

LIVE CAMERA HELMET DETECTION

Submitted in partial fulfilment of the required for the award of

Bachelors of Computer Science

by

S. DIVAKARAN

(REGISTER NO 40290025)



DEPARTMENT OF COMPUTER SCIENCE

SCHOOL OF SCIENCE AND HUMANITIES

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

**Accredited with Grade “A” by NAAC | 12B Status by UGC|
Approved by AICTE**

JEPPIAR NAGAR, RAJIV GANDHI SALAI, CHENNAI 600 199

APRIL 2023



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited "A" Grade by NAAC | 12B Status by UGC | Approved by AICTE

www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **S. DIVAKARAN (40290025)** who carried out the project entitled "LIVE CAMERA HELMET DETECTION" under our supervision from AUGUST 2022 TO NOVEMBER 2022.

INTERNAL GUIDE

Dr.V.ULAGAMUTHALVI, M.E.,Ph.D.

DEAN AND HEAD OF THE DEPARTMENT

Dr.REKHA CHAKRAVARTHI M.E.,Ph.D

Submitted for Viva voce Examination held on 08.05.2023

Internal Examiner

External Examiner

DECLARATION

I am, **S. DIVAKARAN (40290025)** here by declare that the Project Report entitled “**LIVE CAMERA HELMET DETECTION**” done by us under the guidance of **MS.VINODHINI M.sc.**, Department of Computer Science at **SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY**, Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai-600 119 is submitted in partial fulfillment of the requirements for the award of Master of Science degree in Computer Science



DATE: 8.05.2023

PLACE: CHENNAI

S.DIVAKARAN

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to Board of Management of **SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. REKHA CHAKRAVARTHI M.E., Ph.D. Dean of Arts and Science** for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my project guide **Ms. VINODHINI.K M.sc.**, for their valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff member of the Department of Computer Science who were helpful in many ways for the completion of the project.

S.DIVAKARAN

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO
	ABSTRACT	
1	INTRODUCTION	
	PROJECT OVERVIEW	
2	LITERATURE SURVEY	
3	SYSTEM ANALYSIS	
	3.1 FEASIBILITY STUDY	
	3.2 ECONOMIC FEASIBILITY	
	3.3 TECHNICAL FEASIBILITY	
	3.4 SOCIAL FEASIBILITY	
	3.5 OPERATIONAL FEASIBILITY	
	3.6 EXISTING SYSTEM	
	3.7 PROPOSED SYSTEM	
4	SYSTEM SPECIFICATION	
	4.1 HARDWARE REQUIRMENTS	
	4.2 SOFTWARE REQUIRMENTS	
5	SOFTWARE SPECIFICATIONS	
	5.1 MACHINE LEARNING	
	5.2 DEEP LEARNING	
	5.3 OPENCV	
	5.4 NUMPY AND ARRAYS	
	5.5 YOLOV3	
	5.6 HAAR CASCADE FACE DETECTION	

	5.7 PYTHON
	5.8 PYCHARM
	5.9 XML SUPPORT
6	SYSTEM DESIGN
	6.1 SYSTEM ARCHITECTURE
	6.2 CLASS DIAGRAM
	6.3 SEQUENCE DIAGRAM
	6.4 ACTIVITY DIAGRAM
	6.5 DATA FLOW DIAGRAM
	6.6 ER DIAGRAM
7	IMPLEMENTATION
	7.1 MODULE DESCRIPTION
8	SYSTEM TESTING
	8.1 UNIT TESTING
	8.2 INTEGRATION TESTING
	8.3 FUNCTIONAL TESTING
	8.4 SYSTEM TESTING
	8.5 WHITEBOX TESTING
	8.6 BLACKBOX TESTING
	8.7 TEST STRATEGY AND APPROACH
	8.8 TEST OBJECTIVES
	8.9 ACCEPTANCE TESTING
	8.10 TEST RESULTS

9	CONCLUSION AND FUTURE WORK
10	RESULT
11	REFERENCES

LIVE CAMERA HELMET DETECTION USING MACHINE LEARNING

ABSTRACT

In this paper, At present the continuous mobilization of vehicles has led to a surge in the number of road accidents across the world. To get better of this, government is trying to focus on the safest and preventive measures in traffic. So, our main idea is to introduce a helmet detection mechanism as most of the deaths caused are due to the absence of helmet. The practice of direct observation is found to be time taking and a lot of human effort is needed. This project attempts to implement a detection process through a few machine learning algorithms by using pre-defined libraries. This system notices a person with/without a helmet thereby imposing fines on the detected candidates. Further, this research work concludes that the automatic identification of helmets can overcome the challenges faced by manual data collection process. Moreover, this research work assumed that, through data collection, the algorithm can help to track the helmet use and promote its active use by people in order to ensure the road safety. And further Construction is a high-risk industry where construction workers tend to be hurt in the work process. Head injuries are very serious and often fatal.

CHAPTER 1

INTRODUCTION

Helmet detection refers to the use of computer vision and image processing techniques to identify whether a person is wearing a helmet or not. This technology is commonly used in various applications such as road safety, sports, construction, and industrial safety. To perform helmet detection, a camera captures an image or video of a person, which is then analysed by a deep learning algorithm that has been trained on a large dataset of images containing both people wearing helmets and people not wearing helmets. The algorithm uses object detection techniques to identify the position of the person's head and then checks whether a helmet is present in that location. There are several benefits to using helmet detection technology. For example, in sports safety, helmet detection can be used to ensure that athletes are wearing proper safety gear during practice and competitions. In construction and industrial settings, helmet detection can be used to ensure that workers are complying with safety regulations and wearing the necessary protective equipment. However, helmet detection technology also has limitations. For example, it may not work well in low light or noisy environments, or if the helmet is obscured or not easily visible in the image. Additionally, there may be privacy concerns related to the use of cameras to monitor individuals.

Helmet usage in recent times on roads has become increasingly important due to the rise of road accidents and fatalities. Many countries have implemented laws that require the use of helmets while riding a motorcycle or bicycle, in order to promote safety on the road.

According to the World Health Organization (WHO), helmets can reduce the risk of head injuries by 69% and the risk of death by 42% in the event of a motorcycle crash. However, despite the proven safety benefits, many people still choose not to wear helmets while riding. To address this issue, some countries have implemented stricter laws and enforcement mechanisms to encourage helmet usage. For example, in India, the government has made wearing helmets mandatory for all two-wheeler riders and has increased fines for non-compliance.

Helmet reduces the chances of skull getting decelerated, hence sets the motion of the head to almost zero. Cushion inside the helmet absorbs the impact of collision and as time passes head comes to a halt. It also spreads the impact to a larger area, thus safeguarding the head from severe injuries. More importantly it acts as a mechanical barrier between head and object to which the rider came into contact. Injuries can be minimized if a good quality full helmet is used. Traffic rules are there to bring a sense of discipline, so that the risk of deaths and injuries can be minimized significantly. However strict adherence to these laws is absent. Hence efficient and feasible techniques must be created to overcome these problems. To reduce the involved risk, it is highly desirable for bike-riders to use helmet. Worrying fact is that India ranks in top as far as road crash deaths are considered. Rapid urbanization, avoiding helmets, seat belts and other safety measures while driving are some of the reasons behind this trend according to analysis done by experts. In 2015 India signed Brasilia Declaration on Road Safety, where India committed to reduce road crash deaths to 50 percent by 2020. Observing the usefulness of helmet, Governments have made it a punishable offense to ride a bike without helmet and have adopted manual strategies to catch the violators. However, the existing video surveillance-based methods are passive and require significant human assistance. In general, such systems are infeasible due to involvement of humans, whose efficiency decreases over long duration. Automation of this process is highly desirable for reliable and robust monitoring of these violations as well as it also significantly reduces the amount of human resources needed. Similarly, in Thailand, police officers have started using cameras and facial recognition technology to identify riders who are not wearing helmets and issue fines. Overall, helmet usage on roads remains an important topic in promoting safety and reducing the number of road accidents and fatalities. Recent research has successfully done this work based on CNN, R-CNN, LBP, HoG, HaaR features, etc. But these works are limited with respect to efficiency, accuracy or the speed with which object detection and classification is done.

HELMETS IN THE FIELDS

Helmets are protective gear worn on the head to prevent or reduce the severity of injuries in case of an impact or collision. Helmets can be used in various activities and industries to ensure safety and protect against head injuries. Here are some common uses of helmets:

Road safety: Helmets are used by motorcyclists, bicyclists, and scooter riders to protect their heads in case of accidents or collisions.

Sports: Helmets are used in many sports such as football, hockey, baseball, and cycling to protect athletes from head injuries.

Construction: Workers in the construction industry use helmets to protect their heads from falling objects, debris, and other hazards.

Military: Military personnel use helmets to protect their heads from gunfire, shrapnel, and other combat-related hazards.

Industrial safety: Workers in industries such as mining, oil and gas, and manufacturing use helmets to protect their heads from hazards such as falling objects, exposure to chemicals, and other risks.

In each of these contexts, helmets are designed and tested to meet specific safety standards and provide maximum protection against potential hazards.

Helmets play an important role in protecting humans from head injuries. The head is a critical part of the human body, and injuries to the head can have serious and even life-threatening consequences. Helmets can help prevent or reduce the severity of head injuries by absorbing the force of an impact and distributing it over a larger area. Helmets are designed to fit snugly on the head and provide a cushioning layer between the head and any potential impact. They are typically made from materials such as polycarbonate, fiberglass, or carbon fiber, which are strong and lightweight.

The use of helmets can significantly reduce the risk of head injuries in a variety of contexts, such as motorcycle riding, bicycling, sports, and workplace safety. In fact, studies have shown that helmets can reduce the risk of head injuries by up to 88% in some cases.

Construction is a high-risk industry where construction workers tend to be hurt in the work process. Head injuries are very serious and often fatal. According to the accident statistics released by the state administration of work safety from 2015 to 2018, among the recorded 78 construction accidents, 53 events happened owing to the fact that the workers did not wear safety helmets properly, accounting for 67.95% of the total number of accidents .

In safety management at the construction site, it is essential to supervise the safety protective equipment wearing condition of the construction workers. Safety helmets can bear and disperse the hit of falling objects and alleviate the damage of workers falling from heights. Construction workers tend to ignore safety helmets because of weak safety awareness. At the construction site, workers that wear safety helmets improperly are much more likely to be injured. Traditional supervision of the workers wearing safety helmets on construction sites often requires manual work. There are problems such as a wide range of operations and difficult management of site workers. These factors make manual supervision difficult and inefficient and it is difficult to track and manage the whole workers at the construction sites accurately in real time. Hence, it is hard to satisfy the modern requirement of construction safety management only relying on the traditional manual supervision. In this context, it remains a significant issue to study on the automatic detection and recognition of safety helmets wearing conditions.

The automatic monitoring method can contribute to monitoring the construction workers and confirm the safety helmet wearing conditions at the construction site. In particular, considering that the traditional manual supervision of the workers is often costly, time-consuming, error-prone, and not sufficient to satisfy the modern requirements of construction safety management, the automatic supervision method can be beneficial to real-time on-site monitoring.

In this paper, based on the previous studies on computer vision-based object detection, we develop a deep learning-based method for the real-time detection of safety helmet at the construction site.

1.1.PROJECT OVERVIEW

In this Project Work, a Non-Helmet Rider detection system is built which attempts to satisfy the automation of detecting the traffic violation of not wearing helmet. The main principle involved is Object Detection using Deep Learning at three levels. The objects detected are person, motorcycle at first level using YOLOv3, helmet at second level using YOLOv3.

CHAPTER 2

LITERATURE SURVEY

[1] **Title** : " Automated Helmet Detection for Multiple Motorcycle Riders using CNN"

Authors : Madhuchhanda Dasgupta, Oishila Bandyopadhyay, Sanjay Chatterji - 2018

Description : . The ability to continuously monitor vehicle compliance with traffic rules is an important component of any effective traffic management system. In India, motorcycles may be one of the most prominent modes of transportation due to the fact that there are many citizens in urban areas. It has been stated that most motorcyclists have abstained from use of head protection in city traffic or even in the roadway driving. Many studies have shown that using a helmet on motorcycles reduces the likelihood of head and brain injuries when one is involved in a collision. Most traffic and safety rules are now monitored by a traffic video surveillance camera system, which allows the rules to be observed by means of breach of today. This paper offers a practical solution for confirming single or multiple motorcycle passengers, or "dual," as the designers call it, with or verifying their movement. When someone (say, a motorcycle rider) enters the scene at the beginning of the experiment, YOL will be applied to see if an object is present. YOL3, the state-of-the-art, will be used to investigate the starting point. The second neural network architecture, Convolutional Net, has been developed for detection of motorcyclists while using a technique called pattern matching and edge detection. Thus, the results suggest the proposed by the use of a CNN model on the same traffic videos are more promising than those from other models.

[2] **Title** : Helmet and Number Plate detection of Motorcyclists using Deep Learning and Advanced Machine Vision Techniques.

Authors : Fahad A Khan, Nitin Nagori, Dr. Ameya Naik – 2019

Description : presented to me Since the recent increase in use of motorcycles has made it more difficult to keep the roads clear, crashes and injuries are on the rise.

one of the main causes of this is the helmet that wasn't being used by the motorcyclist. Currently, a person must conduct a physical search or have CCTV footage of a different from the junction from that provided by the Department by those motorcyclists inspected by law, in order to locate any who are not wearing helmets. A proposal involves a computer structure to examine photographs of a motorcycle riders to identify those who wear helmets from those who don't wear helmets, allowing more precise identification of the users of motorised cycles. In general, the machine gets objects based on features and then removes them. YOLO-Dark architecture which utilises convolutional neural networks trained on Common Objects in a la Cena offers both convolutional net deep learning models that allow for object recognition and computer vision. YOLO's classifier's wavelet layers are altered to distinguish between three known classes, and the mechanism is implemented as a sliding window. The test results, providing a much more accurate picture of the (to a greater extent) the map's extent, achieved an average precision of 81%.

[3] **Title** : Helmet Detection Using ML & IoT.

Authors : Dikshant Manocha, Ankita Purkayastha, Yatin Chachra, Namit Rastogi, Varun Goel – 2019.

Description : presented to me this paper is focused on predicting unhelmet needs from the data of two-circling cyclists without a centralised authentication. It also helps provide a user experience for imposition fees. Vehicle recognitions start and vehicle instances are initially established on the captured traffic using first-in-one-first-out (FIFO) or best-in-first order, and two-out-first (FIR) methods, and the distinctions are subsequently made using two-in-two-out (TIR) or least-recent-in-first-out (LFO) method. After checking whether the riders and passengers are present in the vehicle, it computes if the pillion riders or the bike is without a helmet with OpenCV. When a motorcycle doesn't have a helmet on, it is scanned and tracked by digital imaging, so that a potential driver, pillion passenger, or motorcycle rider may be flagged as unlicensed (OCR). After getting the vehicle's registration number, a fine will be generated and all information will be mailed to the individual who was cited, along with an E-mail and a text

message sent to the vehicle's owner. An account (an app and a website) may be presented with which allows the user to pay their court fees.

[4] **Title** : Convolutional Neural Network-based Automatic Extraction and Fine Generation.

Authors : Y Mohana Roopa, Sri Harshini Popuri, Gottam Gowtam sai Sankar, Tejesh Chandra Kuppili -2020.

Description : it had helped me to come up with a great idea Humanity, particularly as a human beings, possesses a characteristic tendencies such as these: seeing correlations between causes and effects, ignoring what is related, and neglecting what has little bearing on the occurrence, and also perceiving faults in things that are not present. For those who know the reasons of the most fatalities in motor vehicle crashes, and want to stay home, their helmets and other protective gear are also present an important safety precaution. sustainability could attribute that it is inconsequential because few to no individuals are often using it, or perhaps that no one has bothered to make sure it operational in the past prevention has to remain within limits and increase potential; monitoring must be done to ensure potential does not increases but still maintaining or checking the current capabilities. There is an obvious relationship between traffic flow and human activity, and we thus, for practical purposes, are considered to be the main actors in it. it is physically impractical for a police officer to be part of the flow when he is enforcing the rules of the road Successful undertaking of a large projects will be made possible by a limited group of individuals, and many would be needed to help them. In this situation, the number on the helmet has to be factored in: two people from the pile of plates are likely to emerge. This is to ensure better identification of vehicles, as opposed to the other extreme, which is based on expediency, where multiple numbers from multiple sources are registered on the same vehicle International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056 Volume: 08 Issue: 05 | May 2021 www.irjet.net p-ISSN: 2395-0072 © 2021, IRJET | Impact Factor value: 7.529 | ISO 9001:2008 Certified Journal | Page 3241 would be discarded. watchful wait-and-out

situation Using Exant's method to find cars that donning of helmet camera footage as well as having a vehicle number plate reader readers, we can then expand the search to the collection of unlicensed vehicles to find certain vehicles not with the latter.

[5] **Title:** Improved OCR based Automatic Vehicle Number Plate Recognition using Features Trained Neural Network.

Authors : Bhavin V Kakani, Divyang Gandhi, Sagar Jani – 2020.

Description : presented to me A major portion of the current focus in intelligent transportation is on expanding algorithms and discoveries in the research and development realm. A significant improvement is necessary in traffic and parking/facility regulation, and traffic control with the use of a rapid, integrated and highly reliable automatic recognition plates reading system. This paper is intended to be the starting point of departure for a new OCR techniques that have enhanced the neural trained features for object recognition. In order to raise the accuracy of licence plate recognition, new algorithms are being developed that have a mix of features taken from both training and the raw data. Plate segmentation is concerned with finding; Character expansion focuses on plate recognition; All relevant information can be found in three main modules: License Plate identification, plate segmentation, and plate recognition. This scheme was modelled on 300 motor vehicle photographs from around the world and they could be valid for a large proportion of all motor vechile.

CHAPTER 3

SYSTEM ANALYSIS

3.1. FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential. Three key considerations involved in the feasibility analysis are

- i. Economical Feasibility
- ii. Technical Feasibility
- iii. Social Feasibility

3.2. Economic Feasibility

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus, the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

3.3. Technical Feasibility

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available

technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

3.4. Social Feasibility

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel

threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

3.5 OPERATIONAL FEASIBILITY

Operational feasibility is a measure of how well a proposed system solves the problems, and takes advantages of the opportunities identified during scope definition and how it satisfies the requirements identified in 8 the requirements identified in the requirements analysis phase of system development.

3.6 EXISTING SYSTEM

Currently, there are several helmet detection systems available in the market that use a combination of sensors and image processing techniques to detect helmet usage among riders. Some of these systems use accelerometers to measure the movement of the head, while others use cameras to capture images of the rider. However, these systems have some limitations, such as the need for complex hardware or the inability to detect helmets in low-light conditions.

3.6.1 DISADVANTAGES

- High cost of implementation and maintenance
- Complicated installation process

- Limited accuracy in certain scenarios, such as low light conditions or obscured views
- False alarms or missed detections, leading to reduced effectiveness in promoting road safety
- Dependence on specific hardware and software, which may not be readily available or compatible with other systems

3.7 PROPOSED SYSTEM

This proposed helmet detection system will utilize computer vision and machine learning algorithms to accurately detect whether a rider is wearing a helmet. The system will consist of a camera mounted at a strategic location, which will capture a video of the rider. This image will then be processed using object detection algorithms to identify the presence or absence of a helmet. In addition, this system will be able to promote the safer riding practices.

3.7.1 ADVANTAGES

- Improved accuracy in detecting whether a rider is wearing a helmet or not, which can promote safer riding practices
- Lower cost of implementation compared to existing systems, making it more accessible and feasible for a wider range of applications
- Ease of installation and compatibility with a range of hardware and software, allowing for seamless integration into existing systems

This proposed helmet detection system utilizes video capture and object detection algorithms that potentially offer faster and more efficient helmet detection than systems that rely solely on image capture.

CHAPTER 4

SYSTEM SPECIFICATIONS

4.1.HARDWARE REQUIREMENTS:

CPU type	I5
Ram size	8GB
GPU	Inbuilt GPU, External GPU
Webcam	720p
Monitor type	15 Inch colour monitor

4.2 SOFTWARE REQUIRMENTS:

Operating System	Windows, Linux
Simulation Tool	PyCharm, Machine learning
Core language	Python

CHAPTER 5

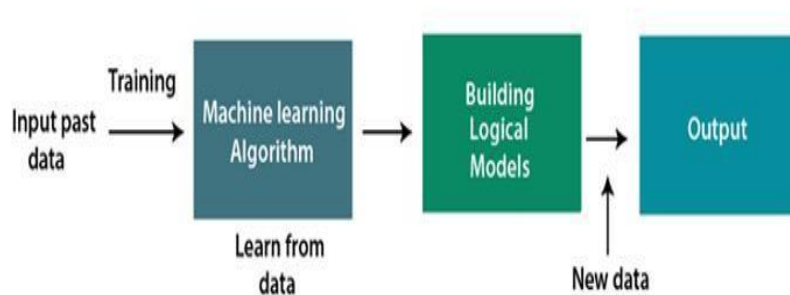
SOFTWARE SPECIFICATION

5.1. MACHINE LEARNING

Machine learning is a growing technology which enables computers to learn automatically from past data. Machine learning uses various algorithms for **building** mathematical models and making predictions using historical data or information. Currently, it is being used for various tasks such as image recognition, speech recognition, email filtering, Facebook auto-tagging, recommender system, and many more.

Machine Learning is said as a subset of artificial intelligence that is mainly concerned with the development of algorithms which allow a computer to learn from the data and past experiences on their own. The term machine learning was first introduced by Arthur Samuel in 1959. We can define it in a summarized way as: “Machine learning enables a machine to automatically learn from data, improve performance from experiences, and predict things without being explicitly programmed”. Suppose we have a complex problem, where we need to perform some predictions, so instead of writing a code for it, we just need to feed the data to generic algorithms, and with the help of these algorithms, machine builds the logic as per the data and predict the output. Machine learning has changed our way of thinking about the problem.

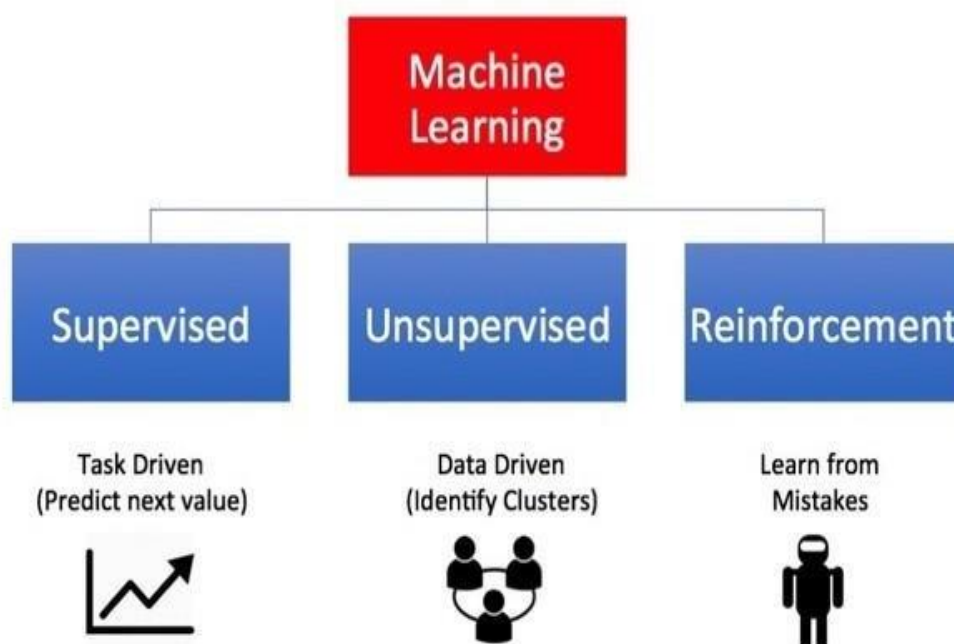
The below block diagram explains the working of Machine Learning algorithm:



5.1.1 Features of Machine Learning:

- Machine learning uses data to detect various patterns in a given dataset.
- It can learn from past data and improve automatically.
- It is a data-driven technology
- Machine learning is much similar to data mining as it also deals with the huge amount of the data.

Types of Machine Learning



5.1.2 Classification of Machine Learning

At a broad level, machine learning can be classified into three types:

1. Supervised learning
2. Unsupervised learning
3. Reinforcement learning

1) Supervised Learning

Supervised learning is a type of machine learning method in which we provide sample labelled data to the machine learning system in order to train it, and on that basis, it predicts the output. The system creates a model using labelled data to understand the datasets and learn about each data, once the training and processing are done then we test the model by providing a sample data to check whether it is predicting the exact output or not. The goal of supervised learning is to map input data with the output data.

Supervised learning can be grouped further in two categories of algorithms:

- **Classification**
- **Regression**

2) Unsupervised Learning

Unsupervised learning is a learning method in which a machine learns without any supervision. The training is provided to the machine with the set of data that has not been labelled, classified, or categorized, and the algorithm needs to act on that data without any supervision. The goal of unsupervised learning is to restructure the input data into new features or a group of objects with similar patterns.

It can be further classified into two categories of algorithms:

- **Clustering**
- **Association**
- **K – means clustering**
- **Hierarchical clustering**

3) Reinforcement Learning

Reinforcement learning is a machine learning training method based on rewarding desired behaviours and/or punishing undesired ones. In general, a reinforcement learning agent is able to perceive and interpret its environment, take actions and learn through trial and error. Some more examples of reinforcement learning in image processing include: Robots equipped with visual sensors from to learn their surrounding environment. Scanners to understand and interpret text. Image pre-processing and segmentation of medical images, like CT Scans.

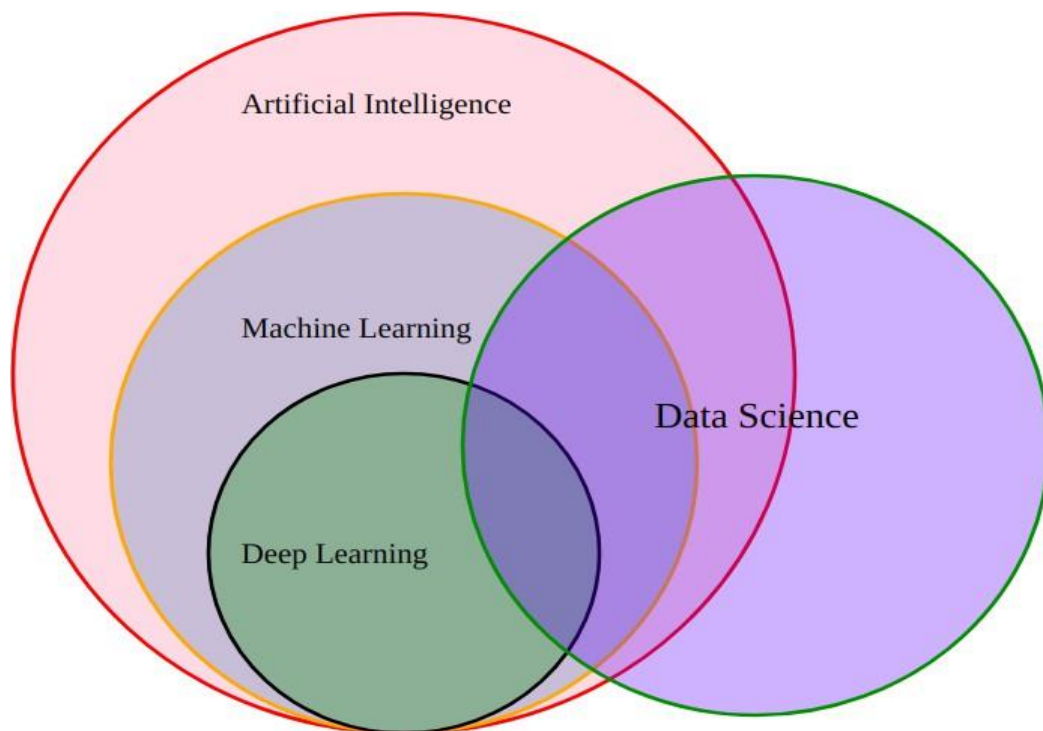
5.1.3 MACHINE LEARNING IMPLIES

1. Algorithm – Set of procedure used to create a Model from data.
2. Model -- A trained Machine
3. Predicted Variable – Variable containing the predicted output.
4. Training data -- Data given to Machine.
5. Testing data -- Data used by Machine to predict.
6. Features -- Input values of data.

5.2. DEEP LEARNING

Subset of MACHINE LEARNING which make the computation of multi layer neural network feasible. These neural networks attempt to simulate the behavior of the human brain albeit far from matching its ability allowing it to “learn” from large amounts of data.

Deep learning eliminates some of data pre-processing that is typically involved with machine learning. These algorithms can ingest and process unstructured data, like text and images, and it automates feature extraction, removing some of the dependency on human experts. For example, let’s say that we had a set of photos of different pets, and we wanted to categorize by “cat”, “dog”, “hamster”, et cetera. Deep learning algorithms can determine which features (e.g. ears) are most important to distinguish each animal from another. In machine learning, this hierarchy of features is established manually by a human expert.



The three most common ways people use deep learning to perform object classification are:

Training from Scratch

To train a deep network from scratch, you gather a very large labelled data set and design a network architecture that will learn the features and model. This is good for new applications, or applications that will have a large number of output categories. This is a less common approach because with the large amount of data and rate of learning, these networks typically take days or weeks to train.

Transfer Learning

Most deep learning applications use the transfer learning approach, a process that involves fine-tuning a pretrained model. You start with an existing network, such as AlexNet or GoogLeNet, and feed in new data containing previously unknown classes. After making some tweaks to the network, you can now perform a new task, such as categorizing only dogs or cats instead of 1000 different objects. This also has the advantage of needing much less data (processing thousands of images, rather than millions), so computation time drops to minutes or hours. Transfer learning requires an interface to the internals of the pre-existing network, so it can be surgically modified and enhanced for the new task. MATLAB has tools and functions designed to help you do transfer learning.

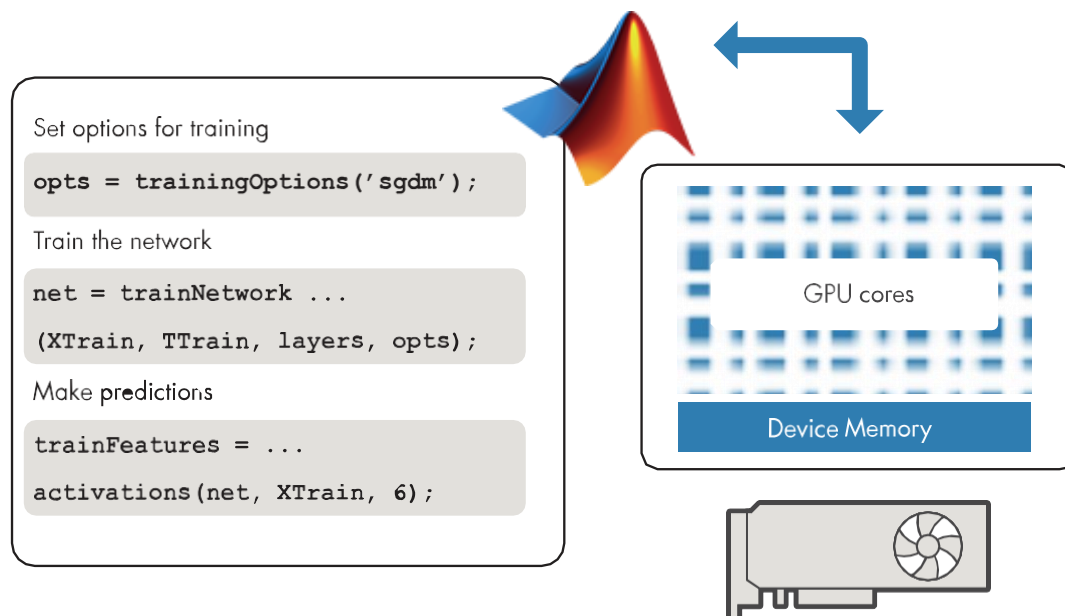
Feature Extraction

A slightly less common, more specialized approach to deep learning is to use the network as a **feature extractor**. Since all the layers are tasked with learning certain features from images, we can pull these features out of the network at any time during the training process. These features can then be used as input to a machine learning model such as Support Vector Machine (SVM).

Accelerating Deep Learning Models with GPUs

Training a deep learning model can take a long time, from days to weeks. Using GPU acceleration can speed up the process significantly. Using MATLAB with a

GPU reduces the time required to train a network and can cut the training time for an image classification problem from days down to hours. In training deep learning models, MATLAB uses GPUs (when available) without requiring you to understand how to program GPUs explicitly.



Deep Learning Toolbox commands for training your own CNN from scratch or using a pretrained model for transfer learning.

Importance of Deep Learning

- Machine learning works only with sets of structured and semi-structured data, while deep learning works with both structured and unstructured data
- Deep learning algorithms can perform complex operations efficiently, while machine learning algorithms cannot
- Machine learning algorithms use labeled sample data to extract patterns, while deep learning accepts large volumes of data as input and analyzes the input data to extract features out of an object

- The performance of machine learning algorithms decreases as the number of data increases; so to maintain the performance of the model, we need a deep learning.

Applications of Deep Learning :

The main applications of deep learning can be divided into computer vision, natural language processing (NLP), and reinforcement learning.

Computer vision:

In computer vision, Deep learning models can enable machines to identify and understand visual data. Some of the main applications of deep learning in computer vision include:

- Object detection and recognition: Deep learning model can be used to identify and locate objects within images and videos, making it possible for machines to perform tasks such as self-driving cars, surveillance, and robotics.
- Image classification: Deep learning models can be used to classify images into categories such as animals, plants, and buildings. This is used in applications such as medical imaging, quality control, and image retrieval.
- Image segmentation: Deep learning models can be used for image segmentation into different regions, making it possible to identify specific features within images.

Natural Language Processing (NLP)

In NLP, the Deep learning model can enable machines to understand and generate human language. Some of the main applications of deep learning in NLP include:

- Automatic Text Generation – Deep learning model can learn the corpus of text and new text like summaries, essays can be automatically generated using these trained models.

- Language translation: Deep learning models can translate text from one language to another, making it possible to communicate with people from different linguistic backgrounds.
- Sentiment analysis: Deep learning models can analyze the sentiment of a piece of text, making it possible to determine whether the text is positive, negative, or neutral. This is used in applications such as customer service, social media monitoring, and political analysis.
- Speech recognition: Deep learning models can recognize and transcribe spoken words, making it possible to perform tasks such as speech-to-text conversion, voice search, and voice-controlled devices.

Reinforcement Learning:

In reinforcement learning, deep learning works as training agents to take action in an environment to maximize a reward. Some of the main applications of deep learning in reinforcement learning include:

- Game playing: Deep reinforcement learning models have been able to beat human experts at games such as Go, Chess, and Atari.
- Robotics: Deep reinforcement learning models can be used to train robots to perform complex tasks such as grasping objects, navigation, and manipulation.
- Control systems: Deep reinforcement learning models can be used to control complex systems such as power grids, traffic management, and supply chain optimization.

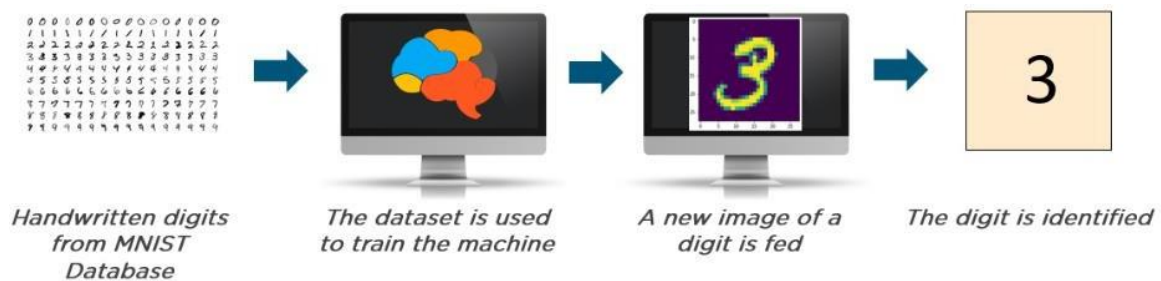
What is TensorFlow?

According to Google, TensorFlow is an open-source library used to create and train computational graphs. It is designed to run on a variety of hardware, including CPUs, GPUs, and TPUs. TensorFlow is used in a wide range of applications, from image and speech recognition to natural language processing and reinforcement learning.

It can't run on a cheap CPU or GPU. It can't process data as fast as GPUs that can process millions of data points per second. TensorFlow is popular for building machine learning applications such as Neural Networks.

Now, let's move on to the final section of our article on Deep Learning with Python, i.e., to build a model that can predict handwritten digits using the MNIST database.

Deep Learning With PyTorch Demo: Predict Handwritten Digits



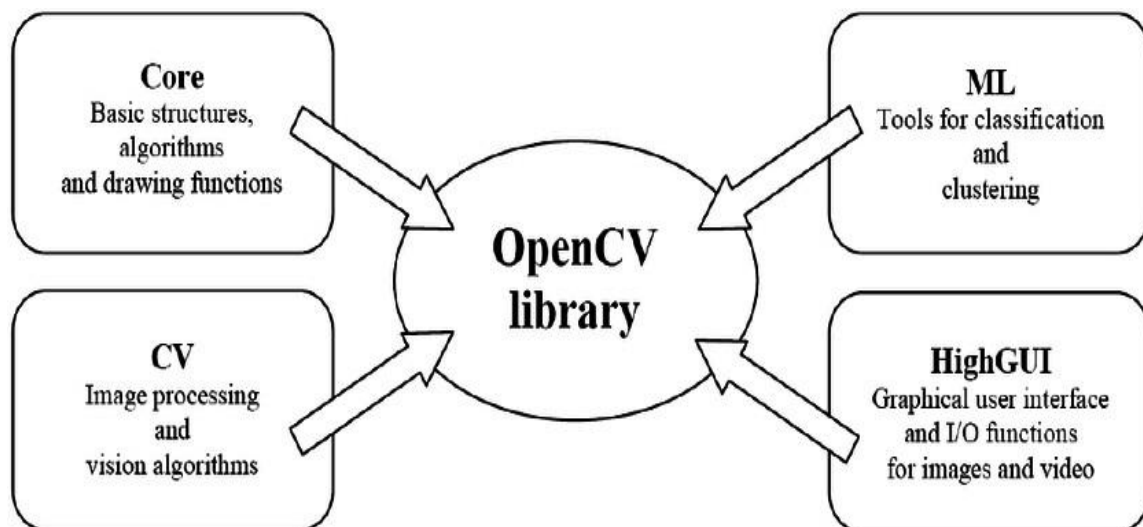
- Import the required libraries.
- Load the dataset.
- Check the total number of training and testing samples.
- Visualize the data.
- Build the model.
- Loss and Optimization.
- Test the model and find the accuracy.

As is evident above, our model has an accuracy of 91%, which is decent.

5.3. OPENCV

OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products.

The library has more than 2500 optimized algorithms, which includes a comprehensive set of both classic and state-of-the-art computer vision and machine learning algorithms. These algorithms can be used to detect and recognize faces, identify objects, classify human actions in videos, track camera movements, track moving objects, extract 3D models of objects, produce 3D point clouds from stereo cameras, stitch images together to produce a high resolution image of an entire scene, find similar images from an image database, remove red eyes from images taken using flash, follow eye movements, recognize scenery and establish markers to overlay it with augmented reality, etc. OpenCV has more than 47 thousand people of user community and estimated number of downloads exceeding 18 million. The library is used extensively in companies, research groups and by governmental bodies.



Along with well-established companies like Google, Yahoo, Microsoft, Intel, IBM, Sony, Honda, Toyota that employ the library, there are many startups such as Applied Minds, VideoSurf, and Zeitera, that make extensive use of OpenCV. OpenCV's deployed uses span the range from stitching street view images

together, detecting intrusions in surveillance video in Israel, monitoring mine equipment in China, helping robots navigate and pick up objects at Willow Garage, detection of swimming pool drowning accidents in Europe, running interactive art in Spain and New York, checking runways for debris in Turkey, inspecting labels on products in factories around the world on to rapid face detection in Japan.

It has C++, Python, Java and MATLAB interfaces and supports Windows, Linux, Android and Mac OS. OpenCV leans mostly towards real-time vision applications and takes advantage of MMX and SSE instructions when available. A full-featured CUDA and OpenCL interfaces are being actively developed right now. There are over 500 algorithms and about 10 times as many functions that compose or support those algorithms. OpenCV is written natively in C++ and has a templated interface that works seamlessly with STL containers.

Features of OpenCV Library

Using OpenCV library, you can

- Read and write images
- Capture and save videos
- Process images (filter, transform)
- Perform feature detection
- Detect specific objects such as faces, eyes, cars, in the videos or images.
- Analyze the video, i.e., estimate the motion in it, subtract the background, and track objects in it.

OpenCV Library Modules

Following are the main library modules of the OpenCV library.

Core Functionality

This module covers the basic data structures such as Scalar, Point, Range, etc., that are used to build OpenCV applications. In addition to these, it also includes the multidimensional array Mat, which is used to store the images. In the Java library of OpenCV, this module is included as a package with the name org.opencv.core.

Image Processing

This module covers various image processing operations such as image filtering, geometrical image transformations, color space conversion, histograms, etc. In the Java library of OpenCV, this module is included as a package with the name `org.opencv.imgproc`.

Video

This module covers the video analysis concepts such as motion estimation, background subtraction, and object tracking. In the Java library of OpenCV, this module is included as a package with the name `org.opencv.video`.

Video I/O

This module explains the video capturing and video codecs using OpenCV library. In the Java library of OpenCV, this module is included as a package with the name `org.opencv.videoio`.

calib3d

This module includes algorithms regarding basic multiple-view geometry algorithms, single and stereo camera calibration, object pose estimation, stereo correspondence and elements of 3D reconstruction. In the Java library of OpenCV, this module is included as a package with the name `org.opencv.calib3d`.

features2d

This module includes the concepts of feature detection and description. In the Java library of OpenCV, this module is included as a package with the name `org.opencv.features2d`.

Objdetect

This module includes the detection of objects and instances of the predefined classes such as faces, eyes, mugs, people, cars, etc. In the Java library of OpenCV, this module is included as a package with the name `org.opencv.objdetect`.

Highgui

This is an easy-to-use interface with simple UI capabilities. In the Java library of OpenCV, the features of this module is included in two different packages namely, `org.opencv.imgcodecs` and `org.opencv.videoio`

5.4. NUMPY

NumPy is a Python library used for working with arrays. It also has functions for working in domain of linear algebra, fourier transform, and matrices. NumPy was created in 2005 by Travis Oliphant. It is an open source project and you can use it freely. Is a library used for working with arrays. It is also a general-purpose array-processing package that provides comprehensive mathematical functions, linear algebra routines, Fourier transforms, and more. NumPy aims to provide less memory to store the data compared to python list and also helps in creating n-dimensional arrays. This is the reason why NumPy is used in Python.



At the core of the NumPy package, is the *ndarray* object. This encapsulates *n*-dimensional arrays of homogeneous data types, with many operations being

performed in compiled code for performance. There are several important differences between NumPy arrays and the standard Python sequences:

- NumPy arrays have a fixed size at creation, unlike Python lists (which can grow dynamically). Changing the size of an *ndarray* will create a new array and delete the original.
- The elements in a NumPy array are all required to be of the same data type, and thus will be the same size in memory. The exception: one can have arrays of (Python, including NumPy) objects, thereby allowing for arrays of different sized elements.
- NumPy arrays facilitate advanced mathematical and other types of operations on large numbers of data. Typically, such operations are executed more efficiently and with less code than is possible using Python's built-in sequences.
- A growing plethora of scientific and mathematical Python-based packages are using NumPy arrays; though these typically support Python-sequence input, they convert such input to NumPy arrays prior to processing, and they often output NumPy arrays. In other words, in order to efficiently use much (perhaps even most) of today's scientific/mathematical Python-based software, just knowing how to use Python's built-in sequence types is insufficient - one also needs to know how to use NumPy arrays.

NumPy – Data Types

NumPy has some extra data types, and refer to data types with one character, like *i* for integers, *u* for unsigned integers etc.

Below is a list of all data types in NumPy and the characters used to represent them.

- *i* - integer
- *b* - boolean
- *u* - unsigned integer
- *f* - float
- *c* - complex float

- m - timedelta
- M - datetime
- O - object
- S - string
- U - unicode string
- V - fixed chunk of memory for other type (void)

Data Type Objects (dtype)

A data type object describes the interpretation of a fixed block of memory corresponding to an array, depending on the following aspects –

- Type of data (integer, float or Python object)
- Size of data
- Byte order (little-endian or big-endian)
- In case of structured type, the names of fields, data type of each field and part of the memory block taken by each field.
- If the data type is a subarray, its shape and data type

The byte order is decided by prefixing '<' or '>' to the data type. '<' means that encoding is little-endian (least significant is stored in smallest address). '>' means that encoding is big-endian (a most significant byte is stored in smallest address).

A dtype object is constructed using the following syntax –

```
numpy.dtype(object, align, copy)
```

The parameters are –

- **Object** – To be converted to data type object
- **Align** – If true, adds padding to the field to make it similar to C-struct
- **Copy** – Makes a new copy of dtype object. If false, the result is a reference to builtin data type object.

Why is NumPy So Popular ?

- NumPy uses **less memory** and is stored in a single continuous block of memory.
- Because of its expressive and compact syntax it is **quick and simple** to comprehend.
- Makes accessing elements and specific portions of the array very easy using **Indexing and Slicing**.
- It has lots of **built-in functions** for mathematical and statistical calculations like power function, addition, deviation, etc. which provides better ways for handling and processing data.

Features of Numpy

NumPy contains many features, some of which are listed below :

- NumPy is a combination of Python and the C language as it is partially written in Python and most of its parts are written in C or C++.
- The **object-oriented** approach is also fully supported by Numpy.
- NumPy functions can be used to work with code that is written in other programming languages and provides tools for integrating with languages such as C, Fortran, etc.
- NumPy is an open-source core Python package for scientific computing.
- NumPy uses less space and stores data in contiguous memory.
- It offers a multidimensional array object with excellent performance as well as methods for working with these arrays.
- Arrays can be reshaped into different dimensions using the reshape function provided by NumPy.
- We can work with different data types using NumPy and can determine the type of data using the dtype function.
- NumPy comes with a plethora of built-in functions such as sum, sort, max, and so on, allowing users to write fewer lines of code while improving the quality of their work.

- NumPy provides a broadcasting technique by which we can perform arithmetic operations on arrays of different shapes.
- NumPy with SciPy is also used as an alternative to MATLAB.

Arrays

A numpy array is a grid of values, all of the same type, and is indexed by a tuple of nonnegative integers. The number of dimensions is the *rank* of the array the *shape* of an array is a tuple of integers giving the size of the array along each dimension.

Operations using NumPy

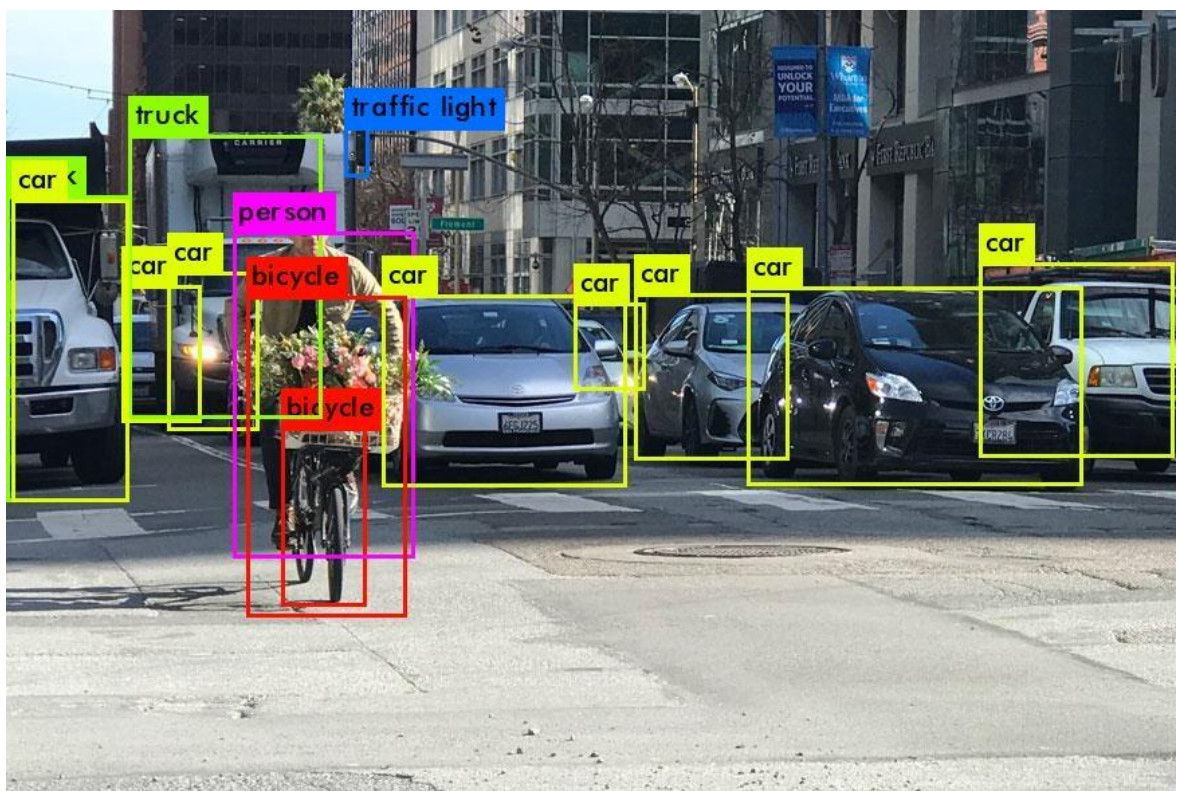
Using NumPy, a developer can perform the following operations

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

5.5. YOLOV3

YOLOv3 (You Only Look Once, Version 3) is a real-time object detection algorithm that identifies specific objects in videos, live feeds, or images. The YOLO machine learning algorithm uses features learned by a deep convolutional neural network to detect an object. Versions 1-3 of YOLO were created by Joseph Redmon and Ali Farhadi, and the third version of the YOLO machine learning algorithm is a more accurate version of the original ML algorithm. The first version of YOLO was created in 2016, and version 3, which is discussed extensively in this article, was made two years later in 2018. YOLOv3 is an improved version of YOLO and YOLOv2. YOLO is implemented using the Keras or OpenCV deep learning libraries. The official successors of YOLOv3 is YOLOv4, and the newly released YOLOv7 (2022), which marks the current state-of-the-art object detector in 2023.

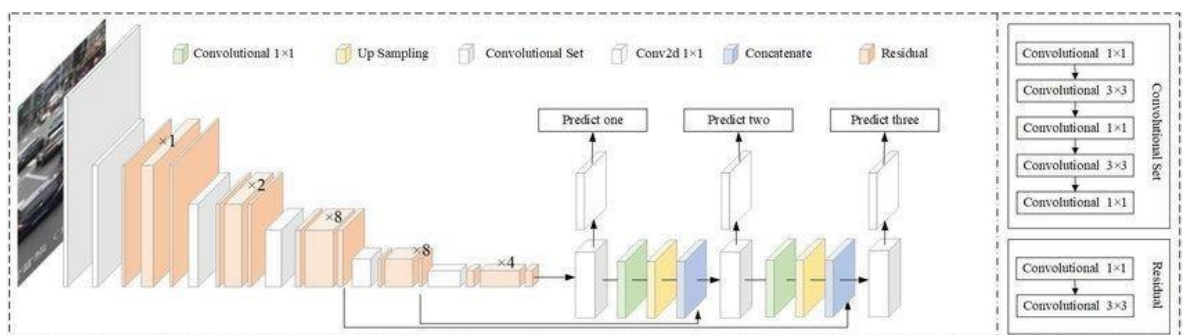
Read more at: <https://viso.ai/deep-learning/yolov3-overview/>



Computer vision example of Yolov3

1.5.1 Working of yolov3

YOLO is a Convolutional Neural Network (CNN) for performing object detection in real-time. CNNs are classifier-based systems that can process input images as structured arrays of data and recognize patterns between them (view image below). YOLO has the advantage of being much faster than other networks and still maintains accuracy. It allows the model to look at the whole image at test time, so its predictions are informed by the global context in the image. YOLO and other convolutional neural network algorithms “score” regions based on their similarities to predefined classes. High-scoring regions are noted as positive detections of whatever class they most closely identify with. For example, in a live feed of traffic, YOLO can be used to detect different kinds of vehicles depending on which regions of the video score highly in comparison to predefined classes of vehicles.



Working diagram of YOLOV3

1.5.2 YOLO ARCHITECTURE

The YOLOv3 algorithm first separates an image into a grid. Each grid cell predicts some number of boundary boxes (sometimes referred to as anchor boxes) around objects that score highly with the aforementioned predefined classes. Each boundary box has a respective confidence score of how accurate it assumes that prediction should be and detects only one object per bounding box. The boundary boxes are generated by clustering the

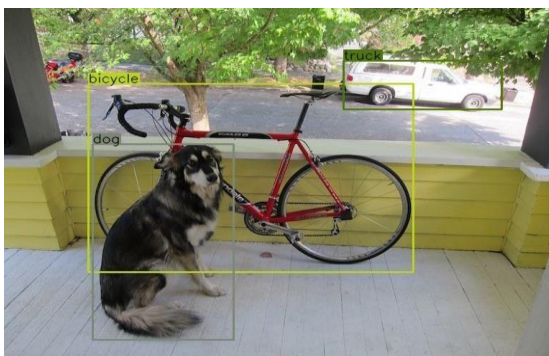
dimensions of the ground truth boxes from the original dataset to find the most common shapes and sizes.

1.5.3 YOLOv3 Resources

The YOLOv3 algorithm has a multitude of credible resources created by the author and makers of the algorithm itself. For any purpose, primary resources are always best for getting accurate information on the topic, but for YOLO v3, these resources are even more important because of all the second-hand information available on its use.

1.5.4 HOW TO USE YOLOv3

The first step to using YOLOv3 would be to decide on a specific object detection project. YOLOv3 performs real-time detections, so choosing a simple project that has an easy premise, such as detecting a certain kind of animal or car in a video, is ideal for beginners to get started with YOLOv3. In this section, we will go over the essential steps and what you have to know for using the YOLO machine learning algorithm successfully. You can also (more easily) use YOLO's COCO pretrained weights by initializing the model with `model = YOLOv3()`. Using COCO's pre-trained weights means that you can only use YOLO for object detection with any of the 80 pretrained classes that come with the COCO dataset. This is a good option for beginners because it requires the least amount of new code and customization.

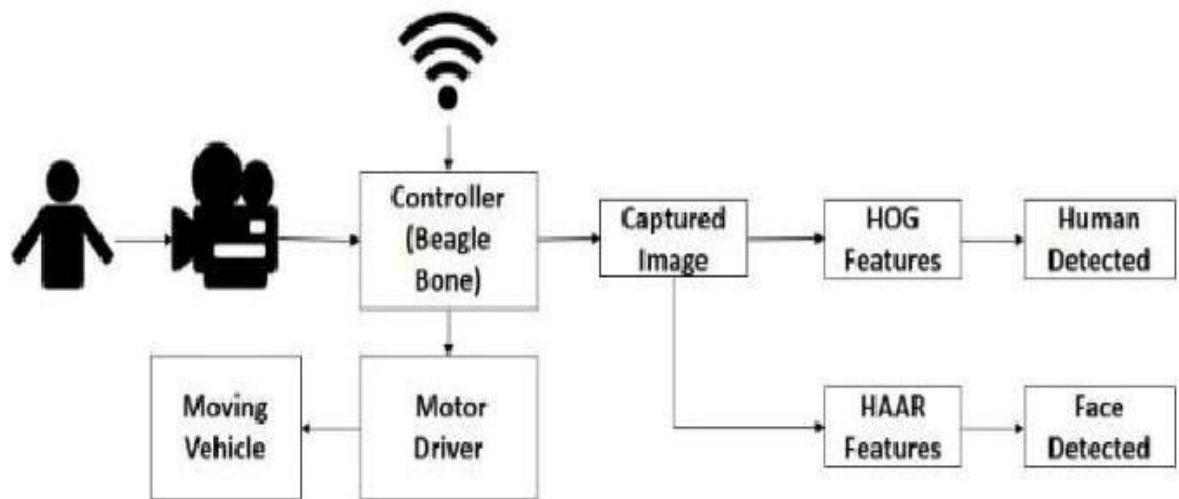


Example for YOLOv3

5.6. HAAR CASCADE FACE DETECTION



Face Detection a widely popular subject with a huge range of applications. Modern day Smartphones and Laptops come with in-built face detection softwares, which can authenticate the identity of the user. There are numerous apps that can capture, detect and process a face in real time, can identify the age and the gender of the user, and also can apply some really cool filters. The list is not limited to these mobile apps, as Face Detection also has a wide range of applications in Surveillance, Security and Biometrics as well. But the origin of its Success stories dates back to **2001**, when **Viola and Jones** proposed the first ever Object Detection Framework for Real Time Face Detection in Video Footage. Haar Cascade is a feature-based object detection algorithm to detect objects from images. A cascade function is trained on lots of positive and negative images for detection. The algorithm does not require extensive computation and can run in real-time.



- Haar cascades can work in real-time. It's blazing fast.
- Haar cascade is not much accurate compared to modern object detection algorithms.
- It detects many false positives. This can be tuned to some extent but can't be completely removed.
- It is very simple to implement.
- The biggest downside of Haar Cascade is its false negative detection.

Haar Cascade can't be used for face recognition since it only identifies the matching shape and size.

Haar cascade uses the cascade function and cascading window. It tries to calculate features for every window and classify positive and negative. If the window could be a part of an object, then positive, else, negative.

5.6.1 Pre-trained Haar Cascades

There are lots of pre-trained haar cascade files that make implementation super easy. We can also train our own haar cascade, but that needs big data to train on. The OpenCV library manages a repository on GitHub for all popular haar cascades pre-trained files that can be used for various object detection tasks, for example:

- Human face detection

- Eye detection
- Vehicle detection
- Nose / Mouth detection
- body detection
- license plate detection

Haar cascade stores its features in an XML file; these files can be directly loaded in OpenCV for object detection using Haar cascade.

5.6.2 Implementing Haar-cascades in OpenCV

If you are working with any of the pre-trained object detection available in the repository provided by OpenCV, you only need to download the pre-trained XML file.

5.6.3 Loading Haar-cascade in OpenCV

We can load haar-cascade XML files using `cv2.CascadeClassifier` function.

Example:

```
face_detector=cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
```

```
eye_detector = cv2.CascadeClassifier('haarcascade_eye.xml')
```

We can call the detector function once the XML file is loaded.

5.6.4 Hierarchical detection

Haarcascade supports Hierarchical detection, which means the Haar cascade can be able to detect multiple objects within a single frame in a hierarchical manner. Suppose we have to detect the faces and eyes of humans. To proceed with the task, we need to follow these steps.

- Detect Faces
- For every face, crop faces and forward them for eye detection
- After finding the coordinates of the eyes (ex,ey,ew,eh) draw a bounding box around the eyes in the original picture.

- draw a bounding box around faces using coordinates(x,y,w,h) on the original picture.

5.6.4 Implementing Face Detection in Real Time

Object detection using Haar Cascade can be used in OpenCV video stream, we only need to read a video or camera feed in OpenCV, and the rest of the thing will be the same. A Video feed is a sequence of frames so that the code will be the same as the single frame. Due to its light computation requirements, Haar Cascade runs at a good time per second.

We will read OpenCV video cam feed input to take images in real-time.

5.6.5 Limitation of Haar Cascade Face Detection

Haar Cascade is still very popular for some objects like faces, cars, etc., where the object is easily distinguishable. Haar Cascade can't work for deep object detection like types of grain, etc. There are a few Limitations of the Haar Cascade Algorithm-

- Lower Accuracy compared to modern object detectors.
- High False-positive detection.
- Manual tuning of parameters is required.
- Training haar cascade for the custom objects is not easy at all.

5.7.PYTHON

Python is a dynamic, high level, free open source and interpreted programming language. It supports object-oriented programming as well as procedural oriented programming. In Python, we don't need to declare the type of variable because it is a dynamically typed language.



For example, `x=10`. Here, `x` can be anything such as String, int, etc.

Python is an interpreted, object-oriented programming language similar to PERL, that has gained popularity because of its clear syntax and readability. Python is said to be relatively easy to learn and portable, meaning its statements can be interpreted in a number of operating systems, including UNIX-based systems, Mac OS, MS-DOS, OS/2, and various versions of Microsoft Windows 98. Python was created by Guido van Rossum, a former resident of the Netherlands, whose favourite comedy group at the time was Monty Python's Flying Circus.

Features in Python

There are many features in Python, some of which are discussed below

- Easy to code
- Free and Open Source
- Object-Oriented Language
- GUI Programming Support
- High-Level Language
- Extensible feature
- Python is Portable language
- Python is Integrated language.

Python Syntax

- Somewhat similar to the English language, with a mathematical influence, Python is built for readability.
- Unlike other languages that use semicolons and/or parentheses to complete a command, Python uses new lines for the same function
- Defines scope (i.e., loops, functions, classes) by relying indentation, using whitespace, rather than braces (aka curly brackets)

Python Flexibility

Python, a dynamically typed language, is especially flexible, eliminating hard rules for building features and offering more problem-solving flexibility with a variety of methods. It also allows users to compile and run programs right up to a problematic area because it uses run-time type checking rather than compile-time checking.

The Less Great Parts of Python

On the down side, Python isn't easy to maintain. One command can have multiple meanings depending on context because Python is a dynamically typed language. And, maintaining a Python app as it grows in size and complexity can be increasingly difficult, especially finding and fixing errors. Users will need experience to design code or write unit tests that make maintenance easier.

Speed is another weakness in Python. Its flexibility, because it is dynamically typed, requires a significant amount of referencing to land on a correct definition, slowing performance. This can be mitigated by using alternative implementation of Python (e.g. PyPy).

Python and AI

AI researchers are fans of Python. Google TensorFlow, as well as other libraries (scikit-learn, Keras), establish a foundation for AI development because of the usability and flexibility it offers Python users. These libraries, and their availability, are critical because they enable developers to focus on growth and building.

5.8. PYCHARM

PyCharm is a dedicated Python Integrated Development Environment (IDE) providing a wide range of essential tools for Python developers, tightly integrated to create a convenient environment for productive Python, web, and data science development. It is cross-platform, working on Microsoft Windows, macOS and Linux. PyCharm has a Professional Edition, released under a proprietary license and a Community Edition released under the Apache License. PyCharm Community Edition is less extensive than the Professional Edition.



FEATURES:

1. Intelligent Code Editor:

- It helps us write high-quality codes!
- It consists of colour schemes for keywords, classes, and functions. This helps increase the readability and understanding of the code.
- It helps identify errors easily.
- It provides the autocomplete feature and instructions for the completion of the code.

2. Code Navigation:

- It helps developers in editing and enhancing the code with less effort and time.
- With code navigation, a developer can easily navigate to a function, class, or file.
- A programmer can locate an element, a symbol, or a variable in the source code within no time.
- Using the lens mode, further, a developer can thoroughly inspect and debug the entire source code.

3. Refactoring

- It has the advantage of making efficient and quick changes to both local and global variables.
- Refactoring in PyCharm enables developers to improve the internal structure without changing the external performance of the code.
- It also helps split up more extended classes and functions with the help of the extract method.

4. Assistance for Many Other Web Technologies:

- It helps developers create web applications in Python.
- It supports popular web technologies such as HTML, CSS, and JavaScript.
- Developers have the choice of live editing with this IDE. At the same time, they can preview the created/updated web page.
- The developers can follow the changes directly on a web browser.
- PyCharm also supports AngularJS and NodeJS for developing web applications.

5. Support for Popular Python Web Frameworks

- PyCharm supports web frameworks such as Django.
- It provides the autocomplete feature and suggestions for the parameters of Django.
- It helps in debugging the codes of Django.
- It also assists web2py and Pyramid, the other popular web frameworks.

6. Assistance for Python Scientific Libraries

- PyCharm supports Python's scientific libraries such as Matplotlib, NumPy, and Anaconda.
- These scientific libraries help in building projects of Data Science and Machine Learning.
- It consists of interactive graphs that help developers understand data.
- It is capable of integrating with various tools such as IPython, Django, and Pytest. This integration helps innovate unique solutions.

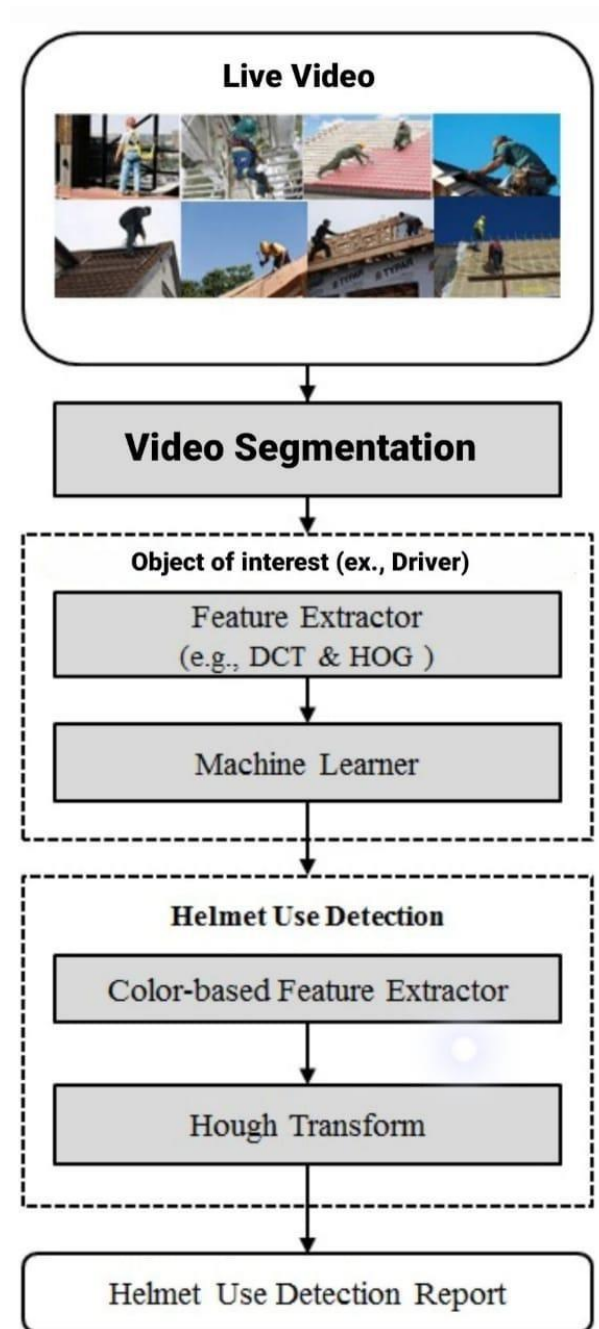
5.9 XML SUPPORT

XML and data access are intimately tied—XML is all about encoding data, and data access is increasingly becoming all about XML. The .NET Framework does not just support Web standards—it is built entirely on top of them.

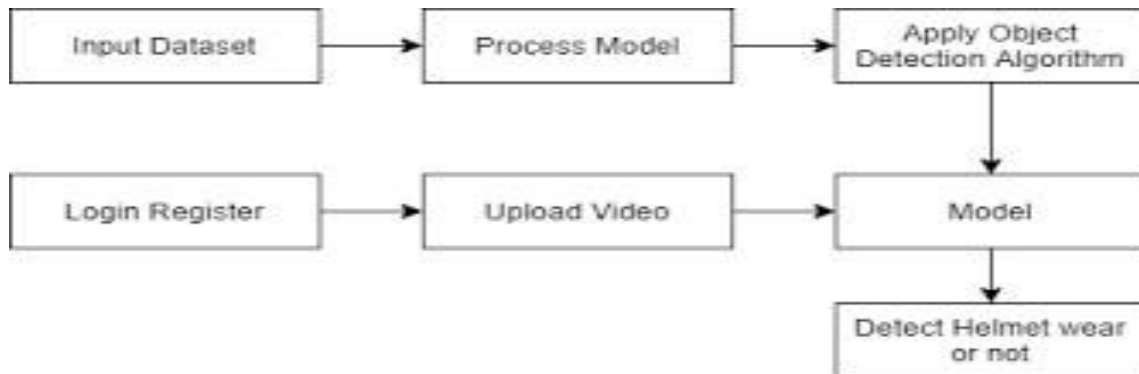
CHAPTER 6

SYSTEM DESIGN

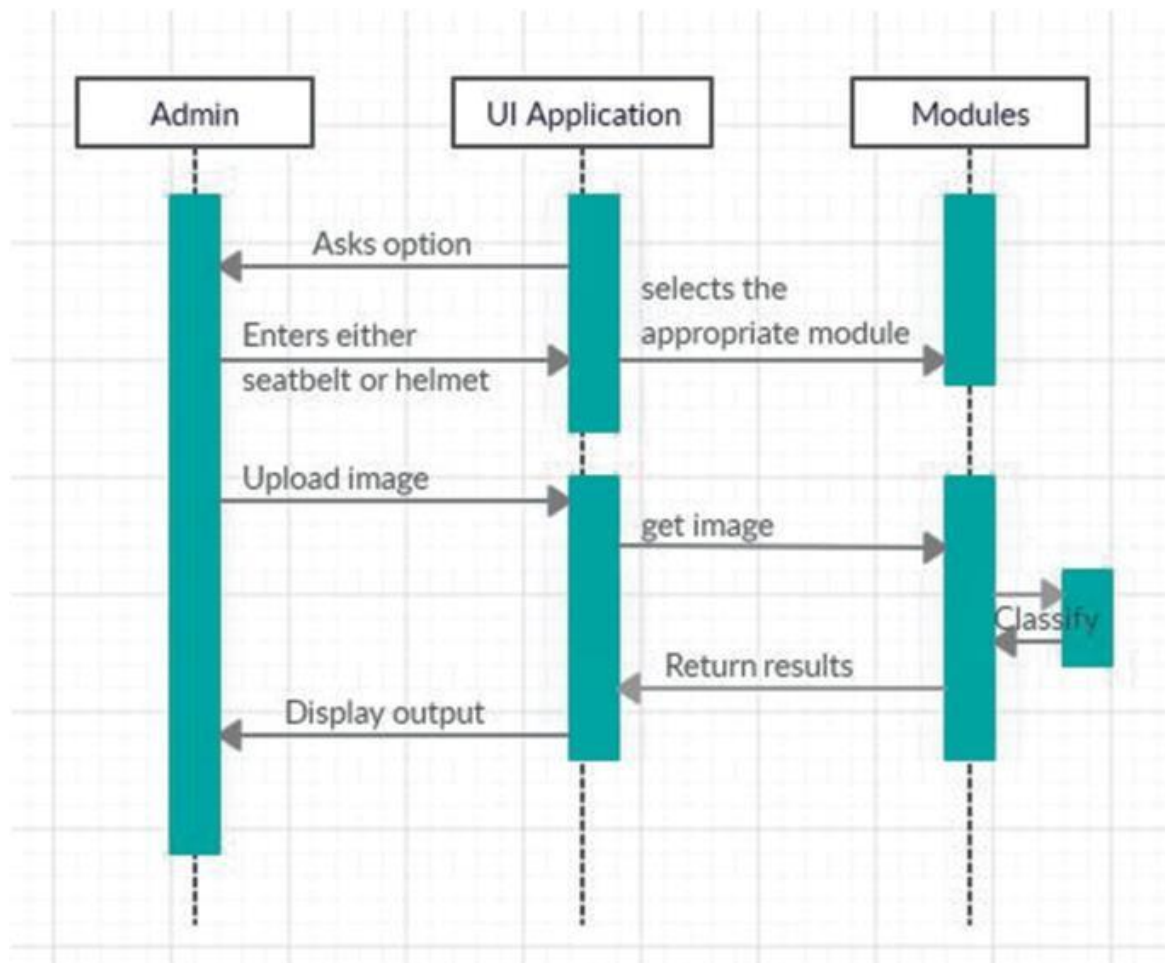
6.1.SYSTEM ARCHITECTURE



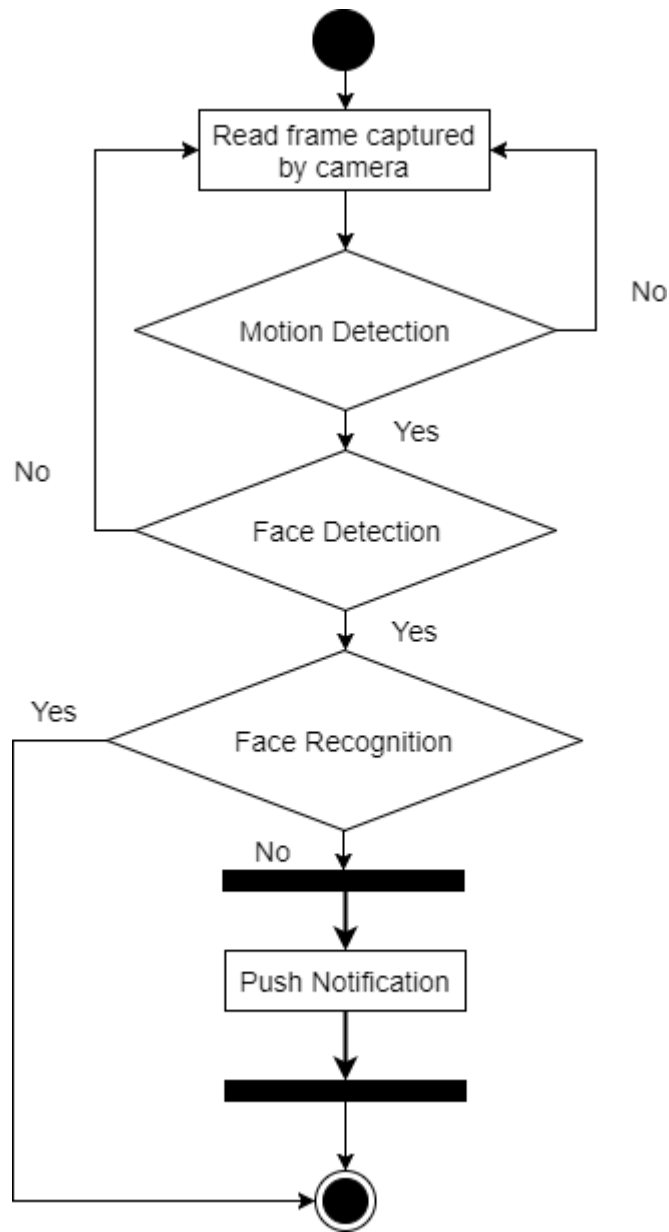
6.2 CLASS DIAGRAM



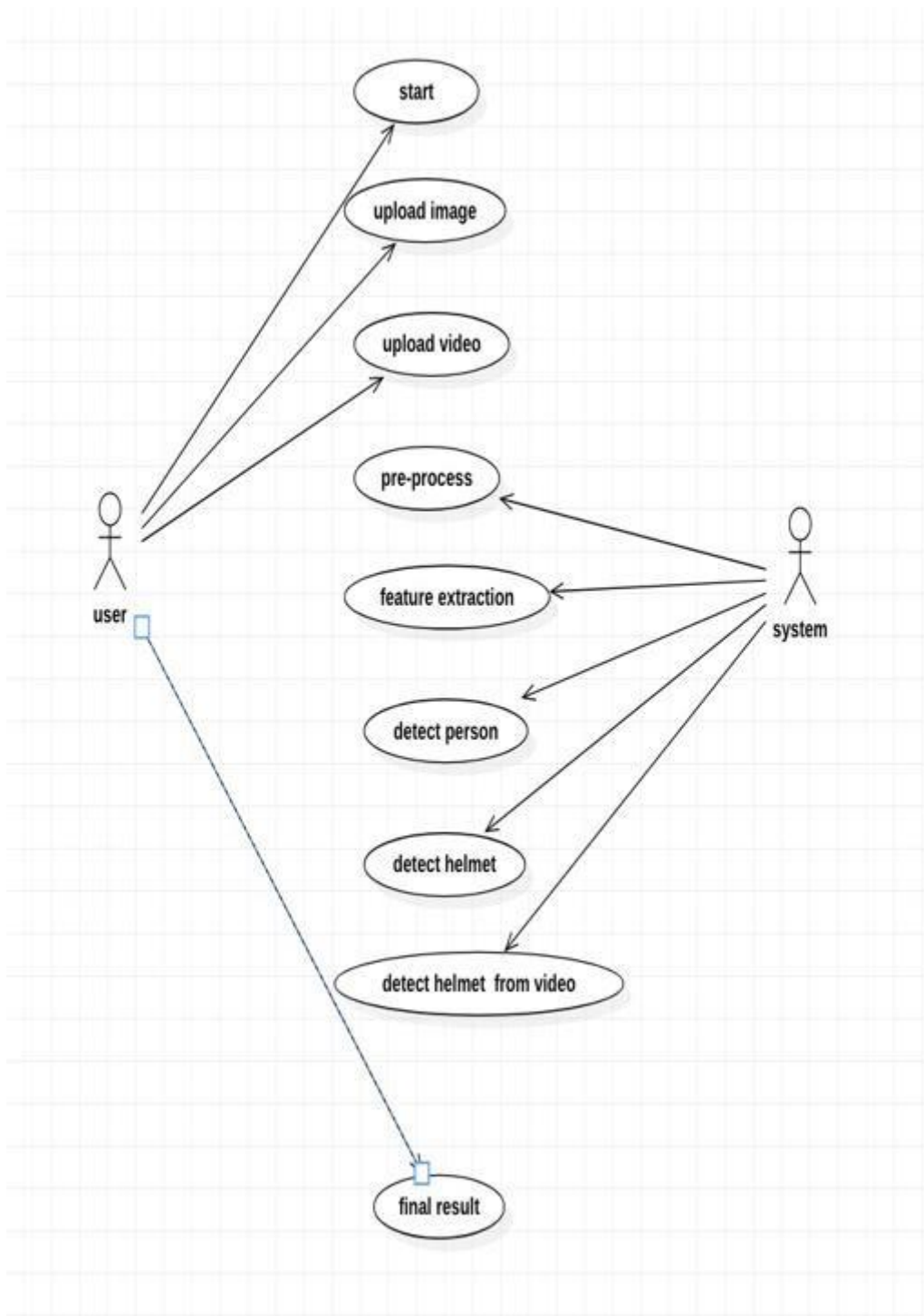
6.3 SEQUENCE DIAGRAM



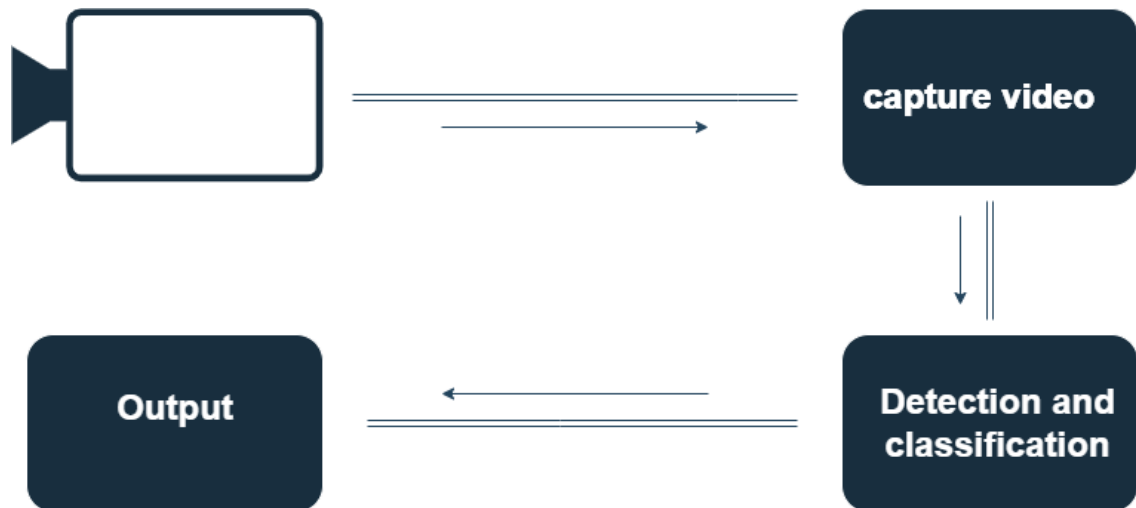
6.4. ACTIVITY DIAGRAM



6.5. ER DIAGRAM



6.6. DATA FLOW DIAGRAM



The presented process used to detect whether wearing helmet or not. The working method of the proposed procedure includes three principal phases;

1. Train the pretrained data.
2. using yolov3 give an computer vision.
3. haar cascade face detection algorithm.

CHAPTER 7

IMPLEMENTATION

5.1.MODULE DESCRIPTION

- ❑ MODULE 1: Live camera
- ❑ MODULE 2: Detect the helmet
- ❑ MODULE 3: Haar cascade face detection (HFD)
- ❑ MODULE 4: Training the pretrained module using cv2 haarcascade Model
- ❑ MODULE 5: Evaluation.

MODULE 1: Live camera

A **webcam** is a video camera which is designed to record or stream to a computer or computer network. They are primarily used in video telephony, live streaming and social media, and security. Webcams can be built-in computer hardware or peripheral, and are commonly connected to a device using USB or wireless protocols. Video frames recorded by the wide-angle CCTV camera are used to detect motorcyclists. If any motorcyclist without helmet is found, then the zoomed (narrow-angle) CCTV is activated to capture image of the violating motorcyclist.

MODULE 2 : Detecting the helmet

In the second module the program is allocated to detect whether the person is wearing helmet or not. By using some of the pretrained module we can get a frame in the face with a green colour text output in the webcam.

MODULE 3 : Haar cascade face Detection (HFD)

A common Haar feature for face detection is a set of two adjacent rectangles that lie above the eye and the cheek region. The position of these rectangles is

defined relative to a detection window that acts like a bounding box to the target object (the face in this case).

MODULE 4 : Training the pretrained module using cv2 haarcascade Model

In the proposed system we are using cv2 haarcascade model which is Haar cascade is an algorithm that can detect objects in images, irrespective of their scale in image and location. This algorithm is not so complex and can run in real-time. We can train a haar-cascade detector to detect various objects like cars, bikes, buildings, fruits, etc. And it is done by the pretrained xml model which it have an allocation of how the helmet size should be un the face. By using haar cascade model draw a bounding box around faces using coordinates (x,y,w,h) on the original picture or video.

MODULE 5 : EVALUATION

In the program the output shown in the webcam itself like (Helmet and No Helmet). And give an accuracy rate also for both then by using the pretrained xml model we can able to detect the helmet in the live cam itself.

CHAPTER 8

SYSTEM TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub – assemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

TYPES OF TESTS

8.1. UNIT TESTING

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application. It is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

8.2. INTEGRATION TESTING

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfactory, as shown by successfully unit testing, the combination of components is correct and consistent.

Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

8.3. FUNCTIONAL TEST

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centred on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised

Procedures : interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

8.4. SYSTEM TEST

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

8.5. WHITE BOX TESTING

White Box Testing is a testing in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is used to test areas that cannot be reached from a black box level.

8.6. BLACK BOX TESTING

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

8.8. TEST STRATEGY AND APPROACH

Field testing will be performed manually and functional tests will be written in detail.

8.9. TEST OBJECTIVES

All field entries must work properly.

- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

8.10. ACCEPTANCE TESTING

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

8.11. TEST RESULTS

All the test cases mentioned above passed successfully. No defects encountered.

CHAPTER 9

CONCLUSION AND FUTURE WORK

In conclusion, the helmet detection project is a crucial application that can help improve safety in various settings such as construction sites, sports, and transportation. By using deep learning techniques such as YOLOv3 and Python, the project can accurately and efficiently detect helmets in real time.

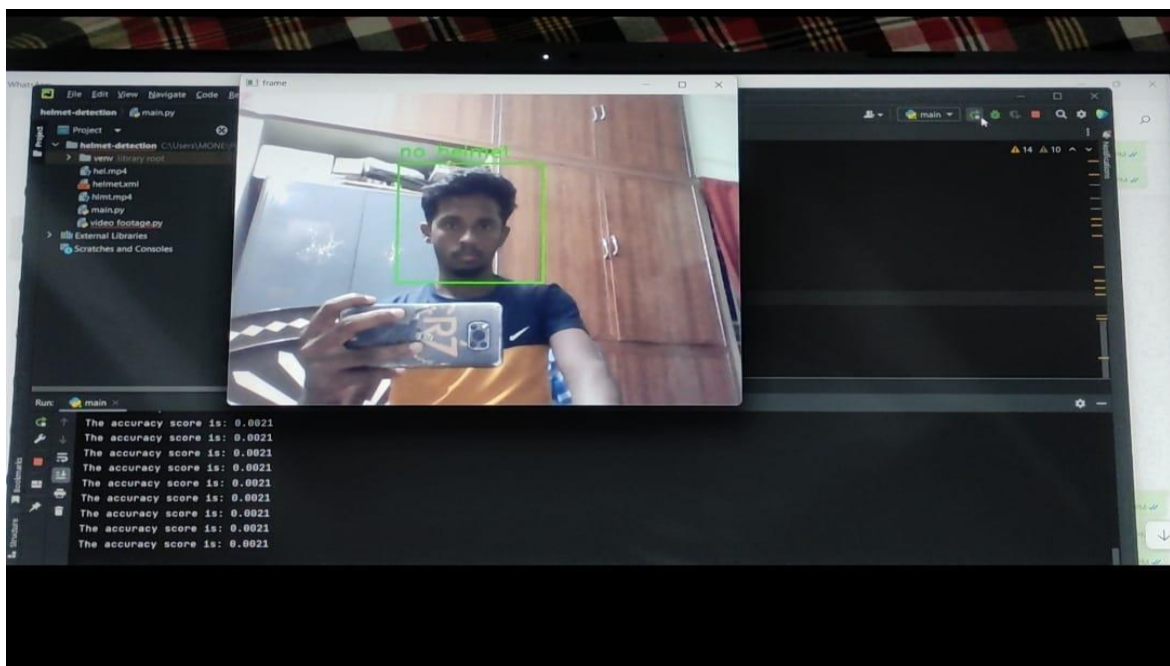
To ensure the project's success, various software testing types such as functional testing, performance testing, integration testing, and usability testing can be employed to verify the system's accuracy, reliability, and user-friendliness.

Overall, the helmet detection project has the potential to significantly improve safety in various settings, and by utilizing advanced machine learning techniques and software testing methodologies, the system can provide accurate and reliable helmet detection results in real time. In the future this project can be developed with the additional feature of extracting vehicle registration numbers from motorcycles that are riding on roads by people without a helmet and then using this registration number the E-challan(digital traffic enforcement system used in India) can be sent to those owners of vehicles. A real time analysis can be made with secure server storage system with tracking system for this future enhancement.

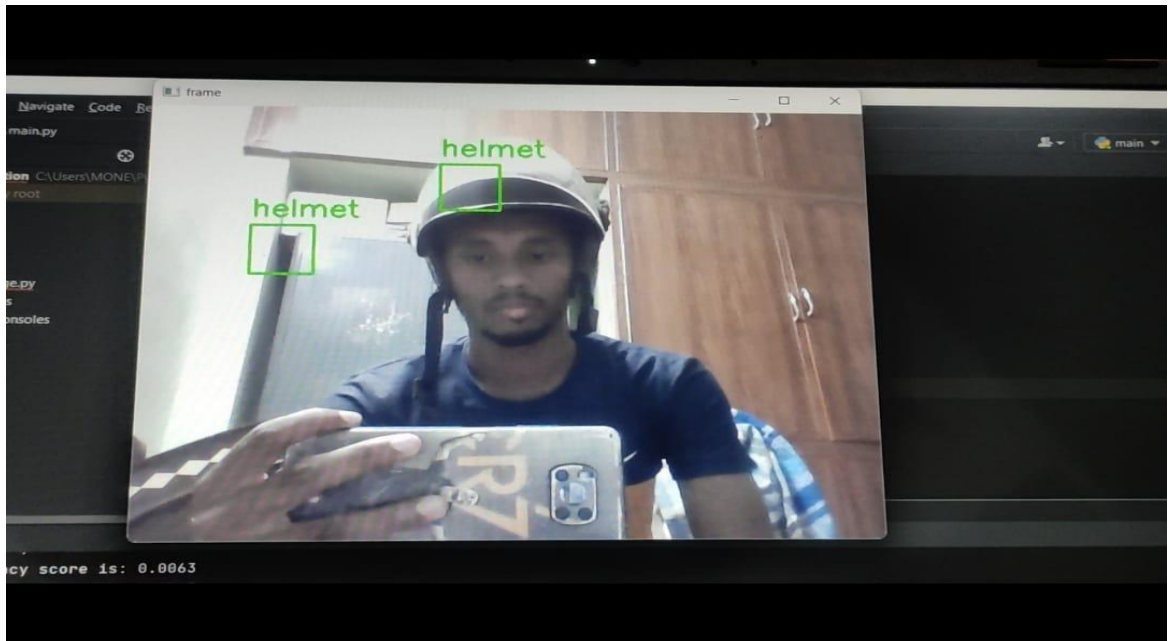
CHAPTER 10

RESULT

OUTPUT



(Output for without Helmet)



(Output for with Helmet)

CHAPTER 11

REFERENCE

1. X. Chang and X. M. Liu, "Fault tree analysis of unreasonably wearing helmets for builders," *Journal of Jilin Jianzhu University*, vol. 35, no. 6, pp. 67–71, 2018.
View at: [Google Scholar](#)
2. Z. Y. Wang, *Design and Implementation of Detection System of Warning Helmets Based on Intelligent Video Surveillance*, Beijing University of Posts and Telecommunications, Beijing, China, 2018.
3. H. Zeng, *Research on Intelligent Helmets System for Engineering Construction*, Harbin Institute of Technology, Harbin, China, 2017.
4. A. Kelm, L. Laubat, A. Meins-Becker et al., "Mobile passive radio frequency identification (RFID) portal for automated and rapid control of personal protective equipment (PPE) on construction sites," *Automation in Construction*, vol. 36, pp. 38–52, 2013.
View at: | [Google Scholar](#)
5. S. Barro-Torres, T. M. Fernández-Caramés, H. J. Pérez-Iglesias, and C. J. Escudero, "Real-time personal protective equipment monitoring system," *Computer Communications*, vol. 36, no. 1, pp. 42–50, 2012.
View at: [Publisher Site](#) | [Google Scholar](#)
6. A. H. M. Rubaiyat, T. T. Toma, M. Kalantari-Khandani et al., "Automatic detection of helmet uses for construction safety," in *Proceedings of the 2016 IEEE ACM International Conference on Web Intelligence Workshops (WIW)*, ACM, Omaha, NE, USA, October 2016.
View at: [Publisher Site](#) | [Google Scholar](#)
7. K. Shrestha, P. P. Shrestha, D. Bajracharya, and E. A. Yfantis, "Hard-hat detection for construction safety visualization," *Journal of Construction Engineering*, vol. 2015, Article ID 721380, 8 pages, 2015.
View at: [Publisher Site](#) | [Google Scholar](#)

8. R. Waranusast, N. Bundon, V. Timtong, C. Tangnoi, and P. Pattanathaburt, "Machine vision techniques for motorcycle safety helmet detection," in *Proceedings of the Image & Vision Computing New Zealand*, IEEE, Wellington, New Zealand, November 2013.
View at: [Publisher Site](#) | [Google Scholar](#)
9. P. Doungmala and K. Klubsuwan, "Helmet wearing detection in Thailand using haar like feature and circle hough transform on image processing," in *Proceedings of the IEEE International Conference on Computer & Information Technology*, IEEE, Nadi, Fiji, December 2016.
View at: [Publisher Site](#) | [Google Scholar](#)
10. J. S. Jia, Q. J. Bao, and H. M. Tang, "Method for detecting safety helmet based on deformable part model," *Application Research of Computers*, vol. 33, no. 3, pp. 953–956, 2016.
View at: [Google Scholar](#)
11. K. Li, X. G. Zhao, J. Bian, and M. Tan, "Automatic safety helmet wearing detection," 2018, <https://arxiv.org/abs/1802.00264>.
View at: [Google Scholar](#)
12. H. Wu and J. Zhao, "Automated visual helmet identification based on deep convolutional neural networks," in *Proceedings of the 13th International Symposium on Process Systems Engineering (PSE 2018)*, vol. 44, pp. 2299–2304, San Diego, CA, USA, July 2018.
View at: [Publisher Site](#) | [Google Scholar](#)
13. S. Xu, Y. Wang, Y. Gu, N. Li, L. Zhuang, and L. Shi, "Safety helmet wearing detection study based on improved faster RCNN," *Application Research of Computers*, vol. 37, no. 3, pp. 901–905, 2019.
View at: [Google Scholar](#)