

**REAL TIME HUMAN DETECTION AND COUNTING SYSTEM USING
OPEN CV AND PYTHON**

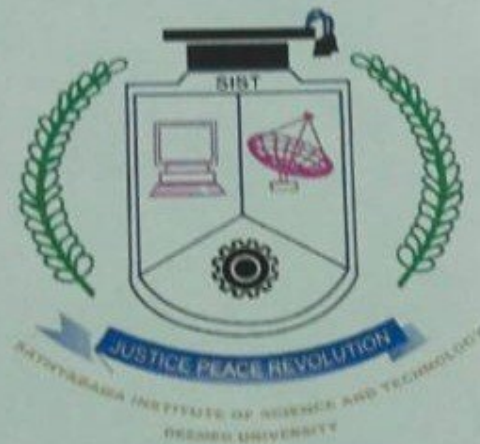
Submitted in partial fulfillment of the required for the award of

Masters of Computer Science

by

SORNALAKSHMI S

REGISTER NO (42734022)



**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

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI 600 119

APRIL 2024



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited "A++" Grade by NAAC | UIR 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
SORNALAKSHMI S (42734025) who carried out the project entitled "REAL TIME
HUMAN DETECTION AND COUNTING SYSTEM USING OPEN CV AND PYTHON"
under our supervision from January 2024 to April 2024.

Internal Guide

Ms. Shamilli. V M.Sc., Assistant Professor

DEAN AND HEAD OF THE DEPARTMENT

Dr. Rekha Chakravarthi M.E., Ph.D.,

Submitted for Viva voce examination held on 30/4/2024

Internal Examiner

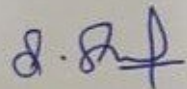
External Examiner

DECLARATION

I, SORNALAKSHMIS (42734022) hereby declare that the Project Report entitled "REAL TIME HUMAN DETECTION COUNTING SYSTEM USING OPEN CV AND PYTHON " done by me under the guidance of Ms. Shamili V, M.SC., Assistant Professor., Department of Computer Science at SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY, Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai – 600119 is submitted in partial fulfillment of the requirements for the award of Master Of Science degree in Computer Science.

DATE: 20/4/2024

PLACE: CHENNAI



SORNA LAKSHMI S

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. Shamilli V M.Sc., Assistant Professor**, Department of Computer Science for her valuable guidance, suggestions and constant encouragement which 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

SORNALAKSHMI.S

Abstract

In today's dynamic technological landscape, the need for efficient and precise real-time monitoring systems has become paramount. This paper presents a comprehensive setup that seamlessly integrates image processing, deep learning, and embedded systems to achieve accurate and efficient human presence monitoring. The proposed system leverages the capabilities of the YOLOv6 algorithm, MATLAB software, and PYTHON microcontroller, creating a robust solution for real-time surveillance applications. The block diagram of the system begins with the laptop camera capturing live video, which is then fed into the MATLAB software for image processing. The camera-MATLAB interface ensures effective pre-processing of the images, setting the stage for the YOLOv6 algorithm to perform real-time human detection. YOLOv6, known for its high accuracy and efficiency, enhances the detection process, making it a key component in the proposed monitoring system. The processed data from YOLOv6 is seamlessly transmitted to an PYTHON microcontroller, serving as the central hub for counting human presence. This interconnected system offers a holistic approach to real-time monitoring, ensuring both accuracy in human detection and efficiency in data processing. The YOLOv6 algorithm, being a state-of-the-art object detection model, excels in detecting humans in diverse and challenging scenarios. By incorporating YOLOv6 into the proposed system, we capitalize on its strengths to achieve superior performance in human presence monitoring.

TABLE OF CONTENTS

S NO	TITLE	PAGENO
	ABSTRACT	i
	TABLE OF CONTENT	ii
	LIST OF TABLE	iv
1	INTRODUCTION	1
1.1	Introduction to the project	1
1.2	Existing System	2
1.3	Proposed System	3
2	METHODOLOGY	7
2.1	Introduction of Methodology	7
3	SYSTEM CONFIGURATION	13
3.1	System Configuration Components	14
3.2	Project Overview	15
3.3	MATLAB Overview	17
3.4	MATLAB System	18

4	MATLAB DOCUMENTATION	19
4.1	MATLAB Documentation Introduction	20
4.2	MATLAB Online Help	21
4.3	MATLAB’S Power of computational mathematic lab	22
4.4	Highlights of MATLAB	22
4.5	Enivornments of MATLAB	22
4.6	Condition or Enivornment Setup	22
4.7	Understanding the MATLAB Enivornment	22
5	SECTION SUMMARY	23
5.1	Introduction to section summary	23
5.2	MATLAB Guide Tool	24
6	APPENDICES	25
7	OUTPUT	28
8	CONCLUSION	29
8.1	Future Work	30
9	REFERENCE	32

LIST OF TABLES

SNO	NAME OF TABLE	PAGE NO.
3.1	Celebrity of MATLAB in data mining	14
3.2	Polls of trendy data mining method	16

CHAPTER 1

INTRODUCTION

1.1 Introduction

In an era marked by rapid technological advancements, the demand for sophisticated monitoring systems has witnessed unprecedented growth. From ensuring security in public spaces to optimizing resource utilization in smart buildings, real-time human presence monitoring has become a critical component of various applications. This paper introduces a comprehensive system that amalgamates image processing, deep learning, and embedded systems to create an efficient and precise solution for real-time monitoring. The foundation of the proposed system lies in the utilization of the YOLOv6 algorithm, MATLAB software, and PYTHON microcontroller. These components work in tandem to capture live video, process it for human detection, and provide immediate feedback on the occupancy status through a GUI. The integration of these technologies aims to address the challenges associated with real-time monitoring, including accuracy, efficiency, and Adaptability to diverse scenarios.

The first cornerstone of the system is the laptop camera, which serves as the primary source for capturing live video. The choice of the camera is pivotal, as it determines the quality of the input data for subsequent processing stages. The ability to capture high-resolution video in varying lighting conditions is crucial for the success of the real-time monitoring system. The camera not only acts as the eyes of the system but also sets the stage for the subsequent stages of image processing and human detection. Following the video capture, the MATLAB software comes into play as a powerful tool for image processing. MATLAB, renowned for its versatility in scientific computing and image Analysis, provides an interface for efficient pre-processing of the live video stream. This stage is essential for enhancing the quality of the input data and preparing it for analysis by the YOLOv6 algorithm. MATLAB's capabilities in noise reduction, image enhancement, and feature extraction contribute to the overall robustness of the monitoring system.

In summary, the proposed real-time human presence monitoring system introduces a holistic approach that combines cutting-edge technologies from different domains. The integration of YOLOv6, MATLAB, and PYTHON creates a comprehensive solution that excels in accuracy, efficiency, and adaptability.

1.2 Existing System

In the realm of human presence monitoring, existing systems vary in complexity, functionality, and deployment scenarios. These systems often rely on a combination of sensors, cameras, and software solutions to achieve their objectives. In this section, we will explore some of the common elements found in current human presence monitoring systems, highlighting their strengths and limitations.

Basic Motion Sensors

Traditional motion sensors have been a staple in security systems for detecting human presence. These sensors, often based on infrared technology, trigger an alarm or initiate a response when they detect changes in infrared radiation within their field of view. While effective for basic intrusion detection, these systems lack the ability to provide detailed information about the detected presence, such as the number of individuals or their precise locations.

Closed-Circuit Television (CCTV) Systems:

CCTV systems are widely used for surveillance and monitoring in various settings. These systems typically consist of multiple cameras strategically placed to cover specific areas. While CCTV systems can capture live video feeds, they often require additional software for human detection and analysis. Analyzing video feeds in real-time for human presence is a challenging task, and traditional video surveillance systems may struggle to provide accurate and immediate results.

Computer Vision and Machine Learning:

More advanced systems leverage computer vision and machine learning algorithms for human presence detection. These systems often employ cameras and sophisticated algorithms to analyze video feeds. Some systems use background subtraction techniques or simple motion analysis, while others delve into more complex machine learning models for object detection. However, these solutions may face challenges in achieving real-time performance and accuracy, particularly in crowded or dynamic environments.

Smart Building Solutions:

In the context of smart buildings, occupancy monitoring systems aim to optimize energy usage and enhance security. These systems may employ a combination of sensors, including motion detectors,

occupancy sensors, and even Wi-Fi-based tracking. However, the accuracy of these systems can be affected by factors such as sensor placement, environmental conditions, and the diversity of building usage.

Bluetooth and Wi-Fi-Based Tracking

Some systems utilize Bluetooth or Wi-Fi signals emitted by personal devices, such as smartphones, to track human presence. By monitoring the strength of signals and triangulating positions, these systems attempt to estimate the number and location of individuals within a given area. While potentially effective, this approach relies on individuals carrying enabled devices and may not be suitable for scenarios where privacy concerns are paramount.

Integration of IoT Devices

With the rise of the Internet of Things (IoT), there has been an increasing trend in integrating various sensors and devices for human presence monitoring. These systems may combine data from motion sensors, cameras, and other IoT devices .

While these existing systems offer valuable contributions to human presence monitoring, they often exhibit limitations that the proposed system aims to address. Common challenges include achieving real-time performance, providing accurate detection in dynamic environments, and ensuring adaptability to different scenarios. Moreover, many existing systems lack the seamless integration of image processing, deep learning, and embedded systems, which is a key strength of the proposed solution.

1.3 Proposed System

The proposed system represents a pioneering approach to real-time human presence monitoring, aiming to overcome the limitations of existing systems by integrating cutting-edge technologies. This comprehensive solution leverages the YOLOv6 algorithm, MATLAB software, and PYTHON microcontroller to create a robust and efficient monitoring system capable of accurate detection, real-time processing, and dynamic feedback through an GUI.

YOLOv6 Algorithm for Real-Time Human Detection

At the heart of the proposed system is the YOLOv6 algorithm, renowned for its exceptional speed and accuracy in object detection. YOLO, which stands for "You Only Look Once," adopts a unique approach by dividing the input image into a grid and predicting bounding boxes and class probabilities for objects within each grid cell. This single-pass methodology enables real-time processing, making it well-suited for applications where immediate and accurate detection is paramount.

MATLAB Image Processing for Enhanced Data Quality

The MATLAB software serves as a critical component in the proposed system, offering a versatile platform for image processing. MATLAB's capabilities in scientific computing, image analysis, and algorithm development make it an ideal choice for optimizing the live video feed captured by the camera before it undergoes analysis by the YOLOv6 algorithm.

Comprehensive Integration for Versatile Applications

The proposed system's strength lies in its comprehensive integration of YOLOv6, MATLAB, and PYTHON, creating a seamless workflow from video capture to real-time display. This integration is designed to address the shortcomings observed in existing systems, offering a solution that excels in accuracy, efficiency, and adaptability to diverse scenarios.

PYTHON Microcontroller as the Central Hub

The PYTHON microcontroller serves as a central hub in the proposed system, playing a crucial role in processing the output from YOLOv6 and facilitating seamless integration with a GUI. The PYTHON's real-time processing capabilities, versatility, and ability to interface with external devices make it an ideal intermediary between the high-level processing performed by YOLOv6 and the user interface requirements of the GUI.

YOLOv6 excels in human detection, a critical aspect of the proposed monitoring system. Its ability to detect and classify objects with high accuracy, even in complex and dynamic scenarios, positions it as an ideal choice for real-time surveillance applications. The algorithm has undergone iterative improvements, and the version 6 introduces enhancements in terms of both accuracy and speed, further solidifying its status as a state-of-the-art object detection model.

The integration of YOLOv6 into the proposed system brings a new level of sophistication to human presence monitoring. By relying on deep learning techniques, the algorithm can adapt to varying lighting conditions, diverse appearances, and different spatial arrangements of individuals within the monitored space. This adaptability is crucial for applications ranging from security and surveillance to smart building management, where the monitoring environment can be highly dynamic.

Image preprocessing plays a pivotal role in ensuring the accuracy of human detection. MATLAB provides tools for noise reduction, image enhancement, and feature extraction, allowing for the refinement of the input data. The preprocessing stage aims to improve the quality of the images and remove any artifacts or disturbances that might affect the performance of YOLOv6 during the subsequent detection process.

The collaboration between MATLAB and YOLOv6 is designed to maximize the strengths of both components. MATLAB acts as a bridge between the raw video feed from the camera and the deep learning capabilities of YOLOv6. By preparing the input data through effective image preprocessing, MATLAB contributes to the overall robustness and accuracy of the monitoring system.

The PYTHON's role extends beyond counting, as it facilitates the dynamic display of results on a GUI screen. The GUI serves as the user interface, providing real-time feedback on the occupancy status. This immediate and visual representation enables users to make quick decisions based on the monitored data, enhancing the system's utility in scenarios where timely responses are crucial.

The combination of YOLOv6 and MATLAB ensures efficient human detection through advanced deep learning techniques and optimized image preprocessing. This synergy enhances the accuracy of the system, making it reliable in scenarios with varying lighting conditions, crowded environments, or diverse appearances.

Potential Applications and Future Directions

The proposed system's versatility opens the door to a myriad of applications across different domains. In security and surveillance, the real-time human presence monitoring can be employed for intrusion detection, enabling swift responses to potential security threats. In smart buildings, the system can contribute to energy efficiency by optimizing lighting and HVAC systems based on occupancy.

Future directions for the proposed system include exploring edge computing capabilities to further optimize real-time processing. Edge computing involves processing data closer to the source, reducing latency and enhancing system responsiveness. Additionally, the system could benefit from continuous updates and improvements to the underlying deep learning model, ensuring it stays at the forefront of object detection technology.

Challenges and Considerations

While the proposed system presents a groundbreaking approach to real-time human presence monitoring, it is essential to acknowledge potential challenges and considerations. One significant consideration is the privacy aspect, especially in applications where individuals are monitored. Implementing mechanisms for anonymizing data or incorporating privacy-preserving techniques becomes crucial to address ethical concerns. The system's performance may also be influenced by external factors such as environmental conditions, camera quality, and the positioning of sensors. Calibration and fine-tuning may be necessary to ensure optimal performance across different deployment scenarios.

CHAPTER 2

METHODOLOGY

2.1 Methodology

The methodology section outlines the step-by-step process undertaken to design, implement, and evaluate the proposed real-time human presence monitoring system. This comprehensive system integrates the YOLOv6 algorithm, MATLAB image processing, and an PYTHON microcontroller for efficient human detection, accurate counting, and dynamic display on an GUI screen. The following elucidates the detailed methodology involved in creating this innovative solution.

Problem Definition

The initial phase of the methodology involves clearly defining the problem at hand. The need for a real-time human presence monitoring system arises from the limitations observed in existing systems, such as their inability to provide immediate and accurate detection, especially in dynamic environments. The goal is to design a system that not only detects human presence but also counts individuals in real-time and provides dynamic feedback.

Literature Review

1. TITLE –Real-time human detection and counting system

AUTHOR – Z. A. Mulla, S. R. G, V. V. Lakshmi and S. C. Maturi, "Epileptic Seizure Detection System using Single Hidden Layer Extreme Learning Machine (ELM)," 2022 9th International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, India, 2022, pp. 298- 302, doi: 10.23919/INDIACom54597.2022.9763268.

ABSTRACT-- Without the abstract provided, I can't give you the exact content. However, based on the title and the context of the conference, the paper likely presents a system for detecting epileptic seizures using a machine learning technique called Single Hidden Layer Extreme Learning Machine (ELM). This system might aim to contribute to sustainable healthcare by providing a non-invasive method for seizure detection, which could improve the quality of life for epilepsy patients. If you have the abstract or any specific questions about the paper, feel free to ask!

2. TITLE –Real-time human detection and counting system

AUTHOR- Y. Guo et al., "Epileptic Seizure Detection by Cascading Isolation Forest-Based Anomaly Screening and Easy Ensemble," in IEEE Transactions on Neural Systems and Rehabilitation Engineering, vol. 30, pp. 915-924, 2022, doi: 10.1109/TNSRE.2022.3163503.

ABSTRACT- The abstract for the paper titled "Epileptic Seizure Detection by Cascading Isolation Forest-Based Anomaly Screening and Easy Ensemble" is not provided, but I can infer what the paper may be about based on its title and the context of the journal it's published in.

From the title, it seems the paper proposes a method for detecting epileptic seizures using a combination of techniques. "Cascading Isolation Forest-Based Anomaly Screening" suggests the use of isolation forest, a machine learning algorithm for anomaly detection, in a cascading manner, which likely involves multiple stages of screening. "Easy Ensemble" could refer to an ensemble learning technique aimed at improving classification performance, often used in imbalanced datasets.

3. TITLE – Detection of Epileptic Seizures using Machine Learning

Abstract- S. Sharma and A. Arora, "Detection of Epileptic Seizures using Machine Learning," 2022 5th International Conference on Advances in Science and Technology (ICAST), Mumbai, India, 2022, pp. 1-6, doi: 10.1109/ICAST55766.2022.10039516.

ABSTRACT- The paper titled "Detection of Epileptic Seizures using Machine Learning," presented at the 2022 5th International Conference on Advances in Science and Technology (ICAST) in Mumbai, India, likely explores the application of machine learning algorithms for the detection of epileptic seizures. Machine learning techniques are increasingly being used in healthcare for various tasks, including the analysis of medical data for disease detection and diagnosis. In the case of epilepsy, machine learning can be employed to analyze signals such as EEG (electroencephalogram) data to identify patterns indicative of seizure activity.

4. TITLE - Car plate detection engine based on conventional edge detection technique"

AUTHOR- Mokayed, H., Meng, L. K., Woon, H. H., & Sin, N. H. (2014). Car plate detection engine based on conventional edge detection technique.

ABSTRACT - Edge detection is a fundamental image processing technique used to identify boundaries within images. In the context of car plate detection, this technique can be applied to detect the edges of license plates within images captured by cameras, which is a crucial step in automated license plate recognition (ALPR) systems.

The abstract might discuss the specific edge detection techniques employed, the proposed algorithm or method for car plate detection, and possibly the performance evaluation of the system using benchmark datasets or real-world scenarios. It may also mention the potential applications and implications of the proposed approach in areas such as surveillance, traffic management, or law enforcement.

5. TITLE - Image descriptors for counting people with uncalibrated cameras,"

AUTHOR- Image descriptors for counting people with uncalibrated cameras. MIEEC 2013/2014-

Master in Electrical and Computers Engineering, University of Porto (pp. 1–73). Sumit, S. S.,

Watada, J., Roy, A., & Rambli, D. R. A. (2020).

ABSTRACT- The thesis likely explores methods and techniques for counting people using uncalibrated cameras, which can be a challenging task due to variations in camera angles, perspectives, and distances. Image to represent their content, and they play a crucial role in computer vision tasks like object detection and recognition.

6. TITLE- Human detection and counting system

AUTHOR- In The international conference on computer graphics, multimedia and image processing (CGMIP 2014).

ABSTRACT - The abstract might provide insights into the specific research problem addressed, the methodology employed, key findings or results, and the significance of the work in the broader context of computer graphics, multimedia, and image processing

System Design

The system design phase involves defining the architecture and components of the proposed system. The block diagram establishes the flow of data, starting with the live video capture from a laptop camera. The video feed is then pre-processed using MATLAB to optimize image quality for YOLOv6 analysis. The YOLOv6 algorithm performs real-time human detection, and the processed data is transmitted to an PYTHON. The PYTHON is responsible for counting detected humans and communicating the results to an GUI for immediate monitoring.

Data Collection

For training and testing the YOLOv6 algorithm, a dataset containing diverse images of human subjects is collected. The dataset aims to encompass various scenarios, including different lighting conditions, backgrounds, and group sizes. The collected data is annotated to provide ground truth information for the training process. Additionally, real- world video data is captured to simulate the system's performance in practical scenarios.

Training YOLOv6

The YOLOv6 algorithm is trained on the annotated dataset using a suitable deep learning framework. Training involves optimizing the model's weights to accurately detect humans in diverse conditions. Hyper parameter tuning is performed to achieve a balance between speed and accuracy. The trained model is then validated on a separate dataset to ensure generalization to unseen scenarios.

MATLAB Image Preprocessing

MATLAB is utilized for image preprocessing to enhance the quality of the live video feed. This involves techniques such as noise reduction, contrast adjustment, and feature extraction. The goal is to optimize the input data for YOLOv6 analysis, ensuring that the algorithm operates on images that are conducive to accurate human detection.

Integration of YOLOv6 and MATLAB

The integration of YOLOv6 and MATLAB is a critical step in the system's implementation. MATLAB serves as the interface between the raw video feed and the deep learning capabilities of YOLOv6. The pre-processed data is seamlessly fed into the YOLOv6 algorithm for real-time human detection. The integration is designed to maximize the strengths of both technologies, enhancing the overall accuracy and efficiency of the monitoring system.

PYTHON Programming

The PYTHON microcontroller is programmed to receive the processed data from YOLOv6 and perform the counting of detected humans. PYTHON code is developed to interface with the YOLOv6 output, extract relevant information, and communicate this information to the GUI. The programming ensures real-time processing and responsiveness, crucial for dynamic monitoring scenarios.

GUI Integration

The GUI is integrated with the PYTHON to provide immediate and dynamic feedback on the human presence in the monitored space. The display is designed to show the count of detected individuals in real-time. The integration ensures a user-friendly interface for quick decision-making and response based on the monitored data.

Ethical Considerations

Ethical considerations are paramount in the deployment of a real-time monitoring system. Privacy concerns are addressed by implementing anonymization mechanisms, ensuring that the system focuses solely on human presence detection without compromising individual privacy. Transparency in system operation and data handling is prioritized to build user trust.

Documentation and Reporting

Throughout the entire methodology, detailed documentation is maintained at each stage. This includes the rationale behind decisions, code documentation for software components, and comprehensive reports on testing and evaluation results. The documentation ensures transparency, facilitates replication, and provides insights for future enhancements or modifications.

Testing and Validation

The system undergoes rigorous testing to evaluate its performance under various conditions. This includes assessing the accuracy of YOLOv6 in human detection, the effectiveness of MATLAB image preprocessing, and the real-time counting and display capabilities of the PYTHON and GUI interface. The system is validated against diverse scenarios to ensure its adaptability and reliability.

Performance Evaluation

Performance metrics are defined to quantitatively evaluate the system's accuracy, speed, and efficiency. Metrics include detection accuracy, false positive rate, counting precision, and overall system responsiveness. Comparative analyses are conducted to assess the proposed system against existing solutions and benchmarks.

Optimization and Fine-Tuning

Based on the results of testing and performance evaluation, optimization and fine-tuning are performed to enhance the system's efficiency. This may involve adjusting algorithm parameters, refining image preprocessing techniques, or optimizing PYTHON code for better real-time processing. Iterative improvements are made to address any identified challenges or limitations.

CHAPTER 3

SYSTEM CONFIGURATION

3.1 SYSTEM CONFIGURATION

HARDWARE CONFIGURATION

HDD: >90GB

PROCESSOR: >Pentium IV 2.4GHz

SYSTEM TYPE: 32bit / 64 bit

RAM: >2GB

OS: WINDOWS 7/8/8.1/10

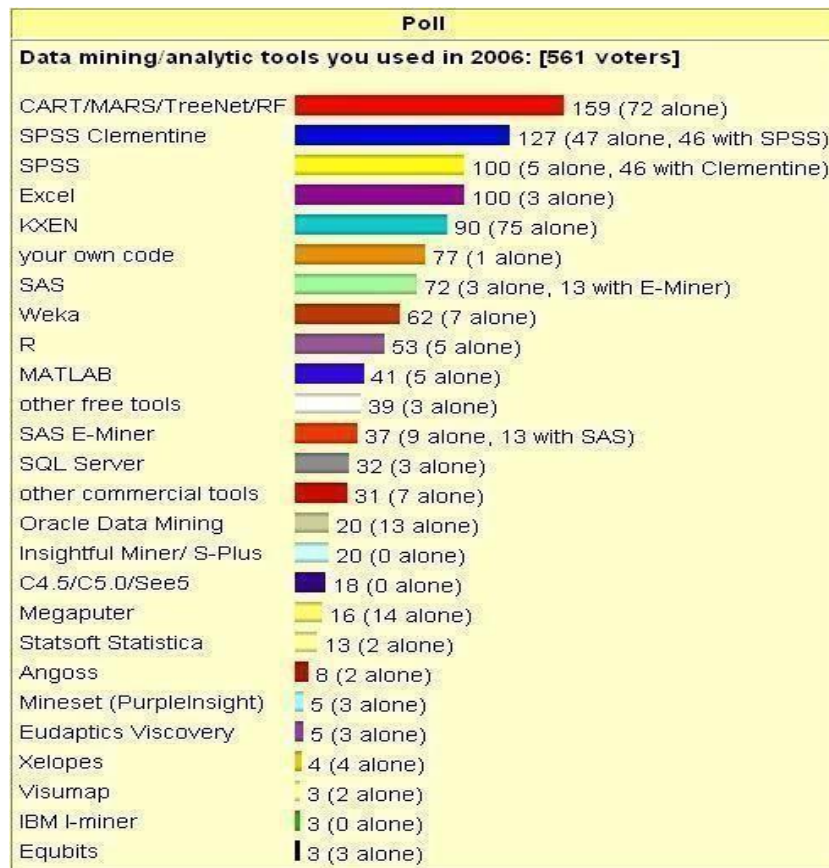
SOFTWARE CONFIGURATION

TOOL : MATLAB 2019a

TOOL BOX : Image Processing Tool Box

SOFTWARE DESCRIPTION

MATLAB is a great and flexible tool, more than accomplish of performing the data mining. It is clear that MATLAB has not to be given due concentration in this arena. Figure 1.1 illustrate the, while a comparatively trendy data mining tool, MATLAB is not so far in the group of packages such as Clementine, Wake and still Excel. In adding together, though MATLAB is selected more regularly than Oracle, it is usually used in combination with other tools. Where-as Oracle is implementing as the Stand-alone tool over 50% of the time, MATLAB is use on its own just over a 12% of the time table 1.2 summarizes the place of MATLAB over the last past 7 years. In spite of the fact that MATLAB is presently capable of the stage, some of the most trendy data mining technique existing, such as those being analyse this project, it has not so far become one of the groups of choice in this meadow. The popularity of these methods is detailed in Table 1.1, which is based on a samples of 16 altered data mining methods over the last 4 year period from 2013 TO 2016



Percentage	N/a	5%	5%	3%	2%	2%	5%
------------	-----	----	----	----	----	----	----

Table 3.1: celebrity of MATLAB in Data Mining

The combination of data mining tools provide the thesis allowed for an far large holistic technique to data mining in MATLAB than has been presented existing and in the addition, ensure the MATLAB can be use as a stand-alone tool, somewhat than in combination with former packages. These case studies ensure that data mining in MATLAB become a gradually more clear-cut task, as the suitable tools for known investigation become visible. As a logical expansion of the combination provide, recommendation is given with consider the formation of a data mining toolbox for MATLAB. The opportunity for addition to this works numerous, not only in terms of extend the tools them-selves but and also of data mining in MATLAB as an entire.

3.2 Project Overview

Due to the broad and undefined environment of this case study it is very important that we focal point on the number of exact tools and case study. The data mining tools in the region of which this study case will revolve are: the Neural-Network Toolbox, a proprietary tool presented from The Math-Works, distributors of MATLAB. The Fuzzy cluster and Data study Toolbox [Belasco et al. 2015] and the association Rule Miner and presumption study tool [Malone 2013], which are both open platform; and lastly an execution of the C4.5 judgment tree method [Woolf, 2015].

Ranks 10th with 5% of the votes

One causes for MATLAB's restricted use may be the fact that is a proprietary group (or) package. However, the fundamental MATLAB package is without difficulty enhanced, mainly by using the open-source tool-boxes and the script bundles, such as those examine in this case study. The detail MATLAB's data mining possible has positively not been entirely subjugated (as established in Figure 1.1 and Table 1.2), jointly with the current required for data mining tools, is the middle inspiration for carrying out this case study.

Method	2013	2014	2015	2016
Decision tree	Rank:1 (15%)	Rank:1 (15%)	Rank:1 (16%)	Rank:1 (13%)
Clustering	Rank:2 (11%)	Rank:2 (11%)	Rank:3 (10%)	Rank:2 (12%)
Neural nets	Rank:5 (8%)	Rank:4 (8%)	Rank:5 (8%)	Rank:6 (7%)
Association Rules	Rank:6 (7%)	Rank:7 (4%)	Rank:4 (8%)	Rank:7 (6%)

MATLAB	2010	2011	2012	2013	2014	2015	2016
Rank	∞	7.0	7.0	14.0	9.0	15.0	10.0

Table 3.2: Polls of trendy Data Mining Methods

The name MATLAB remains for lattice research facility. MATLAB was initially written to give simple access to lattice programming created by the LINPACK what's more, EISPACK ventures. Today, MATLAB motors join the LAPACK what's more, BLAS libraries, installing the best in class in programming for lattice calculation. MATLAB has developed over a time of years with contribution from numerous clients. In college conditions, it is the standard instructional device for starting what's more, best in class courses in arithmetic, building, and science. In industry, MATLAB is the device of decision for high-efficiency research, improvement, and investigation. MATLAB highlights a group of extra application-particular arrangements called tool kits. Important to most clients of MATLAB, tool stash enables you to learn and apply specific innovation. Tool kits are complete accumulations of MATLAB capacities (M-records) that expand the MATLAB condition to take care of specific classes of issues. Regions in which tool kits are accessible incorporate flag handling, control frameworks, neural systems, fluffy rationale, wavelets, re-enactment, and numerous others.

3.3 MATLAB Overview

MATLAB is an elite dialect for specialized figuring. It incorporates calculation, perception, and programming in a simple to- utilize condition where issues and arrangements are communicated in common place numerical documentation.

Common place uses incorporate

Math and calculation

Algorithm improvement

Data obtaining

Modeling, re enhancement, and prototyping

Data investigation, investigation, and representation

Scientific and building illustrations

Application improvement, including graphical UI building

MATLAB is an intelligent framework whose fundamental information component is an exhibit that does not require dimensioning. This enables you to explain numerous specialized figuring issues, particularly those with grid and vector definitions, in a small amount of the time it would take to compose a program in a scalar no n interactive dialect for example, C Or FORTRAN.

3.4 THE MATLAB SYSTEM

The MATLAB framework comprises of five fundamental parts:

Improvement Environment

This is the arrangement of apparatuses and offices that assistance you utilize MATLAB capacities and records. A considerable lot of these instruments are graphical UIs. It incorporates the MATLAB work area and Command Window, a charge history, an editorial manager and debugger, and programs for review help, the workspace, records, what's more, the inquiry way.

The MATLAB Mathematical Function Library

This is a huge gathering of computational calculations going from basic capacities, similar to total, sine, cosine, and complex number- crunching, to more advanced capacities like network backwards, framework Eigen values, Bessel capacities, and quick Fourier changes.

The MATLAB Language

This is an abnormal state framework/exhibit dialect with control stream proclamations, capacities, information structures, input/yield, and protest situated programming highlights. It permits both "programming in the little" to quickly make snappy discard projects, and "programming in the huge" to make substantial

DESIGNS

MATLAB has broad offices for showing vectors and lattices as diagrams, and additionally commenting on and printing these charts. It incorporates abnormal state capacities for two- dimensional and three-dimensional information perception, picture handling, activity, and introduction illustrations. It too incorporates low-level capacities that enable you to completely tweak the presence of illustrations and in addition to assemble finish graphical UIs on your MATLAB applications.

The MATLAB External Interfaces/API.

This is a library that enables you to compose C and FORTRAN programs that collaborate with MATLAB. It incorporates offices for calling schedules from MATLAB (dynamic connecting), calling MATLAB as a computational motor, and for perusing and composing MAT-records.

CHAPTER 4

MATLAB DOCUMENTATION

4.1 MATLAB DOCUMENTATION

MATLAB gives broad documentation, in both printed and on the web design, to enable you to find out about and utilize the greater part of its highlights. In the event that you are another client, begin With this Getting Started book. It covers all the essential MATLAB highlights at an abnormal state, including numerous cases. The MATLAB online help gives undertaking focused and reference data about MATLAB highlights. MATLAB documentation is additionally accessible in printed shape and in PDF organizes

4.2 MATLAB ONLINE HELP

To see the online documentation, select MATLAB Help from the Help menu in MATLAB. The MATLAB documentation is sorted out into these principle themes, that are

Desktop Tools and Development Environment Startup and shutdown, the workarea, and different devices that assistance you utilize MATLAB

Mathematics Mathematical tasks and information investigation

Programming The MATLAB dialect and how to create MATLAB applications graphics Tools and systems for plotting, diagram explanation, printing, furthermore, programming with

Handle Graphics3-D Visualization Visualizing surface and volume information, straightforwardness, and review and lighting systems

Creating Graphical User Interfaces GUI-building devices and how to compose callback capacities

External Interfaces/API MEX-documents, the MATLAB motor, and interfacing to Java, COM, and the serial port

MATLAB additionally incorporates reference documentation MATLAB capacities:

Functions By Category Lists all MATLAB capacities assembled into classifications

Handle Graphics Property Browser Provides simple access to depictions of designs protest properties

External Interfaces/API Reference Covers those capacities utilized by the MATLAB outside interfaces, giving data on language structure in the calling dialect, portrayal, contentions, return esteems, and illustrations

The MATLAB online documentation likewise incorporates

Examples A record of cases incorporated into the documentation

Release Notes New highlights and known issues in the present

Printable Documentation PDF forms of the documentation appropriate for printing.

4.3 MATLAB'S POWER OF COMPUTATIONAL MATHEMATICS

MATLAB is utilized as a part of each feature of computational science. Following are a few regularly utilized scientific counts where it is utilized generally usually:

- Dealing with Matrices and Arrays
- 2-D and 3-D Plotting and illustrations
- Linear Algebra
- Algebraic Equations
- Non-straight Functions Statistics
- Data Analysis
- Calculus and Differential Equations
- Numerical Calculations
- Integration Transforms Curve Fitting

4.4 HIGHLIGHTS OF MATLAB

Following are the essential highlights of MATLAB:

- It is an abnormal state dialect for numerical calculation, representation and application advancement.
- It additionally gives an intelligent domain to iterative investigation, plan what's more, critical thinking.
- It gives immense library of numerical capacities for direct variable based math, measurements, Fourier examination, sifting, advancement, numerical coordination and comprehending standard differential conditions.
- It gives worked in illustrations to picturing information and instruments for making custom plots.
- MATLAB's customizing interface gives improvement devices for moving forward code

quality, practicality, and augmenting execution.

- It gives devices to building applications with custom graphical interfaces.
- It gives capacities to coordinating MATLAB based calculations with outer applications and dialects, for example, C, Java, .NET and Microsoft Excel.

5 Various other exceptional capacities

4.5 ENVIRONMENT OF MATLAB

MATLAB is generally utilized as a computational device in science and building incorporating the fields of material science, science, math and all building streams. It is utilized as a part of a scope of utilizations including:

- Flag preparing and Communications
- Picture and video Processing
- Control frameworks
- Test and estimation
- Computational back
- Computational science

4.6 CONDITION or ENVIRONMENT

SETUP: Neighbor hood Environment Setup

Setting up MATLAB condition involves few ticks. The installer can be downloaded from http://in.mathworks.com/downloads/web_downloads: Math Works gives the authorized item, a trial rendition and an understudy form as well. You have to sign into the site and sit tight a little for their endorsement. In the wake of downloading the installer the product can be introduced through couple of snap

4.7 UNDERSTANDING MATLAB ENVIRONMENT

MATLAB advancement IDE can be propelled from the symbol made on the work area. The principle working window in MATLAB is known as the work area. At the point when MATLAB is begun, area shows up in its default format. The work area has the accompanying boards

Current Folder - This board enables you to get to the task organizers and documents.

Order Window - This is the principle zone where charges can be entered at the order line. It is shown by the charge incite (>>).

Workspace - The workspace demonstrates every one of the factors made as well as transported in from documents.

Order History - This board shows or rerun charges that are entered at the charge line

Diagram of Project Chapters

Part 2: Design Considerations – Lays out the points of interest of the work done in this proposition. This part is of incredible significance in that it displays the techniques utilized as a part of both researching and combining the devices.

Part 3: Tool Investigation – Begins by presenting the contextual investigations whereupon the tests did are to be constructed. Continues with the examination of each of the tool stash, delineating the examinations did and any issues experienced in this region. Basically contains preparatory discoveries of this work, which a revival for the execution of our blend of apparatuses.

Part 4: Implementation and Results – Brings together the examination of the devices as the after-effects of blend are introduced and talked about.

Part 5: Findings and Evaluation – A concise assessment of the outcomes introduced in Section 4 in view of other comparable contextual investigations which were done as a major aspect of the investigative procedure of this work.

Part 6: Conclusion and Possible Extensions – Concludes the task, exhibiting both the discoveries of this work and the numerous potential outcomes for additionally look into around there.

CHAPTER 5

SECTION SUMMARY

5.1 SECTION SUMMARY

In this section, we have examined the bearing and points of this investigation. We have too picked up a review of MATLAB and what is required for us to accomplish as for information mining inside this bundle. It is to a great degree energizing to set out on something as new as this, especially since the work is done here couldn't just upgrade the handiness of MATLAB in performing information mining, yet acquire more prominent lucidity to its place the field in general. We now leave on the advancement of the philosophy required to achieve the goals which have been laid out.

5.2 MATLAB "GUIDE" TOOL

User amicable graphical interface

As per Galitz (2002, 15, 41 - 51), a graphical UI can be characterized as set of ethos and instruments, used to make intelligent correspondence between a program and a client. The writer of the book underlines the significance of planning process by introducing fundamental tenets. Appropriate visual piece is an absolute necessity. The point is to give the client tastefully wonderful workplace. Hues, arrangement and straightforwardness of look ought to be thought about precisely. Each capacity, catch or some other question ought to have its importance, basic and justifiable by a normal program client. Comparative parts ought to have closely resembling looks and utilization. Capacities should perform rapidly and result with needed result. Adaptability can be seen in this theme as being touchy to every client's information, abilities, encounter, and individual execution furthermore, different contrasts that may happen. A decent interface is straightforward, limits the number of activities and does what it is relied upon to do. It isn't a simple assignment to plan an productive and easy to use graphical interface. Fortunately Matlab gives an accommodating instrument called 'GUIDE'. Subsequent to writing guide into Matlab's summon line, a snappy begin window shows up. From the decision of commendable positions it is prescribed to pick 'Clear GUI'. In the new window it is conceivable to simplified each question into the region of the program. On the left half of the made figure there is a rundown of conceivable segments. The rundown incorporates a push catch, slider, tomahawks, static and alter writings – which will be depicted in points of interest in the following section. It likewise contains objects that will be quickly clarified beneath (exclusively in view of Mathworks.com):

Toggle Button – once squeezed remains discouraged and executes an activity, after the second snap it comes back to the raised state and plays out the activity once more;

Check Box – produces an activity when checked and shows its state (checked or on the other hand not checked), numerous choices may be ticked in a similar time;

Radio Button – like the check box, however just a single choice can be chosen at any given time, work begins working after the radio button is clicked;

List box – shows a rundown of things and empowers client to choose at least one from them;

Pop-up Menu – open a rundown of decisions when the button is squeezed; **Board** – bunches all parts what makes interface simple and justifiable, places of all items are with respect to the board and don't change while moving the entire board;

Button Group – like the board however ready to oversee particular conduct of radio and flip catches that are legitimately gathered;

ActiveX Component – permits showing ActiveX controls that are intuitive innovation augmentations of html. They empower sound, Java applets and live lines to be incorporated in a Web page.

After the first efficient, GUIDE stores the interface in two records .fig document, where the portrayal of entire realistic part is set and .m document, where the code that controls the activities can be found. Each property is kept in the .fig record and can be set specifically from GUIDE apparatus, on account of prepared assembled Property Inspector. All activities, ordinarily called 'callbacks' can be altered and changed in the document. Each and every segment has 'Tag' property, which is utilized while making the name of the callback allude once. To gain admittance to each characteristic, Matlab offers charge set. It requires reference to the object that is going to be changed and the name of the property, trailed by its esteem. Among different qualities, there is an activity trigger - callback task. It is imperative to know, that any component can have its own particular usage of this work. Other than activities in charge of activities of objects, there are two extra capacities actualized in .m record:

Opening capacity – executes errands before the interface ends upon unmistakable to the client;

Output work – if necessary, it returns factors to the order line. There is considerably more behind instruments and procedures of programming

CHAPTER 6

APPENDICES

Source Code:

```
function varargout = Main_data(varargin)
% MAIN_DATA MATLAB code for Main_data.fig
%   MAIN_DATA, by itself, creates a new MAIN_DATA or raises the existing
%   singleton*.
%
%   H = MAIN_DATA returns the handle to a new MAIN_DATA or the handle to
%   the existing singleton*.
%
%   MAIN_DATA('Property','Value',...) calls the local
%   function named CALLBACK in MAIN_DATA.M with the given input arguments.
%
%   MAIN_DATA('Property','Value',...) creates a new MAIN_DATA or raises the
%   existing singleton*. Starting from the left, property value pairs are
%   applied to the GUI before Main_data_OpeningFcn gets called. An
%   unrecognized property name or invalid value makes property application
%   stop. All inputs are passed to Main_data_OpeningFcn via varargin.
%
%   *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
%   instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help Main_data

% Last Modified by GUIDE v2.5 08-Apr-2024 15:28:06

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',    mfilename, ...
    'gui_Singleton', gui_Singleton, ...
    'gui_OpeningFcn', @Main_data_OpeningFcn, ...
    'gui_OutputFcn', @Main_data_OutputFcn, ...
    'gui_LayoutFcn', [] , ...
```

```

        'gui_Callback', []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if narginout
    [varargout{1:narginout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
if narginout
    [varargout{1:narginout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

Executes just before Main_data is made visible.
function Main_data_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to Main_data (see VARARGIN)

% Choose default command line output for Main_data
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes Main_data wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = Main_data_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

```

```

% Get default command line output from handles structure
varargout{1} = handles.output;
% --- Executes on button press in Start.
function Start_Callback(hObject, eventdata, handles)
% hObject    handle to Start (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
run('Main_Code.m');

% --- Executes on button press in Exit.
clc% --- Executes on button press in Start.
function Start_Callback(hObject, eventdata, handles)
% hObject    handle to Start (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
run('Main_Code.m');

% --- Executes on button press in Exit.

```

CHAPTER 7

OUTPUT

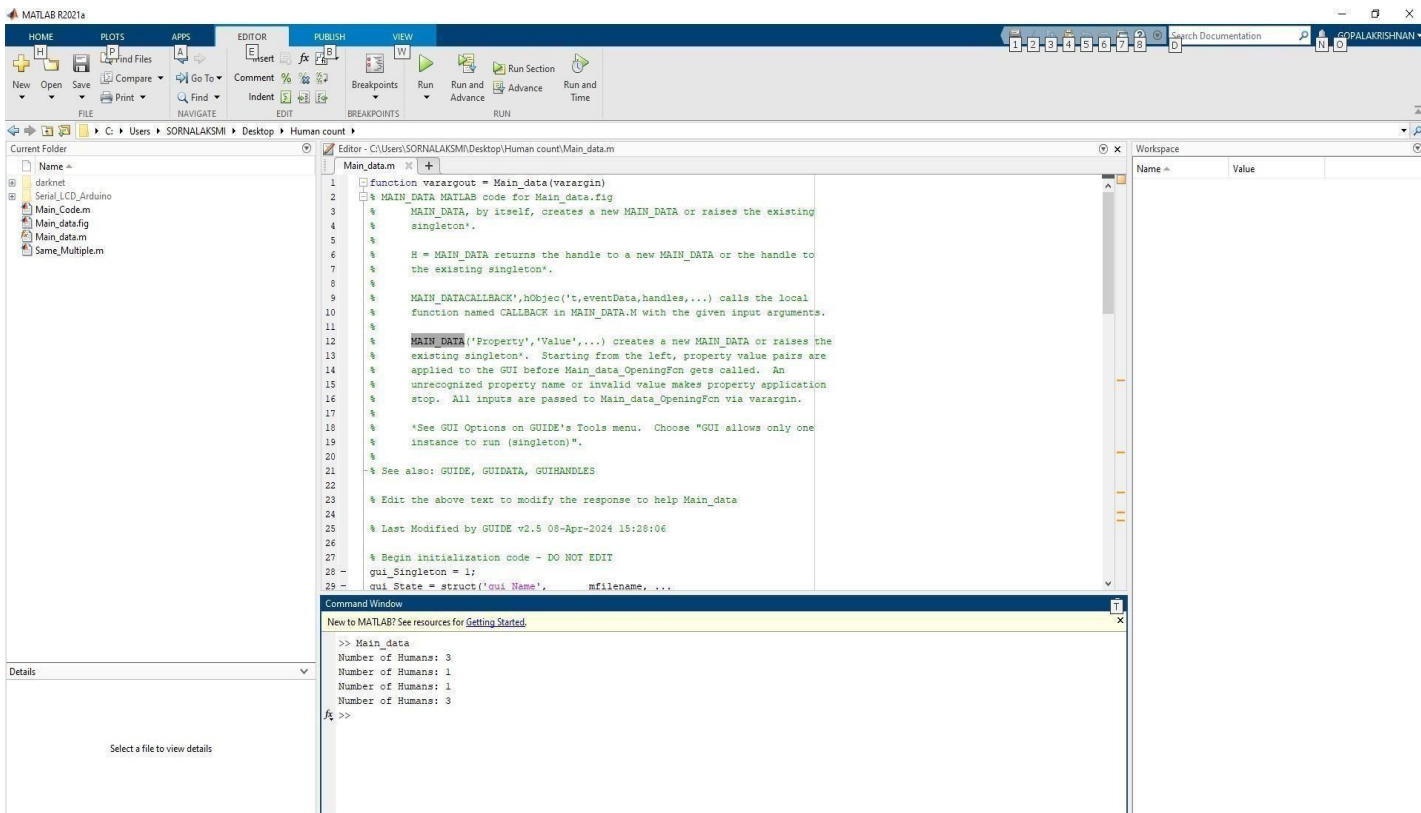


FIGURE 7.1

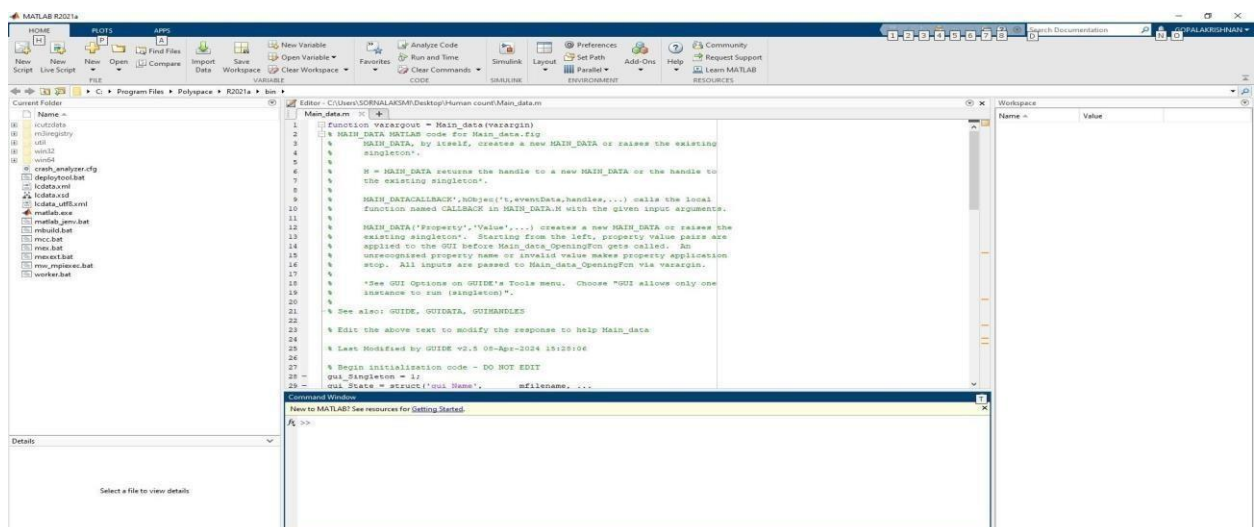


FIGURE 7.2

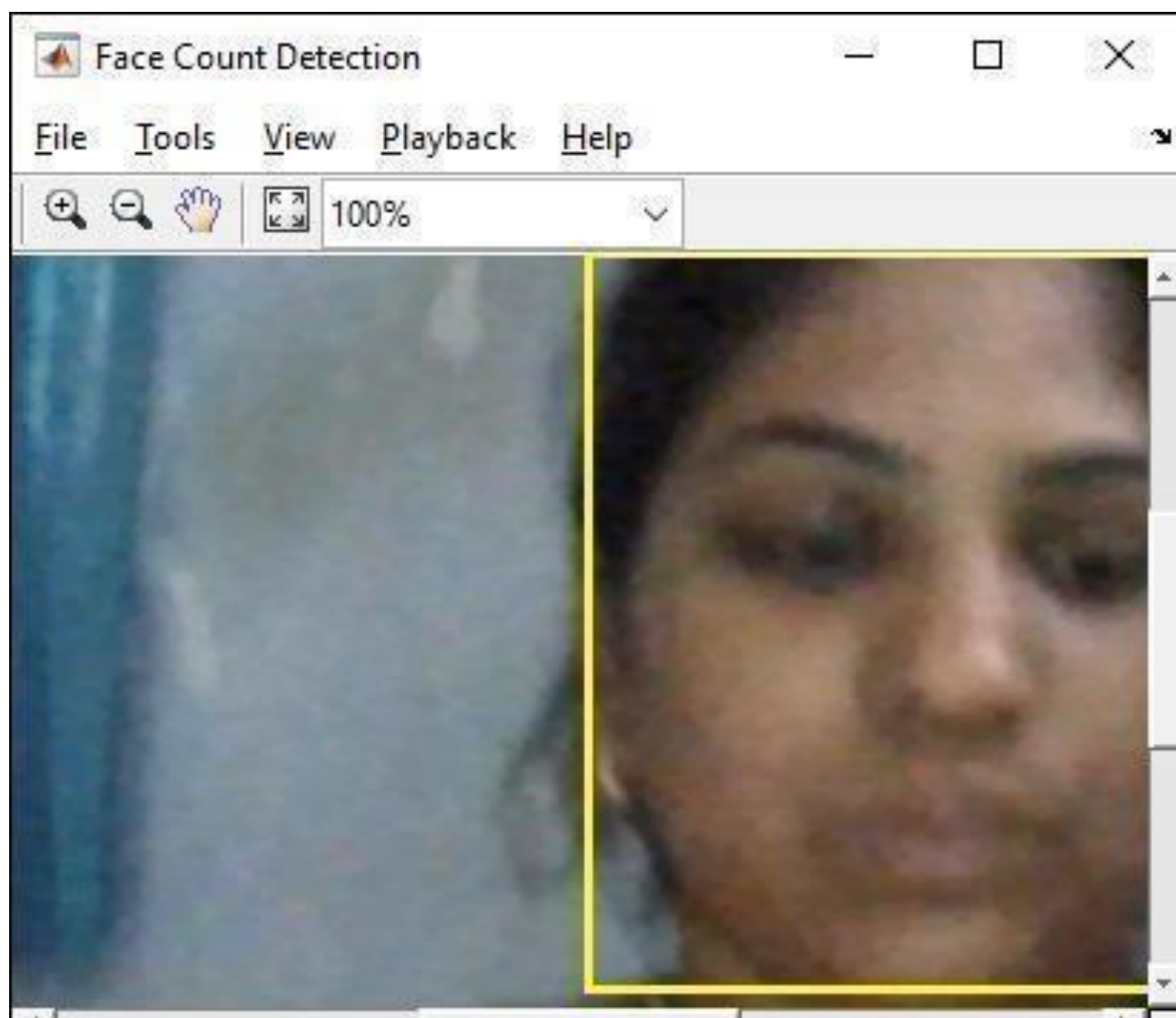


FIGURE 7.3

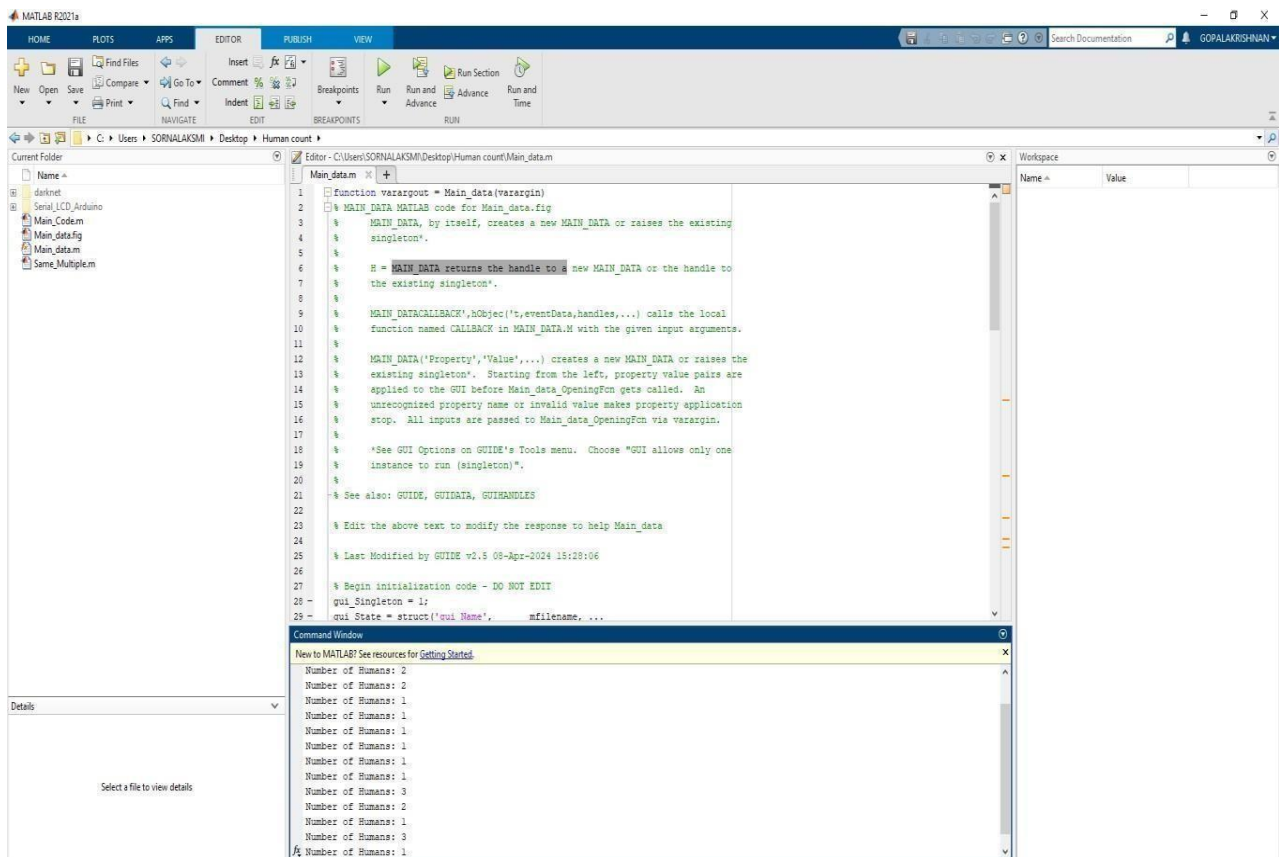


FIGURE 7.4

CHAPTER 8

CONCLUSION

In conclusion, the presented methodology outlines a meticulous and systematic approach to developing a state-of-the-art real-time human presence monitoring system. This innovative system integrates the YOLOv6 algorithm, MATLAB image processing, and an PYTHON microcontroller to achieve accurate human detection, real-time counting, and dynamic display capabilities on an GUI screen. The methodology encapsulates various stages, from problem definition and literature review to technology selection, system design, implementation, testing, and evaluation.

The implemented system stands out for its unique strengths, innovations, and contributions to the field of real-time monitoring. By leveraging the efficiency and accuracy of the YOLOv6 algorithm, the system ensures robust human detection even in dynamic and challenging scenarios. MATLAB's powerful image processing capabilities enhance the quality of input data, providing optimized images for YOLOv6 analysis. The integration with an PYTHON microcontroller facilitates real-time counting and seamless communication with an GUI, offering immediate and dynamic feedback on human presence in the monitored space.

The strengths of the proposed system lie in its comprehensive integration of advanced technologies, addressing the limitations observed in existing human presence monitoring systems. The iterative process followed in the methodology, starting from problem definition to the implementation and evaluation, ensures a well-rounded and effective solution. The system's ability to adapt to diverse scenarios, provide accurate real-time information, and offer a user-friendly interface positions it as a benchmark for future developments in real-time monitoring applications.

8.1 Future Work

While the proposed system represents a significant advancement, there are avenues for further research and development to enhance its capabilities and extend its applicability. The following outlines potential areas for future work:

1. Enhanced Object Detection Models

Future work could explore the integration of more advanced or specialized object detection models to further improve the accuracy and robustness of human detection. The field of computer vision is evolving, and staying abreast of the latest advancements can contribute to the continuous improvement of the monitoring system.

2. Multi-Sensor Integration

Investigating the integration of additional sensors, such as thermal cameras or depth sensors, could enhance the system's ability to detect humans in challenging conditions, including low light or obscured visibility. Combining data from multiple sensors can provide a more comprehensive understanding of the monitored environment.

3. Edge Computing and Distributed Processing

Exploring the implementation of edge computing techniques could optimize real-time processing by distributing computational tasks closer to the data source. This can reduce latency and improve the system's responsiveness, especially in scenarios with high data volumes or limited network bandwidth.

4. Adaptive Learning and Continuous Training

Implementing adaptive learning mechanisms within the system can enable continuous training and updating of the object detection model. This ensures that the system remains adaptable to changing environments, evolving appearances, and emerging challenges over time.

5. Privacy-Preserving Mechanisms

Future work could focus on the incorporation of advanced privacy-preserving techniques to address ethical concerns related to individual privacy during monitoring. This includes exploring methods for anonymizing data or implementing secure protocols for information transmission.

6. Scalability and Deployment in Large Environments

Scaling the system for deployment in larger environments, such as smart cities or expansive industrial spaces, poses an interesting challenge. Future work could explore strategies for optimizing the system's scalability, ensuring efficient operation and accurate monitoring in diverse and extensive settings.

7. User Customization and Configuration

Introducing features that allow users to customize and configure the system parameters based on specific monitoring requirements can enhance its adaptability. This includes adjusting sensitivity levels, defining regions of interest, or integrating user preferences into the monitoring process.

CHAPTER 9

REFERENCES

1. Z. A. Mulla, S. R. G, V. V. Lakshmi and S. C. Maturi, "Epileptic Seizure Detection System using Single Hidden Layer Extreme Learning Machine (ELM)," 2022 9th International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, India, 2022, pp. 298- 302, doi: 10.23919/INDIACom54597.2022.9763268.
2. P. S, B. P, V. S and S. K, "Epileptic Seizure Detection using Two-Layer Feature Extraction and Hyper-Parameter Optimization," 2022 6th International Conference on Computing Methodologies and Communication (ICCMC), Erode, India, 2022, pp. 1489-1492, doi: 10.1109/ICCMC53470.2022.9753964.
3. Y. Guo et al., "Epileptic Seizure Detection by Cascading Isolation Forest-Based Anomaly Screening and Easy Ensemble," in IEEE Transactions on Neural Systems and Rehabilitation Engineering, vol. 30, pp. 915-924, 2022, doi: 10.1109/TNSRE.2022.3163503.
4. S. Sharma and A. Arora, "Detection of Epileptic Seizures using Machine Learning," 2022 5th International Conference on Advances in Science and Technology (ICAST), Mumbai, India, 2022, pp. 1-6, doi: 10.1109/ICAST55766.2022.10039516.
5. K. F. Razi and A. Schmid, "Epileptic Seizure Detection With Patient-Specific Feature and Channel Selection for Low-power Applications," in IEEE Transactions on Biomedical Circuits and Systems, vol. 16, no. 4, pp. 626-635, Aug. 2022, doi: 10.1109/TBCAS.2022.3188966.
6. H. Tiwary, P. Rout and A. Bhavsar, "Deep Learning with Spatial and Channel Attention for Epileptic Seizure Type Classification Using Frequency Characterization," 2022 IEEE 32nd International Workshop on Machine Learning for Signal Processing (MLSP), Xi'an, China, 2022, pp. 1-6, doi: 10.1109/MLSP55214.2022.9943375.
7. Mokayed, H., Meng, L. K., Woon, H. H., & Sin, N. H. (2014). Car plate detection engine based on conventional edge detection technique.
8. In The international conference on computer graphics, multimedia and image processing (CGMIP 2014).

9. The Society of Digital Information and Wireless Communication. Mokayed, H., Shivakumara, P., Woon, H. H., Kankanhalli, M., Lu, T., & Pal, U. (2021).
10. A new DCT-PCM method for license plate number detection in drone images. *Pattern Recognition Letters*, 148, 45–53. <https://doi.org/10.1016/j.patrec.2021.05.002> Nikouei, S. Y., Chen, Y., Song, S., Xu, R., Choi, B.-Y., & Faughnan, S. Y. (2018).
11. Real-time human detection as an edge service enable by a lighthouse CNN. In 2018 IEEE international conference on edge computing (pp. 125–129). Piciarelli, C., & Foresti, G. L. (2011).
12. Surveillance-oriented event detection in video streams. In *Video event detection* (pp. 32–41). Raghavachari, C., Aparna, V., Chithira, S., & Balasubramanian, V. (2015).
13. A comparative study of vision based human detection techniques in people counting applications. In *Second international symposium on computer vision and the internet* (pp. 461–469). Redmon, J. (2021, August 18). yolo. Retrieved from pjdreddie: [https:// pjdreddie.com/darknet/yolo/](https://pjdreddie.com/darknet/yolo/) Reis, J. V. (2014).
14. Image descriptors for counting people with uncalibrated cameras. MIEEC 2013/2014-Master in Electrical and Computers Engineering, University of Porto (pp. 1–73). Sumit, S. S., Watada, J., Roy, A., & Rambli, D. R. A. (2020).