current field and prepare me to work in "Big Four" companies. However, I honestly do not mind about the size of my future company, as long as I learn something at the end of the day.

I must also say that I have adapted a different approach in my studies after going through internship, which I believe will be beneficial to me if I ever decide to pursue further studies in the future.

9 REFERENCES

1. TUTORIALS

https://youtu.be/oXlwWbU8l2o
https://youtu.be/N81PCpADwKQ
https://youtu.be/01sAkU_NvOY
https://youtu.be/PAuco4ehQUE
https://youtu.be/WQeoO7MI0Bs

2. ARTICLES AND BLOGS

https://docs.opencv.org/4.x/
https://medium.com/analytics-vidhya/introduction-to-computer-vision-opencv-in-python-fb722e805e8b
https://medium.com/@rinu.gour123/ai-python-computer-vision-tutorial-with-opencv-b7f86c3c6a1a
https://www.analyticsvidhya.com/blog/2021/07/performing-computer-vision-task-with-opencv-and-python/
https://www.javatpoint.com/opencv
https://medium.com/analytics-vidhya/computer-vision-and-image-processing-with-opencv-8868876618c3
https://medium.com/analytics-vidhya/introduction-to-image-processing-with-opencv-2a1bbef8eda0
https://towardsdatascience.com/guide-to-car-detection-using-yolo-48caac8e4ded
https://medium.com/analytics-vidhya/object-detection-using-yolo-and-car-detection-implementation-1ec79e882875

THANK YOU