Abstract

The human detection robot is a state-of-the-art device designed to detect the presence of humans in a given area. Equipped with advanced sensors and algorithms, the robot is able to accurately detect human presence and provide real-time alerts to the user. One of the key features of the human detection robot is its ability to operate in a variety of environments, including both indoor and outdoor settings. The robot's robust design allows it to withstand harsh weather conditions and operate reliably in a wide range of temperatures. Additionally, the human detection robot is easy to use and requires minimal training to operate. The user can easily configure the robot's settings and adjust its sensitivity to accurately detect human presence in the desired area. Overall, the human detection robot is a valuable tool for any company looking to enhance security and improve safety in their premises. The project's domain is 'Robot Fencing with motion sensing camera' where the system detects any human occurrence and stops working if there is any human detected in its vicinity .The proposed system is developed to enhance the security of the existing system. The system is designed based on object detection algorithm such as yolo v5 model. The system performs and detects human efficiently.

Acknowledgement

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without the mention of a number of individuals whose professional guidance and encouragement helped me in the successful completion of this report work. We take this opportunity to thank Dr. Ashok Shettar, Vice-chancellor and to Dr. B S.Anami, Registrar, KLE Technological University, Hubballi. We also take this opportunity to thank Dr. Meena S M, Professor and Head, School of Computer Science and Engineering for having provided us with an academic environment which nurtured our practical skills contributing to the success of our project. We sincerely thank our guide Ms. Neha Tarannum, Assistant Professor, Department of Computer Science and Engineering for her guidance, inspiration and wholehearted co-operation during the course of completion.

We sincerely thank our project co-ordinator Prof.Mahesh Patil for his support, inspiration and wholehearted co-operation during the course of completion.

Our gratitude will not be complete without thanking the Almighty God, our beloved parents, our seniors and our friends who have been a constant source of blessings and aspirations.

Vaibhav Kumar
Aman Raj
Anand Doddamani
Yajas Menon

Contents

1 Introduction

Introducing the human detection robot - a state-of-the-art device designed to accurately detect the presence of humans in a given area. Equipped with advanced sensors and algorithms, the robot is able to quickly and reliably identify human presence, providing real-time alerts to the user.

The human detection robot is an invaluable tool for any company looking to enhance security and improve safety in their premises. With its ability to operate in a variety of environments, including both indoor and outdoor settings, the robot is versatile and durable, making it ideal for use in a wide range of settings.

In addition to its versatility and durability, the human detection robot is also easy to use and requires minimal training to operate. The user can easily configure the robot's settings and adjust its sensitivity to accurately detect human presence in the desired area. This user-friendly design makes the robot accessible to individuals with a wide range of technical expertise, allowing companies to deploy the robot quickly and efficiently.

The human detection robot is not only versatile and easy to use, but it also provides real-time alerts to the user. When the robot detects human presence, it sends an alert to the user, allowing them to quickly respond to any potential security or safety concerns. This real-time notification system allows companies to react quickly to potential threats and ensures that their premises are always protected.

In the proposed system , we have developed a system that identifies Humans and distinguishes them from other objects in the environment and track the movements. The system operates inside the cage of the industry and can to communicate its findings to a central control system, which is used to trigger alarms or other actions in response to the presence of humans.

1.1 Motivation

There are several key motivations for using a human detection robot in a company. First and foremost, the robot can enhance security and improve safety in the company's premises. By accurately detecting the presence of humans in a given area, the robot can alert the user to potential security threats and help prevent accidents or other safety incidents.

Another key motivation for using a human detection robot is its versatility and durability. The robot is able to operate in a variety of environments, including both indoor and outdoor settings, and is built to withstand harsh weather conditions. This makes it an ideal tool for use in a wide range of settings, including industrial facilities, warehouses, and other environments where human presence may be important to monitor.

Additionally, the human detection robot is easy to use and requires minimal training to operate. This user-friendly design makes the robot accessible to individuals with a wide range of technical expertise, allowing companies to deploy the robot quickly and efficiently.

Overall, the human detection robot is a valuable tool for any company looking to enhance security and improve safety in their premises. Its advanced sensors and algorithms, versatile design, and user-friendly interface make it an essential component of any security system.

1.2 Literature Survey

The paper "Human Detection Robot using YOLOv5 Algorithm". According to the abstract of the paper, the main focus of the research was to design and develop a human detection robot using the YOLOv5 object detection algorithm. The paper describes the design and implementation of the robot, including the hardware and software components used. It also discusses the performance of the robot in detecting humans, and compares the results to other object detection algorithms. The paper concludes that the YOLOv5 algorithm is a suitable choice for implementing a human detection robot, as it has a high detection accuracy and a fast processing speed. It also suggests that further work could be done to improve the performance of the robot, such as by adding additional sensors or using a larger training dataset. Overall, this research paper provides a detailed overview of the design and implementation of a human detection robot using the YOLOv5 object detection algorithm. It may be of interest to researchers and practitioners working in the fields of robotics, computer vision, and artificial intelligence.[1]

The paper "A comparative study of YOLOv5 models' performance for image localization and classification." According to the abstract of the paper, the main focus of the research was to compare the performance of different YOLOv5 models for image localization and classification tasks. The authors trained and evaluated several YOLOv5 models on the COCO dataset, a widely used dataset for object detection tasks. They compared the models' performance in terms of the mean average precision (mAP) metric, which measures the accuracy of object detection.

The paper found that the YOLOv5-SPP model had the best performance, followed by the YOLOv5-Tiny and YOLOv5-Large models. The authors also observed that the YOLOv5 models had a faster processing speed compared to other state-of-the-art object detection models such as RetinaNet and Faster R-CNN.

Overall, this research paper provides a detailed comparison of the performance of different YOLOv5 models for image localization and classification tasks. It may be of interest to researchers and practitioners working in the fields of computer vision and artificial intelligence.[2]

In the paper "Finding RTSP addresses for IP cameras, NVRs, and DVRs." This paper Provides information and instructions on how to locate the RTSP (Real-Time Streaming Protocol) address of a network-based camera, such as an IP camera, NVR (Network Video Recorder), or DVR (Digital Video Recorder). The RTSP address is a URL that allows a client device, such as a computer or smartphone, to connect to the camera and view the video stream in real-time. The article explains that the RTSP address may be listed in the camera's documentation or on the manufacturer's website, and provides steps for accessing the address through the camera's web interface.

The article also provides tips for troubleshooting common issues that may arise when trying to access the RTSP address, such as incorrect login credentials or network connectivity problems. It also offers suggestions for using third-party software or online services to view the video stream, if the RTSP address is not available or not working.[3]

In the paper "Configure IP Camera on Network." . This paper provides information and instructions on how to set up and configure an IP (Internet Protocol) camera on a network.

An IP camera is a type of network-based camera that is connected to a network using an Ethernet cable or WiFi connection. The article explains the steps involved in setting up an IP camera, including connecting the camera to the network, configuring the camera's settings, and accessing the video stream.

The paper also provides information on how to access the camera's web interface, which is a web-based control panel that allows the user to configure the camera's settings and view the video stream. It describes the various settings and options available in the web interface, such as resolution, frame rate, and network settings.[4]

In the research paper "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors." According to the abstract of the paper, the main focus of the research was to evaluate the performance of the YOLOv7 object detection algorithm on various benchmarks. YOLOv7 is a real-time object detection algorithm that has achieved state-of-the-art performance on several benchmarks.

The authors of the paper trained and evaluated the YOLOv7 algorithm on the COCO, VOC, and Open Images datasets, which are widely used for object detection tasks. They found that YOLOv7 outperformed other state-of-the-art object detection algorithms in terms of the mean average precision (mAP) metric, which measures the accuracy of object detection.

The paper also describes the "trainable bag-of-freebies" approach used in YOLOv7, which is a set of techniques that can be applied to improve the performance of the algorithm. The authors found that the use of these techniques significantly improved the performance of YOLOv7 on the COCO dataset.[5]

In the paper "Application of an Improved YOLOv5 Algorithm in Real-Time Detection of Foreign Objects by Ground Penetrating Radar." According to the abstract of the paper, the main focus of the research was to evaluate the performance of the YOLOv5 object detection algorithm for detecting foreign objects using ground penetrating radar (GPR) data. GPR is a non-invasive method of detecting objects and features underground using radar waves.

The authors of the paper used an improved version of the YOLOv5 algorithm, which they developed by adding additional layers to the model and fine-tuning it on a dataset of GPR images. They evaluated the performance of the improved YOLOv5 algorithm on a dataset of GPR images and found that it had a high accuracy in detecting foreign objects.

The paper also discusses the potential applications of the improved YOLOv5 algorithm in real-time detection of foreign objects using GPR data, such as in the fields of archaeology, civil engineering, and environmental monitoring.[6]

1.3 Problem Statement

To design and develop a security system that detects and tracks the movements of humans in real-time captured through IP camera to control the operation of robot installed in the cage of industry. .

1.4 Applications

There are several key applications for a human detection robot in a company. One of the primary uses of the robot is to enhance security and improve safety in the company's premises. By accurately detecting the presence of humans in a given area, the robot can alert the user to potential security threats and help prevent accidents or other safety incidents.

One specific application of the human detection robot is in industrial facilities and warehouses. In these environments, the robot can be used to monitor areas where human presence may be important, such as restricted access areas or areas with hazardous materials. By accurately detecting human presence in these areas, the robot can help prevent unauthorized access and ensure that employees are not exposed to potential dangers.

In addition to its security and safety applications, the human detection robot can also be used for research and development purposes. By providing accurate data on human presence in a given area, the robot can help researchers study human behavior and develop new technologies and strategies for improving safety and security.

Another applications of the human detection robot is a versatile and valuable tool for any company looking to enhance security and improve safety in their premises. Its advanced sensors and algorithms, versatile design, and user-friendly interface make it an essential component of any security system.

1.5 Objectives

The objectives for a human robot detection system for a company may vary depending on the specific needs and goals of the company. Some possible objectives for such a system could include:

- To extract the the Region of Interest(ROI) from the given input video.
- To build a model that can detect humans from the detected Region of Interest (ROI)
- To trigger an alarm and stop the robot working if an human is detected inside the cage.
- To restart the robot when there is no human inside the cage.

2 Requirement Analysis

Requirement analysis provides an overview of the entire system with purpose, scope, definitions, acronyms, abbreviations, references, and overview of the software requirements specification.

2.1 Functional Requirement

This section gives specific information about the functional requirements of the Robot Fencing Project. Functional requirements define a system or its component along with the main functionality of the system that defines the system as a whole unit. Below are the functional requirements of the system.

- The system shall be able to identify and isolate the region of interest (ROI) from the input video.
- The system shall be able to detect the presence of humans within the detected ROI.
- The system must be able to stop its movement when a human is detected within the designated area (e.g. a cage).
- The robot should be able to resume movement when no humans are detected within the designated area.
- The system should be able to communicate the presence or absence of humans to a central control system.
- The system should be able to adapt to changes in lighting and background.
- The system should be able to operate in a variety of environments, including indoor and outdoor spaces.

2.2 Non Functional Requirements

This section gives specific information about the Non-functional requirements of the Robot Fencing Project. Non-functional requirements deal with the performance attributes of the system. It justifies how the system should exist in terms of its usability, performance, reliability, maintainability, etc. Below are the Non-functional requirements of the system.

- The system should be able to detect objects with a high degree of accuracy, minimizing the number of false positives and false negatives.
- The model should be able to process the video stream in real-time, without introducing significant delays or lag.
- The model should be able to handle a wide range of input conditions, such as different lighting conditions and object poses, without degrading performance.
- The model should be able to handle a large number of object classes and detections without significantly slowing down.
- The model should be compatible with the IP camera and the device running YOLOv5, as well as any libraries or frameworks used to access the camera's video stream and display the model's output.
- The model should be easy to maintain and update, with clear documentation and a straightforward configuration process.
- The model should be secure, with measures in place to prevent unauthorized access and protect sensitive data.

2.3 Hardware Requirements

This section gives information about the Hardware Requirements for the Robot Fencing Project. This is related to the devices on which the algorithm is compatible. Just to run the algorithm, minimal hardware requirements are employed. Below are the Hardware requirements of the system.

- 8GB Ram.
- Intel i5 or i7 processor.
- 256 GB SSD and if there is a graphics card then it would be better

2.4 Software Requirements

This section gives information about the Software Requirements for the Robot Fencing Project. This is related to the operating systems on which this Project is compatible and technologies used.

2.4.1 Operating System to run

Machine Learning algorithm requires a platform or specifically, operating system to execute the program. This section includes various operating systems that are compatible . Below are the operating systems that share the software compatibility with the current project.

- Windows.
- Mac OS.
- Linux.

2.4.2 Technologies Used

This section includes various technologies that proved helpful to build the system effectively. Algorithms indeed need some powerful technologies to stand out in terms of efficiency, maintainability and every other aspect. The different technologies used in building the system are as below.

- Google Colab.
- Jupyter Notebook.

3 System Design

In this section gives us the basic View of the System Design and Model used in this project.

3.1 Architectural Design

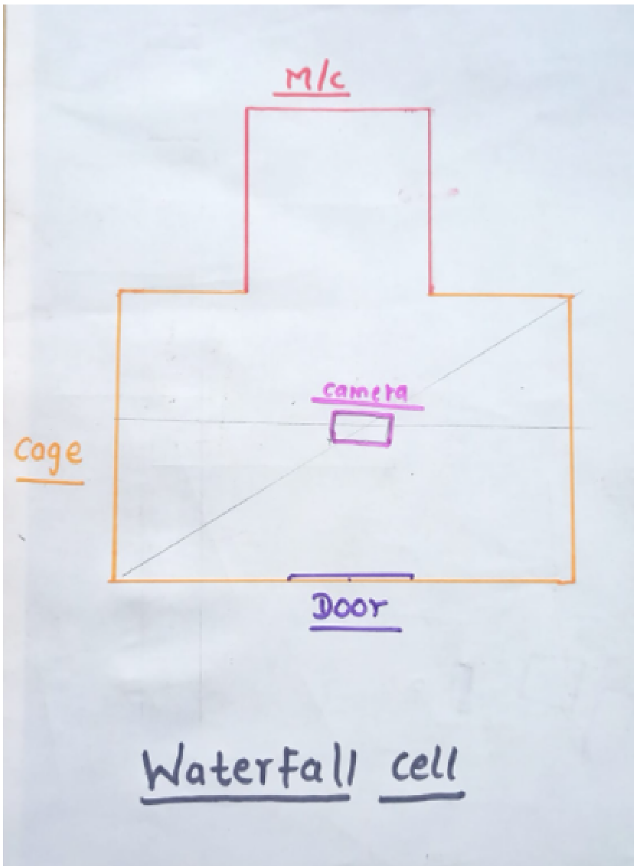

Figure 1: A Basic Process of the system

The fig 1. is the Architectural Design of Project . There will be a cage in which a robot is working . The Robot doesn't know to stop when there is an Human presence .

So there will be an IP camera attached at the top of the cage . So our model will take the video as an input provided by the IP camera and will tell the robot whether there is an Human presence or not .

If there is a Human presence inside the cage the robot will stop working and will again start moving when there is no Human inside the cage.

3.2 Proposed Methodology

In this section we will discuss about the models which we have used for this Project.

3.2.1 Yolo v5 Model

YOLO (You Only Look Once) is a popular real-time object detection system that is fast and accurate.YOLO v5 is the latest version of YOLO, which was released in 2021. It is an improvement over the previous versions, and it is faster and more accurate than its predecessors.

The YOLO v5 model works by dividing the input image into a grid of cells, and each cell is responsible for predicting a set of bounding boxes and class probabilities. The model makes a prediction for each cell in the grid, and these predictions are combined to form the final object detections

YOLO v5 is trained on large datasets of annotated images, and it uses a variant of the Darknet architecture, which is a convolutional neural network (CNN). The model is trained to predict the class probabilities and bounding boxes of objects in an image, and it is optimized using an objective function that measures the accuracy of the predictions.

There are many applications for YOLO v5, including object tracking, video surveillance, and autonomous vehicles. It is also used in a variety of industries, including retail, security, and transportation.

Model	size (pixels)	mAP^{val} 50-95	mAP^{val} 50	Speed CPU b1 (ms)	Speed V100 b1 (ms)	Speed V100 b32 (ms)	params (M)	FLOPs @640 (B)
YOLOv5n	640	28.0	45.7	**45**	**6.3**	**0.6**	**1.9**	**4.5**
YOLOv5s	640	37.4	56.8	98	6.4	0.9	7.2	16.5
YOLOv5m	640	45.4	64.1	224	8.2	1.7	21.2	49.0
YOLOv5l	640	49.0	67.3	430	10.1	2.7	46.5	109.1
YOLOv5x	640	50.7	68.9	766	12.1	4.8	86.7	205.7
YOLOv5n6	1280	36.0	54.4	153	8.1	2.1	3.2	4.6
YOLOv5s6	1280	44.8	63.7	385	8.2	3.6	12.6	16.8
YOLOv5m6	1280	51.3	69.3	887	11.1	6.8	35.7	50.0
YOLOv5l6	1280	53.7	71.3	1784	15.8	10.5	76.8	111.4
YOLOv5x6 + TTA	1280 1536	55.0 **55.8**	72.7 **72.7**	3136 -	26.2 -	19.4 -	140.7 -	209.8 -

Figure 2: Yolo v5 Model and its version

We have Used Yolo v5 Model For our Project.

4 Implementation

4.1 Video Capturing

We are taking video as an input from an IP Camera Provided by the industry .

Algorithm:

Step 1: Initialize the cv2 library and the numpy library.

Step 2: Create a video capture object using the cv2.VideoCapture() function, with an optional argument specifying the video source.

Step 3: Load a pre-trained YOLO object detection model using the cv2.dnn.readNetFromONNX() function and the specified model file.

Step 4: Read a file containing the list of class labels "coco.txt" using the built-in open() function and the file object's read() method. **Step 5:** Split the class labels into a list by splitting the string at each newline character (' ').

Step 6: Print the list of class labels to the console.

4.2 Training

Algorithm:

Step 1: Begin an infinite loop.

Step 2: Use the video capture object's read() method to retrieve the next frame of the video. If the frame is empty (i.e. None), break out of the loop.

Step 3: If the frame is empty (i.e. None), break out of the loop.

Step 4: Resize the frame to a specified width and height using the cv2.resize() function.

Step 5: Convert the frame to a blob (binary large object) using the cv2.dnn.blobFromImage() function, with various optional arguments specifying the scale factor, size, mean, and channel swap.

Step 6: Set the blob as the input to the object detection model using the model's setInput() method.

Step 7: Run the model's forward() method to get the detections.

Step 8: Assign the detections to a variable.

The Below Algorithm is for the video that we have sent after video capturing are of continuous human detection, which means the code will always run and will never end.

Step 1: Initialize empty lists for storing class IDs, confidences, and bounding boxes.
Step 2: Get the number of rows in the detections array.
Step 3: If the frame is empty (i.e. None), break out of the loop.
Step 4: Get the width and height of the image from the shape of the image array.
Step 5: Calculate the scale factor for the x-axis and y-axis based on the original and resized image dimensions
Step 6: Iterate over the rows in the detections array:
Step 7: Get the class ID, confidence, and bounding box coordinates for the current detection. Append the class ID and confidence to their respective lists. Scale the bounding box coordinates using the x and y scale factors. Append the scaled bounding box coordinates to the boxes list.

4.2.1 Reducing Overfitting

Here we are checking the confidence that in the range from 0 to 1 if the value in greater then 0.5 then only it is detecting the person properly

Algorithm:

Step 1: Iterate over the rows in the detections array. Get the confidence and class scores for the current detection.
Step 2: If the confidence is greater than 0.5: Find the index of the class with the highest score.
Step 3: If the score for this class is also greater than 0.5: Append the class index and confidence to their respective lists. Get the bounding box coordinates for the detection. Calculate the scaled bounding box coordinates using the x and y scale factors. Append the scaled bounding box coordinates to the boxes list.
Step 4: Use the cv2.dnn.NMSBoxes() function to filter the bounding boxes and confidences using non-maximum suppression, with optional arguments specifying the overlap threshold and confidence threshold. Assign the resulting indices to a variable.

4.3 Testing

This is to create frames from a video, which will run and the video will continue until the 'q' key is pressed.

Algorithm:

Step 1: Iterate over the indices returned by the non-maximum suppression function: Get the bounding box coordinates and class index for the current index. Use the class index to get the label for the class from the list of class labels. Get the confidence for the detection. Construct a string containing the label and confidence, formatted to two decimal places. Use the cv2.rectangle() function to draw a rectangle around the bounding box on the image. Use the cv2.putText() function to write the label and confidence string on the image.

Step 2: If the confidence is greater than 0.5: Find the index of the class with the highest score.

Step 3: Display the image using the cv2.imshow() function and a window title.

Step 4: Use the cv2.waitKey() function to wait for a key press event, with an optional argument specifying the time delay in milliseconds.

Step 5: If the key pressed is the 'q' key, break out of the loop.

5 Results

Using the YOLO v5 model for human detection in a robot would likely yield good results, as the model is designed to be fast and accurate at detecting a wide variety of objects in real-time. However, the specific results might as well depend on a number of factors, including the quality of the training data used to train the model, the performance of the hardware on which the model is running, and the specific application for which the model is being used.

5.1 Connection of IP Camera

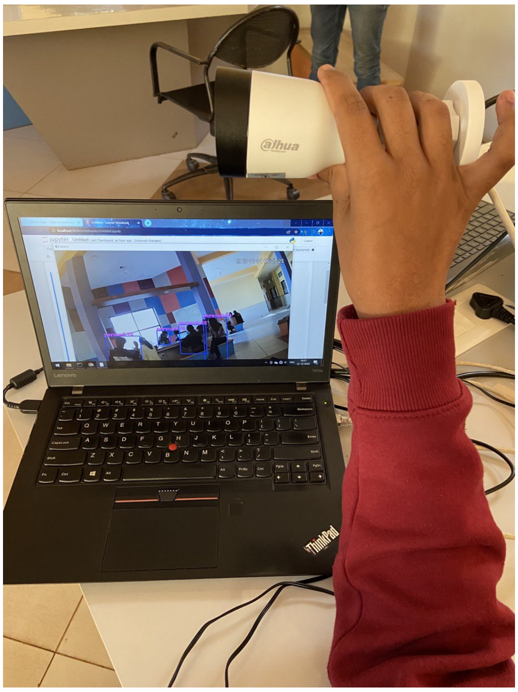

Figure 3: Setup

5.2 Frame that IP Camera Captured

Figure 4: Frame of IP Camera

5.3 Detection of Human using IP camera

Through IP camera we have trained the model for detecting a person . So this is the frame where the Persons Are Getting Detected.

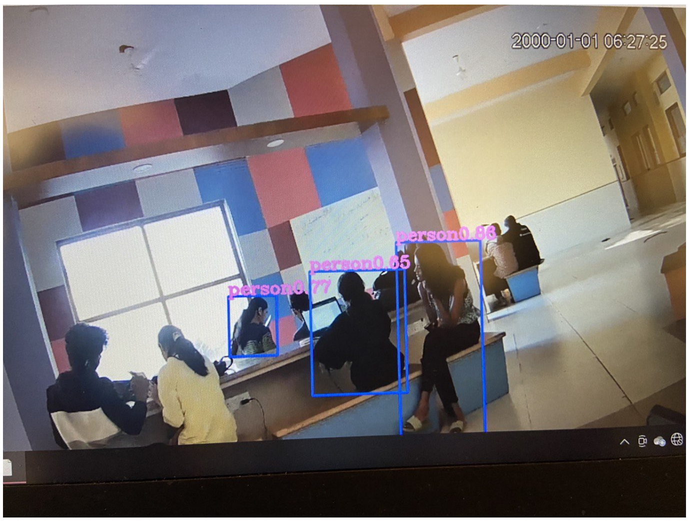

Figure 5: Detection of person

5.4 Final Output

After detecting the person our model will give 1 as a output and if no person is there then it will give 0 as output.

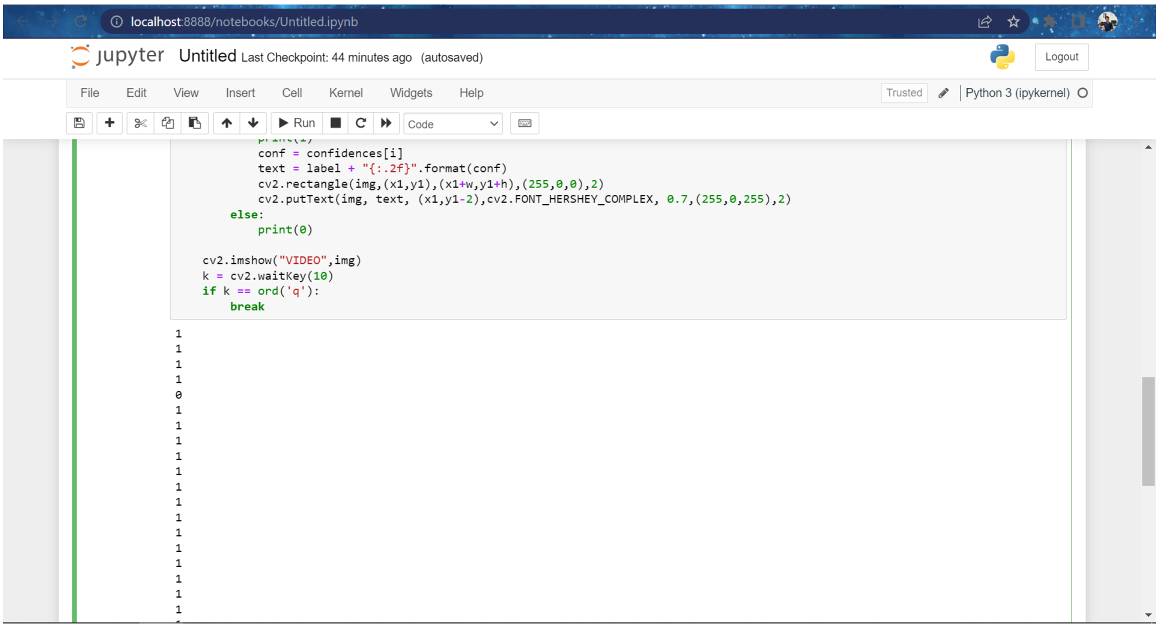

Figure 6: Final Output

5.5 Our Graphical User Interface

This is our Graphical user interface .

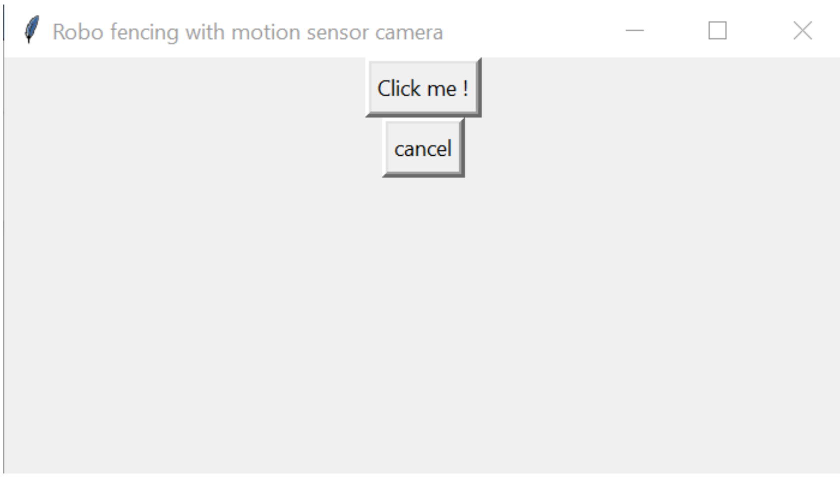

Figure 7: Final Output

6 Conclusion

The system is able to detect human using the IP camera given and based on that it gives result as 0's if there is no human present. and 1's if there was a human detected by the IP camera .

In general, the YOLO v5 model is efficiently detecting humans with high accuracy, provided that it has been trained on a sufficient amount of annotated data containing human images. It is also important to consider the specific environment in which the robot will be operating, as this can impact the model's performance.

References

[1] Literature Survey on Object Detection using YOLO Rekha B. S.1, Athiya Marium2, Dr. G. N. Srinivasan3, Supreetha A. Shetty

[2] A comparative study of YOLOv5 models performance for image localization and classification Marko Horvat, Gordan Gledec

[3]Finding RTSP addresses for IP cameras NVRs/DVRs

[4] To configure an IP camera on the network

[5] YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors Chien-Yao Wang1 , Alexey Bochkovskiy, and Hong-Yuan Mark Liao , Institute of Information Science, Academia Sinica, Taiwa

[6] Application of an Improved YOLOv5 Algorithm in Real-Time Detection of Foreign Objects by Ground Penetrating Radar Zhi Qiu 1,2 , Zuoxi Zhao 1,2,*, Shaoji Chen 1,2, Junyuan Zeng 1,2, Yuan Huang 1,2 and Borui Xiang