RnC Project Report-Height Detection

By-Anisha Deb & Shashank Rao  
*Robotics-n-Circuits*Manipal ,Kanranataka

*Abstract*— This paper outlines a real-time object and height detection system implemented using the YOLOv3 model and OpenCV library. The methodology involves image preprocessing with the cv2.dnn.blobFromImage() function to standardize inputs for neural network inference. Additionally, the system incorporates distance estimation algorithms to calculate object distances from the camera. The combined approach enables efficient distance estimation and object and its height detection, suitable for diverse applications including robotics and surveillance.

# Objective of the project

The project aims to create an efficient system mainly used in e-commerce websites which would further help customers choose better clothing options and choose the correct size for their equipment or apparel.

.

## WORKING OF THE PROJECT

*1.Loading the YOLOv3 Model: The YOLOv3 model is loaded using the cv2.dnn.readNet() function with the pre-trained weights and configuration files.*

*2. Loading Class Names: The names of the classes that the model can detect are loaded from a file named "coco.names".*

*3. Preprocessing Images: The cv2.dnn.blobFromImage() function preprocesses each frame by normalizing pixel values, resizing to match the network's input size, and performing mean subtraction.*

*4. Object Detection: The preprocessed frames are passed through the YOLOv3 model, and object detections are obtained along with their confidence scores.*

*5. Non-Maximum Suppression: Non-maximum suppression is applied to filter out overlapping bounding boxes and retain only the most confident detections.*

*6. Distance Estimation: The script calculates the distance of detected objects from the camera using the provided formula and focal length.*

*7. Displaying Results: Detected objects, along with their class labels, distances, and real heights, are annotated on the frame and displayed in real-time.*

*8. Looping Through Frames: The script continuously captures frames from the camera, processes them for object detection and distance estimation, and displays the results until the user quits by pressing 'q'.*

### B. CONTRIBUTION BY EACH MEMBER :

### The project was divided on the basis of each other strengths in Python and field of Machine learning.Aspects of YOLO and inner workings of The Convolutional neural network were understood using various resources by both members

|  |  |
| --- | --- |
| Anisha Deb | Producing the data set for shoulder width and known Camera DepthCoding Yolo Components, Height detection by calculating focal length and Depth from above dataset |
| Shashank Rao | Coding And Testing Various Algortihms For Better Calibration, such as Aruco Boards, Masking, etc.Making the PowerPoint for presentation, Brainstorming newer ideas for Product usage |

# Problems faced during the process

# Lack of Diverse Data for accurate results

# Poor processing power for clear updating of height

# Lack of RealSense camera for better z axis detection

# POSSIBLE SOLUTIONS

# Introduction of better ML language to get more

#### accurate results

# Having a physical component like a 3D camera for better distance estimation

# Usage of Tensor Flow to detect faces and get ratios of the face to get a better height estimation

# Potential Real-World applications Of Height Detection Software:

1. Retail Optimization:

Virtual Fitting Solutions: Integration of height detection software into virtual fitting rooms enables customers to obtain accurate measurements, facilitating the selection of clothing items that fit appropriately.

1. Security Enhancements:

Access Control Systems: Height detection software serves as a key component in access control mechanisms, verifying individuals' height against predefined criteria to regulate entry to secure facilities.

1. Construction and Architecture:

Precision Planning and Design: Architects and urban planners leverage height detection software to accurately measure buildings, structures, and terrain, facilitating meticulous planning and design processes.

1. Obstacle Avoidance and Navigation:

Height detection software enables robots to perceive obstacles accurately by assessing their height profile. By incorporating this information into their navigation algorithms, robots can navigate complex environments more effectively, avoiding collisions and optimizing path planning.

##### References

1. <https://www.youtube.com/watch?v=TGIpzLwkKLk&ab_channel=autopy>
2. <https://www.youtube.com/watch?v=fu2tfOV9vbY&ab_channel=RobMulla>
3. <https://pjreddie.com/projects/coco-mirror/>
4. <https://www.youtube.com/watch?v=ag3DLKsl2vk&ab_channel=codebasics>
5. <https://www.youtube.com/watch?v=IfRMV2MY9n0&ab_channel=codebasics>
6. <https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>