Computer Vision Methods for Advanced Smart Wheelchairs Using DNN

(深層ニューラルネットワークを用いた知的車いすのためのコンピュータビジョン手法)

2019 年 9 月

埼玉大学大学院理工学研究科（博士後期課程）

理工学専攻 (主指導教員　久野 義徳)

SARWAR ALI
(サロワロ アリ)

I would like to dedicate this thesis to my loving parents.

Acknowledgments

I would like to thank my supervisor Professor Yoshinori Kuno, Saitama University, Japan, for his supervision, suggestions, trust, patience, and continuous support throughout the development of this thesis. I am sincerely grateful to him for his constant encouragement, and his stimulating ideas.

I am incredibly grateful to Professor Yoshinori Kobayashi, Assistant Professor Antony Lam and Hisato Fukuda for their guidance, comments, valuable advice, and feedback during evaluation of my thesis work. I am forever thankful to them for giving me their valuable time.

I am also thankful to the Computer Vision Lab members. With their support, my research was much more enjoyable.

My biggest gratitude goes to my family for their endless love and support and trust in me. Without them, I would have never reached this stage.

Finally, thank you all for your kind support and encouragement. So this thesis is especially dedicated to you all.

Abstract

The advancement of technology in research is increasing with several applications in robotics. Smart robotic wheelchair studies is one of them, which could provide user-friendly interfaces and/or autonomous functions that meet the needs of severely impaired users along with an aging society. Powered wheelchairs have been developed for people that lack muscle control (e.g., due to spinal cord injuries) and have difficulties in operating unpowered wheelchairs. However, many people have struggled in operating powered wheelchairs and in practice, are often accompanied by companions. For some wheelchair users, traveling by using wheelchairs may be a difficult task, especially when they are in dynamic environments where pedestrians move around in dense public areas such as at bus stops, airport terminals, shopping malls, and so forth. In addition, performing tasks such as boarding buses or climbing up a certain height is of great importance. Thus, to provide a better quality of life for individuals like the elderly, physically disabled, and mentally handicapped people, there are growing demands for advanced Smart Robotic wheelchairs that have independent mobility functions that can navigate and sense information from their environment and respond in useful ways without caregiver support.

A large majority of the robotic wheelchair research to date has focused on indoor areas. Outdoor navigation should also be considered among environments people frequent. There are many possible functionalities for an advanced Smart Wheelchair in outdoor environments like the detection of outdoor terrain to run steadily, static and dynamic obstacles to avoid them, steps to climb up/down, and so forth. This thesis is specifically focused on navigation in outdoor crowded

environments to move smoothly among people along with bus and bus door detections to get on/off buses for transportation while considering practical issues such as sensors and the computational cost of the robotic wheelchair with real-time autonomous operations. To develop such wheelchairs, Computer Vision techniques are essential for detection and further analysis and also our system should run in real-time. Thus the system should have (1) High detection speed, which can be obtained by using simple algorithms using less computations and it should use a notebook laptop for portability since wheelchairs have a limited power source, (2) High accuracy, which we achieve by using a Neural Network, and (3) High precision measurement for detecting exact locations of necessary objects, relative to the wheelchair.

To achieve these goals, this thesis first proposes a smart robotic wheelchair system that is able to detect pedestrians and also control the wheelchair movements to avoid pedestrians smoothly including user comfort in mind. This thesis presents a method for our Smart Wheelchair to maneuver around individual and multiple pedestrians by detecting and analyzing their interactions and predicted intensions with the wheelchair. Our Smart Wheelchair can obtain head and body orientations of pedestrians by using TensorFlow based OpenPose. Using a single camera, we infer the walking directions or next movements of pedestrians by combining face pose and body posture estimation with our collision avoidance strategy in real-time. Experimental results show our approach is promising. We collected data from a train station and also from our campus to evaluate our method. Moreover, we determine the relative distance between pedestrians and the wheelchair by detecting how much of the image frame the pedestrians occupy (we call this "coverage"), to maintain a safe distance from pedestrians. If the distance between a given pedestrian and wheelchair is nearly 1m (which indicates 60% coverage of the pedestrian in the camera image), the wheelchair stops and processes the next frame and gets directions until it sees a clear path. For an added layer of safety for the user, we also use a LiDAR sensor for detection of any obstacles out of the camera field of view to avoid collisions in advance. The final system results in autonomous navigation that generates wheelchair movements that are safe and comfortable to the wheelchair user and other people in real-time multi-person scenarios.

In addition, we propose a bus boarding wheelchair system that can precisely detect bus and bus doors (open-door and close-door) using Convolutional Neural Networks (CNN) based image recognition. This is an extension of our ongoing work on a bus boarding wheelchair system in terms of the camera processing, vision component. For that, the YOLO Dark-net based object detection approach is employed. The system needs to operate in real-time for proper detection despite our wheelchair platform having limited power. Therefore, we modified one of the YOLO versions called Tiny-YOLO to run at a fair amount of speed around 10FPS on our notebook PC. For accurate detection of buses with open and closed bus doors, we trained our image recognition system with 1,800 different images for each of these classes. The overall precision of the system was measured using the Intersection of Union (IoU) method and we achieved an overall 70% average result in detection. Once our system can detect a given bus with an open door using our fast and modified Tiny-YOLO, we then apply a Hough line transform algorithm to get accurate and precise localization of the open door lines. To evaluate the performance of our proposed method, we also compare the accuracy of our modified Tiny-YOLO and our proposed combined detection method with the original ground truth. Moreover, we achieved an average 90% IoU, which was a significant improvement over the modified Tiny-YOLO. This information is indispensable for our bus-boarding robotic wheelchair to board buses. In real experiments with a bus, we also see the effectiveness in detecting the buses and bus doors. Consequently, we propose a Smart Wheelchair that can detect buses and precisely recognize bus doors, whether they are opened or closed for automated boarding before receiving any bus boarding commands.

Finally, we demonstrated successful operation of the proposed smart robotic wheelchair system using our method, which grants freedom of movement through pedestrian in urban environments.

Contents

List of Figures

List of Tables

Chapter 1 Introduction

1.1 Motivation

In today's world, the elderly and disabled are increasing day by day. According to [1], 8.5% of the world's population are aged over 65 and not all of them live a healthy life. Moreover, about 15% of the people in the world are disabled [2]. So, there are growing demands for wheelchairs that can support them. Powered wheelchairs have been developed for people that lack muscle control (e.g., spinal cord injury) and have difficulties in operating unpowered wheelchairs. Most people with cognitive/motor/sensory impairments, whether due to disability or disease or aging, rely on power wheelchairs for their mobility needs. However, clinical studies show a significant number of such people have found difficulties in operating their powered wheelchairs [18]. Among 200 practicing clinicians in the U.S. 40% of their patients find it hard or impossible to control powered wheelchairs [6].

Nowadays, there are a vast number of technologies, which improve the quality of life for many by providing independent mobility for individuals like the elderly, physically disabled, and mentally handicapped [3, 4, 5, 16, 17]. For some wheelchair users, traveling by using wheelchairs may be a difficult task, especially when they are in dynamic environments where pedestrians move around in dense public areas such as airport terminals, shopping malls, train stations, and so forth (Figure 1.1). Researchers have come up with "Smart Robotic Wheelchairs" that have user-friendly interfaces and autonomous functions that can navigate and sense information from their environment and respond in useful ways [19]. Therefore, Smart Wheelchairs should have high maneuverability and navigational intelligence with autonomy, which will provide safety and ease of operation to both users and pedestrians along with avoiding obstacles and maneuvering through doorways or any narrow spaces, reducing the workloads of caregivers (Figure 1.2). Most recent Smart Wheelchairs are used in indoor environments but Smart Wheelchairs should provide autonomous functions in outdoor areas such as roads and other places to move around too.

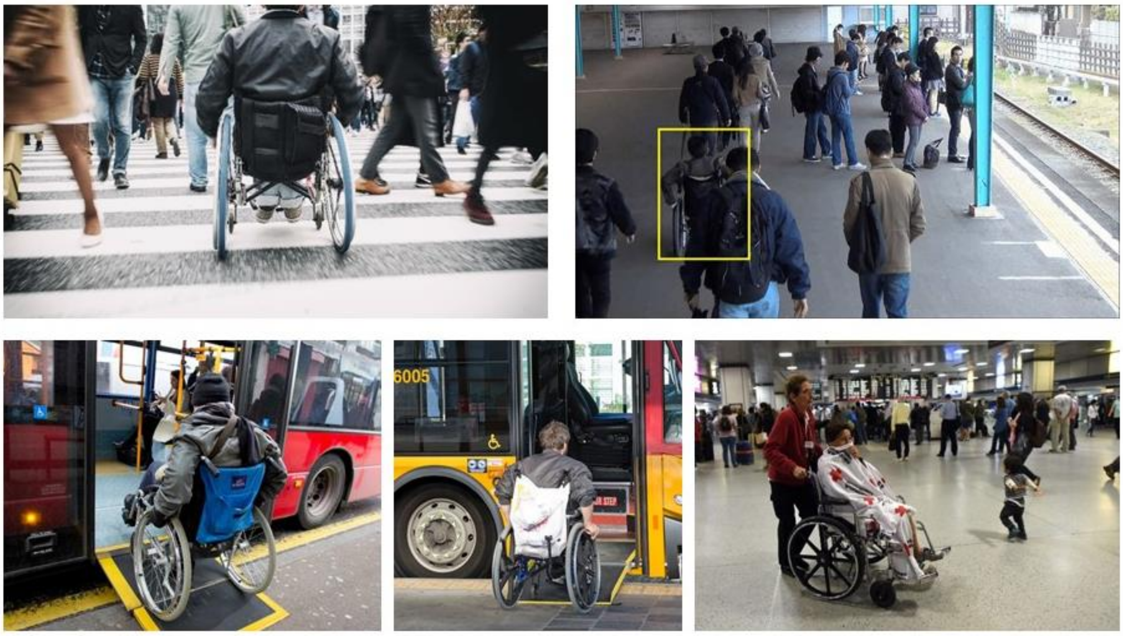


Figure 1.1– Wheelchairs in outdoor environments.

Figure 1.2– To advance Smart Wheelchairs.

There are many possible functionalities for an advanced Smart Wheelchair in outdoor environments like detection of outdoor terrain to run steadily, avoid static and dynamic obstacles, steps to climb up/down, and so forth. The method in [20] investigated the primary concerns in developing outdoor safe navigation for Smart Wheelchairs by detecting any given terrain's smoothness and recognizing categories of the terrain for a specified given path. One of the important functions of autonomous wheelchairs is to detect and avoid pedestrians in outdoor crowded environments to move smoothly among people.

Researchers have developed many ways to detect people and avoid them for wheelchairs. For example, the approaches in [7, 8], detect pedestrians as static obstacles but only in indoor environments and a 2D grid safety map was given for navigation. Other approaches have assumed simple independent motions for pedestrians (e.g. constant velocity) [9] and avoid them using simultaneous localization and mapping (SLAM). In actual human occupied environments, pedestrian motions can exhibit a lot of variation. One of the main drawbacks of these methods is that their Smart Wheelchairs are not familiar with pedestrian behaviors and movements in real-world environments. Pedestrian behaviors in the real-world should be taken into account. One observation we have made is that when a wheelchair encounters pedestrians that are aware of it, the pedestrians try to make way for the wheelchair. Hence, when pedestrians are aware of the wheelchair, the wheelchair can generally maintain its current trajectory. Therefore consideration of how pedestrians interact with the wheelchair is essential to determining how the wheelchair should plan its path for navigating crowds.

In our previous works [10, 11, 12], we considered the awareness levels of pedestrians by detecting face/head orientations towards the wheelchair and devised a Smart Wheelchair navigation strategy where head detection and the orientations of pedestrians was used for collision avoidance in indoor environments, using both RGBD and laser sensors. The experimental results in those studies showed that the wheelchair could detect and avoid a single pedestrian by observing his/her awareness of the wheelchair. However, there are also situations in real-world environments where some pedestrians might not be aware of the wheelchair even though they are facing towards wheelchair. Therefore, to drive smoothly and successfully in dynamic environments among multiple pedestrians, using only awareness from face detection and orientations is insufficient for obtaining an acceptable amount of information for human avoidance. There are two ways to obtain information on the pedestrians' movements. The first way is through interactions that indicate pedestrian awareness of the wheelchair and another is through pedestrian intensions, which can be obtained from body orientations predicting the next movements of the pedestrian. When people walk in any given environment, the personal spaces (PS) proxemics concept [13] needs to be considered. The PS proxemics concept indicates that a certain amount of boundary around people exists and people use this boundary space to protect themselves from collisions. Thus as long as people are aware of other pedestrians (or wheelchairs) in the environment, they will make an attempt at avoiding collisions by reorienting their bodies and moving along a new direction. We are interested in exploring such important circumstances, where we consider pedestrian awareness of the wheelchair along with predicting their intended future movements so that the wheelchair maneuvers smoothly and safely in an effective manner in crowded real-world environments, where pedestrians and the wheelchair can collaboratively pass each other.

Additionally, in this thesis we investigate navigation in outdoor environments with bus and bus door detections to get on/off the buses for transportation while considering practical issues such as sensors and the computational cost of the robotic wheelchair with real-time autonomous operations. Furthermore, moving in an outdoor environment, wheelchair users sometimes also need to travel long distances and so might need an easy mode of transportation like riding buses or trains. If we consider this type of situation, there are still some difficult barriers for Smart Wheelchairs to overcome for stable autonomous operation. For autonomous operations in such environments where the wheelchair would have to climb up or down to certain heights, LiDAR processing is essential for accurate measurement of heights and distances. In bus boarding, the wheelchair system needs precise detection of the bus door's distance and the height of the bus doorstep for boarding on the bus.

In our previous work [23], a single laser bidirectional sensing based approach to step detection and step height measurement was used by our Smart Wheelchair to detect bus doorsteps. Our wheelchair provides for autonomous movement in outdoor terrains and can climb onto steps like stairways or at bus doorways without any need for additional support from lifts/ramps.

Nevertheless, using only LiDAR processing has some drawbacks. Before measurement of doorstep heights and distances, our wheelchair should be able to perform real-time operations like detecting bus doors and door positions accurately and precisely. Performing such real-time detection using only LiDAR would be difficult. Therefore, Smart Wheelchairs require a sensing system that can make use of cameras as well. This is because they could provide fast, precise, and accurate detection in conjunction with LiDAR processing for further reliability support. One of the main goals in this thesis is to focus on the vision part, involving camera processing for the bus boarding aspect of our Smart Wheelchair. In the camera processing part, the Smart Wheelchair at first needs to detect the presence of buses at bus stops. After detection of the bus, it must identify the bus doors and which door is open. If the wheelchair recognizes the open door accurately and precisely with approximate estimates of the bus door width, then further detection of appropriate and exact locations of that door with respect to itself can be performed by LiDAR processing for accurate boarding of the bus.

1.2 Objectives

In this thesis we aim to acquire two of the most important functions for our advanced Smart Wheelchair.

1. Pedestrian Avoidance: The Smart Wheelchair should work in crowded environments to move smoothly among many people.
2. Bus and Bus Door Detection: The Smart Wheelchair should also have the ability to get on/off the bus for transportation (we focus on the Computer Vision part our wheelchair project in regards to bus boarding).

To develop such a wheelchair, Computer Vision techniques are essential for detection and further analysis. In addition, our system needs to operate in real-time. Therefore, the system should have (1) high detection speed which can be achieved by using a simple algorithm for less computations and using a notebook laptop for portability since Smart Wheelchairs have a limited and low power source, (2) high accuracy which can achieved by using a deep neural network, and (3) high precision measurement of exact locations of detected objects with respect to the wheelchair.

For pedestrian avoidance, we propose a method to enhance the movements of Smart Wheelchairs for severely impaired users by analyzing and detecting the intensions and interactions of individual and multiple pedestrians. In our experiments, we employ a system that is mostly based on one RGB camera for frontal observations. We use a Tensorflow based OpenPose [14] system to detect and obtain human skeleton data of the full body and different body parts, and estimate any given pedestrian's body and face orientations. From this, we create a model for estimating the pedestrians' awareness and next movements. Our study consists of three steps for analyzing both individual and multiple moving pedestrians. In the first step, we determine the head orientations to see if any given pedestrian is aware of the wheelchair or not. In the second step, we determine the body orientations for estimating the intended future movement directions of pedestrians. Finally, we combine these two orientations (head and body) to determine a strategy for which direction our Smart Wheelchair should move in to avoid collisions with pedestrians in real-time. Moreover, we determine the relative distance between pedestrians and the wheelchair to check if the pedestrians are too close to the wheelchair, to maintain a safe distance away from pedestrians. During its maneuvers, for added safety, we also use a laser sensor that scans for surrounding objects that might collide with the wheelchair and generate steering commands for avoiding them by using our previous method described in [11].

For Bus and Bus Door detection, we propose a bus boarding wheelchair system that can get onto a bus using Convolutional Neural Network (CNN) based image recognition for reliable and precise localization of bus doors. This is a work on a bus boarding wheelchair system in terms of the camera processing, vision component. The visual processing work presented here is used in a smart six wheeled Bus-boarding Mobility Robot (BMR) wheelchair we developed in collaboration with Toyota Motor Corporation and the University of Tokyo [29]. Specifically, we use deep learning for image recognition to identify buses, open doors, and closed doors. Before boarding a given bus, for real-time detection operations, we modified the Tiny-YOLO model [30] to run faster than the original Tiny-YOLO in CPU mode. Once our system can detect a given bus with an open door using our fast and modified Tiny-YOLO, we then apply a Hough line transform algorithm to get accurate and precise localization of the door line. In the end, after determining the bus door's position, we feed the information to our BMR wheelchair for LiDAR processing for fine-tuned estimates of bus doorway dimensions to complete the bus boarding process.

1.3 Research Contributions

This research resulted in an advanced Smart Wheelchair system with people avoidance features for maneuvering through crowed environments and bus door detection features for bus boarding. The system works by letting the user just sit on the wheelchair around bus station. The detection of people and movement tasks of the wheelchair are taken care of by the vision camera, notebook laptop and built-in motorized system, reducing the workload of the user. Research contributions include the following:

- An experimental paradigm for designing a detection system to detect people by only a vision system. (Chapter 4)
- An experimental paradigm for studying how robotic wheelchairs can detect people by detection system. (Chapter 4)
- An experimental paradigm for studying how to design a pedestrian avoidance system. (Chapter 4)
- An experimental paradigm for studying how to measure the distance between the wheelchair and pedestrians. (Chapter 4)
- An experimental paradigm for studying how a vision system uses classifications of different objects and this classification approach could be used. (Chapter 5)
- An experimental paradigm for designing a new modified convolutional neural network for detecting Buses and Bus Doors in bus stations for boarding onto a given bus. (Chapter 5)
- Finally, an experimental paradigm for describing the operation of the autonomous wheelchairs to find a bus and its bus door with precise measurement of the door for boarding. (Chapter 5)

1.4 Organization of Sections

Chapter 2 - Background

Chapter 2 provides the background. This part concerns current state-of-the art techniques in Smart Wheelchairs. We provide an overview and discussion of different functions needed for advanced Smart Wheelchair.

Chapter 3 - Detection and CNN Networks

In chapter 3, we explain our selected CNN networks among different types of CNN networks and their detections in terms of speed and accuracy. At the end of the chapter, we specifically focus on two types of objectives, one is on human pose and the other is on object detection techniques and methods.

Chapter 4 - Pedestrian Avoidance

In this chapter, our wheelchair maneuverability approach for pedestrian avoidance is discussed. People detection techniques and methods using their interactions and intensions with the wheelchair are also explained in this chapter.

Chapter 5 - Bus and Bus Door Detections

This chapter describes CNN based detection of a bus boarding Smart Wheelchair that can detect buses and bus doors precisely using a vision system. Our proposed method supports the Smart Wheelchair for precisely localizing the bus door, is also explained in this chapter.

Chapter 6 - Conclusions and Future Work

Finally, we conclude the thesis with a summary of our Computer Vision method using a deep neural network framework. This framework can be adopted in Smart Wheelchairs to make it more convenient to users, followed by a discussion on future works and feasible applications.

Chapter 2 Background

The overall goal of this research is to create a smart robotic wheelchair that moves in a manner that is safe and socially acceptable to the humans and also detects buses and bus doors for boarding. As such, this thesis draws on work from many fields, including Smart Wheelchairs, human-robot interaction, and human information retrieval.

We review the concepts and showcase the latest human-computer interface hardware, and Computer Vision innovations made in recent years. These tools grant people with disabilities not only mobility but also the necessary help and support to handle activities in daily life. We hope that the information gathered in this study will enhance awareness of the status of modern Smart Wheelchair technology and eventually increase the functional mobility and productivity of people who use powered wheelchairs (PWs). The rest of this chapter is organized to describe robotic wheelchairs with their hardware, and object detection functionalities.

2.1 Smart Wheelchair Review

The different types of wheelchairs can be broadly separated into manual wheelchairs and automatic wheelchairs (Figure 2.1). Manual wheelchairs need to be propelled or pushed by the user or a caregiver. When it comes to manual wheelchairs, there are both non-motorized and motorized models (powered or electric wheelchairs) (Figure 2.1). As the name suggests, motorized wheelchairs have a motor and rechargeable battery and are moved using a hand control on the armrest, requiring only minimal effort from the user. Automatic wheelchairs, provide the highest level of independence and convenience. They have a lever on the armrest to control the speed and direction, the inclusion of a motor makes navigating slopes easy, while the small turning circle allows use indoors and outdoors. Among automatic wheelchairs, there are two types, one is the semi-autonomous type and the other is the autonomous type (Figure 2.1). The autonomous type lets the wheelchair move automatically in known environments to a final destination that is pre-selected by the user. The semi-autonomous type on the other hand lies in between both controllers in a sense that the computer performs short-term route planning and reactively avoiding obstacles, and the users' only interrupt it when they wish to deviate from the plan. In such automatic wheelchairs, some researchers adds essential features to fully develop them into Smart Wheelchairs.

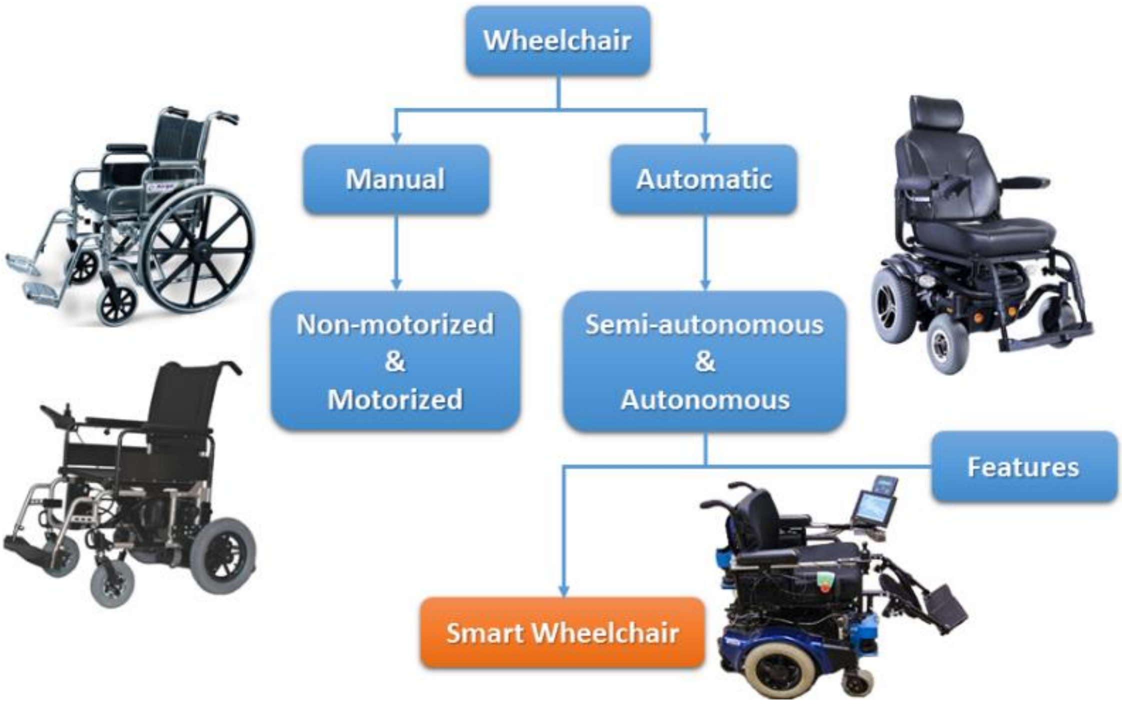


Figure 2.1– Wheelchair Categories

In recent years, numerous methods have been introduced for developing smart robotic wheel chairs to accommodate the disabled. The trends in development can be broadly classed into three main areas: 1) Improvements to the assistive technology mechanics, 2) Improvements to the user-machine physical interface, 3) Improvements to shared control between the user and the machine. One of the key aspects of smart robotic wheelchairs is to provide independent mobility for users with severe impairments who cannot control the wheelchair by means of a standard joystick. Generally, the developed Smart Wheelchair platform highly depends on the user's profile (i.e. abilities and disabilities), and there is no single solution that is suitable for all users. Based on knowledge about what type of input medium that the users are able to run, we can gain insight about the appropriate level of assistance.

2.2 Smart Wheelchair Hardware

The basic components of a Smart Wheelchair are shown in Figure 2.2 and described as follows:

- Control Unit: This controls the wheelchair movements (i.e. steering commands).
- Wheels: Which have installed motor(s) in them. The drive system, which may be front-wheel, rear-wheel, center-wheel, or all-wheel-drive.
- LiDAR: This is used for robust localization and map building to allow for high-level decision making in environments.
- Human Interface: The Smart Wheelchair users have a choice between hand joysticks, sip-n-puff controls, chin joysticks, or head joysticks for manually controlling the wheelchair. Higher Level Input for setting the goal, the path, and restrictions such as the speed of driving are also provided. Moreover their might be a touchscreen to further improve the ease-of-use of the Smart Wheelchair.

- Vision: We propose a realistic system (no lasers, just cameras) for navigation in outdoor environments with occlusion detection and avoidance. The goal is find the fastest and safest path that will prevent our wheelchair from getting stuck and injuring the user.

Apart from the basic components of the Smart Wheelchair, our wheelchair also has common components like batteries and a seating system, which has also been upgraded. The first powered wheelchairs derived their power from two 24V wet cell batteries. But these batteries have to be removed from the wheelchair during travel on airplanes. They were eventually replaced by dry cell batteries. Seating systems are typically upgraded to include cushions that use foam, or air to prevent pressure sores. Backrests are typically padded with foam and can be motorized to tilt and recline. Lateral supports keep the user from tilting side-to-side. Footrests are either removable or motorized to accommodate a more comfortable reclining position.

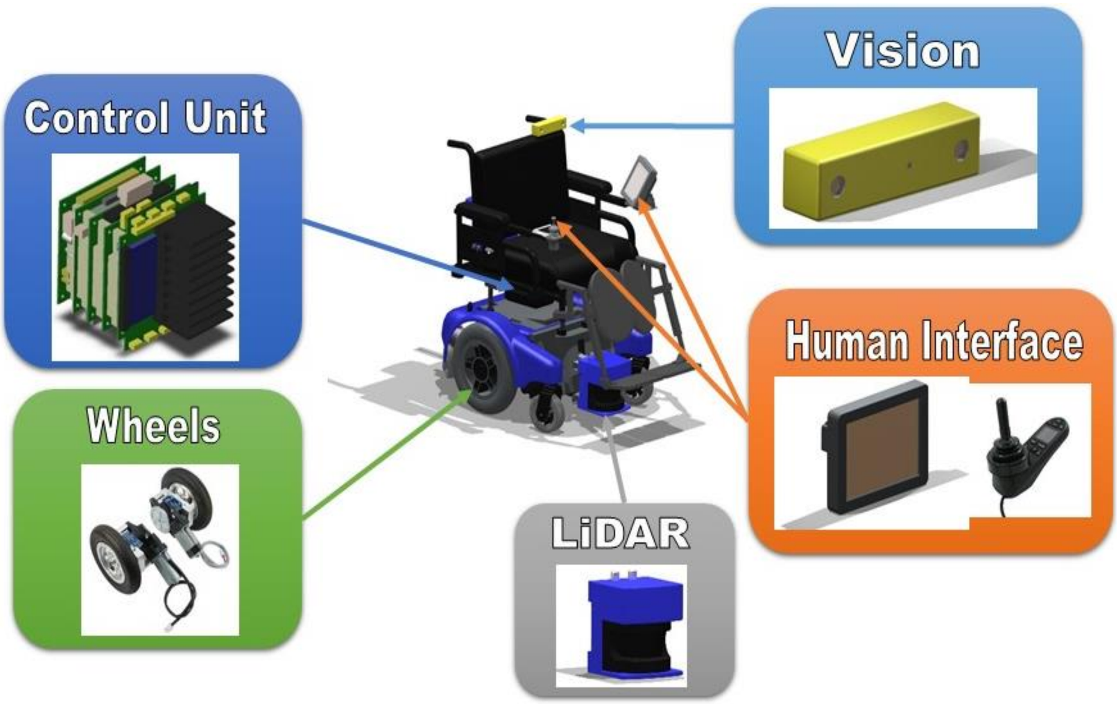


Figure 2.2– Smart Wheelchair Hardware

To avoid obstacles, Smart Wheelchairs need sensors to acquire surrounding object or structure information around them. The ultrasonic acoustic range finder (i.e., sonar) is the most frequently used sensor by Smart Wheelchairs. Sonar sensors are very accurate for objects at an angle or head-on if there are no other sound frequencies around. However, sonar sensors are vulnerable to "cross-talk," which happens when the signal generated by one sensor produces an echo to a different sensor. We can find more accurate obstacle and drop-off detection with laser range finders (LRFs), which provide a 270° field of view. There are many experiments using LRF sensors. However, these LRFs are expensive, are large, and consume a large amount of power, which will make the Smart Wheelchair very difficult for production at a cheap rate for purchasing by the average user.

A complete system with intelligent mobility aids to impaired users needs sensors that are accurate, small, lightweight, cheap and impervious to environmental conditions (e.g., lighting, precipitation, temperature) with also have low power requirements.

Computer Vision is perhaps the most promising sensor technology. Cameras are much smaller than LRFs and, thus, much easier to mount in multiple locations on a wheelchair. Cameras can also provide much greater field of view coverage. The cost of the Smart Wheelchair with Computer Vision hardware will be significantly cheap by use of simple web cameras and Computer Vision software. These software continue to improve and can be updated, which makes successful implementation of a Smart Wheelchair based on Computer Vision increasingly likely.

2.3 Detection Fundamentals

There are many models, approaches, and techniques used in Computer Vision object detectors. Often times, various trade-offs for speed and accuracy between these factors need to be considered for real-life applications. Below, we detail some of these factors:

- Feature extractors: Takes an input dataset, propagates each example through a Convolutional Neural Network (in our case), and returns an array of dense feature vectors. Output strides for the extractor controls how the filter convolves around the input image pixel matrix.
- Input image resolutions: It describes the image's level of detail. Higher resolution mean more image detail.
- Matching strategy and IoU threshold: This allows the network to predict high scores for multiple overlapping default boxes rather than requiring it to pick only the ones with maximum overlap. This is used to show how predictions are excluded in calculating loss.
- Bounding box regression: The bounding box regressor, instead of predicting the bounding box location on the image, predicts the offset of the ground-truth/predicted bounding box to the anchor box.
- Training dataset: The training dataset or learning set is the material through which the computer learns how to process information. Data augmentation may also be used to get more data, by making minor alterations to the existing dataset. Examples of minor alterations include changes such as flips, translations, and rotations.
- Prediction function: It takes an arbitrary set of measurements as a data-series and returns a prediction value generated by the trained DNN classifier.
- Feature map layer for object detection: The Convolution layer uses a filter matrix over the array of image pixels and performs a convolution operation to obtain a convolved feature map layer.
- Localization loss function: The localization loss is the loss between the predicted bounding box correction and the true values. The localization loss for the bounding box offsets the prediction and the classification loss for conditional class probabilities.

- Training configurations: Includes batch size, input image resize, learning rate, and learning rate decay.

Detections and appropriate CNN network selections are discussed in the next chapter.

2.4 Summary

This chapter presented different types of wheelchairs. It focused on two main things: the hardware configuration of Smart Wheelchairs and our required properties and keypoints for detection of objects to avoid them. We also reviewed other types of Smart Wheelchairs and their sensor systems to detect obstacles.

Chapter 3 Detection and CNN Networks

3.1 Overview

CNNs are a class of deep artificial neural networks that are used primarily to analyze images. They can classify images, cluster them by similarity, and perform object recognition within scenes. For object recognition, these Neural Network algorithms can identify faces of individuals, different animals, households, vehicles, and many other aspects of visual data. Therefore, nowadays, convolutional nets in image recognition use a lot with the help of deep learning. Thus Computer Vision applications like self-driving cars, robotics, drones, security, medical diagnoses, and treatments for the visually impaired, CNNs are a major component.

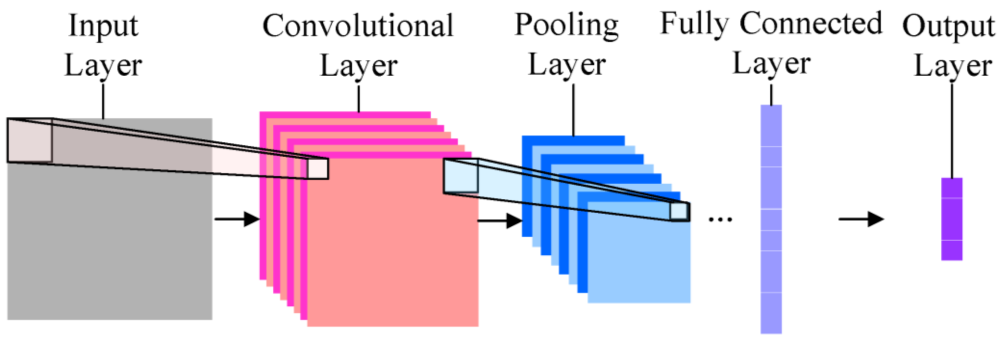


Figure 3.1– A convolutional neural network [34].

Deep Neural Networks differ from classical single-hidden-layer neural networks in that they consist of multiple layers where data is processed in a multistep process for pattern recognition [34]. One example of layering of neural network is depicted in Figure 3.1

In this chapter, we present details on our CNN framework for detection of people and distinct objects. We also discuss our how we adapt existing a conventional deep neural networks to our needs.

3.2 Pose Detection (OpenPose)

Human pose detection is necessary for our wheelchair to detect of the actions and motions of pedestrians. Furthermore, different people may show different poses in executing the same action. All these factors will result in large intra-class appearance and pose variations, which confuse many existing action recognition algorithms. Human actions, particularly those involving whole-body and limb (e.g., arms and legs) movements, and interactions with their environment, which contain rich information about them. Therefore, if we know the pose of a person, we can further train machine learning models to make autonomous systems for our required applications.

In this section, we investigate several open source deep learning models and code for pose estimation to find the most appropriate pose estimation method.

1. **DensePose**

   It is based on DenseReg. (Dense Shape Regression) [33]. DensePose adopts the architecture of Mask-RCNN with the Feature Pyramid Network ((FPN) features, and ROI-Align pooling. Additionally, they introduce a fully-convolutional network on top of ROI-pooling. DensePose does not give limb (e.g., arms and legs) movements, it only provides outputs in terms of the whole body shape.

The goal of the model is to determine the surface location of each pixel, and its corresponding 2D parameterization of the part it belongs to (Figure 3.2).

Figure 3.2– DensePose Detections [33].

2. AlphaPose

AlphaPose [35] is a multi-person pose estimator, and claims to be the first open source system (Figure 3.2). AlphaPose performs both pose estimation and pose tracking on images, videos, or lists of images. AlphaPose uses a regional multi-person pose estimation (RMPE) framework to facilitate pose estimation in the presence of inaccurate human bounding boxes. AlphaPose has some limitations like its processing speed is very slow compared to other detectors and uses lots of GPU power.

The limitation of this detection method is that it cannot detect occulated very well. And the missing person detector also cause missing detection of human pose.

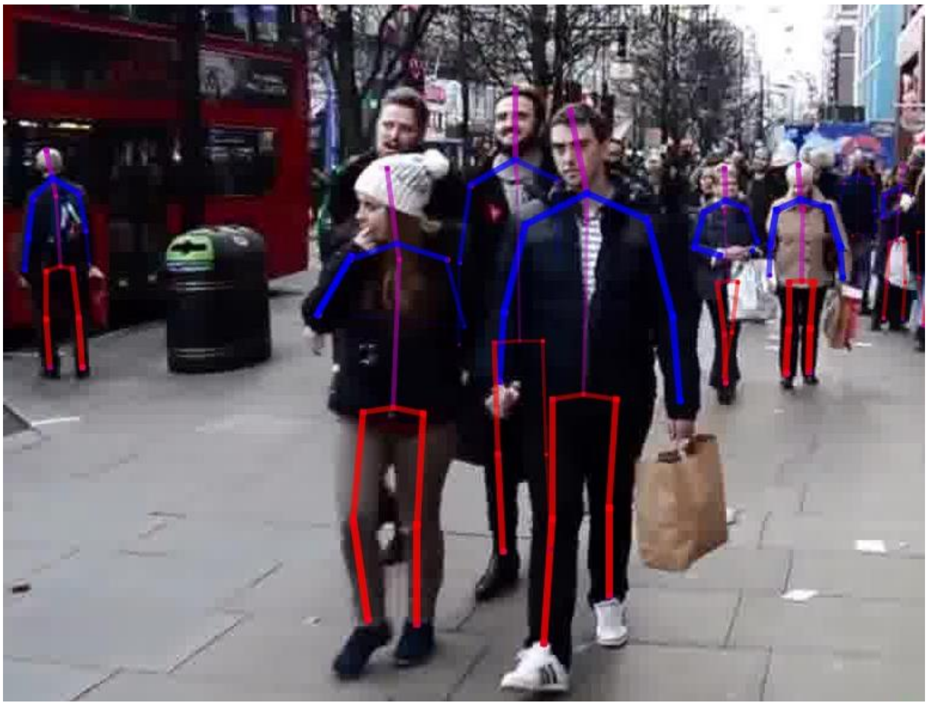


Figure 3.3– AlphaPose Detections [35].

Figure 3.4– OpenPose Detections [14].

3. OpenPose

OpenPose [14] in Figure 3.4 is a real-time multi-person key-point detection library for body, face, and hands estimation by the CMU Perceptual Computing Lab. It also has TensorFlow based implementation of the Human Body Pose which is much faster than other detector. This approach uses Part Affinity Fields (PAFs) [36], to learn to associate body parts with individuals in the image.

The OpenPose algorithm pipeline is shown in Figure 3.5. The process starts form grabbing an image form the camera and processing it in a neural network. The output of the network returns a tensor consisting of 57 matrices. In the next process, a heatmap and PAFs are used to extract the location of the body parts containing 18 matrices and couples the parts into pairs covering 38 matrices. After receiving body parts, non-maximum suppression (NMS) applied to get peak values. Lastly, a bipartite graph is used to get the right connections between pairs with the help of line integral weights and using merging operations, we get the output detected image. Here, the assignment algorithm is used to get the highest score.

For our pedestrian avoidance system we used the OpenPose detector as our base system. Details of our methods are described in Sec. 4.3.

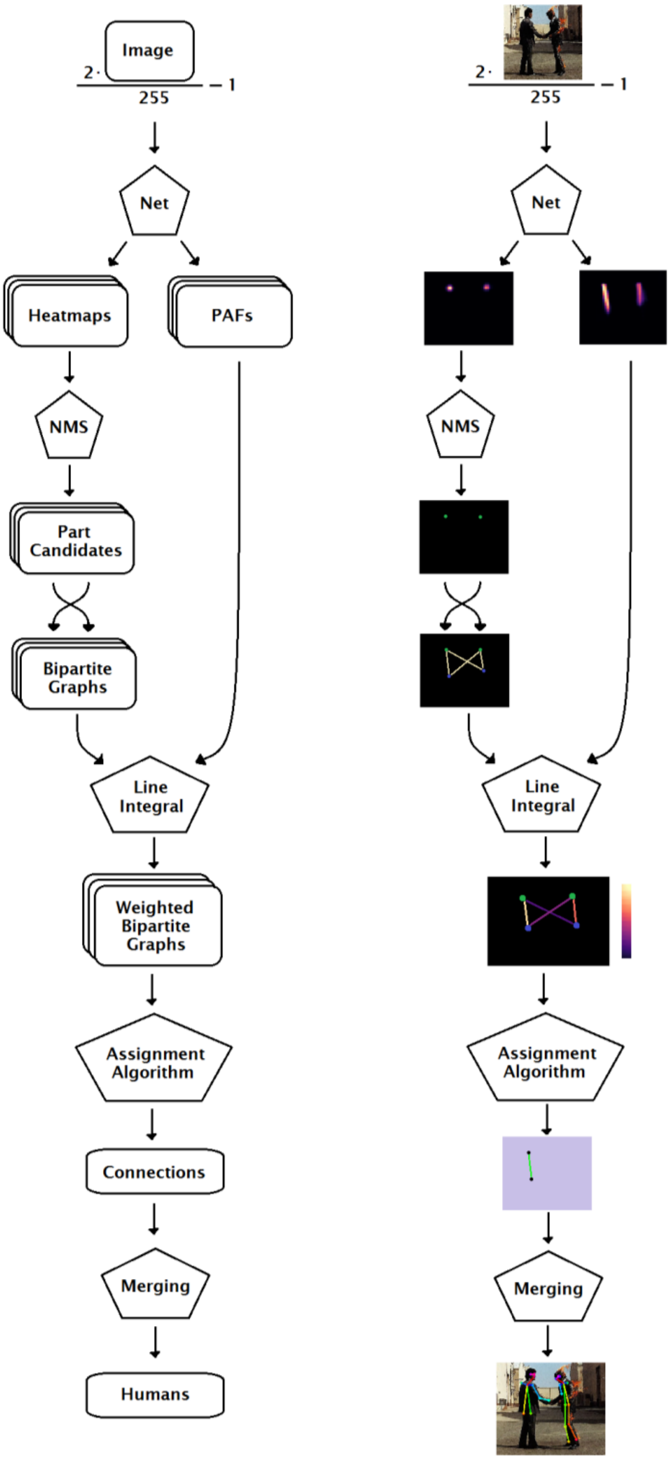


Figure 3.5– OpenPose Pipeline [36].

3.3 Object Detection (YOLO)

In recent times, Convolutional Neural Networks have been used in many applications of Computer Vision and have outperformed many algorithms in visual perception or object recognition [24,25], but most of these networks need specialized, high-powered, costly hardware (e.g. NVIDIA GPUs) to achieve high performance [26]. There are various kinds of CNN algorithms for object detection, such as R-CNN, Faster R-CNN, Mask R-CNN, SSD, YOLO (Figure 3.6), and from their speed vs. accuracy trade-offs we found that YOLO (You Only Look Once) is best for fast detection [27,28], compared to other.

Figure 3.6– Various kinds of CNN algorithms for object detection.

Standard YOLO version requires GPUs but there is a tiny version for the CPU called, Tiny-YOLO that can be implemented in low-powered machines like those of Smart Wheelchairs with a tolerable limited speed in processing and it has 30 layers of network structure as shown in Figure 3.7.

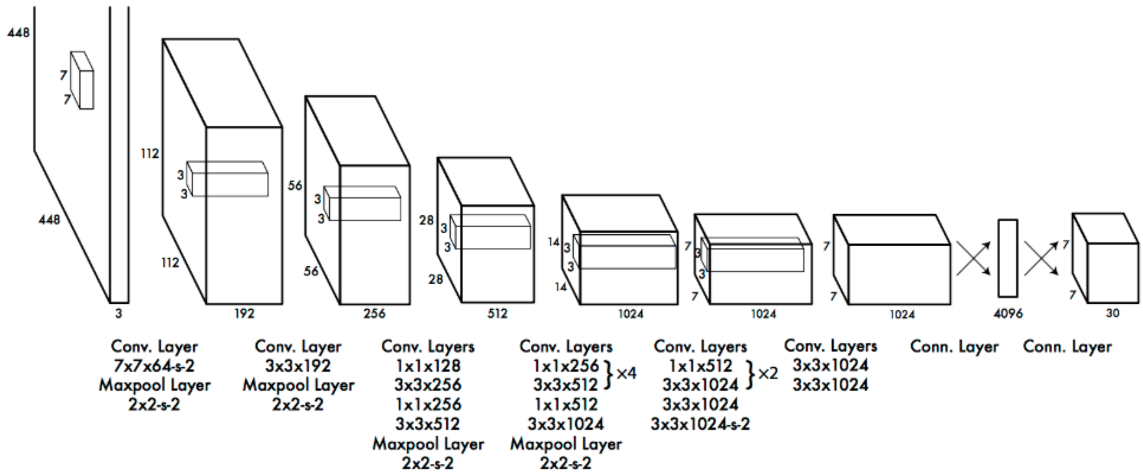


Figure 3.7– Tiny-YOLO network structure.

YOLO model has several advantages over classifier-based systems. It applies a single neural network to the full image. This network divides the image into regions and predicts bounding boxes and probabilities for each region as shown in Figure 3.8. These bounding boxes are weighted by the predicted probabilities. This makes it extremely fast, more than 1000x faster than R-CNN and 100x faster than Fast R-CNN. The YOLO model can processes images in real-time at 45 FPS on GPUs like the GeForce GTX Titan X by NVIDIA.

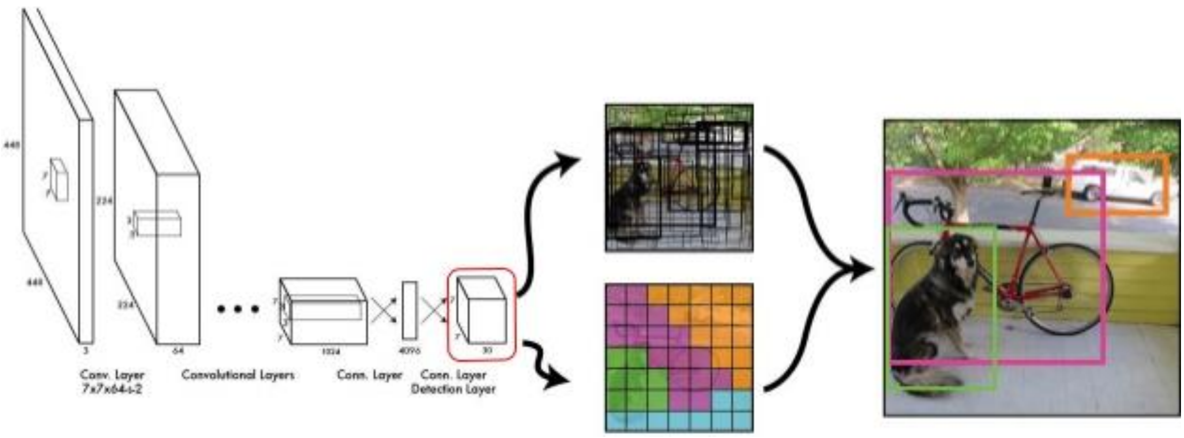


Figure 3.8– Bounding box predictions by YOLO.

We note that although YOLO has faster detection speeds, the bounding boxes of detected objects sometimes do not give accurate and precise values of the object locations according to ground truth in real-time applications. In addition Tiny-YOLO detection accuracy and precision is less than original YOLO version. Therefore, we approach a modification to the Tiny-YOLO network layers. Our modified Tiny-YOLO network is explained briefly in the Sec. 5.3.1.

3.4 Summary

In this chapter we discussed about different types of detections. Our focus on human pose detection and several pose detection methods are described in this chapter. Nevertheless, we found that Tensorflow based OpenPose detection model is best suit for the base of our solution to the pedestrian avoidance issue. In addition, we also discover the suitable object detector model as Tiny-YOLO version which we will use as a base network for our bus and bus door detection problem for our bus boarding Smart Wheelchair.

Chapter 4 Pedestrian Avoidance

4.1 Overview

In this chapter, we explain our proposed method to enhance the movements of Smart Wheelchairs for severely impaired users by analyzing and detecting the intensions and interactions of individual and multiple pedestrians. We use an RGB camera to take images and process the images with Tensorflow based on the OpenPose system to detect the intensions and interactions of pedestrians with our proposed method.

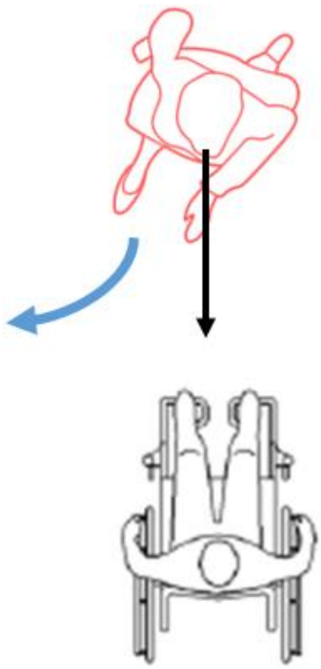


Figure 4.1– Gaze and body orientation of a pedestrian.

Our study consists of three steps for analyzing moving pedestrians. In the first step, we determine the human gaze for pedestrians' awareness of the Smart Wheelchair and in the second step, we determine the body orientations for estimating the intended future movement directions of pedestrians (Figure 4.1). Finally, we apply a strategy for which direction our Smart Wheelchair should move to avoid collisions, based on these orientations. The system architecture, methods, and results are given below.

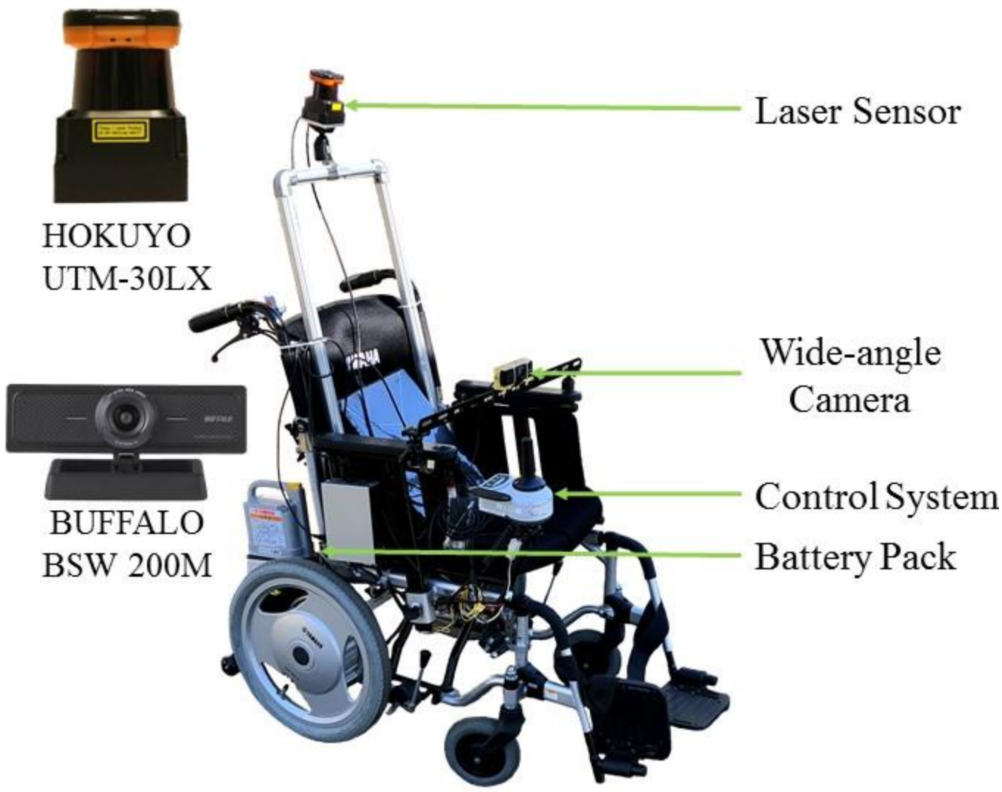


Figure 4.2– Our wheelchair system for pedestrian avoidance.

4.2 System Outline

In our proposed system, the Smart Wheelchair needs to have a wide enough range to sense the environment so that it can detect the various types of hazards in real environments. We use a combination of an RGB camera and laser sensor and evaluate its effectiveness in perceiving all possible types of obstacles that exist in outdoor environments and developed our robotic wheelchair as shown in Figure 4.2.

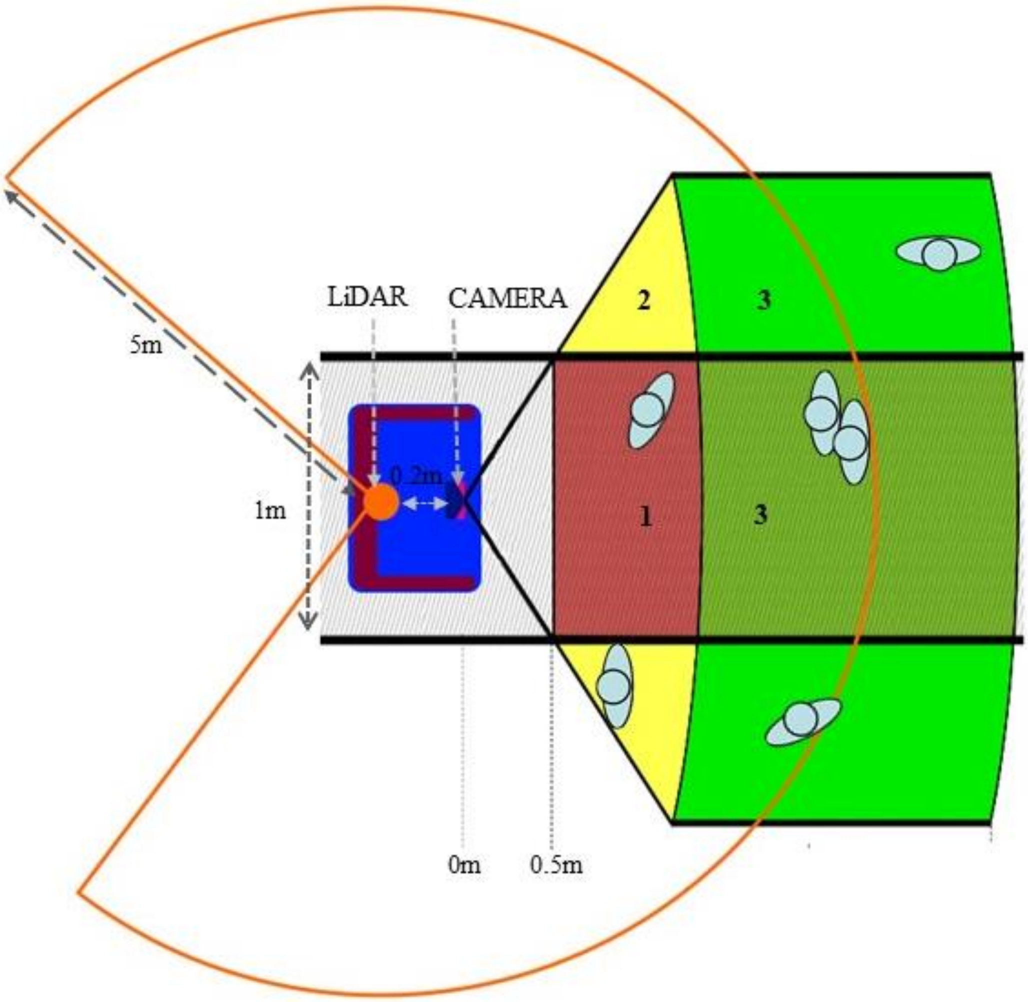


Figure 4.3– Our wheelchair sensor detection ranges and our defined “zones” to determine potential collisions. The LiDAR's range is shown as the orange arcs.

We use a BUFFALO BSW 200M series camera as a major sensing unit for a wide-viewing angle of 120° and mount it 75cm above the ground in front of the wheelchair with a 0° inclination to get an aligned picture with the horizontal plane. This camera has a focal length of f=1.8mm with a max resolution of 1920×1080 at 30fps. For our experiments, we use 800×600 resolution image frames.

Moreover, the wheelchair is equipped with a Laser Range Sensor (UTM-30LX by Hokuyo) with a coverage angle of 270° installed on the rear frame at 1.05m from the ground. In our experiment, we use our previous detection method in [11] as an added layer of safety to avoid unexpected obstacles and consider the maximum range of the laser sensor to be 5m and set a buffer for free space with a minimum range of 0.5m around the wheelchair to avoid collisions like in Figure 4.3.

In Figure 4.3, we divided the field of view of our vision camera equally into three sections as middle (1 and 3), right (2 and 3) and left (2 and 3). Here, area number 1 is considered as the area with the most potential for collisions, while area number 2, is considered as the "moderate zone". Area number 3 is considered less prone to collisions. Our Smart Wheelchair mainly runs in the forward, left, and right directions.

4.3 Methodology

Our Smart Wheelchair system primarily uses two procedures to control its speed along with avoidance of obstacles. At first, our system estimates the pedestrians' head and body orientations and with the help of those estimates, we determine their intensions and interactions with the wheelchair. Secondly, we use our direction finding technique to select the direction at which the wheelchair will move autonomously to reduce the user's workload.

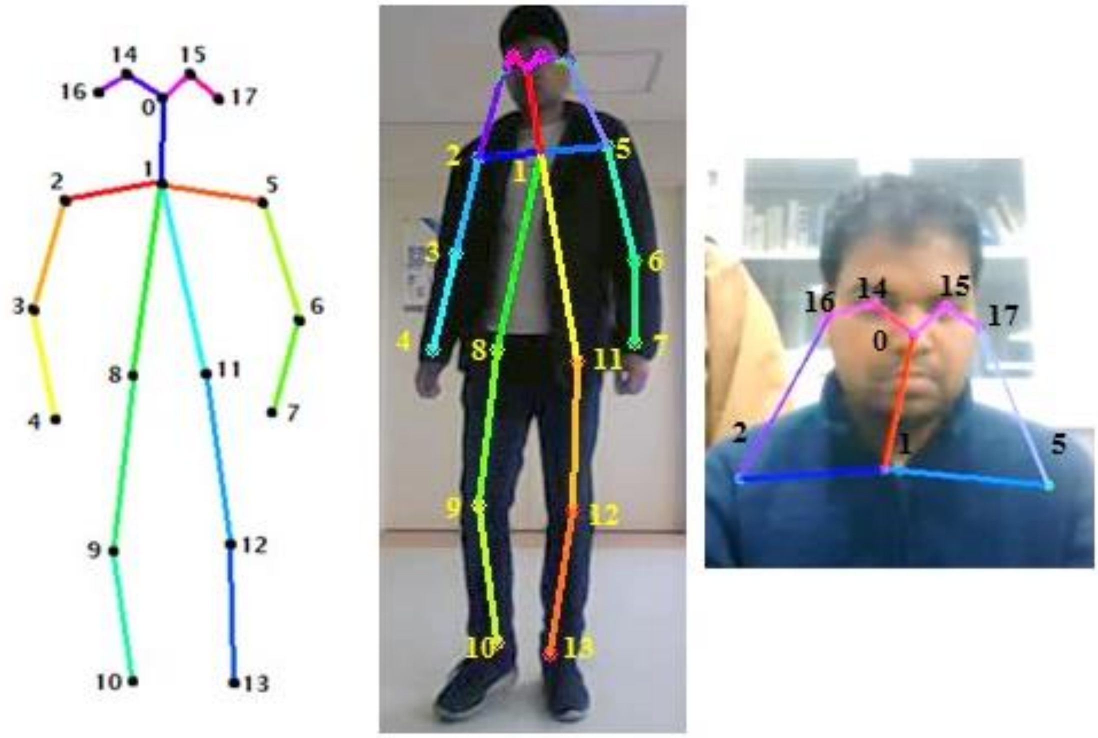


Figure 4.4– Human keypoints extracted by OpenPose based on the COCO dataset model.

4.3.1 Pedestrians' Pose Estimation

We used the "Tensorflow based OpenPose" [14] as discussed in Sec. 3.2, for pose estimation . The original OpenPose library runs on the Caffe deep learning framework, which is computationally intensive. On the other hand, Tensorflow is a very fast framework for real-time performance. The Tensorflow port of OpenPose allows for high-speed performance at the cost of some loss in accuracy. However, we have found that this slight decrease in accuracy is a good trade-off for our real-world application. We used the pre-trained COCO dataset model [15] of OpenPose in our pro-gram with Tensorflow in Python on a Notebook GPU (NVIDIA GTX1070), which gives us detections speeds of around 10fps. The result is a network with a depth of 57 layers including 18 layers for body parts localization as shown in Figure 4.4, 1 layer for the background, and 19 layers for limbs information in each of the x and y directions.

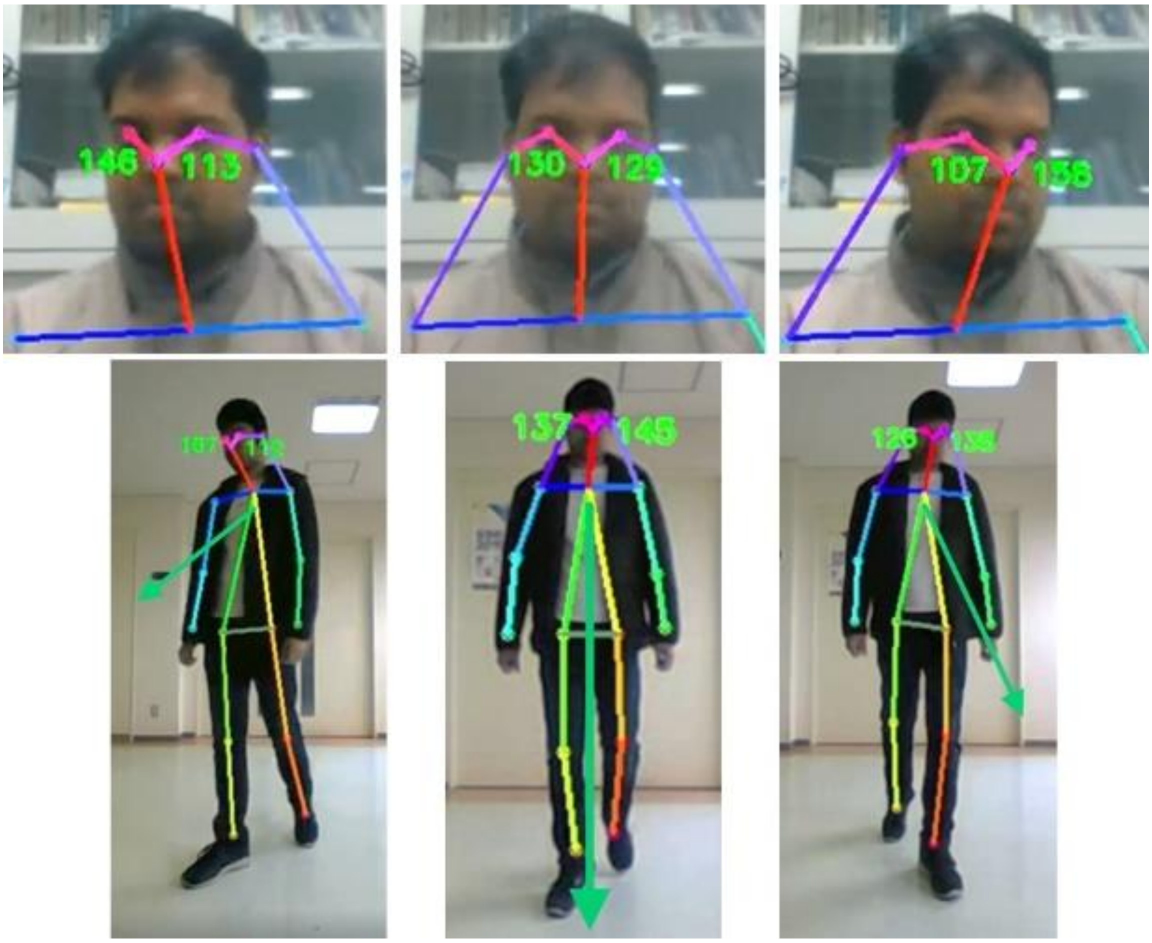


Figure 4.5– Measuring head orientation (top) and body pose directions (bottom).

We use the output data of detected body parts to determine head and body orientations. For head orientations, the detection step generates possible regions of the head using the neck, nose, two eyes, and two ears. In our detections, we take the nose, eyes, and neck to estimate various angles and make the observation that when a person turns his head left, the left eye angle is greater than the right eye angle with the connecting line from the left shoulder to left ear also vanishing and vice versa as in Figure 4.5 (top). Whereas, Figure 4.5 (bottom) shows the body orientations, where the left, middle, and right images show that the person's body is posing in the left, forward, and right directions respectively. We use the coordinates of the neck, two shoulders, and two hips to estimate body orientations.

4.3.2 Intension and Interaction Evaluation

In our system, we have found it effective to first divide the frame equally into three sections, namely, the middle, right, and left based on our defined zones in front of the camera in Figure 4.3. We then determine human awareness of the wheelchair using head orientations and the wheelchair's movement direction is determined by combining the pedestrians' awareness with their body positions and orientations.

Whenever a wheelchair encounters pedestrians, the wheelchair can most likely maintain its path because the pedestrians usually give way for the wheelchair. However, this may depend on the pedestrians' awareness of the wheelchair. If we assume a person does not tilt his/her head towards the ground or is not using a phone while walking, we can interpret that he/she may be clearly aware of the wheelchair's presence. We used this strategy for all persons detected by OpenPose and with their head orientations estimated by our method (Sec. 4.3.1). We point out some basic interactions of pedestrians to simplify our problem and assume the types of interactions with the wheelchair based on each frame and for how long any given pedestrian interacts with the wheelchair. The basic concept of our algorithm for determining the pedestrians' awareness of the wheelchair depending upon the position in the frame and head orientations of pedestrians can be stated as:

- Three conditions, when (1) the pedestrian's position is in the centre of the frame and the pedestrian is facing straight, (2) the pedestrian's position is in the right of the frame and facing left, and (3) the pedestrian's position is in the left of the frame and facing right, then our wheelchair upon detecting these conditions, con-siders the pedestrian as looking at the wheelchair i.e. aware of it and that the pedestrian will try to avoid the wheelchair.
- For the rest of the positions and head orientations of the pedestrians, the wheel-chair will assume that pedestrians are not looking at the wheelchair and unaware of it.

- If the pedestrian is looking at the wheelchair for a fraction of a second, the wheelchair will assume the pedestrian is unaware of it and will stop and process the next frame to get the head orientations to try to determine awareness.

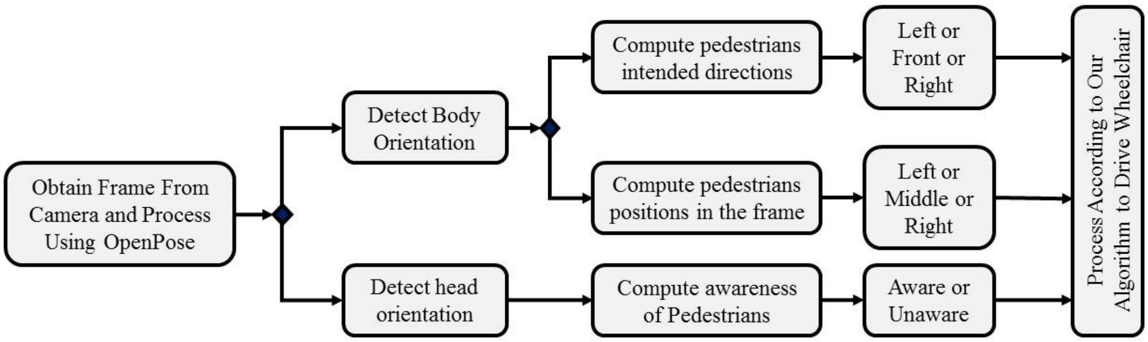


Figure 4.6– Detections process diagram of our Smart Wheelchair.

Next, we compute body orientations from skeleton data and also count the pedestrians' positions in the frame, for determining human intensions. Basically, using the combination of pedestrian awareness and body positions in the frame and orientations, we determine the probable locations that pedestrians will move to in order to determine where the wheelchair should move as shown in Figure 4.6.

4.4 Autonomous Navigation

Our Smart Wheelchair uses discrete actions (forward, right, left, slightly right, slightly left, slow down, and stop): at each time step, in order to avoid collisions and navigate efficiently and smoothly towards the goal. After detecting the awareness of the wheelchair and body orientations of the pedestrians with their position in the frame we determine which direction the wheelchair should move according to the decisions Table 4.1.

Table 4.1 – Decision matrix of wheelchair steering procedure for different scenarios.

<table>
<tr><th></th><th>Person Location In Frame
Body Orientation</th><th>Right</th><th>Middle</th><th>Left</th></tr>
<tr><td rowspan="3">Aware</td><td>Right</td><td rowspan="2">Move Forward</td><td>Move Slightly Left</td><td>Slow Down</td></tr>
<tr><td>Straight</td><td>Slow Down</td><td rowspan="2">Move Forward</td></tr>
<tr><td>Left</td><td>Slow Down</td><td>Move Slightly Right</td></tr>
<tr><td rowspan="3">Unaware</td><td>Right</td><td rowspan="2">Move Forward</td><td>Move Left</td><td>Stop</td></tr>
<tr><td>Straight</td><td>Stop</td><td rowspan="2">Move Forward</td></tr>
<tr><td>Left</td><td>Stop</td><td>Move Right</td></tr>
</table>

We also calculate the partially relative distance by how much the pedestrian is covering the frame, for some special cases where the pedestrian is unaware of the wheelchair and his/her position in the camera frame appears to be too close to the wheelchair. By that we can determine the closest potential collision and avoid the collision by stopping the wheelchair. To determine the distance (D) from the camera to a person, we consider the usual width of a pedestrian (Y), which is around 400mm and the number of pixels the person covers in the image frame (X) as in Figure 4.7. Then, the distance equation is:

$$D = fYA/sX \tag{4.1}$$

where, f = 1.8mm is the focal length and s = 4.5mm is the sensor width of the camera. A = 800 pixel, is the camera frame width in pixels and X is the pedestrian's width in pixels. We consider the 25% portions of the image frame on both the right and left sides as a safe area and the middle as the risky area (Figure 4.7).

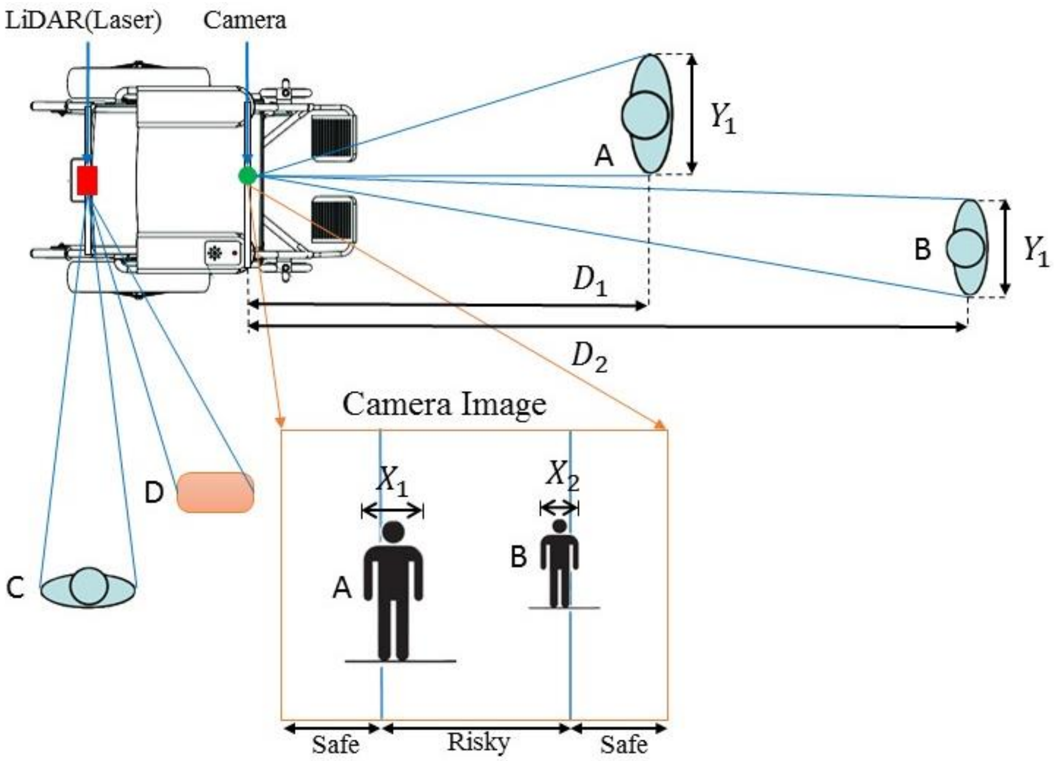


Figure 4.7– Distance Measurement Using the Camera

While operating, if a pedestrian occupies the risky area for example in cases where pedestrian has his/her back facing the wheelchair and is not aware of it, then the wheelchair stops and waits for the person to clear the way. However, if pedestrians occupy the safe areas, the wheelchair will continue to move while performing our detection and manoeuvring algorithms. Finally, we use the laser sensor detection process from [11] for additional safety, to avoid any pedestrians or objects that are out the camera's range as shown in Figure 4.3. Experimental results of our method and discussions are given below.

4.5 Experimental Results and Discussion

We first tested our method inside our lab as in Figure 4.8. We tested our method for a single person and attempted to estimate the interactions and intensions of his/her next move. Then, we conducted our tests for two persons in a group and analysed their behaviours as in Figure 4.8. To test the prediction accuracy, we extracted 2D trajectories from real-world pedestrian videos shot in a train station. In Figure 4.8, the estimated human trajectory is represented as the yellow colored arrows in the frames and the wheelchair moving direction is represented in red. Head directions are represented in blue. For planned slight movements of the wheelchair, we denote the arrow as curved otherwise, the movements are denoted as straight arrows. Black arrows indicate the slowing down or stopping of the wheelchair.

We now show how our approach works when a pedestrian is too close to the wheelchair. For example, the measurement of distance is given in Figure 4.9. Here, the distance of the pedestrian is too close, almost 0.7m in front of the wheelchair. In this situation, the wheelchair would stop and process the next frame and get directions until it sees a clear path which is same as our previous work [10]. Figure 4.10 and Figure 4.11 shows the detections by a laser sensor and our method respectively.

In Figure 4.10 we used a laser sensor for our first safety priority and for testing our method in outdoor environments. However, later we did not use the laser sensor after we reprogramed our method. In our last setup, we used only vision camera to do the all sensing operations. Since we developed the distance measurement equation, now we can measure the distance of the person in front of the wheelchair. We experimented with our method to get quantitative results for evaluation. Our quantitative results are discussed below.

We experimented on 50 trials and our smooth avoidance rate for our wheelchair movement is 75.4% as shown in Table 4.2, our wheelchair stopes 4.87% times and people avoiding on their own is about 19.73% times. Therefore, we address this 24.6% results of our 50 trials as errors for our wheelchair. There are 18 basic types of scenarios from the Table 4.1 where we used all the possible scenarios at least 2 times.

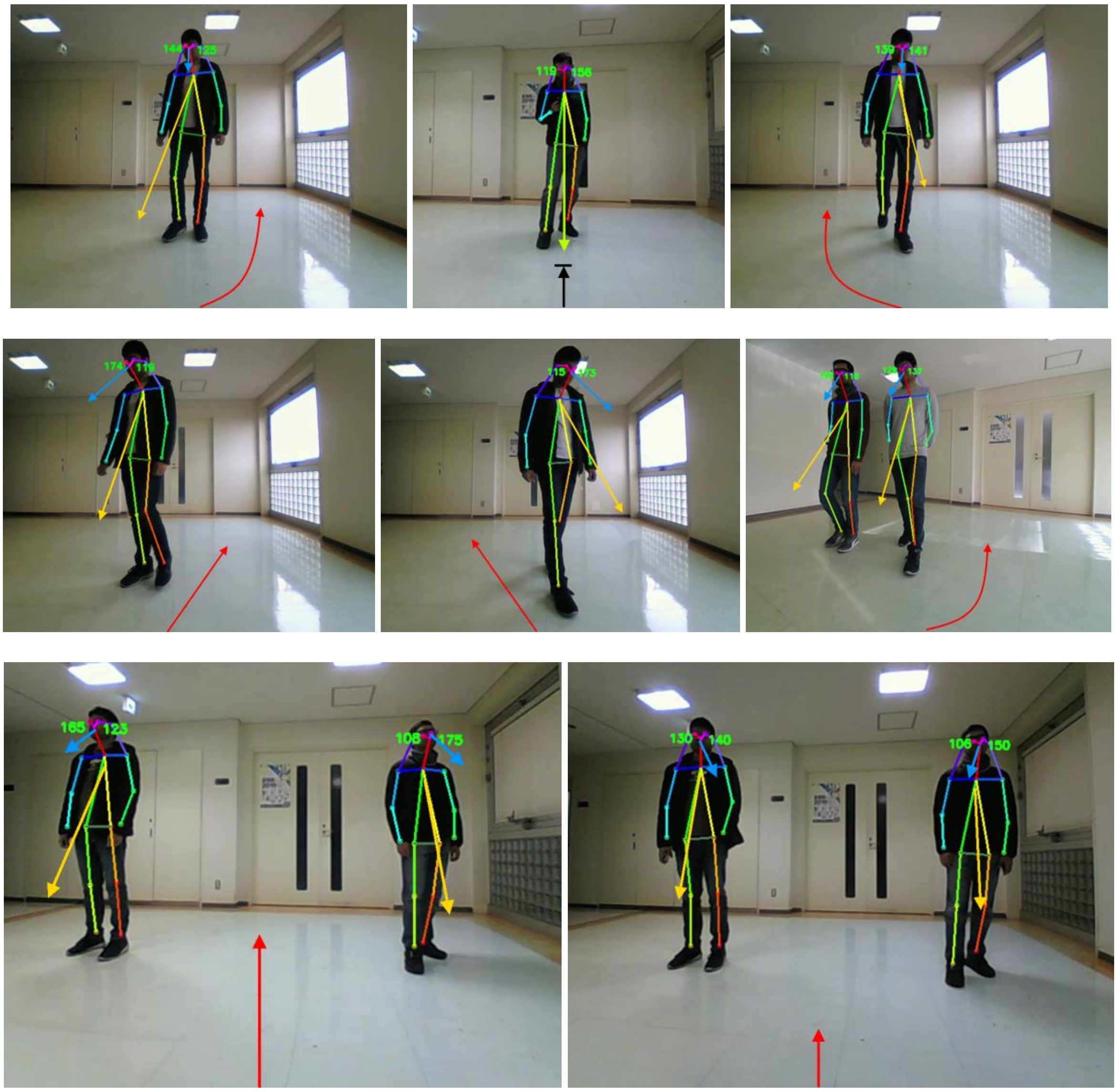


Figure 4.8– Wheelchair maneuvering directions for different combinations of human head and body orientations.

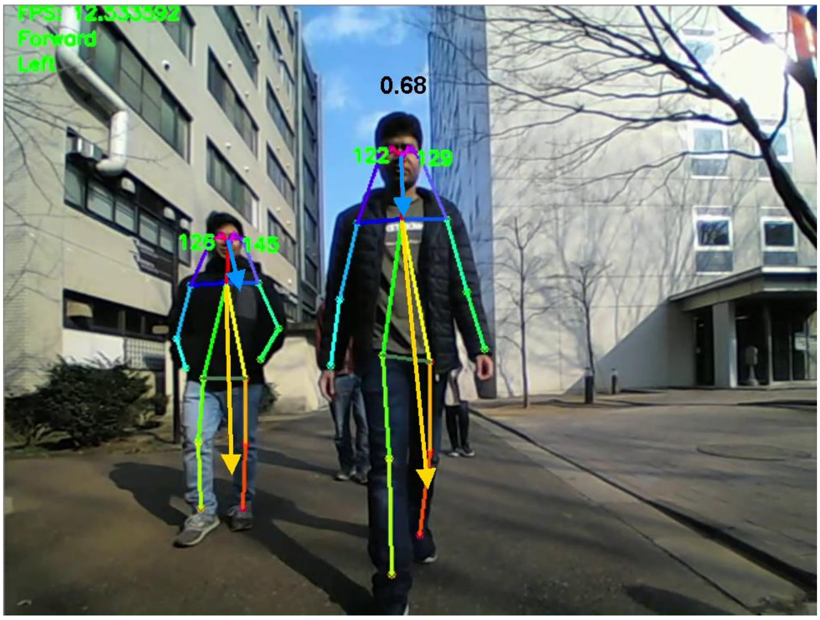


Figure 4.9– Distance measurement using camera when a pedestrian is too close.

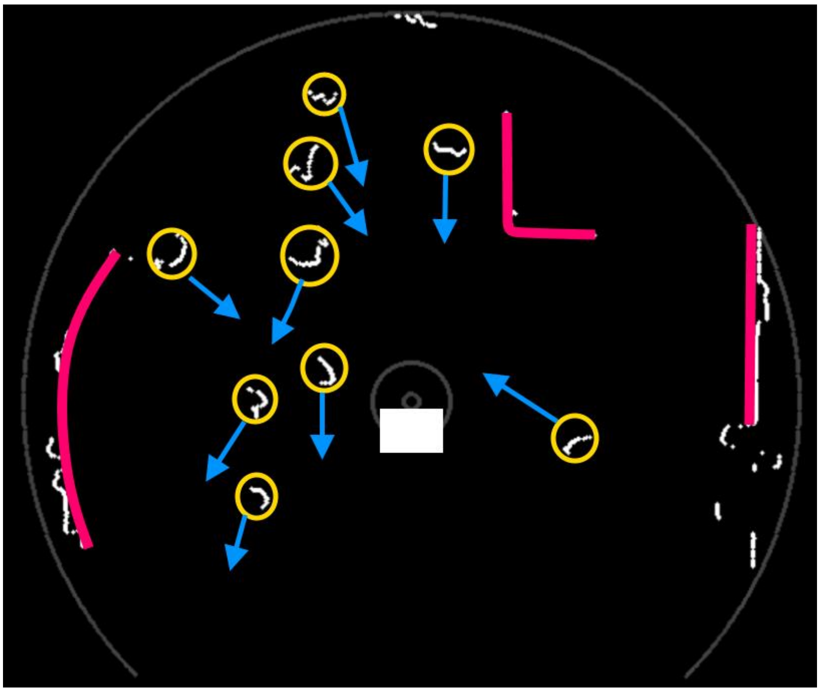


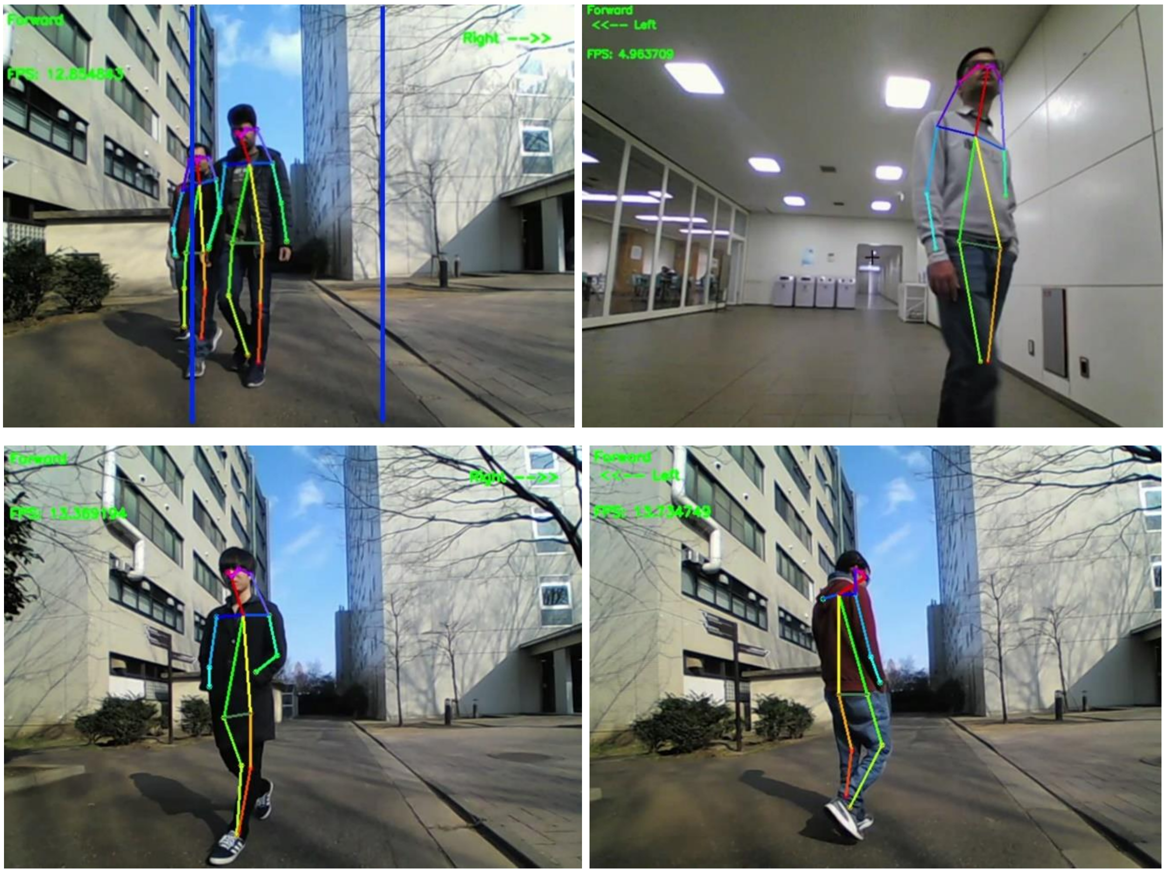
Figure 4.10– In a train station, detection of pedestrians with a laser sensor. Here the wheelchair is in the middle, yellow circle denotes the detected pedestrian, the light blue arrows represent their directions and red is for static obstacles like walls.

Figure 4.11– In a train station, detection of pedestrians with our method.

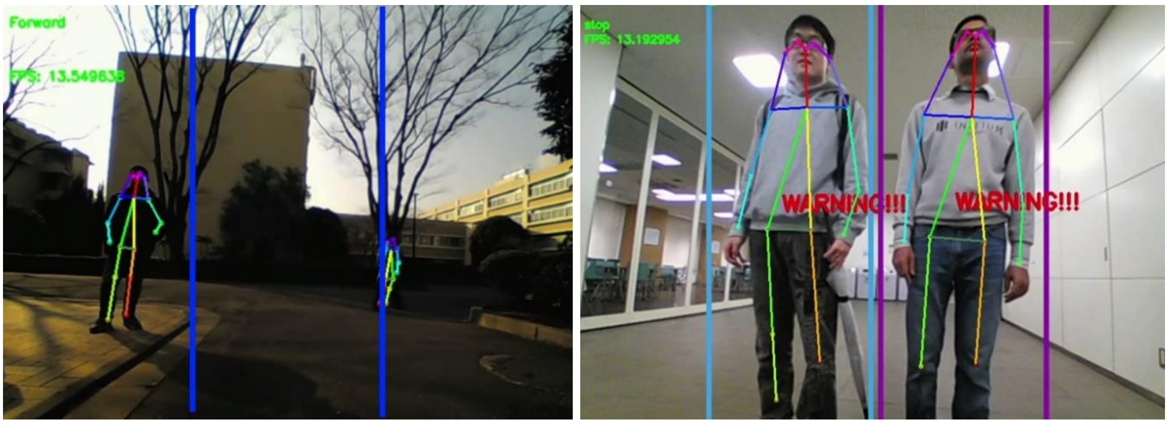
In Table 4.2, the errors are due to delays during the acquisition of data/images. Another reason for the false detection of the TensorFlow based OpenPose is due to the failure to obtain gaze along with body orientation data. Also, sometimes this OpenPose version yields fluctuating body parts data in each frame. This fluctuation might have been caused by the resolution of the video frame, since we used 800×600 resolution images. Therefore, further improvements need to be devised in later research.

Table 4.2 – Results of 50 trials on outdoor experiments.

Trials	**On Average**		
	Smooth Avoidance of Wheelchair	**ERROR**	
		People Avoiding on Their Own	**Wheelchair Stops**
50	75.4%	19.73%	4.87%



(a) Right and Left



(b) Forward and Stop

Figure 4.12–Detection Final Results with Smooth Avoidance

Figure 4.12 shows the final detection result of our experiments where the wheelchair avoids pedestrians smoothly and carefully. Figure 4.12(a) shows the turning moments of the wheelchair to the right or left. It also can be seen, that our system can avoid people from behind as shown in the Figure 4.12(a)(bottom right). Figure 4.12(b) displays the forward and stop moments in our experiments. We have modified our distance measuring program to show us warnings in the image as in Figure 4.12(b)(right) when a person is too close to the wheelchair, for example, *1m to 0.7m*. This function is for the safety of the user to stop the movement of the wheelchair. Figure 4.12 also shows the division of frames into 3 parts where in the middle part, most of our operations were successfully executed.

Figure 4.13– Example of our experiment with a user.

Moreover, we conducted our experiment with users also to observe user understanding as shown in Figure 4.13. Here, our autonomous wheelchair speed was a constant as set to 0.285 m/s, which was very comfortable for users. The frame sequence in the figure shows that the wheelchair runs smoothly without stopping.

Figure 4.14– Our experimental errors.

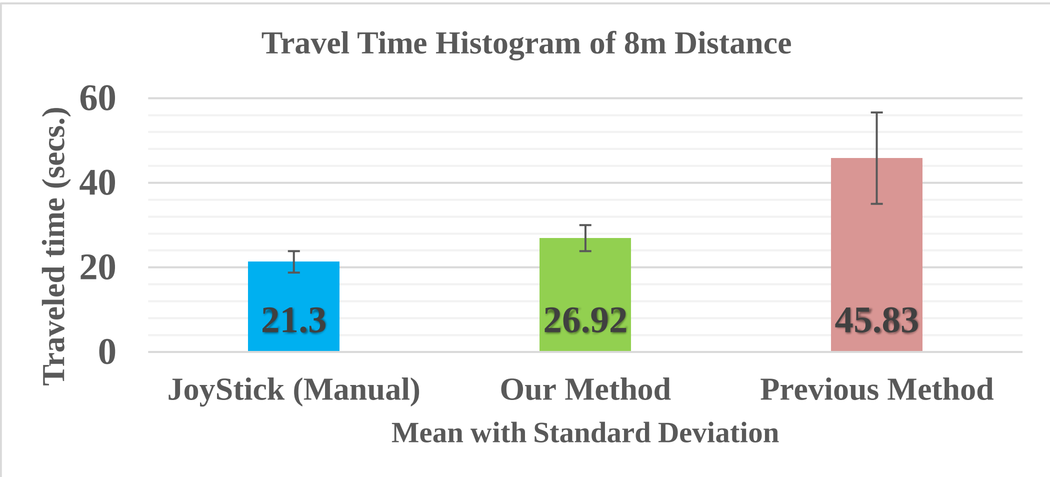


Figure 4.15– Average with standard deviation time difference when using joystick (manual), our method and our previous method for 25 trials.

Although our avoidance method was almost perfect, it does gives some errors results also as shown in Figure 4.14. From the figure, we can see that it gives the wrong direction for wheelchair to move towards. Sometimes it also gives warnings for stop where the Smart Wheelchair does not needs itself to stopping, which can be overcome by modifying the algorithm.

In terms of task completion time using two pedestrians back-to-back, we found the time it takes to navigate to the goal by using the joystick (manual control) was the fastest and averaged 21.3 secs., as shown in Figure 4.15. We compare our new method with a previous one [10] (the HFI-Hands Free Interface method). From the Figure 4.15, we can determine that the previous method required almost double the time when joysticks (manual control) were used, with an average time of 45.83 secs. for travel over an 8m meter distance where the pedestrian stands as an obstacle in the way. Whereas, our method required 26.92 secs. for the same kind of scenario, which is much closer to the speed when using a joystick. In terms of standard deviation, we can also see that the deviation of 3.0782 secs. is low, compared to the previous method which is 10.8087 secs. where for manual drive it is 2.5385 secs. Our approach is much faster and the wheelchair does not stop and wait for the person to move from its way as in our previous method. Our method runs smoothly like how a normal user would operate the wheelchair while navigating crowed environments. In short, it shows that our method is better than the previous method and can work as well as manual control with a joystick in terms of the generated motion, time and the distance it travelled.

To evaluate the effectiveness of our method, we conducted an experiment with 20 participants to use our wheelchair system in two modes:

1. With autonomous functions.
2. With user only operations for avoiding pedestrians.

For pedestrians, we appointed some volunteers. Each participant evaluated the systems with a questionnaire, providing their feedback on a scale of 1 to 5 representing "I do not think so at all", "I do not think so", "Neither", "I think so", and "I definitely think so". The question are:

- Q1. Is it easy to predict the wheelchair's behavior?
- Q2. Was it possible to predict in advance when the wheelchair would turn to the right, left, or stop?
- Q3. Was the ride comfortable?
- Q4. Was it easy to move the wheelchair when in manual mode?/ Was the operation easy in automatic mode of wheelchair?
- Q5. Is it safe in crowded places?

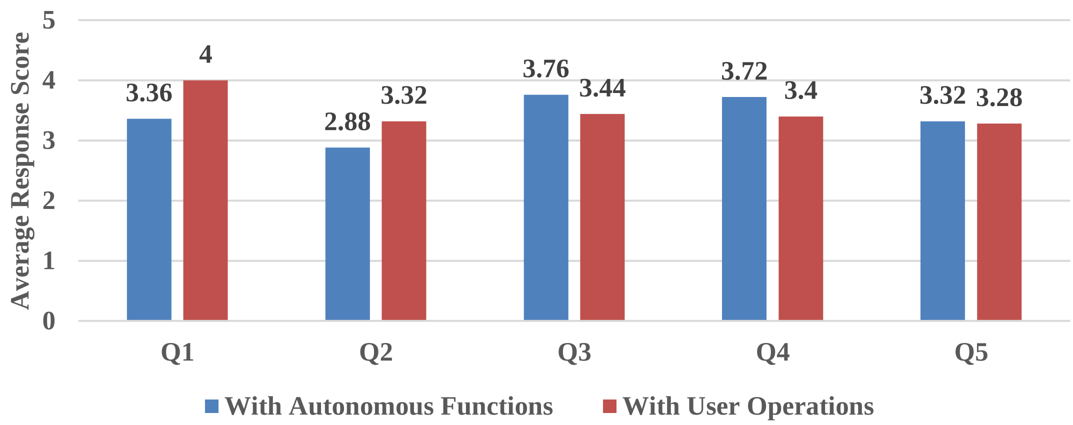


Figure 4.16– Average response scores from the questionnaires in the experiments.

Figure 4.12 shows the average participants' responses to the questionnaire. We calculated the scores for autonomous functions and also for user only operations (manual operation).

Question number 1 and 2 are for getting user assumption before riding the wheelchair into the autonomous functions. Question 3, 4, 5 are for getting the wheelchair performance in autonomous mode. Therefore if these 3, 4, 5 questions receive better scores in autonomous compared to user manual operation then our approach is acceptable. From the results, we conclude that for questions 3, 4 and 5, which are the most important survey of our method has greater score in autonomous functions than the user operations.

We have proposed a robotic wheelchair that observes pedestrians and the environment. It can recognize the pedestrians' intensions from his/her behaviors and its surrounding environment. Experimental results show our approach is promising. We collected data from a train station and also from our campus to evaluate our methods.

4.6 Summary

In summary, we have developed a wheelchair maneuverability approach for severely impaired users. We have found that our system, which utilizes Tensorflow based OpenPose data can detect pedestrian interactions with the wheelchair and their intended next movements. We investigated the combination of two possible orientations of pedestrians (head and body) and evaluated its effectiveness in perceiving various types of wheelchair movement directions in response to pedestrian behaviors in crowded outdoor environments.

Chapter 5 Bus and Bus Door Detections

5.1 Overview

Our main goal in this chapter is to focus on the vision part, involving camera processing with the help of a CNN. We propose a bus boarding wheelchair system that can get onto a buses using CNN based image recognition for reliable and precise localization of bus doors. This is a work on a bus boarding wheelchair system in terms of the camera processing, vision component. We used YOLO dark-net as primary detection frame work as stated in Sec. 3.3. Some example of detections using YOLO detector is shown in Figure 5.1.

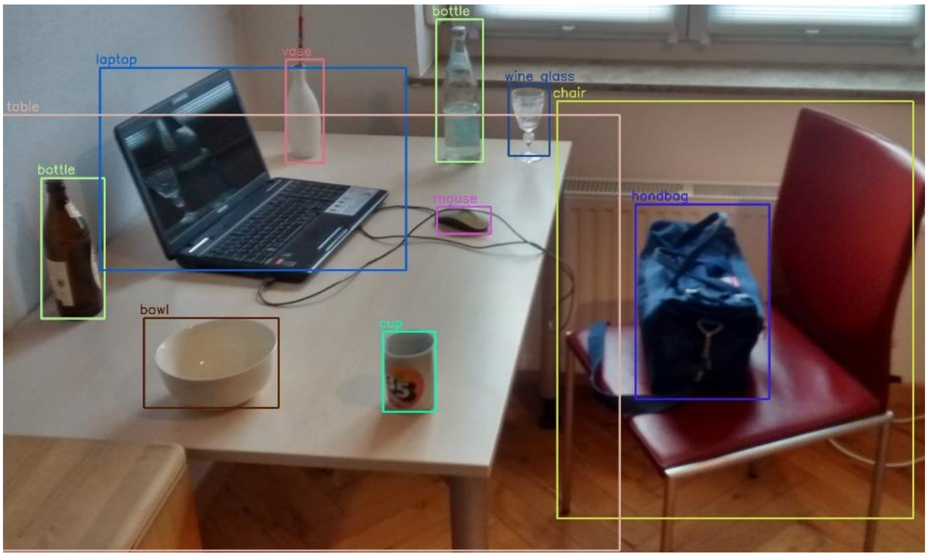


Figure 5.1– Detections Using YOLO Darknet

For real-time processing using our notebook, we modified the Tiny-YOLO version provided by the author to run at a fair amount of speed of around 10FPS entirely on a CPU. Once our system detects an open door, a Hough line transform algorithm is applied to get accurate and precise localization of the door lines.

The subsequent sections describe our wheelchair system, our proposed methodology, and shows the precision over the YOLO bounding boxes on detected class objects. In the last sections, we discuss our results and conclusions.

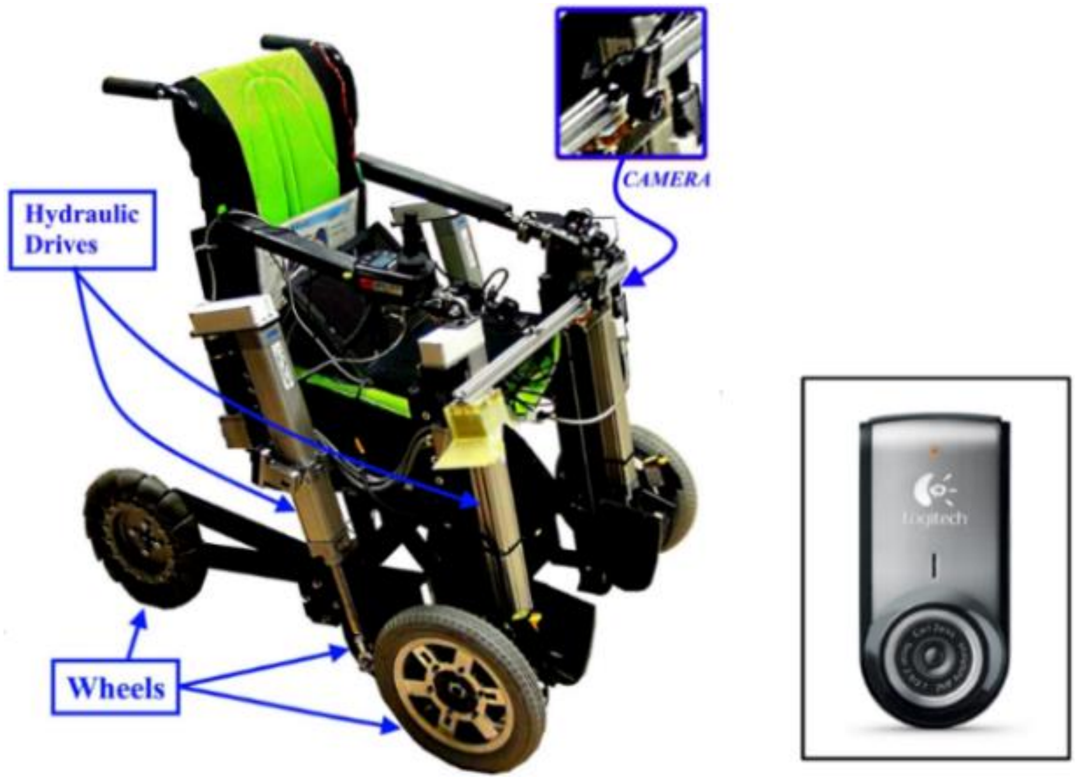


Figure 5.2– Bus boarding Smart Wheelchair system.

5.2 Our BMR Wheelchair System

We collaborated with Toyota Motor Corporation and the University of Tokyo to develop a new autonomous six wheeled Smart Wheelchair that can overcome the steps ahead of its path like bus doorsteps or escalators. Figure 5.2 shows our proposed Smart Wheelchair, which is called the Bus-boarding Mobility Robot (BMR). This configuration makes it so the wheelchair can step onto a specific height with the front wheels and simultaneously balance itself with the rear wheel. Once the front wheel has a good grip on the upper portion of the step, it moves forward, while the middle wheel is lifted up and the wheelchair is balanced with the help of the rear wheel. Moreover, the wheelchair is equipped a wide vision camera (Logitech c905) for capturing frames. The camera is mounted at 75 cm above the ground plane along with our bidirectional sensing system [23].

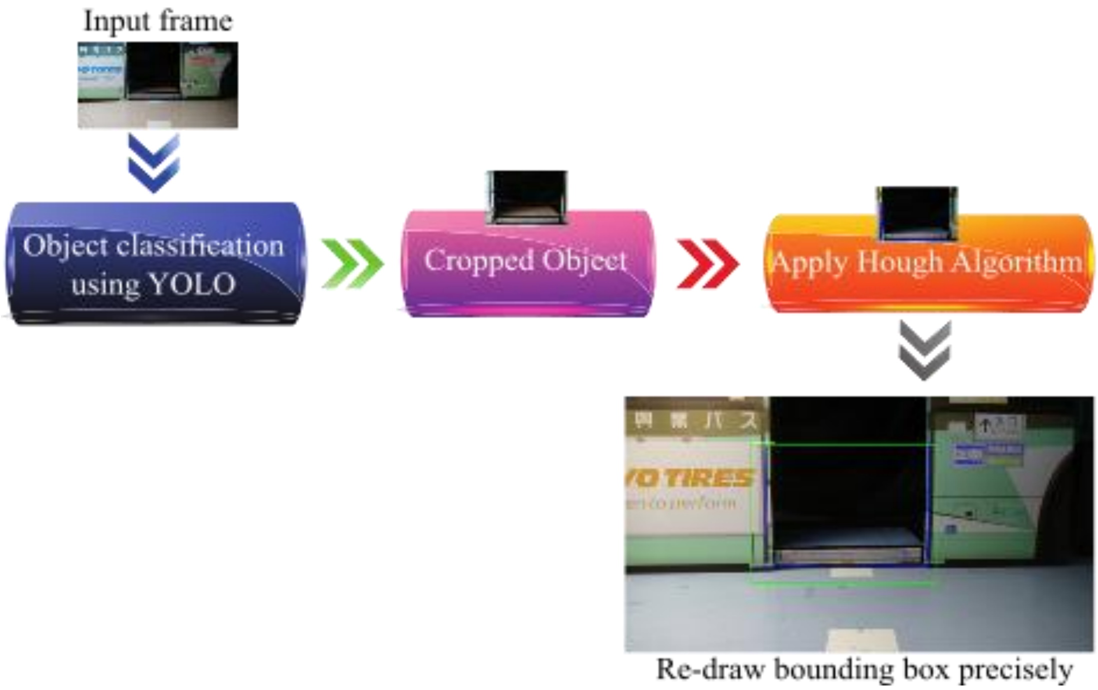


Figure 5.3– Basic system process diagram.

5.3 Proposed Methodology

Our proposed system in Figure 5.3 illustrates the basic system process diagram where, we use the modified Tiny-YOLO version for primary detection of our predefined class of objects, namely, “bus”, “open-door” and “close-door” of the bus. Afterward if an open door is detected the system crops the area around the bounding box and executes our Hough line transform algorithm to get accurate and precise localization of the door lines.

As the localization is not perfectly performed by Tiny-YOLO version, we introduce our refinement method to get the precise localization of the bus door. Subsequent sections briefly describe our methodology for getting precise bounding box information using our system.

5.3.1 Bus and Bus Door Detection

In real-time visual object detection, speed is a significant factor to consider along with the accuracy of the detection system. Our selected detector model (YOLO model) from Sec. 3.3 can process images in real-time. In addition, devised a simplified architecture of the Tiny-YOLO network that can process images at nearly 5 FPS without using any GPU and still maintains almost the same accuracy and precision of YOLO. However, despite that fact, such speeds are still too slow for our detection process. Some researchers in [30, 31] have devised techniques to reduce runtime by changing the network’s filter sizes and layers according to object features. Therefore, we first doubled the numbers of filters in the first convolutional layer to extract enough local information and visual features and replaced some 3×3 filters with 1×1 filters for reducing the filter size in the following 2 layers. This boosted the network’s speed but reduced the accuracy and precision. To increase the accuracy and precision we down sampled the image to make sure the last layer remains the same as Tiny-YOLO. Our modified network is implemented in Python using Tensorflow with a 3.6GHz Intel Core i7 CPU.

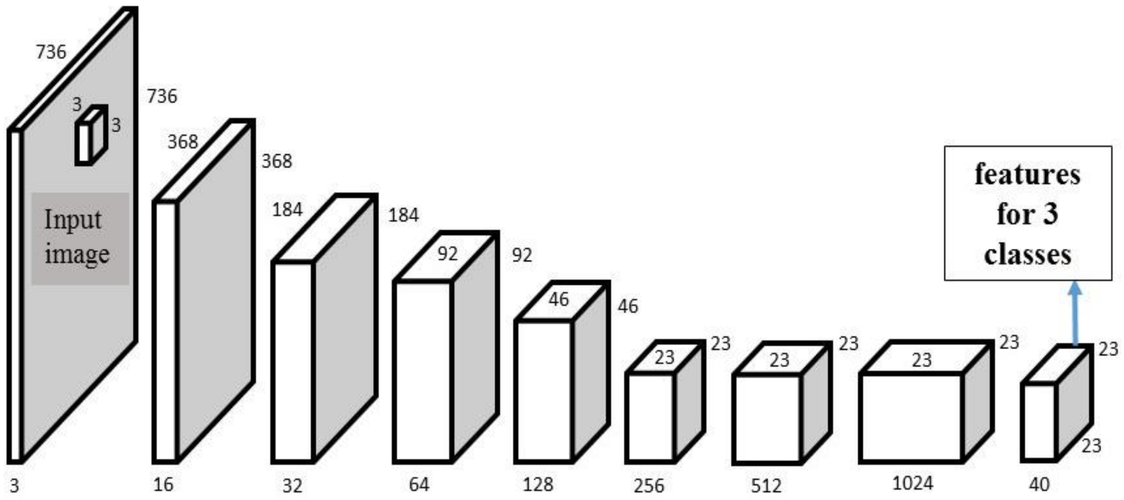


Figure 5.4– Modified Tiny-YOLO layer structure.

For our research purposes, we trained the network with 3 classes: “bus” and the bus door with “closed door” and “opened door” states. The conventional Tiny-YOLO model has 30 layers as stated in Sec. 3.3, but in our modified tiny-YOLO network, we reduced the layers to 15, since our object class number is less and similar. The network layer structure and the layer descriptions are represented in Figure 5.4 and Table 5.1, respectively.

For training, we collected images from different sources like Google, the VOC 2007 and VOC 2012 datasets, and also from our camera and arranged them such that around 600 images would be annotated for each class to train the system. So the total images are 1800 for training and 600 images for validations. In the training period we used the default learning algorithms provided by the author of YOLO Darknet. Moreover, we changed the batch size to 128, and subdivision to 4 to achieve an improved learning rate. To generate better visual features, the training was done on an NVIDIA GeForce GTX Titan X for 40000 epochs. During training, the image was divided into S×S splits/grids and the output of the last layer gives a feature vector, which represents region predictions. These predictions were encoded in the last layer as an S×S×(B×5+C) tensor. Where, B is the collection of predicted bounding boxes in each of the grid cells and their locations are rep-

Table 5.1 – Modified Tiny-YOLO Layers descriptions

Layers	Filters	Size	Input	Output
1. Conv.	16	3 x 3 / 1	736 x 736 x 3	736 x 736 x 16
2. Maxpool		2 x 2 / 2	736 x 736 x 16	368 x 368 x 16
3. Conv.	32	3 x 3 / 1	368 x 368 x 16	368 x 368 x 32
4. Maxpool		2 x 2 / 2	368 x 368 x 32	184 x 184 x 32
5. Conv.	64	3 x 3 / 1	184 x 184 x 32	184 x 184 x 64
6. Maxpool		2 x 2 / 2	184 x 184 x 64	92 x 92 x 64
7. Conv.	128	3 x 3 / 1	92 x 92 x 64	92 x 92 x 128
8. Maxpool		2 x 2 / 2	92 x 92 x 128	46 x 46 x 128
9. Conv.	256	3 x 3 / 1	46 x 46 x 128	46 x 46 x 256
10. Maxpool		2 x 2 / 2	46 x 46 x 256	23 x 23 x 256
11. Conv.	512	3 x 3 / 1	23 x 23 x 256	23 x 23 x 512
12. Maxpool		2 x 2 / 1	23 x 23 x 512	23 x 23 x 512
13. Conv.	1024	3 x 3 / 1	23 x 23 x 512	23 x 23 x 1024
14. Conv.	1024	3 x 3 / 1	23 x 23 x 1024	23 x 23 x 1024
15. Conv.	40	1 x 1 / 1	23 x 23 x 1024	23 x 23 x 40

resented by 5 location parameters x, y, w, h, and class label confidence c. C represents the number of classes. In our framework, $S = 7, B = 2$, and $C = 3$. Each bounding box (BBox) requires 6 parameter values: x, y, w, h, class (C), and confidence (c) as shown in Equation 5.1.

$$BBox = (C, x, y, w, h, c) \tag{5.1}$$

Here (x, y) and (w, h) represent the minimum coordinate values of the bounding box and width and height of the bounding box, respectively. From these values, the maximum coordinate is $(x + w, y + h)$.

Finally, the confidence score for each box represents the intersection over the union (IoU) between the predicted box and ground truth box with the probability of the detected object. The class scores for detecting our three class labels (Bus, Close-door, Open-door) were approximately 70%. Figure 5.5, shows the detected class labels on frames for different types of buses and their orientations. The state of the door of the bus is vital information for our boarding wheelchair.

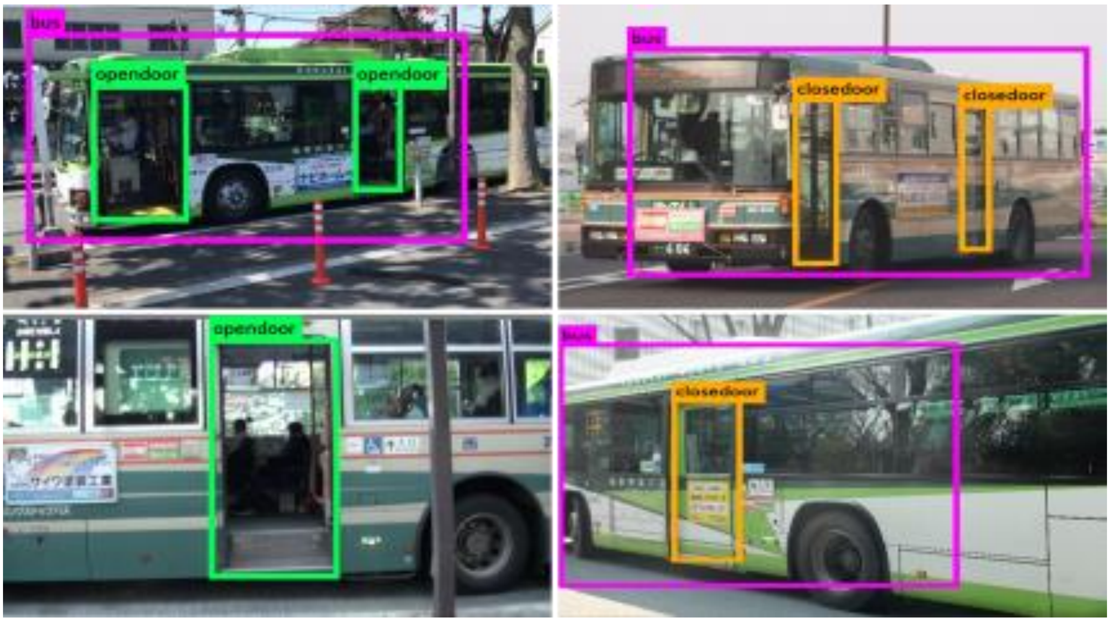


Figure 5.5– Detection of our classes for different buses.

5.3.2 Precise Bounding Box Method

YOLO is fast, but then again it is not capable of providing accurate shape information beyond a rectangular bounding box for localizing the object in the frame. This is because different camera angles can distort the shape of even doors, which have straight line features. Fortunately, the Hough line transform is effective for recognizing basic line shapes in image [32]. Therefore, we use the Hough line transform for refined estimation of the door's shape.

Whenever the CNN network detects any open door of the bus, the system crops the image within the bounding information (BBox) to our next layer. Now, the image contains mostly bus door pixels where we apply the Hough line transform algorithm to get straight lines. To avoid redundant lines, a mask is applied as shown in Figure 5.6 so that we can focus on only the lines near the door's edge. After applying the mask, the layer processes the image and uses the Hough transform to find the best three lines that fit with the Open-door shape. Moreover, we redraw the bounding box with the Hough detected lines to detect the bus door precisely before boarding onto the bus. For precise boarding, we use the sensor framework for measuring the door width and height of the steps [23].

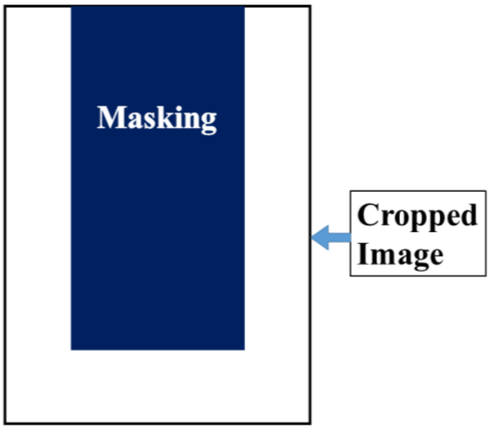


Figure 5.6– Masking in cropped image.

5.4 Experimental Results and Discussion

Performing bus boarding experiments are always time consuming and difficult to manage so we set up a mock bus structure in our laboratory to confirm the effectiveness of our proposed method for fast and precise detection of buses and bus doors (in Figure 5.7).

Figure 5.7– Experimental setup for detection of bus with our BMR wheelchair.

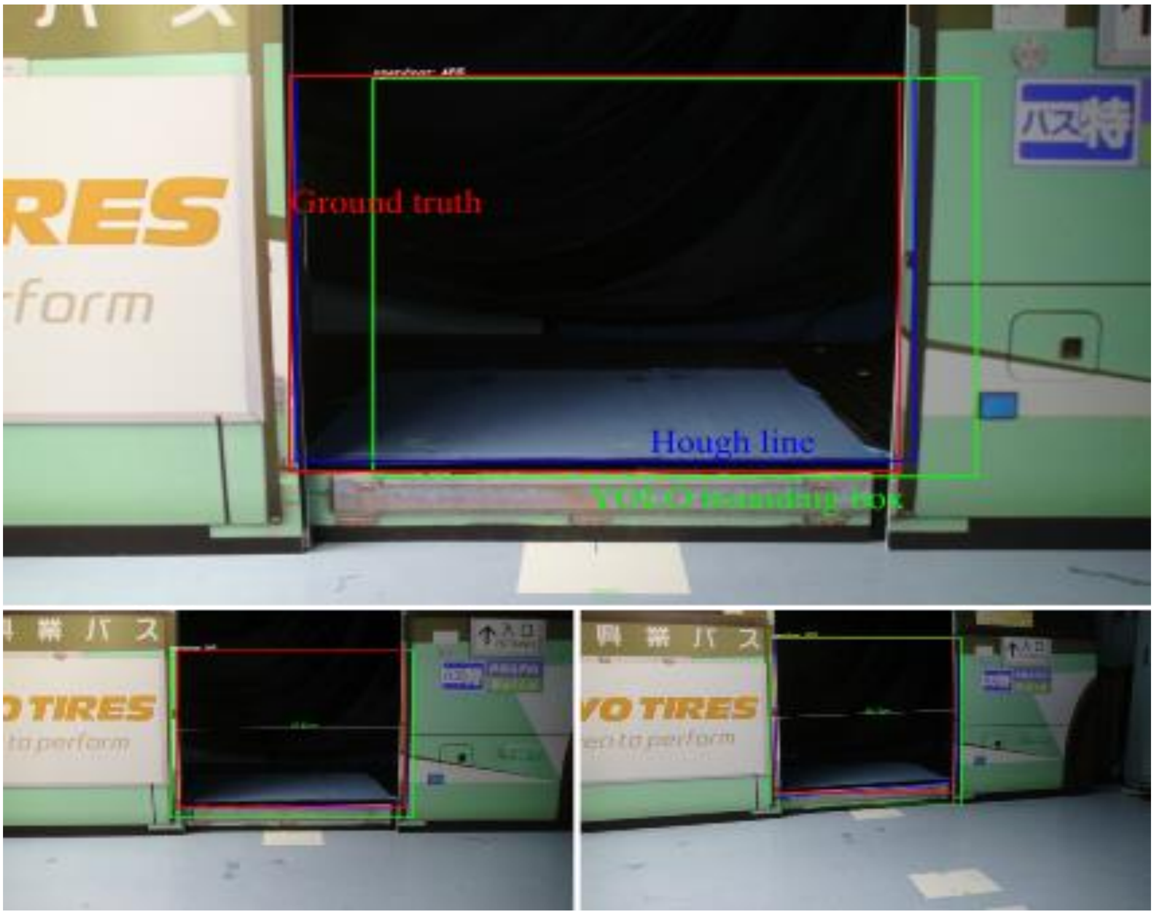


Figure 5.8– Demonstration of ground truth, YOLO, and our Hough line approach.

Our final object detection system runs at 10FPS, which is fast enough to run in real-time. From Figure 5.7, we can also see the detected class as "open door" and "close door" in the experiment's setup. But, the detected bounding boxes for the classes are displaced from the actual locations of the objects. Considering these inaccuracies, we applied the Hough line method to correct the localization of the object. Figure 5.8 demonstrates the effectiveness of our method for improved bounding boxes.

In Figure 5.8, the red, green, and blue boxes represent the ground truth, modified Tiny-YOLO, and our proposed method respectively. We conducted an experimental evaluation has been conducted, for comparing the performance of our proposed method and YOLO with ground truth shown in Figure 5.9 and Figure 5.10. In our experiments, we conducted 22 trials with our mock-up to get a dataset of bounding box values for our tests.

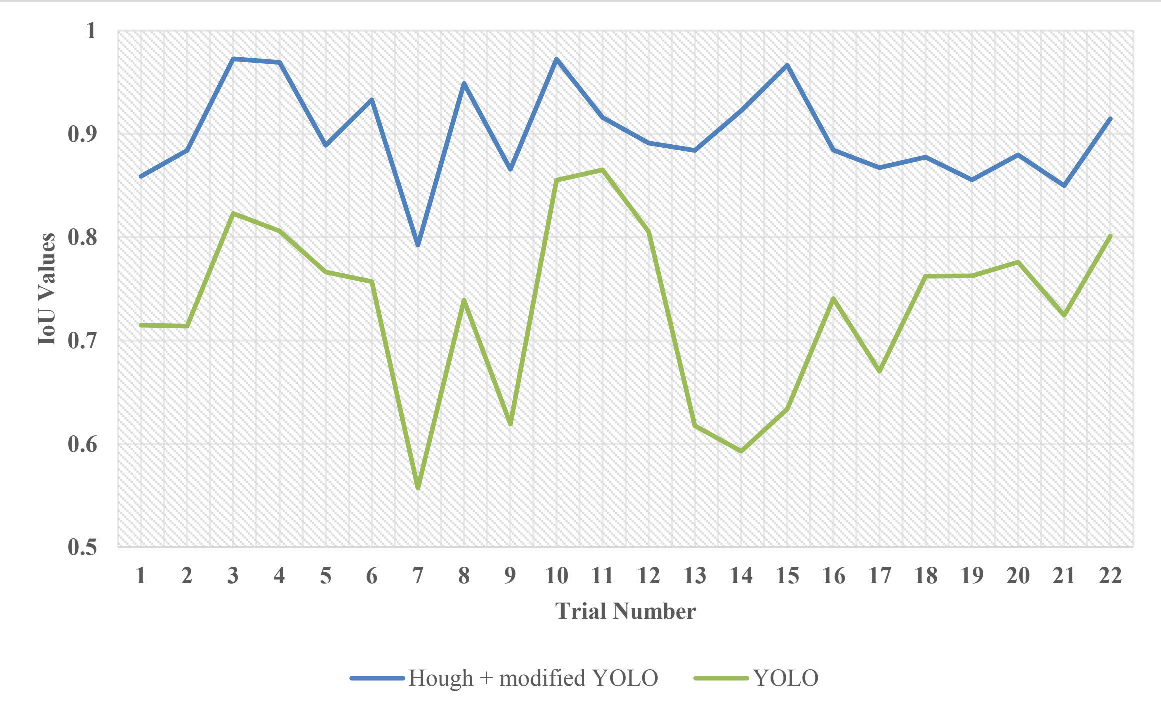


Figure 5.9– IoU score comparison over 22 trials.

For comparing the effectiveness, we calculated the intersection over union (IoU) between bounding box values from the ground truth with the modified Tiny-YOLO, and our proposed method using equation 5.2.

$$IoU \ = (Area\ of\ Overlap)/(Area\ of\ Union) \tag{5.2}$$

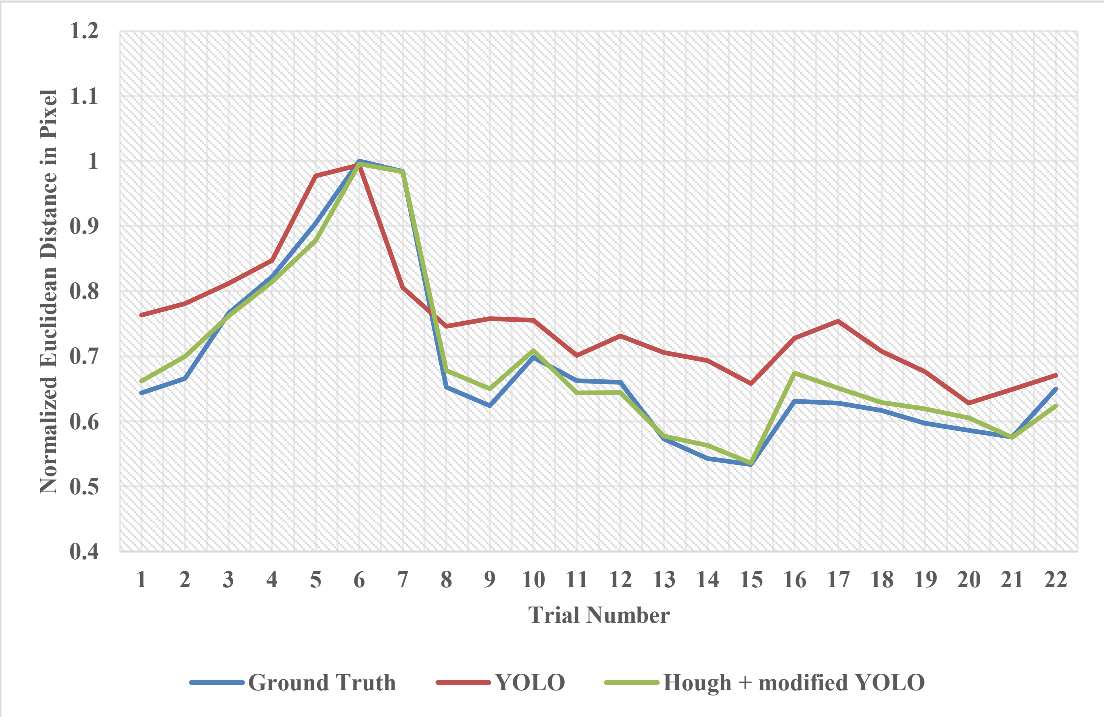


Figure 5.10– Normalized Euclidean distance comparison over 22 trials.

Figure 5.9 illustrates the IoU scores of all 22 trials for our modified Tiny-YOLO and our proposed Hough Line method with respect to the ground truth. Moreover, Table 5.2 shows the average value of IoU of our image recognition system and proposed method. The average value of IoU of our proposed method indicates an increase in bounding box accuracy by nearly 20% over the default image recognition system with respect to the ground truth.

In addition, we measured the Euclidean distance between the minimum and maximum values of the coordinates for the bounding boxes from the ground truth, modified Tiny-

YOLO, and our proposed method. From Table 5.2, we can see from the average Euclidean distances, our method is very close to the ground truth values. This is also true for the standard deviation and variance of the Euclidean distances, which are also very similar to the ground truth. Moreover, Figure 5.10 shows a Normalized Euclidean distance comparison and we can also see that our method is very close to the ground truth. So our proposed method works better than conventional CNN for localizing accurate door positions.

Table 5.2 – Different types of comparisons between three types of bounding box values.

Compared terms	**Ground Truth**	**Modified Tiny-YOLO**	**Our proposed method**
Average of IoU value	1.00	0.73	**0.90**
Average of Euclidean distance of open bus-door width	2067.29	2277.04	2088.80
Standard deviation of Euclidean distance of open bus-door width	400.45	283.47	379.90
Variance of Euclidean distance of open bus-door width	160364.24	80354.40	144326.80

Table 5.3 shows the comparison between Tiny-YOLO and our proposed method with ground truth values in terms of error percentage. Form Table 5.3, the average values of IoU have a 10% error rate. In addition, the average of the Euclidean distance of the open bus-door width has only a 1.04% error rate, which indicates that our method is better than Tiny-YOLO in terms of accurately detecting the bus door.

Table 5.3 – Comparison between Tiny-YOLO and our proposed method with ground truth values in terms of error.

Compared Terms	**Error with ground truth**	
	Tiny-YOLO	**Our proposed method**
Average of IoU value	27%	**10%**
Average of Euclidean distance of open bus-door width	10.14%	**1.04%**

5.5 Summary

In summary, since visual detection is typically costly in terms of time complexity, we aimed to reduce the computational cost of CNN based detection for running a real-time system. We also showed that a purely CNN based detection method based on bounding boxes has some inaccuracies in terms of object localization. Our method of localizing a class object (bus door) significantly improves over this. Moreover, we compared the Tiny-YOLO detection approach and our proposed combined detection method with the original ground truth to show that our method performs better localization of the bus door.

Chapter 6 Conclusions and Future Work

6.1 Conclusions

This thesis presented an interdisciplinary research approach that combines techniques and methods from several research domains, including sociology, psychology, and human-robot interaction. Furthermore, it also presents an incremental study in designing the methods to estimate important cues for human robot interaction and smooth navigation of smart robotic wheelchairs in urban environments. To make Smart Wheelchairs usable in outdoor environments, we worked on two important scenarios, (1) maneuvering among pedestrians and (2) detection of buses and bus doors for boarding buses. Using gaze combined with body orientation, we can avoid pedestrians smoothly and for bus boarding, we used simple image processing in the vision part to achieve fast operation of our bus boarding wheelchair for precisely detecting open bus doors. Our first priority was the user's experience in outdoor environments with an autonomous and user-friendly Smart Wheelchair.

In outdoor environments, there is an uncertainty concerning the pedestrians' motions and in dynamic environments. Thus it is quite difficult for autonomous Smart Wheelchairs to detect the pedestrians' interactions and intensions. In this thesis, we have developed a wheelchair maneuverability approach for severely impaired users. We have found that our system, which utilizes the Tensorflow based OpenPose detector, can successfully detect pedestrian head and body orientations, which in turn, provides feedback on pedestrian interactions with the wheelchair and their intended next movements.

In summary, we investigated the combination of two possible orientations of pedestrians (head and body) and evaluated its effectiveness in perceiving various types of wheelchair movement directions in response to pedestrian behaviors in crowded outdoor environments. The final system results in auto-navigation that generates wheelchair movements that are safe and comfortable to wheelchair users and other people in real-time multi-person scenarios.

In addition, another primary goal of this thesis is to propose a bus boarding Smart Wheelchair that can localize detected bus doors precisely using a vision based system. Visual detection is typically costly in terms of time complexity. Therefore, we are aimed to reduce the computational cost of CNN based detection for running a real-time system. We also showed that a purely CNN based detection method based on bounding boxes has some inaccuracies in terms of object localization. Our method of localizing a class object (bus door) significantly improves over this. Moreover, we achieved a 90% IoU result with respect to the ground truth, which was a significant improvement over using only our modified YOLO network. Additionally, we successfully boarded our wheelchair onto a bus using our bidirectional sensing system using a single LiDAR. Our proposed method supports the Smart Wheelchair for precisely localizing the bus door so that the Smart Wheelchair can board with less computation cost.

Moreover, our system also has the capability to sense the environment to detect objects like people, buses, and other trained objects according to the system configuration so that the wheelchair user would be free to roam around the city independently and comfortably.

6.2 Future Work

In the future, we will build a more sophisticated method for very busy environments to achieve robust performance with more precise estimates of the pedestrians' next move and free space for our wheelchair to move through.

- More complex experiments will be conducted to detect the pedestrians' movements like if the pedestrian were to suddenly appear in front of the wheelchair.
- Complete control free AI movement of the wheelchair. One main point would be that after avoiding the people the Smart Wheelchair would reposition itself onto its original track.
- In the future, we will add a navigation system for reaching the user's preferred location.

For the future work of our bus boarding wheelchair, we are also planning to speed up our computational process for detection by using a simpler version of a CNN that can be run on small hardware like a low-cost CPU.

List of Publications

Published Journal Articles:

Sarwar Ali, S. Al Mamun, H. Fukuda, A. Lam, Y. Kobayashi, Y. Kuno, "Smart Robotic Wheelchair for Bus Boarding Using CNN Combined with Hough Transforms", Lecture Notes in Artificial Intelligence, vol. 10956, pp 163-172, Springer, 2018.

Sarwar Ali, A. Lam, H. Fukuda, Y. Kobayashi, Y. Kuno, "Smart Wheelchair Manoeuvring Among People", Lecture Notes in Artificial Intelligence, vol. 11645, pp 32-42, Springer, 2019.

Conference Proceeding:

Sarwar Al Mamun, **Sarwar Ali**, H. Fukuda, A. Lam, Y. Kobayashi, Y. Kuno, "Companion following robotic Wheelchair with bus Boarding Capabilities", International Conference on Informatics, Electronics & Vision (ICIEV 2018), Japan.

References

1. National Institute of Health, https://www.nih.gov/news-events/news-releases/worlds-older-population-grows-dramatically, March 28, 2016, Accessed date- November 1, 2017.
2. Overview of Disability, http://www.worldbank.org/en/topic/disability, September 20,2017, Accessed date- November 15, 2017.
3. Gomi, T., Griffith, A.: Developing intelligent wheelchairs for the handicapped. In: Assistive Technology and Artificial Intelligence, pp. 150-178). Springer, Berlin, Heidelberg (1998).
4. Al Mamun, S., Lam, A., Kobayashi, Y., Kuno, Y.: Single laser bidirectional sensing for robotic wheelchair step detection and measurement. In: ICIC, pp. 37-47. Springer (2017).
5. Ali, S., Al Mamun, S., Fukuda, H., Lam, A., Kobayashi, Y., Kuno, Y.: Smart Robotic Wheelchair for Bus Boarding Using CNN Combined with Hough Transforms. In: ICIC, pp. 163-172. Springer (2018).
6. Fehr, L., Langbein, W.E., Skaar, S.B.: Adequacy of power wheelchair control interfaces for persons with severe disabilities: A clinical survey. Journal of rehabilitation research and development, 37(3), pp.353-360 (2000).
7. Bauer, A., Klasing, K., Lidoris, G., Mühlbauer, Q., Rohrmüller, F., Sosnowski, S., Xu, T., Kühnlenz, K., Wollherr, D., Buss, M.: The autonomous city explorer: Towards natural human-robot interaction in urban environments. International journal of social robotics, 1(2), pp.127-140.
8. Tomari, M.R.M., Kobayashi, Y., Kuno, Y.: Development of Smart Wheelchair System for a User with Severe Motor Impairment. In: IRIS (2012).
9. Kümmerle, R., Ruhnke, M., Steder, B., Stachniss, C., Burgard, W.: Autonomous robot navigation in highly populated pedestrian zones. Journal of Field Robotics, *32*(4), pp.565-589 (2015).
10. Tomari, R., Kobayashi, Y., Kuno, Y.: Enhancing Wheelchair Maneuverability for Severe Impairment Users. International Journal of Advanced Robotic Systems, Vol.10, 92-105 (2013). DOI: 10.5772/55477
11. Tomari, R., Kobayashi, Y., Kuno, Y.: Socially Acceptable Smart Wheelchair Navigation from Head Orientation Observation. International Journal on Smart Sensing & Intelligent Systems, *7*(2) (2014).

12. Murakami, Y., Kuno, Y., Shimada, N., Shirai, Y.: Collision avoidance by observing pedestrians' faces for intelligent wheelchairs. Journal of the Robotics Society of Japan, 20(2), pp.206-213 (2002).
13. Hall, E.T.: Proxemics. Current Anthropology, 9(2-3), 83-108 (1968).
14. Human pose estimation using OpenPose with TensorFlow, https://arvrjourney.com/human-pose-estimation-using-openpose-with-tensorflow-part-1-7dd4ca5c8027, last accessed 2019/1/22
15. COCO dataset, http://cocodataset.org, last accessed 2018/10/5.
16. Miller, D.P., Slack, M.G.: Design and testing of a low-cost robotic wheelchair prototype. In: Autonomous robots, *2*(1), pp.77-88(1995).
17. Simpson, R.C.: Smart wheelchairs: A literature review. Journal of rehabilitation research and development, 42(4), p.423(2005).
18. L. Fehr, W.E. Langbein, S.B. Skaar,: Adequacy of power wheelchair control interfaces for persons with severe disabilities: a clinical survey, Journal of Rehabilitation Research and Development, 37 (3), pp. 353–360, June, 2000.
19. Timothy Bourke,: Development Of A Robotic Wheelchair, http://www.tbrk.org/papers/uowhonours01.pdf, November, 2001, Accessed date- November 20, 2017.
20. S. Al Mamun, R. Suzuki, A. Lam, Y. Kobayashi, Y. Kuno,: Terrain Recognition for Smart Wheelchair, Volume 9773 of the series Lecture Notes in Computer Science, Intelligent Computing Methodologies, pp 461-470,ICIC 2016.
21. Sharma, V. Simpson, R.C. Lopresti, E.F. Schmeler,: M. Clinical evaluation of semiautonomous smart wheelchair architecture (Drive-Safe System) with visually impaired individuals, J. Rehabil. Res. Dev., Vol 49(1), pp 35–50, 2012.
22. M. Burhanpurkar, M. Labbé, X. Gong, C. Guan, F. Michaud, J. Kelly,: Cheap or Robust? The Practical Realization of Self-Driving Wheelchair Technology, In Proceedings of the IEEE International Conference on Rehabilitation Robotics, July, 2017.
23. S. Al Mamun, A. Lam, Y. Kobayashi, Y. Kuno.: Single Laser Bidirectional Sensing for Robotic Wheelchair Step Detection and Measurement, Volume 10363 of the series Lecture Notes in Computer Science, Intelligent Computing Methodologies, pp 37-47, ICIC 2017.
24. Krizhevsky, I. Sutskever, and G. E. Hinton,: Imagenet classification with deep convolutional neural networks. In NIPS, volume 1, page 4, 2012.
25. Hinton G, Deng Li, Yu D,: Deep neural networks for acoustic modeling in speech recognition: the shared views of four research groups, IEEE Signal Processing Magazine, Volume 29, Issue 6, pp 82-97, Nov, 2012.
26. F. Abuzaid,: Optimizing CPU Performance for Convolutional Neural Networks, http://cs231n.stanford.edu/reports/2015/pdfs/fabuzaid_final_report.pdf, 2015, Accessed date- November 22, 2017.

27. Zero to Hero: Guide to Object Detection using Deep Learning: Faster R-CNN,YOLO,SSD, http://cv-tricks.com/object-detection/faster-r-cnn-yolo-ssd, Accessed date- January 2, 2018.
28. J. Redmon, S. Divvala, R. Girshick, and A. Farhadi,: You only look once: Unified, real-time object detection, CVPR, June, 2016.
29. M. Ishikawa et al. , Travel Device, Patent Number: WO2016/006248 A1, 2016.
30. G. Ning,: YOLO CPU Running Time Reduction: Basic Knowledge and Strategies. http://guanghan.info/blog/en/my-works/yolo-cpu-running-time-reduction-basic-knowledge-and-strategies, March 7, 2016, Accessed date- January 10, 2018.
31. F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, K. Keutzer,: Squeezenet: Alexnet-Level Accuracy With 50x Fewer Parameters And <0.5mb Model Size, In ICLR, Nov 4, 2016.
32. X. Zhao, P. Liu, M. Zhang: A novel line detection algorithm in images based on improved Hough Transform and wavelet lifting transform, IEEE International Conference on ICITIS, December 17-19, 2010.
33. R. A. Guler, N. Neverova, I. Kokkinos,: DensePose: Dense Human Pose Estimation In The Wild, Computer Vision and Pattern Recognition (CVPR), June 18, 2018.
34. M. Peng, C. Wang, T. Chen, G. Liu, Nirfacenet,: A convolutional neural network for near-infrared face identification, Information pp.61, Vol.7, April, 2016.
35. H. Fang, S. Xie, C. Lu,: RMPE: Regional multi-person pose estimation, IEEE International Conference on Computer Vision (ICCV), pp. 2353-2362, 2017.
36. Z. Cao, T. Simon, S.-E. Wei et al., "Realtime multi-person 2d pose estimation using part affinity fields", Computer Vision and Pattern Recognition, 2017.