# 3D localization of objects of daily living activities for rehabilitation applications

Assisting paralysed people for conducting activities of daily living (ADL) is a major challenge in the area of collaborative robotics. The first predicament of wheelchair mounted robotic arms and/or exoskeletons is to identify and localize the objects for ADL in a 3-dimensional (3D) space so that they can be manipulated according to the user's need. This requires robust integration and adaptation of state-of-the-art algorithms in a seamless framework for faster, accurate, and reliable manner ensuring the user's safety. To this end, the current work focuses on the adaptation of the popular YOLO algorithm for image based object identification for detecting objects for ADL. In order to facilitate 3D localization of these objects we integrated the YOLO with a stereo vision based algorithm for depth perception. Thus the integrated system can provide both object identification and 3D localization in real-time so that a wheelchair mounted robotic arm and/or exoskeleton can be semi-autonomously guided to perform ADL for the disabled people. The proposed solution is resource efficient as it can be implemented using low-cost hardware so that the overall product can be affordable, environmentally efficient (as it may consume less power due to low computational complexity). This will lead to enhanced environment perception for the robots which can drastically improve the human robot interaction models for optimal assistance.

Additional Key Words and Phrases: YOLO, Stereo Vision, Object Detection

**ACM Reference Format:**

. 2023. 3D localization of objects of daily living activities for rehabilitation applications. 1, 1 (March 2023), 10 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

## 1 INTRODUCTION

Integration of robotic help into daily human life has been an avid area of research in the field of Artificial Intelligence. Researchers have been exploring to combine robots and artificial intelligence since the very early stages of their development in order to suit distinct demands in various fields. These intelligent robots are currently being utilised in a variety of industries to carry out a variety of tiresome tasks with fewer errors thanks to the deployment of various artificial intelligence techniques like computer vision, brain-computer interface, etc. Another illustration is the robotic nurse, who helps patients with daily activities. In the aggrotech industries, where automated fruit harvesting, disease detection, and plant nurturing robots are widely utilised to promote vegetation development more effectively, this field has also experienced tremendous expansion. Numerous robotic solutions have been introduced in this age of automation to simplify routine dexterity tasks. One such solution is the robotic hand, which has demonstrated to give a human being an extra hand to quickly do the routinely arduous home activities, such as Samsung's Bot-Handy for washing dishes and the robotic cleaner for cleaning homes. Contrarily, little has been done to improve the lives of those groups of people who are dependent on others due to a variety of medical issues. Although there are sophisticated robots that can schedule medication and assist the hand or walking of the crippled, both of these functions require sophisticated algorithms and approaches. Though many scholars have recently been enthusiastic about this field, and both the government and many healthcare industries have launched numerous proposals and studies in this area.

Author's address:



Manuscript submitted to ACM

The vOICe technology supports binocular vision with suitable stereoscopic camera hardware, aimed at assisting the blind.[8] G, Balakrishnan and G, Sainarayanan and Nagarajan, Raghul and Sazali, Yaacob [2007], deviced an Electronic Travel Aid(ETAs) to assist and improve the blind user's mobility in terms of safety and speed. The project aims to create an object detection system along with its 3D coordinates and robust object selection application for the robotic hand.The base of the work is set up with Two-Dimensional Real Time object detection using the YOLO algorithm, followed by Three Dimensional Real Time Object Detection system, implemented using Stereo Vision. The Three Dimensional Object Detection method uses the concepts and tools of the Two dimensional object detection, where two frames of two dimensional images is used. Both object detection algorithms are implemented on python, using the OpenCV library.

## 2 METHODOLOGY

### 2.1 The You Only Look Once (YOLO) Algorithm

The YOLO algorithm is a regression-based algorithm that can predict classes and bounding boxes for entire images as opposed to other algorithms that do so only for a specific region of intrigue for the image. It is also highly effective due to its speed, accuracy and learning capabilities. YOLO is preferred to other algorithms such as Region-based Convolutional Neural Network (RCNN), as RCNN relies on multiple iterations, whereas YOLO implements all of its predictions using a single fully connected layer. In this project, YOLO is implemented on Python, using the OpenCV library.

YOLO uses three techniques -

- The residual block, where the image is divided into N grids each with the same dimensions of S × S.
- Bounding box regression, where a bounding box gives an outline for the image.
- Intersection over Union (IoU), which describes how boxes overlap. This is used to create and output box the fits the image perfectly. [1] The predicted bounding boxes and confidence scores are computed at each grid cell. If the bounding box and the actual box match, then the IoU is 1. This helps eliminate bounding boxes that are not equivalent to the actual box. Each of these N grids is responsible for detecting and locating the object it contains.

### 2.2 Two Dimensional Real Time Object Detection

The image is segmented into cells, on a $19 \times 19$ grid. Then, each cell to forecast K bounding boxes. Four descriptors can be used to describe each bounding box. These are

- center of the box $(b_x, b_y)$
- width of the box $(b_w)$
- height of the box $(b_h)$
- value matching to the object's class $(c)$

The other quantity that matters is $p_c$, which is the likelihood of an object that is present within the bounding box. Anchor boxes are predefined bounding boxes with a certain height and width, typically chosen based on the object sizes in training datasets to capture the scale and aspect ratio of specific object classes to detect. Only when the anchor box's center coordinates falls within a given cell is an object regarded to be in that cell. This parameter causes the height and width to be determined relative to the entire image size, while the center coordinates are calculated relative to the cell. The likelihood that a specific class is present in the cell during the one pass of forward propagation, is given by -

$$score = p_c \times c \tag{1}$$

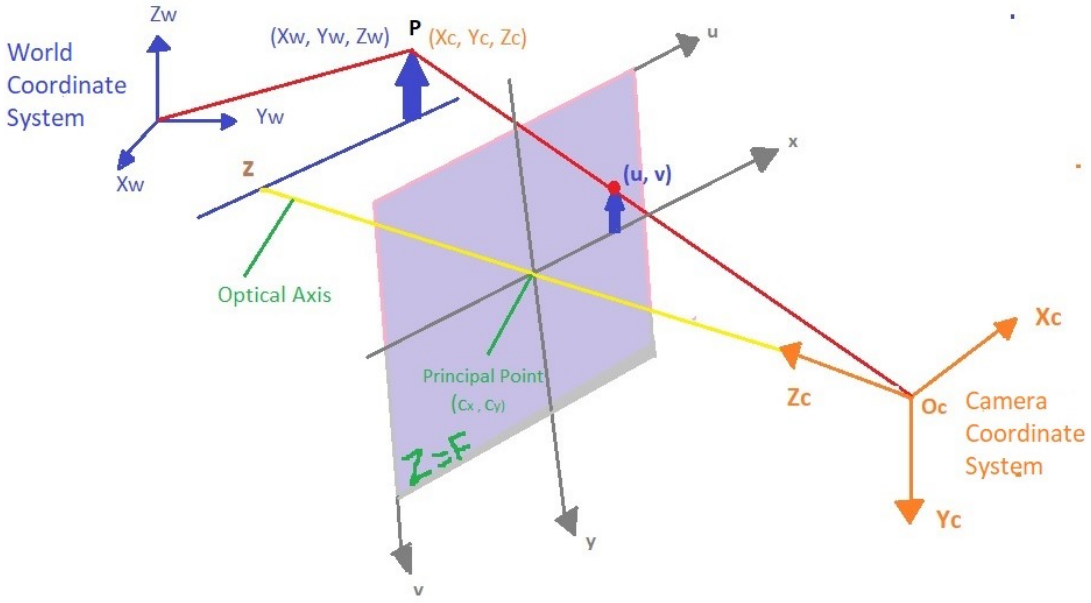


Fig. 1. The three coordinate systems, and the projection of the point P on each of the coordinate systems.

Non Maximum Suppression (Non-Max) is a technique that is used in a variety of tasks. It is a type of algorithm that selects one entity, in this case the bounding box, from a set of overlapping entities. To achieve the desired results, we can choose the selection criteria, which in most cases is a probability of overlap such as what is achieved by IoU.

Non-Max is performed on each of the bounding boxes, and it eliminates the bounding boxes that have overlap.[2] This determines the IoU value for each bounding box in relation to the one with the highest class probability before excluding those whose IoU value is lower than a certain threshold. This compared two bounding boxes that are enclosing the same object and removes the one with a lower probability of enclosing the same. The process is then repeated for the bounding box with the next highest class probability, and so on, until all of the bounding boxes have been found. The vector containing information about the bounding box of the respective class is then produced.

### 2.3 Three Dimensional Real Time Object Detection

Two-Dimensional Object Detection, implemented using YOLO, can accurately assess the class to which the object belongs, but cannot perceive the depth of the object, i.e. how far the object is from the camera. To assess the depth of the object, we use Stereo Vision, which perceives the depth of an object using two two-dimensional images.

Before we go on to perceive the depth of the object, the cameras in use must be calibrated, to remove any distortion. There are two types of distortion in a pinhole camera - radial distortion, and tangential distortion. There are three coordinate systems in play when an image is captured - the world coordinate system, the camera coordinate system and the image coordinate system. The World Coordinate System is the coordinate system attached to the room or the workplace where the image is taken. Let the coordinates of a point, P, in the world coordinate system are given by $(X_w, Y_w, Z_w)$. Let's assume that the camera is placed at some random location $(t_x, t_y, t_z)$ in the room, this is in relation to the world coordinates. The camera may also be rotated with regard to the world coordinate system and be pointed in any

direction. In order to capture rotation in 3D, yaw, pitch and roll are used. The rotation matrix $R$ and the three-element translation vector $t$ connect the world coordinates and the camera coordinates. In the camera coordinate system, the point P, which has coordinate values of $(X_w, Y_w, Z_w)$ in world coordinates, will have the coordinate $(X_c, Y_c, Z_c)$. The two coordinate values have the following relationship

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = R \begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix} + t \tag{2}$$

The Extrinsic Matrix, is obtained by appending the $3 \times 1$ translation vector as a column at the end of the rotation matrix, a $3 \times 3$ matrix.

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = \begin{bmatrix} R|t \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \tag{3}$$

where, the extrinsic matrix is given by

$$P = \begin{bmatrix} R|t \end{bmatrix} \tag{4}$$

Clearly, the Extrinsic Matrix has dimensions of $3 \times 4$. The image plane is placed at a distance $f$, which is the focal point from the optical centre of the camera. Using similar triangles, the projected image $(x, y)$ of the 3D point $(X_c, Y_c, Z_c)$ is given by

$$x = f\frac{X_c}{Z_c} \quad \text{and} \quad y = f\frac{Y_c}{Z_c} \tag{5}$$

Which can be written in matrix form

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} \tag{6}$$

The intrinsic matrix $K$, contains the intrinsic parameters of the camera. The matrix is

$$K = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{7}$$

Only the focal length is displayed in the simple matrix above. However, it's possible that the image sensor's pixels aren't square, in which case we have two distinct focal lengths, $f_x$ and $f_y$. The optical centre $(c_x, c_y)$ of the camera and the centre of the coordinate system for the image are not the same. Additionally, the camera sensor's $x$ and $y$ axes may have a slight skew, denoted by $\gamma$. Considering all the above, the camera matrix now has the following form

$$K = \begin{bmatrix} f_x & \gamma & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{8}$$

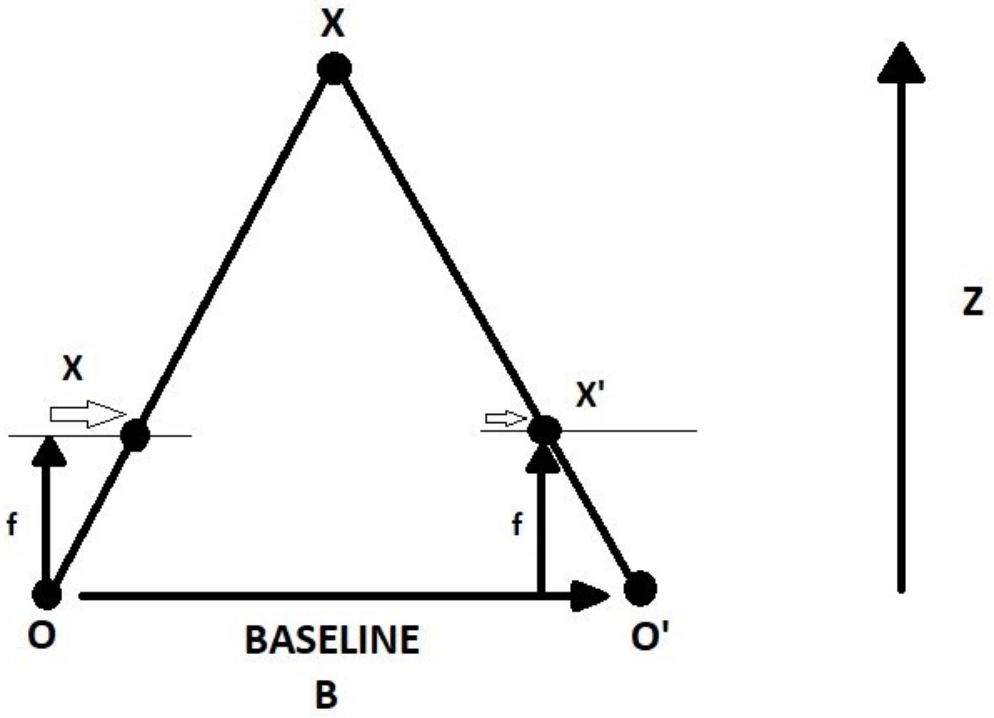


Fig. 2. Epipolar geometry of a depth map

The $x$ and $y$ pixel coordinates in the equation above are in relation to the image's centre. When working with images, the origin is in the image's upper-left corner.

Let the coordinates of the image be $(u, v)$, then

$$\begin{bmatrix} u' \\ v' \\ w' \end{bmatrix} = \begin{bmatrix} f_x & \gamma & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} \tag{9}$$

where,

$$u = \frac{u'}{w'} \quad \text{and} \quad v = \frac{v'}{w'} \tag{10}$$

The camera parameters are thus calculated using a set of known 3D points $(X_w, Y_w, Z_w)$ and their corresponding pixel location $(u, v)$ in the image during the calibration process. On calculating the camera parameters, the distortion parameters and the parameters are passed through OpenCV. For the calibration, we use a chess board, whose checkerboard pattern is distinct and easy to spot in its image. Because of sharp gradients in both directions, the corners of squares on the checkerboard are ideal for localization. Furthermore, these corners are linked by the fact that they are

at the intersection of checkerboard lines. All of these facts are used to precisely locate the corners of squares in a checkerboard pattern.

Once the cameras are calibrated, the 3D depth map can be constructed from the 2D images of the two cameras.The geometry is represented visually in Figure 2. Let the point in 3D space whose depth we are to determine be the point X, and the optical centres of the two cameras be $O$ and $O'$. The focal length of our cameras is length $f$, which we know because of the intrinsic values. The point X appears in each image at positions $x$ and $x'$ on the 2D image plane, respectively. The depth $Z$ of point X can be determined by comparing the similar triangles, written as $Ofx$ and $O'fx'$. This is accomplished using the following equation

$$\text{disparity} = x - x' = \frac{Bf}{Z} \tag{11}$$

The difference in distance between points x and x' and the distance Z of point X from the cameras' optical centres are inversely proportional (the image points of our object as they appear on the 2D plane). Using the concept of epipolar constraint, OpenCV is utilised to quickly and efficiently discover matched points in each of our photos. The depth of the object in question is then determined by calculating the difference between the matching locations. This method for 3D reconstruction is applied to every pixel in the stereo images by repeating this procedure, thus yielding the depth map.

## 3 RESULTS AND DISCUSSION

Firstly, the YOLO algorithm was tested using a laptop webcam. Pictures were taken of everyday objects, and the confidence of their classification were recorded. The confidence of a class is the probability that the object belongs to the certain class multiplied by 100, thus yielding a percentage. Firstly, a single object was placed before the camera, then two and so on, to check the variance in the confidence of the classes. To find how the confidence of detection varies with number of objects, first we consider detection of them individually, and then clustered together. In everyday life, rarely do we find objects isolated. Therefore, it is essential to make sure that the confidences do not vary heavily when the object is present with other objects in the vicinity.

Table 1. Confidence of classes for singular objects

| Class | Confidence |
|---|---|
| Bottle | 89.94 |
| Chair | 89.49 |
| Cup | 90.63 |

Table 2. Confidence of classes when objects are clustered together

| Class | Confidence |
|---|---|
| Bottle | 84.04 |
| Chair | 75.01 |
| Cup | 90.58 |

Table 2 represents the confidence when multiple objects are cluttered together in the same image, first 2 objects in the same image, then 5 and so on. What we notice from Figure 3 is that even though the objects are clustered together,


Fig. 3. Classification of objects clustered together

each object is classified accurately, with a confidence close to that of Table 1. For example, the two bottles on the right of Figure 3, we notice that a portion of two bottles and the chair are not visible, causing the confidence value to drop. On the other hand, the bottle on the left is clearly visible, and is classified with a confidence value similar to when a bottle is placed singularly. Table 2 considers outliers, i.e. cases when the object is not clearly visible to the camera as well, leading to the drop in confidence value for that respective class. Overall we can conclude that, in an ideal scenario, given each object is clearly visible, in a clutter as well, the algorithm detects the object with a high accuracy. But naturally, in a clutter, some objects will hide portions of the other, and thus we can conclude that in a real scenario, the confidence of an objects when together with other objects in an image decreases. In the above experiment, we have considered three objects, but the results are applicable for all class of objects.

For testing the 3D object detection system, are using a few known specifications of the cameras to estimate the Depth and Height of an object. Firstly, the two cameras are set up parallel to each other on a flat base to reduce the error caused by the uneven alignment of the lenses. The distance between the two cameras and the focal length of the cameras are fed to this module for calculation of the dimensions. Since two cameras are utilized, only those objects will

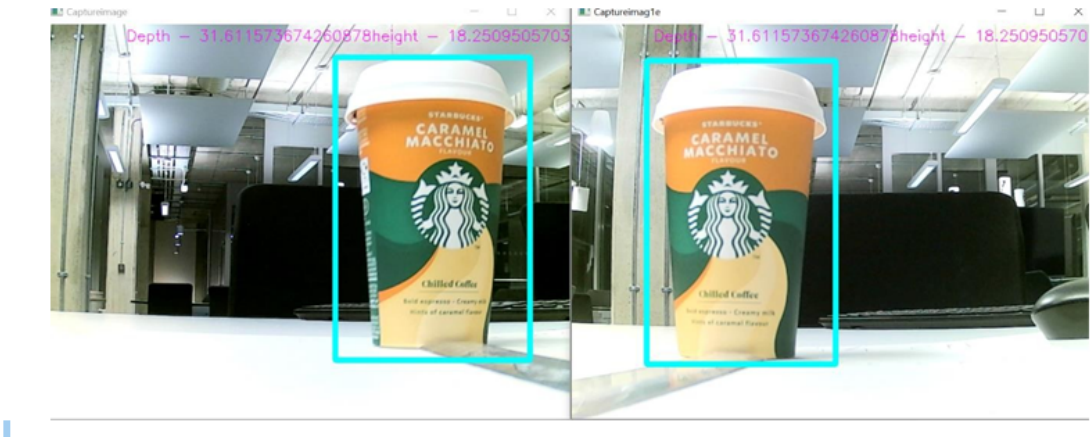


Fig. 4. Depth and Height of a cup

be identified that fall in the common view range of both cameras. The system predicts depth more accurately compared to the height, which can be seen in Table 3.

Table 3. Confidence of classes when objects are clustered together

| Object | Original Depth | Predicted Depth | Original Height | Predicted Height |
|---|---|---|---|---|
| Mouse | 30 | 28 | 5 | 15 |
| Cup | 30 | 31 | 14 | 17 |
| Bottle | 45 | 48 | 18 | 20 |

In the case of depth, the average calculated error is around 2 to 3 centimetres whereas, in the case of height it is 4 centimetres, but in the case of a small object, it is 10 centimetres. This could be because of the height of lenses from the ground. While calculating the height and depth, lens's height from the ground is ignored. Moreover, since two cameras are in use, keeping them in exact parallel is hard. Also, The known values like angle and focal length which are considered are mostly approximate values given in the handbook of the cameras. When the object is too small the depth is measured with less error than height. Similarly, when the number of objects increases, the accuracy of the object decreases, as shown in Table 2, which also applies for 3D object detection. Performing object detection using one camera instead of two cameras increases the system's accuracy, which might be because of mainly two reasons. Firstly, the object will only be detected when it falls in the frame of view of both cameras. Secondly, when two cameras are in use, the frame of view is reduced as shown Figure 5, in which objects C,D, E and F will not be detected as the images as the images fall in any one camera's frame of view, but not both. This is the horizontal aspect only. The same problem applies to the vertical aspect as well. Also, certain objects in the image where only a portion of the object is visible in the image, is not detected, as well as in certain rare scenarios, where a certain object is identified incorrectly. For example, in 1 out of 50 images, a portion of the back rest of a chair was identified as a TV monitor.

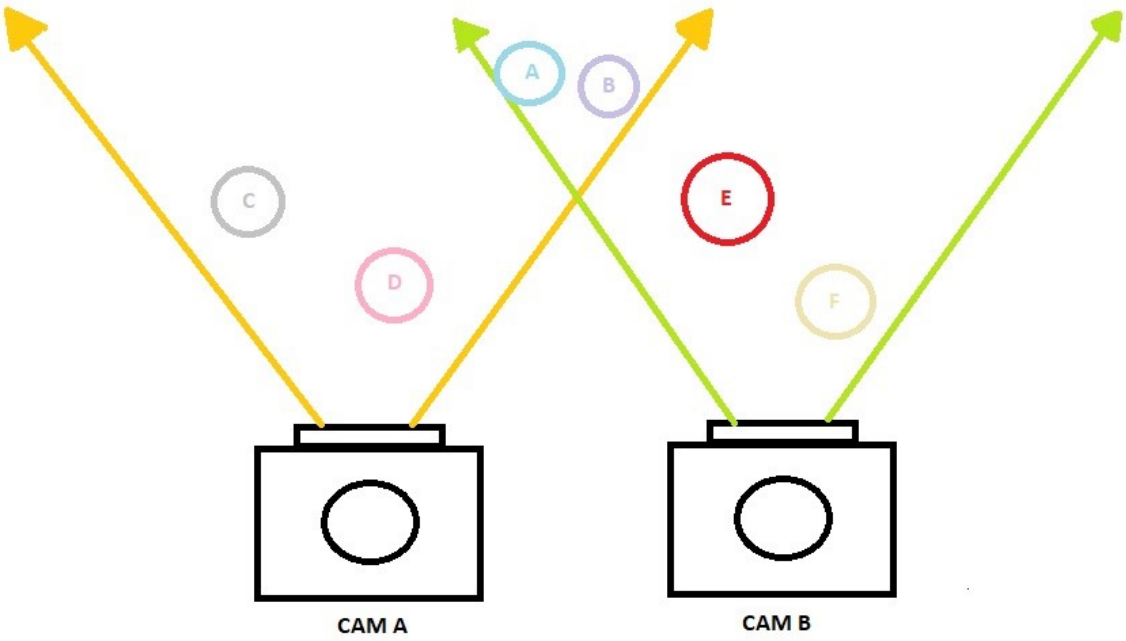


Fig. 5. Common frame of view between the two cameras

## 4 CONCLUSION AND FURTHER APPLICATIONS

Implementing the three dimensional real time object detection on a robotic arm, the system can pick up daily objects, a prototype that can aid the differently-abled in their daily activities. There are various other applications of the algorithm for rehabilitation purposes, like comparative analysis of human motion. Comparative analysis of human motion is necessary for application areas like automated rehabilitation and/or assessment of stroke, Spinal Cord Injury (SCI), Parkinson's Disease (PD) or patients with other physical impairments.[7] Object detection can also aid industrial machines and robots to identify and pick up objects with greater accuracy, an approach several organizations are currently implementing. To ensure safe and robust driving performance, autonomous vehicles rely on the perception of their surroundings. The three dimensional object detection can be implemented by the perception system, which employs object detection algorithms to accurately identify objects in the vehicle's vicinity, a step towards making the vehicle safer to deploy on the road. The work also has potential applications in sports, like football, helping Goal-Line technology and Video Assistant Referee (VAR) become more precise. This work has potential to impact various industries and fields.

## REFERENCES


[1] Introduction to YOLO Algorithm for Object Detection - Section.io. Retrieved January 31, 2023 from https://www.section.io/engineering-education/introduction-to-yolo-algorithm-for-object-detection/

[2] YOLO — You Only Look Once. A State of the Art Algorithm for… | by . Retrieved January 31, 2023 from https://towardsdatascience.com/yolo-you-only-look-once-3dbdbb608ec4

[3] YOLO Algorithm for Object Detection Explained [+Examples]. Retrieved January 31, 2023 from https://www.v7labs.com/blog/yolo-object-detection

[4] 3D Reconstruction with Stereo Images -Part 1: Camera Calibration . Retrieved January 31, 2023 from https://medium.com/@dc.aihub/3d-reconstruction-with-stereo-images-part-1-camera-calibration-d86f750a1ade

[5] Camera Calibration using OpenCV | LearnOpenCV. Retrieved January 31, 2023 from https://learnopencv.com/camera-calibration-using-opencv/

[6] Bappaditya Debnath, Mary O'Brien, Motonori Yamaguchi, and Ardhendu Behera. 2021. A review of computer vision-based approaches for physical rehabilitation and assessment. Multimedia Systems 28, 1 (2021), 209-239. DOI:https://doi.org/10.1007/s00530-021-00815-4

[7] Augmented Reality: Stereoscopic Vision for the Blind. Retrieved January 31, 2023 from https://www.seeingwithsound.com/binocular.htm

[8] Won-Kyung Song, Heyoung Lee, and Zeungnam Bien. 1999. KARES: Intelligent wheelchair-mounted robotic arm system using vision and force sensor. Robotics and Autonomous Systems 28, 1 (1999), 83-94. DOI:https://doi.org/10.1016/s0921-8890(99)00031-7

[9] Chen Zhihong, Zou Hebin, Wang Yanbo, Liang Binyan, and Liao Yu. 2017. A vision-based robotic grasping system using deep learning for garbage sorting. 2017 36th Chinese Control Conference (CCC) (2017). DOI:https://doi.org/10.23919/chicc.2017.8029147

[10] Ahmed Fawzy Elaraby, Ayman Hamdy, and Mohamed Rehan. 2018. A Kinect-Based 3D Object Detection and Recognition System with Enhanced Depth Estimation Algorithm. 2018 IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON) (2018). DOI:https://doi.org/10.1109/iemcon.2018.8615020

[11] Rumin Zhang, Yifeng Yang, Wenyi Wang, Liaoyuan Zeng, Jianwen Chen, and Sean McGrath. 2018. An Algorithm for Obstacle Detection based on YOLO and Light Filed Camera. 2018 12th International Conference on Sensing Technology (ICST) (2018). DOI:https://doi.org/10.1109/icsenst.2018.8603600

[12] H Lipson and M Shpitalni. 2007. Optimization-based reconstruction of a 3D object from a single freehand line drawing. ACM SIGGRAPH 2007 courses (2007). DOI:https://doi.org/10.1145/1281500.12815546

[13] Shuran Song and Jianxiong Xiao. 2016. Deep Sliding Shapes for Amodal 3D Object Detection in RGB-D Images. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016). DOI:https://doi.org/10.1109/cvpr.2016.94

[14] Guha Balakrishnan, Adrian Dalca, Amy Zhao, John Guttag, Fredo Durand, and William Freeman. 2019. Visual Deprojection: Probabilistic Recovery of Collapsed Dimensions. 2019 IEEE/CVF International Conference on Computer Vision (ICCV) (2019). DOI:https://doi.org/10.1109/iccv.2019.00026

[15] Nii Mante and James D. Weiland. 2018. Visually Impaired Users can Locate and Grasp Objects Under the Guidance of Computer Vision and Non-Visual Feedback. 2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC) (2018). DOI:https://doi.org/10.1109/embc.2018.8512918

[16] Kodai Takeda, Kazuyuki Ishihara, and Takushi Kawamorita. 2018. A Sense of Distance and Augmented Reality for Stereoscopic Vision. SAE Technical Paper Series (2018). DOI:https://doi.org/10.4271/2018-01-1036