HumanDetector: Real-Time Human Detection and Counting

LAB REPORT

Submitted by

Shaurya Singh Srinet (RA2111032010006)

Shounak Chandra (RA2111032010026)

Parth Galhotra (RA2111032010029)

Under the Guidance of

Dr. C. Fancy

Assistant Professor, Department of Networking and Communications

In partial satisfaction of the requirements for the degree of

BACHELOR OF TECHNOLOGY
in
COMPUTER SCIENCE AND ENGINEERING

with specialization in Internet of Things

SCHOOL OF COMPUTING

COLLEGE OF ENGINEERING AND TECHNOLOGY

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

KATTANKULATHUR - 603203

MAY 2023

COLLEGE OF ENGINEERING & TECHNOLOGY SRM INSTITUTE OF SCIENCE & TECHNOLOGY
S.R.M. NAGAR, KATTANKULATHUE – 603 203
Chengalpattu District

BONAFIDE CERTIFICATE

Register No. RA2111032010006 Certified to be the bonafide work done by Shaurya Singh Srinet of III Year/IV Sem B. Tech Degree Course in the **Practical Course – 18CSC305J – Artificial Intelligence** in **SRM INSTITUTE OF SCIENCE AND TECHNOLOGY,** Kattankulathur during the academic year 2022 – 2023.

SIGNATURE

Faculty In-Charge
Dr. C. Fancy
Assistant Professor
Department of Networking and Communications
SRM Institute of Science and Technology

SIGNATURE

HEAD OF THE DEPARTMENT
Dr. Annapurani Panaiyappan. K
Professor and Head,
Department of Networking and Communications
SRM Institute of Science and Technology

TABLE OF CONTENTS

ABSTRACT

This project explores the development and assessment of algorithms for the real-time detection and enumeration of humans in images, videos, and camera feeds. The ability to accurately identify and count humans holds significant utility in diverse applications within image processing and computer vision domains. The implementation of these algorithms leverages Python programming language alongside essential tech-stacks such as OpenCV and TensorFlow. Through systematic experimentation and evaluation, this report presents benchmarks and insights crucial for advancing human detection and enumeration methodologies in computer vision research and practical applications.

Keywords: Computer Vision, Human Detection, Enumeration.

LIST OF ABBREVIATIONS

No.	WORD	ABBREVIATION
1	GUI	Graphical User Interface
2	CNN	Convolutional Neural network
3	HOG	Histogram of Oriented Gradients

INTRODUCTION

The field of Computer Vision stands as a cornerstone of artificial intelligence, aiming to imbue machines with the ability to comprehend and interpret digital images and videos. Through the extraction of meaningful insights from visual data, Computer Vision enables a wide array of applications across various domains. At its essence, Computer Vision encompasses techniques such as image segmentation, object detection, facial recognition, and pattern detection, fostering interdisciplinary collaboration and technological advancement.

Within Computer Vision, the detection and enumeration of objects represent fundamental tasks. Detection involves the identification of specific objects within digital imagery or video streams, while enumeration entails quantifying the instances of these objects accurately. This project focuses on the detection and enumeration of humans, leveraging real-time image, video, and camera feeds for comprehensive analysis.

The significance of this project lies in its contribution to advancing methodologies for human detection and enumeration within the realm of Computer Vision. By implementing algorithms and techniques using Python programming language and essential tech-stacks like OpenCV and TensorFlow, this project seeks to provide valuable insights and benchmarks crucial for both research and practical applications in the field.

HUMAN DETECTION & COUNTING

Human detection involves locating all instances of human beings present in an image or video. This task is typically achieved by searching all locations in the image, at various scales, and comparing small areas with known templates or patterns of people. Several methods exist for human detection, each offering different levels of accuracy and efficiency.

Common methods for human detection include:

1. Haar Cascade Classifier: Utilizing a predefined XML file, this method detects humans in real-time images and videos.

2. Histogram of Oriented Gradients (HOG): This method relies on predefined functions to achieve human detection, often providing improved accuracy compared to Haar Cascade Classifier.

3. TensorFlow: Leveraging this open-source API, which specializes in deep neural networks, can further enhance accuracy in human detection tasks.

This project focuses on implementing human detection and counting for various scenarios, including images, videos, and camera feeds.

In the case of images, users can select a real-time image from their local system and detect humans within it, accompanied by a count of detected individuals. Similarly, for videos, users can select a real-time video, and the system will detect humans continuously, providing counts for each frame per second. Additionally, the project facilitates human detection and counting through a camera feed, where users open the webcam, and the system detects and counts humans appearing in the feed.

ACCURACY

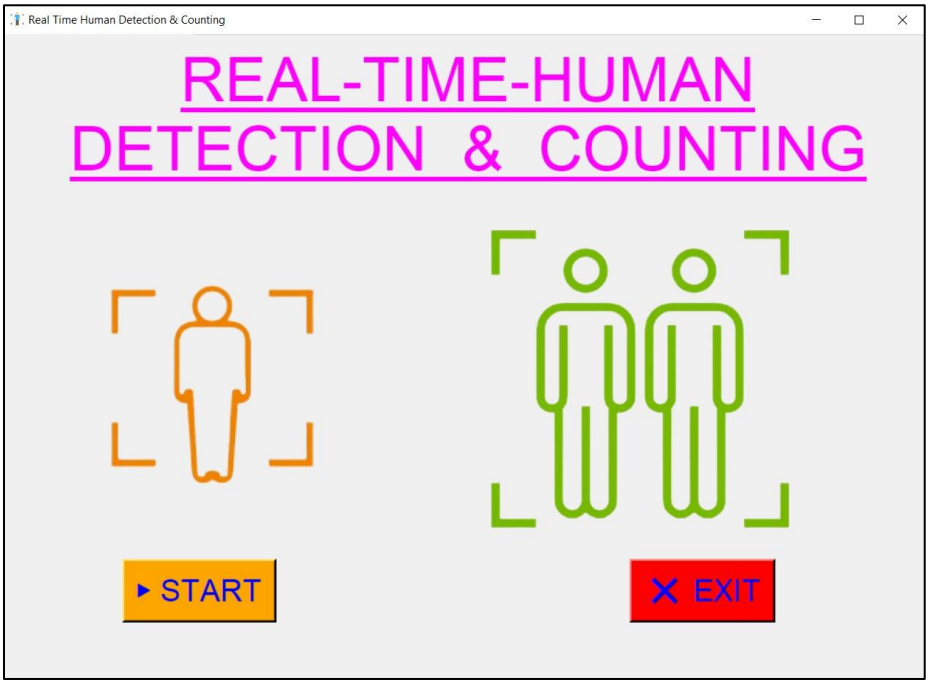
Accuracy is a critical aspect of any computer vision project, including human detection. Throughout the detection process in images, videos, and camera feeds, maintaining accuracy is paramount. In this project, we have set a threshold accuracy of 70% for human detection. Any object detected with an accuracy exceeding this threshold is considered a well-detected human, and a detection indicator is displayed around it during the process. This threshold serves to mitigate false detections during the process.

The concept of maximum accuracy delves into determining the highest level of accuracy achieved during the detection process. Tracking the maximum accuracy provides insights into the precision of our implemented application.

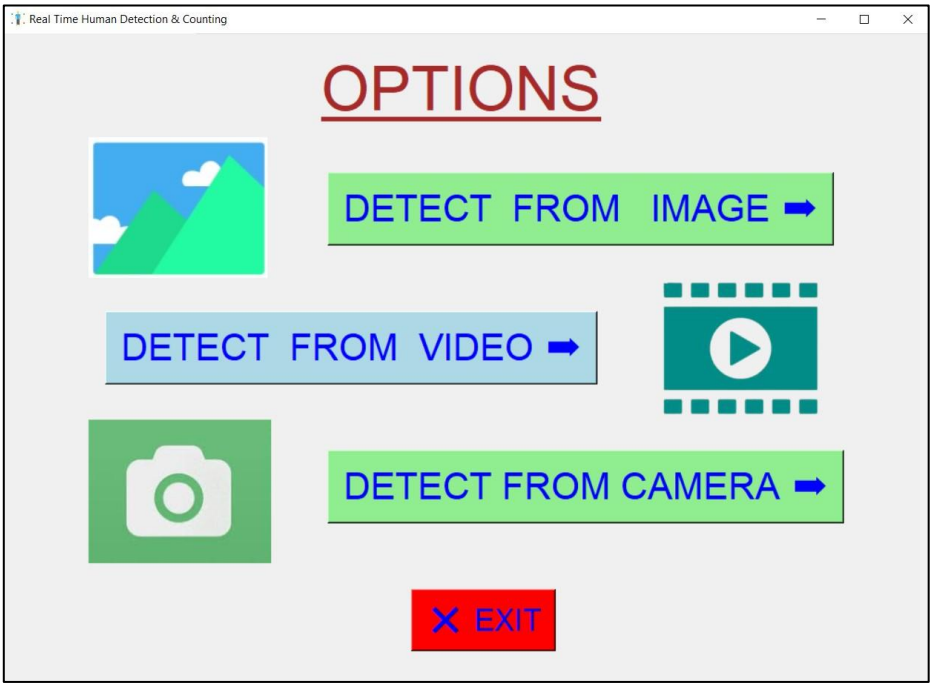
In the context of videos and camera feeds, maximum average accuracy becomes relevant. This metric reflects the highest average accuracy obtained across all frames in a running video or webcam feed. It is particularly useful for plotting purposes, as discussed in the subsequent chapter.

OUTPUT (SCREENSHOTS)

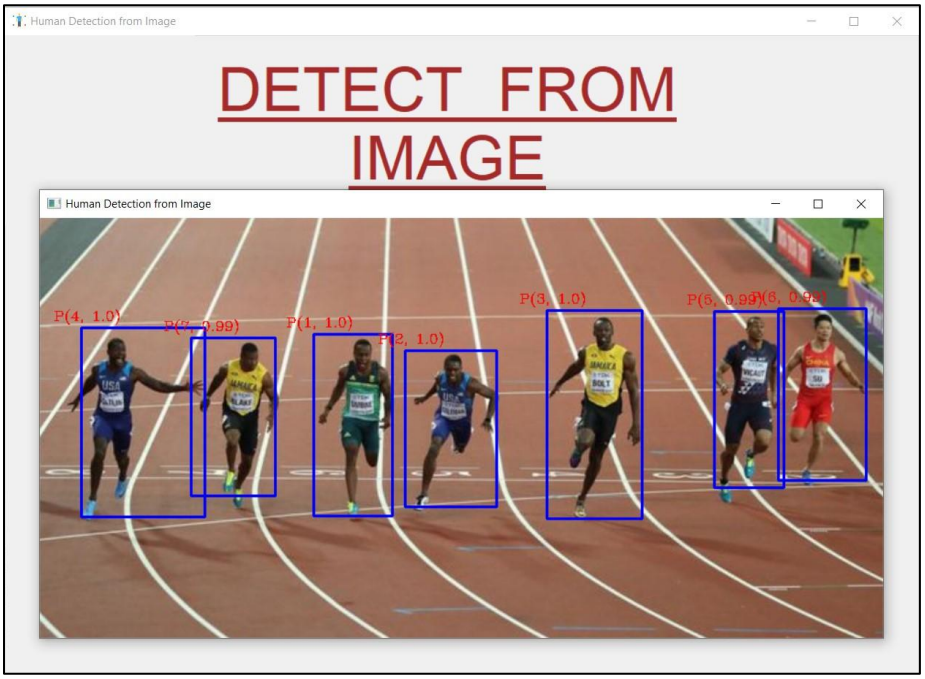
1. HOME PAGE:



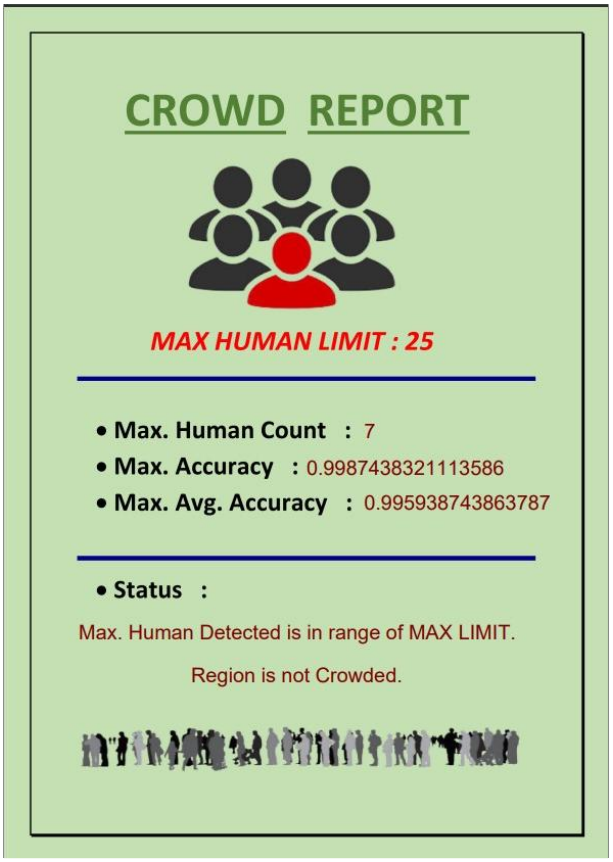
2. OPTIONS PAGE:



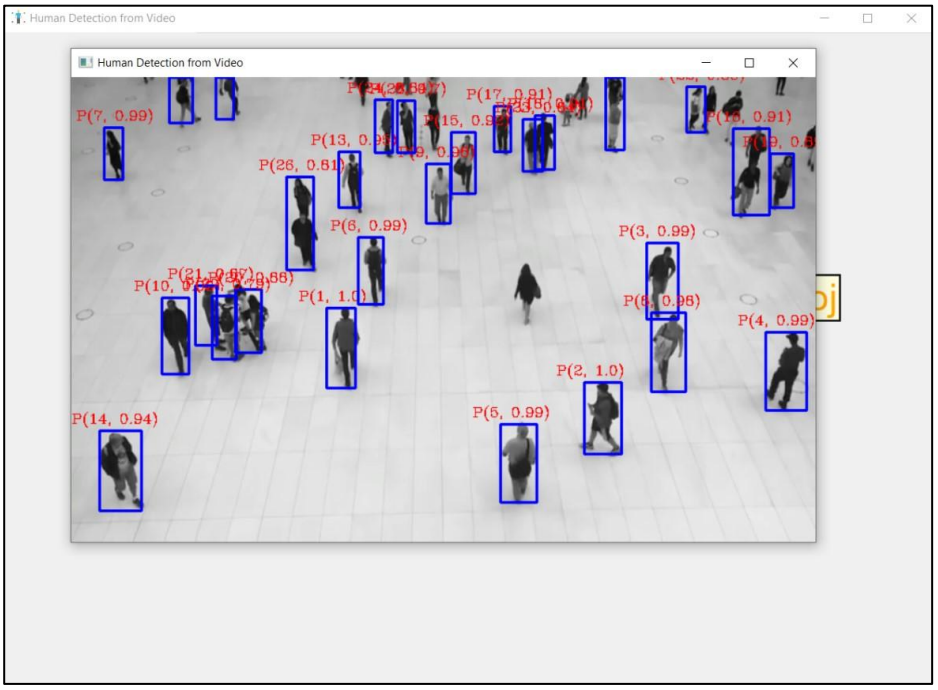
3. DETECT FROM IMAGE:



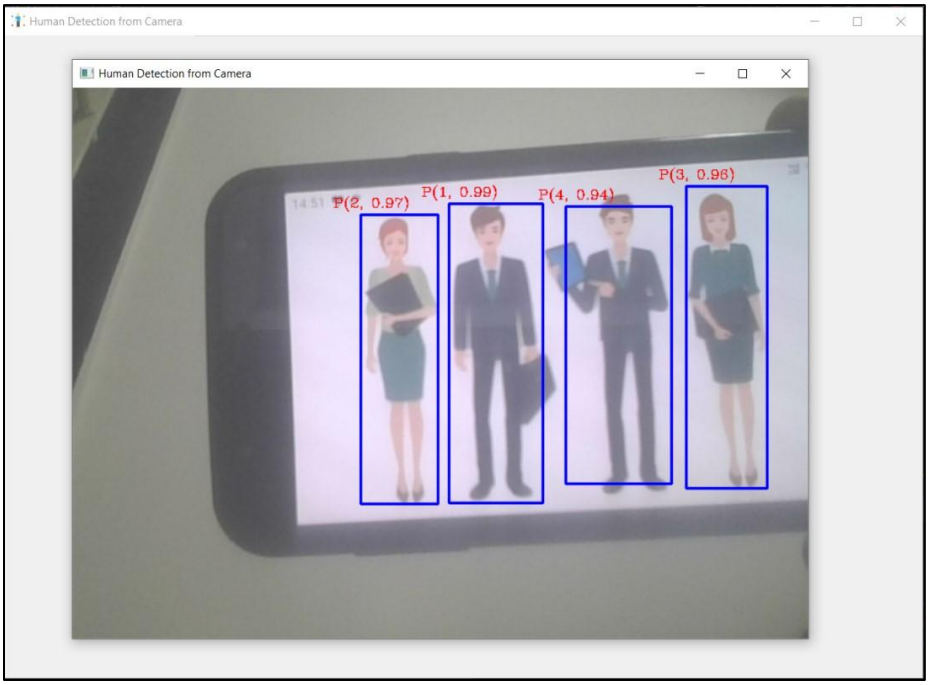
4. SAMPLE CROWDFUND REPORT GENERATED:



5. DETECT FROM VIDEO:



6. DETECT FROM CAMERA:



SCOPE

1. Implementation in Public Spaces: This project holds promise for deployment in malls and other public areas for analyzing crowd density. By monitoring maximum people counts and setting restrictions accordingly, it can contribute to effective crowd management strategies.

2. Automation of Manual Tasks: The application of this technology can extend beyond crowd analysis, potentially replacing various manual tasks with more efficient machine-driven processes. This could lead to increased productivity and resource optimization in various sectors.

3. Crowd Control Measures: As the project evolves, it may pave the way for the implementation of crowd control measures in specific areas or events. By leveraging real-time human detection and counting capabilities, proactive measures can be taken to mitigate overcrowding and ensure safety.

These future scopes highlight the potential of this project to not only revolutionize current practices but also address emerging challenges in crowd management and automation. Continued research and development in this field hold promise for further advancements and applications.

CONCLUSION

In the final phase of this project, we generated a Crowd Report based on the results obtained from the detection process. By setting a threshold human count, we were able to derive various messages corresponding to different human count scenarios detected. This aspect adds a practical dimension to our findings, enabling potential applications in crowd management and analysis.

REFERENCES

[1] J. E. Solem, "Programming Computer Vision with Python," 1st ed. O'Reilly, 2012.

[2] A. Kaehler and G. R. Bradski, "Learning OpenCV," 1st ed. O'Reilly, 2008.

[3] G. Zaccone, M. R. Karim, and A. Menshawy, "Deep Learning with TensorFlow," 1st ed. Packt Publishing, 2017.

[4] A. D. Moore, "Python GUI Programming with Tkinter," 1st ed. Packt Publishing, 2018.

[5] F. Lundh, "Python Standard Library," 1st ed. O'Reilly, 2001.