Bachelor of Engineering with Honours in Mechanical Design and Manufacturing Engineering

MME3191 Capstone Project AY2020/21

Final Report

Machine Vision for waiter robots in smart restaurants

Date of Submission: 30 July 2021

Name	Quek Jun Liang, Justin
Admin No.	1801029

Blank Page

SINGAPORE INSTITUTE OF TECHNOLOGY – NEWCASTLE UNIVERSITY
MME3191 Capstone Project Report Submission Form
Declaration of Authorship

Name of Student: Quek Jun Liang Justin		Student ID Number: 1801029	
I am submitting the report/thesis as: ☑ an individual work			
Degree: Mechanical Design and Manufacturing Engineering			
Capstone Project Title: Machine Vision for waiter robots in smart restaurants			
Faculty Supervisor	: Michael Wai Shing LAU	Organization	: NU
Industry Supervisor (*if applicable*)	: nil	Organization	: nil

I hereby confirm that:

1. this work was done wholly or mainly while in candidature for the SIT-NU joint degree programme;
2. where any part of this capstone project has been previously submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated;
3. I have acknowledged the use of all resources in the preparation of this thesis / report;
4. the thesis / report contains/does not contain* Company's proprietary and/or confidential information. I have sought approval from my Industry Supervisor for the use of the Company's information (if any) for my capstone project;
5. the work was conducted in accordance with the Research Integrity Policy and ethics standards of SIT and that the research data are presented honestly and without prejudice. The SIT Institutional Review Board (IRB) approval number is _____ (where applicable);
6. I have read and understood the University's definition of plagiarism as stated in the SIT Academic Policy Section 14: Academic Integrity

   Plagiarism is the copying, using or passing off of another's work as one's own work without giving credit to the author or originator, and also includes self-plagiarism. For example, reusing, wholly or partially, one's previous work in another context without referencing its previous use.

7. the thesis / report has been checked by Turnitin. I attach the signed Turnitin Originality Report.

Signature of Student

29/07/2021

Date

Acknowledgment

I would like to express my gratitude and appreciation to my supervisor, Dr. Michael Lau and Dr. Edwin Foo for their continuous support and encouragement. Their guidance helped me in all the time of research and experiment, which lead to a conclusive result. Thank you for allowing me to take on this project and grow as an engineer.

I would also like to express my gratitude towards A/Prof Choong Zi Jie for sharing his knowledge on object detection and providing me with the ultrasonic sensor for my experiment. Thank you for your valuable guidance.

Blank Page

Abstract

In this report, Covid-19 has cause restaurant to rely on robots and automation to cut cost while adhering to pandemic safety measures. However, current waiter robots are not that ready for the industry. Unable to avoid dynamic obstacle such as unpredictable human movement has cause the robot to collide and fail as a waiter. In addition, these robots are unable to serve dish without the help of human being.

To improve the current version of waiter robot, real-time object detection will be implemented. YOLO version 4 will be use as the detector's algorithm due to its community, popularity, and performance. The hardware will be Jetson Nano from Nvidia and a Logitech C525 as the camera. To detect object and avoid it successfully, the segment will be split into 2-part, detection, and angular positioning.

For detection, 3 customed trained model were generated, Human, Gender and Wheelchair with Crutches. Each classes have 100 dataset and 10% of the dataset as a validation set. The performance of all 3 model score above 75% mAP for 50% IoU, which is a decent performance. Followed by angular position of the object, using the camera AOV and the pixel density of the image to triangulate the angle with respect to the camera. This angle is then used to activate the corresponding ultrasonic sensor as a proof.

The second part of the report targets the issue on serving dishes without human inference. Using fiducial marker such as Apriltag, the robot was able to identify its location with minimum error. Furthermore, the marker can be used as a alignment or docking tool. By using far object appear smaller than nearer object theory, the robot was able to differential the tilted direction or when its centred.

In conclusion the report was a proof of concept to use object detection as a form of obstacle avoidance. In addition, the fiducial marker was able to predict the direction of tilt. However, it is not commercially ready as the platform, Nvidia Jetson Nano is under power for real-time operation.

Table of Contents

Chapter 1 Introduction

Today, the food and beverages industry are greatly affected by covid-19. With the implementation of new safety measures such as distancing, the industries face difficulties to stay afloat with the new rules.[1] The profit margins would be greatly affected due to limited customer being serve due to covid-19. To keep a competitive edge and the survivability of the restaurants, owners are force to reduce operators such as waiters.[2]

Due to covid-19, restaurants are looking for ways to boost the economic while adhering to the covid-19 regulations. This resulted in heavily reliant on automation such as robots, which will be a key to safely eating out in a restaurant.

1.1 Waiter robots

Compared to human waiters, waiter robot does not need breaks, leaves or even human error. This could offer a consistent experience for the diners. With robot, labour costs are kept minimum as they run on electricity. In China, robot waiter is replacing human due to cheaper alternative. In additional robot are immune to covid-19 virus and do not falling sick.[3]

However, waiter robots have its own cons. Many waiter robots uses sensors like infrared and ultrasonic sensor to detect obstacle.[4] This simple technique may work on stationary object, avoiding stationary object with ease. But for a dynamic object similar to human, the robot sometime fails to detect and often collide. Quoting an article about waiter robots in Guangzhou, China, waitering involve dynamic interaction with humans. The robot may seem to work with physical non-moving object, but an object like human that will move without warning pose an issue for the robot.[5]

Apart from avoiding obstacle, waiter robots have issue serving the diners. Often waiter robot navigates to the target table through the help of line following technique or estimation with odometry or dead-reckoning. This technique only gets the robot near the table and require a human to collect the dishes. This is due to robot unable to align or dock onto the table, allowing the robot to serve the dishes. Without the robot or human waiter to serve the dishes, it would defeat the purpose of dining in a restaurant.[6]

1.2 Machine Vision System

In this project, waiter robot will be developed, by implementing machine vision it will allow the waiter robot to navigate to its destination along the restaurant. Machine vision enables the computers to perceive the environment. Multiple sensors such as cameras, sent images or data to the computer or robot controller for instruction processing. By digitisation of the images, the computer can process the information into some type of action.[7]

Machine vision system uses computer with the help of sensor from the robot for viewing and recognizing of an object. Machine vision can also be used in different type of industrial processes, such as object or pattern recognition. This allows food and beverage (F&B) industries to implement machine vision system to their business, which result in better efficient and effectiveness.[7]

Figure 1: Haidilao waiter robot

An example in Haidilao Beijing, a smart restaurant which uses robot to deliver meals. It took around 3 minutes from the time an order has been placed to the moment the dishes arrive at the correct table. In figure 1 above, cameras and sensors enable waiter robot are used to deliver dishes to the customer. The robot uses machine vision to understand the restaurant's layout and real-time sensor for avoiding human or obstacle. This removes the need for human waiter to deliver and serve the customer, saving labour cost.[8]

1.3 Problem statement

In the world of Covid-19, the food and beverage industry implementing automation such as waiter robots to reduce operators. Current waiter robots are not great at avoiding dynamic object and serving the customer. To improve the versatility of the waiter robot, it should be able to avoid dynamic obstacle such as human at ease. The robot will be able to align or dock

onto the table for ease food transfer to the table. By implementing these, the robot will be more independent and be more aware of its surrounding.

Chapter 2. Literature Review

The literature review consists of the categorisation of machine vision tasks. The various object detection algorithms, and depth perception using for machine vision. This researched information is deemed useful to aid in the execution of the project.

2.1 Categorisation of Machine Vision Tasks

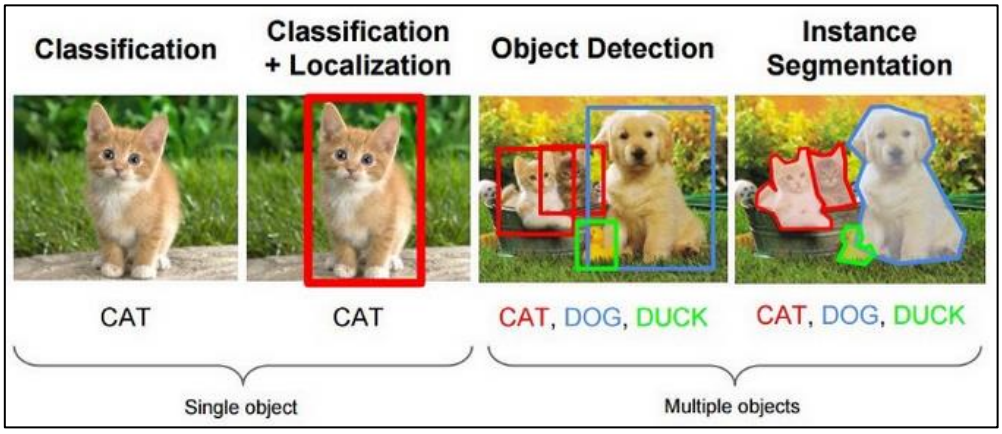


Figure 2: Categorisation[9]

Machine vision uses image and pattern mappings to replicate human complex vision system, which enables machine to detect and identify objects in videos or images. It was made possible with the advancement of Artificial Intelligence (AI), Deep learning and Neural networks.[9]

In this project, the application involves recognition of object within a restaurant. Taking an example from Figure 2, it illustrates the categorisation of the common tasks done by machine vision. An Image Classification uses algorithm to look at the image and classifies the object base on the machine data base. Such classification is typically solve using Convolutional Neural Nets (CNNs). Object classification and localisation not only detect the category of an object, but also the location of an object by drawing bounding box around it. These 2 classifications only classify and detect single object. As compared to previous 2 classification, object detection deals with multiple object classification and its localisation within an image. In the figure 2, object detection of the 3 animals where done, even with overlapping or intersection.[10]

Instance segmentation algorithms is the extraction of an actual object from the image. Similarly, to object detection, both are able to detect multiple objects. However, instead of drawing bounding box, instance segmentation draws a bounding outline of the actual object shape.[11] In some industries, object detection algorithms like YOLO that uses bounding boxes to indicate the object are not sufficient. Image segmentation allows detailed information of an object's shape which provide higher accuracy and reduce uncertainty of the object location.[12]

2.2 Convolution Neural Networks (CNN)

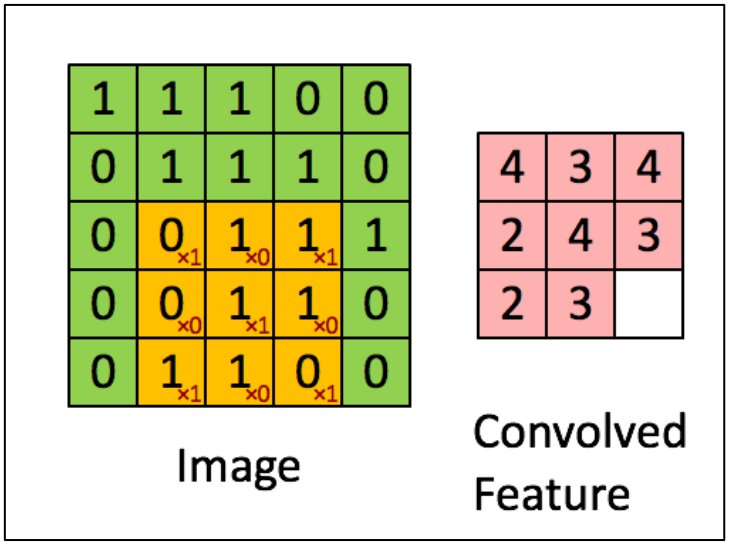


Figure 3: Pixels of an image[13]

CNN is the key foundation of most machine vision application. It transforms input images into outputs through the convolution operation. In convolution layer, figure 3 shows an example of how computer perceive images in greyscale, 1 representing white pixel and 0 representing black pixel. Filter matrix can perform operation such as sharpen, blur, edge detection. By convolution operation, the image matrix with the yellow 3x3 filter matrix, a convolved feature will be output for recognising of unique or special feature of the image.[13]

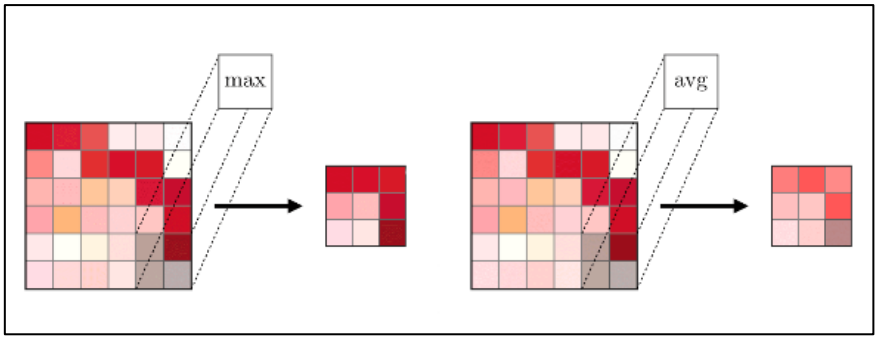


Figure 4: Max pooling and Average pooling[14]

After the convolution layer, pooling layer is added to the next stage. It is usually a down sampling operation, which reduces the pixel count of the image. As shown in figure 4, there are 2 type of pooling, Max pooling and Average pooling. Max pooling operation selects the maximum value of the viewing region, while Average pooling averages the value of the viewing region. [14]

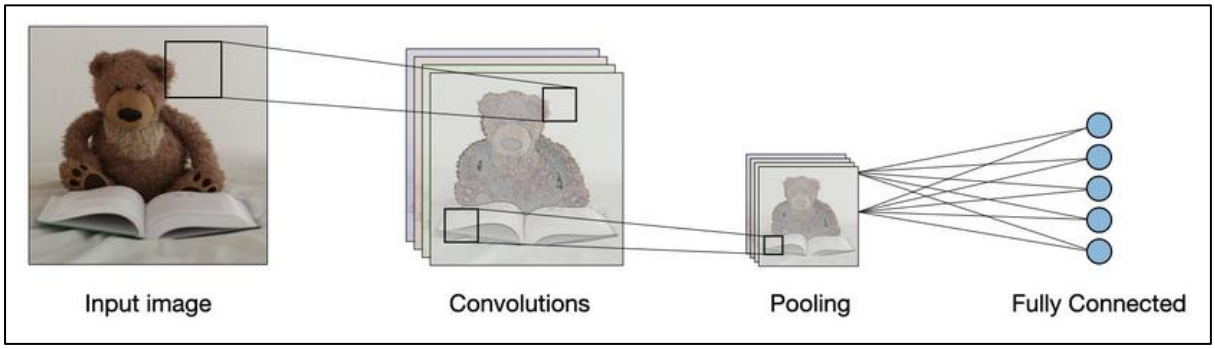


Figure 5: Overview of CNN[14]

Figure 5 shows an overview of the process of convolutional neural network. It ends of at fully connected, right after pooling layer. This layer consists of flattened input from previous layer and eventually generate a result through the nodes.[14]

2.3 Object detection algorithm

In this section, a brief explanation of the different object detection technique used.

2.3.1 Regional Convolution Neural Networks (R-CNN)

In CNN, bounding box scan through entire image and apply convolutions and pooling to extract features. This resulted in high computational resources and time to complete the object detection. R-CNN solution was to proposal regions that are likely to have object to be fed into CNN for features extraction. The proposed method selects a hand full of around 2000 regions from the image, which are also known as the region proposals.[15]

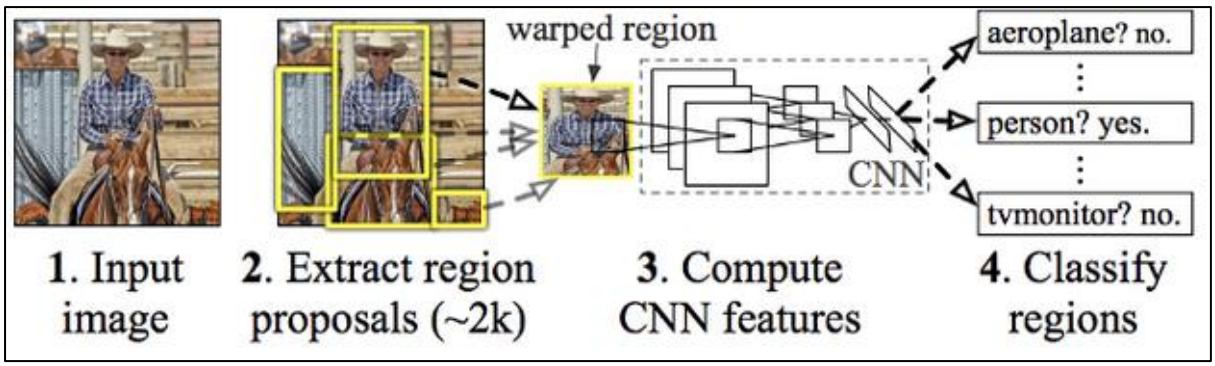


Figure 6: R-CNN[15]

Figure 6 illustrate the process flow of R-CNN. Extraction of region proposals removes regions that do not have any object for detection, reducing the amount of region needed for computation. These set of proposed regions are fed into the CNN, acting as a feature extractor. The output of the data will then be able to classify regions. Although it reduces the regions to 2000 per image, training the network would take huge amount of time. Real time implementation is impossible as it takes around 47 seconds per image. The proposed region for selective computation are generated by a fixed algorithm. This imply that no machine learning is being applied and could lead to generation of bad candidate of region proposals.[15]

2.3.2 Histogram of Oriented Gradients (HOG)

Histogram of oriented gradients is a feature descriptor it uses computer vision and image processing to detect object. The technique takes part of an image, to counts the occurrences of the gradient orientation. This is similar to edge orientation histograms descriptor; HOG uses the histogram of the gradients to determine part of the image's features.[16]

2.3.3 Single Shot Multibox Detector (SSD)

Single shot multibox detector is a deep learning model for detecting object from picture or video. SSD usually has 2 important components, Backbone model and SSD head. Backbone model are pre-trained network convolutional layer, which acts like a feature extractor. While SSD head is another convolutional layer added to the backbone model.[17]

SSD do not use the traditional sliding window technique, but it divides the image into grids. In each grid cell, it is responsible for detecting object within the grid region. If no detection were found, the output for the grid will be "0". Multiple detection within a grid can be solve by assigning anchors.

2.3.4 You Only Look Once (YOLO)

YOLO is one of the famous regression-based algorithms. YOLO is a type of transfer learning, it was pre-trained on a large dataset to solve a particular task. By changing layers within the CNN and training on a different, smaller dataset to solve. This removes the need to train from scratch and a lot of time and computing power.[18] Unlike R-CNN, YOLO do not select region of interest or any part with high probabilities that contain object in the image.[15] These

algorithms predict the classes and generate bounding boxes for the entire image. These allow faster detection than the previous algorithms.[11]

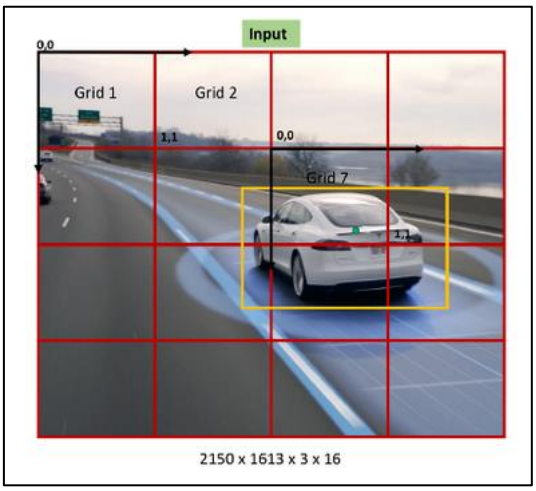


Figure 7: Bounding Boxes[10]

Figure 7 shows how YOLO algorithm works. By dividing the image supplied through the camera into grids. In each grid, localisation and classification algorithm is implemented to detect and provide probability values.[10] YOLO allow a faster detection than other object detection algorithm; YOLO operates at 45 frames per second as compare to R-CNN which operate 50 second per image.[15]

YOLO might not be a perfect solution to all our machine vision idea. It does have its cons, therefore some techniques are use to support YOLO in achieving the end goals. To facilitates the finding of the exact boundaries of an object, non-max suppression is being used.[13] This techniques removes bounding boxes with low probability among the high probability bounding boxes.[10]

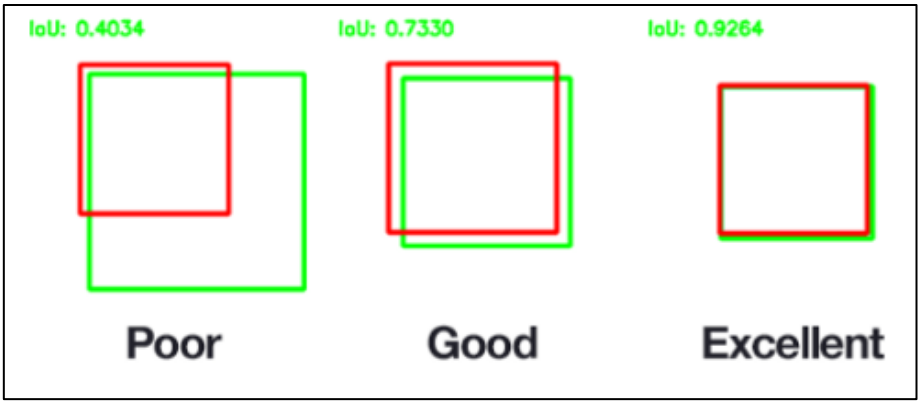


Figure 8:Intersection over union[13]

To achieve this outcome, intersection over union (IoU) are being used. In figure 8 shows how IoU determine the probability of an object containing in the boundary box. The formula is to find the ratio of intersection area of these 2 boxes to the total area. With IoU > 0.5, it will determine that the object is within the box.[13]

Another limitation YOLO encounter was the issue of detecting multiple objects within the same grid. Generally, it can be solved by generating smaller grid size. However, it is not full proof, as cases where objects are very close like a flock of birds or overlapping can still fail.[10]

Figure 9: Anchor box in action

In the figure 9 above, there is person overlapping the car. These 2 objects have its centre point of the bounding boxes and grid cell to be the same. Due to the algorithm, the grid cell has to decide on classifying as a car or the person. With anchor boxes, it is possible to resize multiple grid cell and associate multiple objects with each grid cell. Anchor box will redefine the aspect ratio of the grid cell to fit nicely with the actual object size. After determining these anchor boxes, for each object it will pass through another non-max suppression to remove any bounding boxes with low probability.[19]

2.3 Depth Perception for Machine Vision

Calculating the distance with relation to the machine camera is important for 3D mapping. It enables the waiter robot to understand its surrounding. By only implementing object detection in a 2D image, the object's actual size or distance away from the waiter robot cannot be determine. [20]

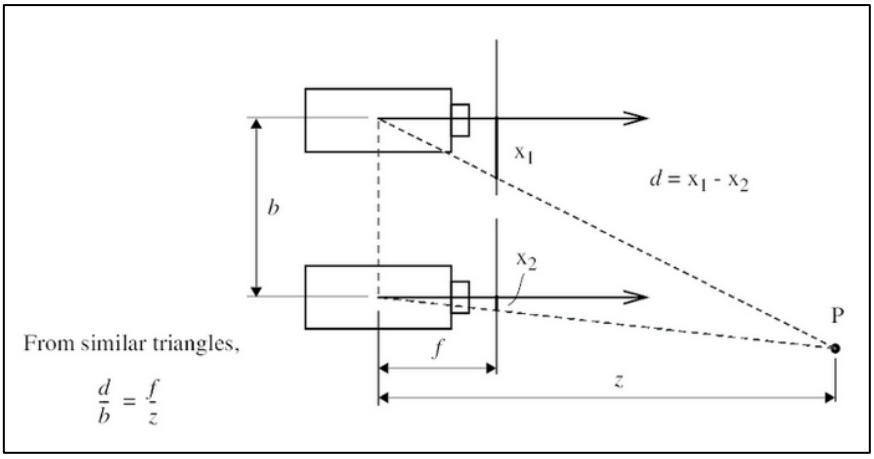


Figure 10: Stereo camera geometry[20]

The proposed idea for this project to determines its depth is to use stereo camera. This involves the concept of triangulation and stereo matching. Figure 10 show a simple example of how distance can be calculated. By matching pixel with the correspondences between the 2 images, the distance can be calculated by finding the disparity, d. By using the formula in the figure 10, we can calculate and obtain the depth (z) from disparity.

Chapter 3 Methodology

In this section, the processes to understanding and application of machine vision will be stated.

3.1 Hardware

Here is the list of hardware being used during the project. The purpose and reasoning for each hardware will also be included.

3.1.1 Nvidia Jetson Module

Figure 11: Nvidia Jetson Nano

In figure 11 shows a Nvidia Jetson Nano. It is a small yet powerful computer that requires as low as 5 watts. It is design to run neural networks such as, object detection and speech processing. Compared to Nvidia jetson, the current waiter robot's processing unit do not have

any graphical processing unit (GPU). Without a GPU, it is inefficient to train a neural network with large dataset.[21]

For this project, Jetson Nano has been used due to budget limit. Having a small form factor and high power-efficient, it is easier to integrated to the current waiter robot. Jetson Nano will be using the JetPack package provided by Nvidia. This package includes Ubuntu 18.04 version, Linux kernel 4.9 and Nvidia drivers that are required.

3.1.2 Camera

Figure 12: IMX219-83 stereo camera

A Stereo Camera was used for object detection. The camera module has a built-in accelerometer, gyroscope, and magnetometer. Being a stereo camera, it has 2 cameras for depth perception. Each camera has 8 megapixels with 3280 by 2464 resolution with a field of view of 83.

Figure 13: Logitech C525

Another camera used for object detection is from Logitech C525 webcam. The camera has 8 megapixels and up to 1280 by 720 video resolution. As compared to stereo camera, Logitech C525 webcam was selected for its low resolution. A low resolution was preferring was due to the performance limit of the Jetson Nano.

Running object detection algorithm such as YOLO resolution of the video or picture from the camera can affect the speed of detection rate. With smaller resolution, there will be lesser data

on each frame, thus improving the speed of the detection rate. Another solution for allowing higher resolution is to purchase higher GPU performance such as Jetson AGX Xavier.

3.2 YOLO object detection

In this project, YOLO detection system was selected to facilitate the recognition of object. As compared to other detectors, YOLO is extremely fast and accurate. In additional, the accuracy of the trained model can be increase by changing its resolution. This would result in increase of time take per detection or frame.

Not only being the top few detectors used, in terms of performances. The system is open source and have a large community sharing about it. This could help to troubleshoot errors occurred in this project.

In this section, the system and file used will be downloaded from "darknet" (https://github.com/AlexeyAB/darknet). The program language used will be mostly C++ and version 4 of the YOLO detector will be used. By using Jetson Nano installed with Linux operating system, the codes can be executed through "Terminal".

3.3 Pre-trained model

Before training custom model for our project purpose, a demo was done to confirm the functionality of the file used and Jetson Nano performance issue. At the same time, this step can help understand what the requirement are to run and detect real-time object.

There are 2 important file that are require before the demo, configuration file (.cfg) and weights (.weights). Configuration file are the parameters for the neural networks, this configuration can affect the performance of the detection. Similarly, the configuration file can also affect the trained model. Parameters for data augmentation are located under the "training" section.

Base on the configuration parameters, the training model can be trained into weights. For this part of the project, a pre-trained model will be downloaded. The version of this file will be a scale down version call YOLO tiny. This would allow faster training and detection rate due to lesser convolutional layers compared to YOLO. With a simpler neural network structure, it is more feasible for deploying onto Jetson Nano.

3.4 Custom trained model

After verifying the performance of YOLO tiny, a custom model will be trained for the purpose of waiter robot. 3 different model has been trained, human, gender, and mobility aids such as wheelchair and crutches.

3.4.1 Preparing dataset

The first step for this section is to generate photos for training. A simple python script was used to download from Google and social media. For each object classes, 100 images were selected and verified if it's the correct class. The dataset may seem insufficient, but for each model the number of unique classes are at most 3. Nevertheless, having more datasets will yield better result. Furthermore, with more unique classes or complex object would need more diversity of data.

After obtaining the dataset, each image has to be mark with boundary boxes of the object and create annotation files. By using the repository "Yolo mark", the boxes can be drawn onto the image and annotation will be generated. The annotation files contain class name and the X and Y axis of the boundary box.

Object class	X centre	Y centre	Width	Height
0 to (classes – 1)	$\frac{\lvert x \rvert}{image\ width}$	$\frac{\lvert y \rvert}{image\ width}$	$\frac{\lvert width \rvert}{image\ width}$	$\frac{\lvert height \rvert}{image\ height}$

Table 1: Annotation code format:

<0> <0.687109> <0.379167> <0.255469> <0.158333>

Figure 14: Example of annotation code

The above example shows the value store within each annotation files. The table 1 explain the value of each position. Object class are the name represented by number. X and Y centre are values calculated based on the centre of annotated box's coordinates. While the width and height are the size of the annotated box.

3.4.2 Setting the parameters

In this segment, the parameters are being modified in preparation for training new models. Here are the list of parameter that are adjusted with its purpose. Under the section "Training", the

"batch" parameter indicates the number of images will be process within 1 iteration. The purpose is to generalise the dataset by having different batches trained with every iteration. Therefore, batch size and dataset should be different.

Unlike "Batch", the parameter "Subdivision" represents the number of sub-divided batches. After every subdivision are done processing within a batch, the iteration complete. This setting affects the training time. Unlike high-end GPU, Jetson Nano has low graphical memory which resulted in small subdivision and higher process time.

$$\underset{\text{batches}}{\text{Max}}\ (\text{min}) = \text{classes} \times 2000$$

$$\underset{\text{batches}}{\text{max}} \geq \text{Number of image}$$

For "max_batches" the rule of thumb is to follow the equations above as a guide. This indicates the number of iterations and will contribute to the training time. If the number of images is greater than the calculated value, a number greater than or equal to the number of images.

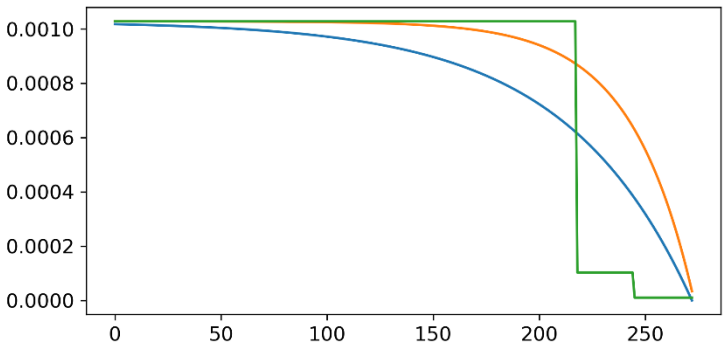


Figure 15: Example of Learning rate Vs Iteration

$$Current\ learning\ rate = learning\ rate \times scales$$

Figure 16: Learning rate

The 15 figure shows an example of different type of learning rates, the X axis represent the number of iteration and Y axis represent the learning rate (LR). The green line indicates a decreasing learning rate by step and rest are decreasing by polynomial. Learning rate controls the speed of the model learns. With a large learning rate, the model will learn fast but may result in large weights changes and affecting the performance. Furthermore, a small learning rate maybe overfitting or even stuck at suboptimal model. In general, there are no possible

ways to obtain the optimal learning rate for a given model. Instead, a good estimated learning rate can be discovered via trial and error.[22]

Lastly, the "width" and "height" parameters represent the network size. Every image will be resized to pre-set value for training and detection purpose. The effect of resizing the images is to reduce data and allow faster computation. While this may be good for non-high accurate prediction as it will reduce its accuracy in replace for speed. For this project, it will be best to resize into smaller resolution ratio for fast detection rate.

3.5 Evaluating custom model

The evaluating pointer described below, especially mean average precision and F1 score, is used to determine the performance of the model.

3.5.1 Precision and Recall

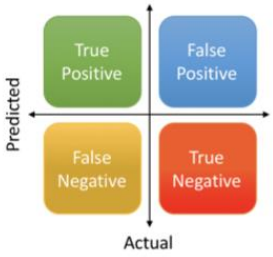


Figure 17: Predicted vs Actual

The figure 17 shows a simple representation of precision and recall. True positive and true negative means the actual and prediction is same. While false positive and false negative means the actual and prediction are not the same. Using the equation 18 and 19 the precision and recall value can be calculated. The higher the value the better the model.

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

Figure 18: Precision equation

$$Recall = \frac{TP}{True\ Positive + False\ Negative}$$

Figure 19: Recall equation

3.5.2 Mean Average Precision (mAP)

$$IoU = \frac{Area\ of\ Overlap}{Area\ of\ Union}$$

Figure 20: IoU equation

Average precision (AP) is the finding the area under the curve of precision and recall. For object detection task such as YOLO, IoU is used to calculate of precision and recall. The mAP will be relatively high at a low IoU threshold. The threshold value determines the overlapping of predicted box and the actual box over the entire area of union. The explain that with a small IoU threshold, small data representing the actual image will deem as true positive. In general, the mAP is solved by calculating individual class's AP and averaging it.

3.5.3 F1 score

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

Figure 21: F1 score equation

F1 score is the weighted average of both precision and recall as shown in equation 21. The value ranging from 0 to 1 with higher score determine a better model performance. Depending on the scenario indicating the importance of precision or recall, the equation can be adjusted to better suit the model.

3.5.4 Frame Per Second (FPS)

In addition to the previous few evaluation points, FPS is one of the importance factors for object detection. The purpose for such detection to for avoid obstacle and prevent collusion. Having high FPS allow the machine to process more frames, which result in faster respond to detection.

In an ideal scenario, waiter robot has to respond to obstacle in real time. Roughly 25 frame per second is the standard for real time, but more is better. The frame rates can be improved by reducing the resolution but sacrifices the mAP of the models. Thus, a balance of these 2 factors has to be made.

3.6 Angular position

After successfully train a custom model, the next step is to determine the object's location with respect to the camera. The camera will act as the waiter robot's eye and will react based on the input. Angle of view (AOV) and the camera input resolution are needed to calculate the estimate location of the object. AOV is define as the maximum angle of an image captured by the camera. The unit is in degree and there are many ways to calculate.

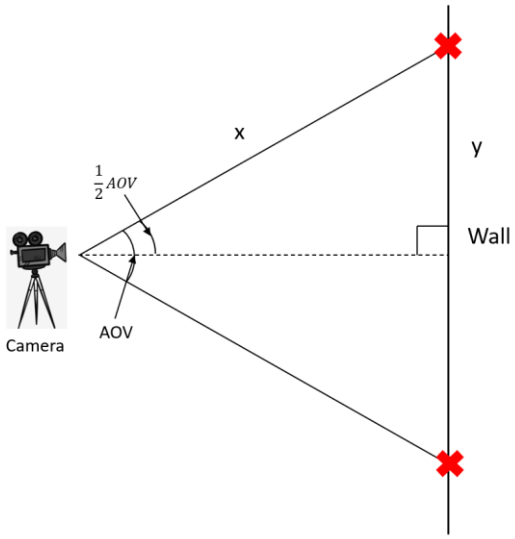


Figure 22: Layout for AOV measurement

The figure 22 shows a simple setup of how AOV is being obtain. The camera is first setup facing a wall. Base on the camera observable image, the left and right most position is mark onto the wall. X is the distance from camera to previously marked left position and Y is the half the distance between the 2 points. Using Trigonometry, AOV of the camera can be calculated.

Figure 23: Example of real-time detection

The figure 23 shows a single frame of real time detection. The camera is detecting at 1280 by 720 pixel resolution. Aside from displaying the object location within an image, the algorithm is able to determine the centre coordinate of the boundary box.

$$\frac{x\ pixel}{AOV} = a$$

Figure 24: Pixel per degree equation

$$\frac{x\ coordinate}{a} = Angular\ position$$

Figure 25: Angular position equation

The equation above is then added into the system to calculate the angular position. X pixel is the image resolution of its x axis, and x coordinate is the centre of the boundary box. With this angle, waiter robot can act or respond to this input.

3.7 Fiducial mark

Figure 26: QR code and Apriltag

This section will address the issues of the waiter robot. At the same time, how fiducial marker can be a proposed solution to improve the capability of the robot. Fiducial markers are pictures place along the working area for camera enabled robot to recognise. QR codes and Apriltag are one of the fiducial markers used by many industries. The figure 26 shows a QR codes and Apriltag, the Apriltag is relatively small as compared to QR codes. This is due to lesser information are being store onto the Apriltag. Placing a large fiducial marker is not viable solution for restaurants, as it will be not aesthetically pleasing for the diners. Therefore, Apriltag was used as an alternative.

3.7.1 Localisation

Localisation is a technique used by robot to triangulate and estimate its opposition with respect to its environment. As of the writing of this report, many waiters robot are having issues with the accuracy of the localization. Since it is being estimated, there is a possibility there are errors in it. With respect to time, the errors will get large, and the waiter robot will be disorientated. Reports have shown by incorporating RFID tag on the robot and working area to reduce such errors.[23] RFID tag may seems to be good. However, Apriltag is a cheaper alternative as it can be replaced easily.

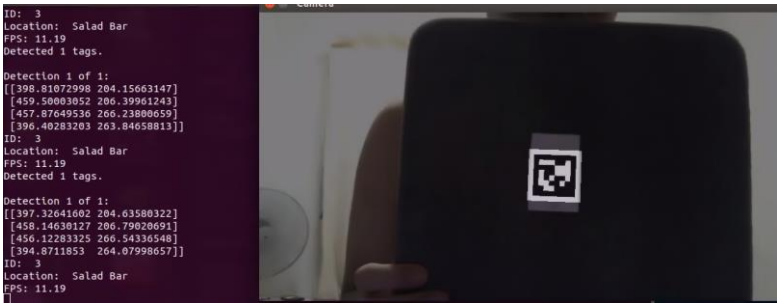


Figure 27: Example of an actual detection

As shown in figure 27, a tag ID was captured by the camera. The information on the tag can be store on it, showing the location. Such information can be customized into coordinates of the locations for localisation purposes.

Figure 28: Apriltag on the floor

The proposed idea was to place these tags on the floor or celling. The white background colour can be change to the floor's colour for better transition and aesthetic. However, changing the colours of the tag will require re-training of the model for detection. The figure 28 shows an example of a 3cm-by-3cm Apriltag on the floor. Damage prevention such as lamination of the Apriltag can be applied to prevent damaging it beyond recognitions.

3.7.2 Docking

The purpose of docking the robot onto the diner's table is to remove the need of human interference, to place dishes onto the table. As of writing of this report, restaurants such as Hai Di Lao (HDL) are using human to place dishes on the tables.

Similar to the previous idea of localisation, the tables can be identified by tags. After identifying the correct table, the tags can be used as a form of alignment of robot to the table.

The detectors are able to generate the 4 corner's coordinates of the tag itself. By using these coordinates, estimated alignment can be done.

Figure 29: Tilted Apriltag to the right

The figure 29 shows an example of the detection system capturing the tag at slanted angle, the arrow shows the direction the tag is facing. The left and right green line correspond to the height of the boundary box. This can be calculated using the coordinates of the 4 corners. The 2 lines have different height, as the nearer line appear longer than the shorter line. Using the theory of far object appear small in a picture, the system can tell it is slanted toward the shorter length of the boundary box.

Chapter 4 Results

The results from all the experiment done will be shown here. It complements the previous segment of Methodology, and support the proposed solution.

4.1 Experimental set-up

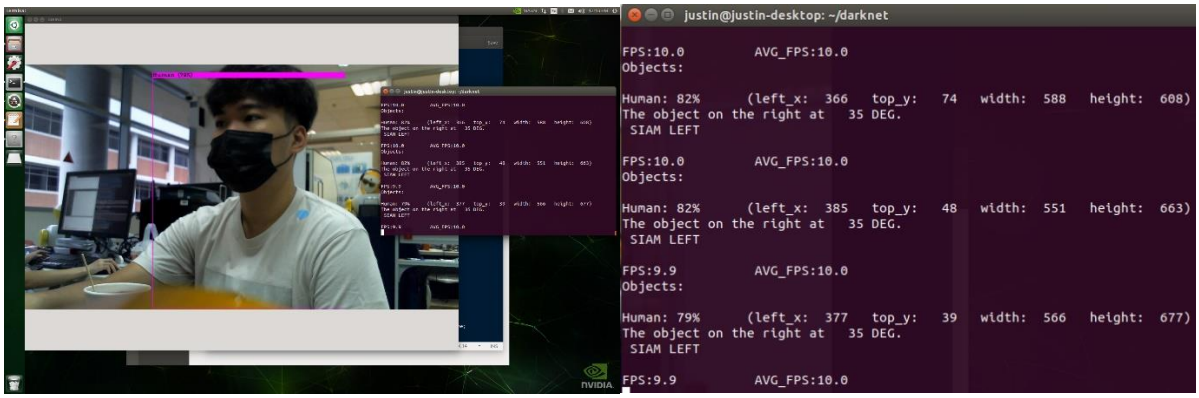


Figure 30: Real-time detection with angular positioning

The first experimental set-up was done to verify the system, both the model and Jetson Nano. In figure 30 the pre trained model was able to detect the human with a confidence of around 80%. The FPS of this model hit an average of 10. This was achieved by adjusting the camera parameters to 1280 by 720 resolution. As the setting was 1920 by 1080 resolution, causing it to achieve as low as 2 FPS which is a poor performance.

4.2 Custom trained model

In this section, customed model will be evaluated by the mAP values. The mAP is supported by the precision and recall , IoU threshold and F1 score. 10 random images per classes will be obtain from Google to test the model. The image use are not same as the training dataset. The evaluated result will roughly gauge the accuracy of the model.

4.2.1 Human model

The figures below show the result of human as the object to be detected. 100 images of human were annotated and process for training.

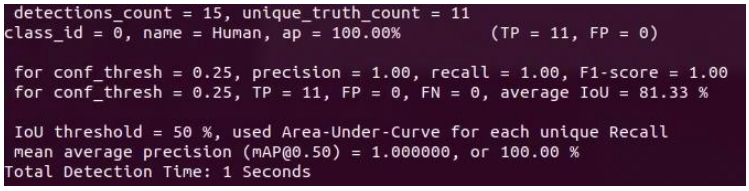


Figure 31: Human model, 50% IoU

The figure 31 are scores result for IoU threshold with 50%. 100% of the images are true positive with more than 50% IoU. Precision , recall and F1 are scored for 1.00. The mean average precision are 100% .

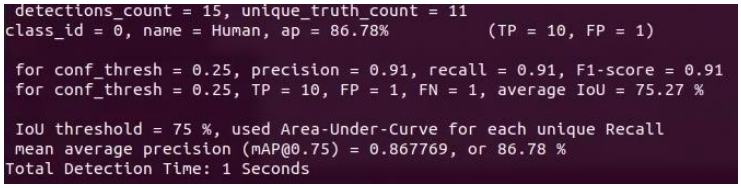


Figure 32: Human model, 75% IoU

By setting IoU threshold to 75%. The precision, recall and F1 score of 0.91 across the board. Leaving the mean average precision was decrease to 86.78%.

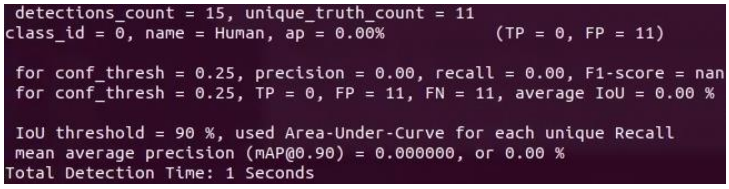


Figure 33: Human model, 90% IoU

With a IoU threshold at 90%, the score for precision, recall, F1 and mean average precision is 0. This show none of the 10 images were recognise as human with a 90%.

4.2.2 Gender model (face)

For the custom model of gender, 3 different classes were trained. To determine the gender of the person, the faces of human were used to train the model. The 3 classes human, female and male, has a combine of 300 image.

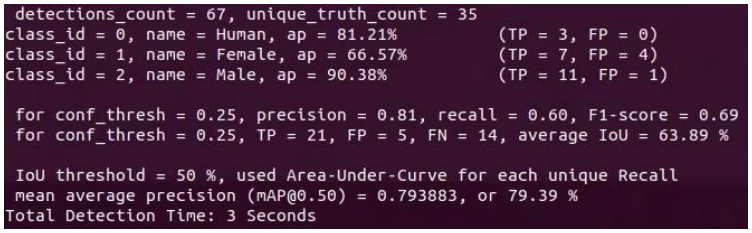


Figure 34: Gender & Human model, 50% IoU

The figure 34 shows the result for IoU threshold of 50%. The precision, recall and F1 score is 0.81, 0.6 and 0.69 respectively. Leaving a mean average precision of 79.39%.

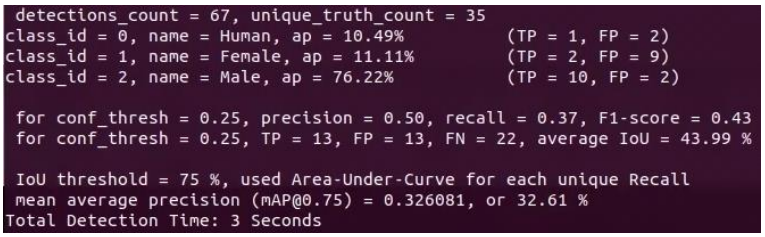


Figure 35: Gender & Human model, 75% IoU

For IoU 75% threshold, the result starts to drop. The precision, recall and F1 scored 0.50, 0.37 and 0.43 respectively. Leaving a mean average precision of 32.61%.

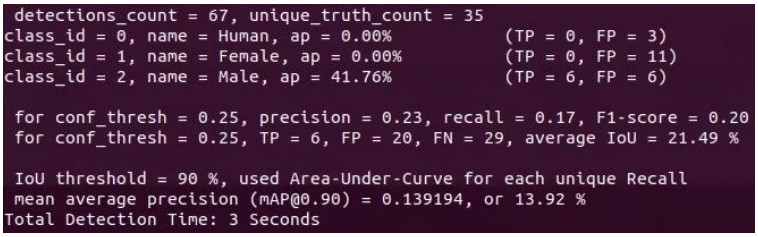


Figure 36: Gender & Human model, 90% IoU

Figure 36 show changes in result for IoU threshold of 90%. Affecting the precision, recall and F1 score to be 0.23, 0.17 and 0.20 respectively. The mean average precision is at 13.92%.

4.2.3 Wheelchair and Crutches (mobility aid)

For this section, 3 classes were trained, human, wheelchair, and crutches. Combining the same dataset for human model, the total dataset are 300 images.

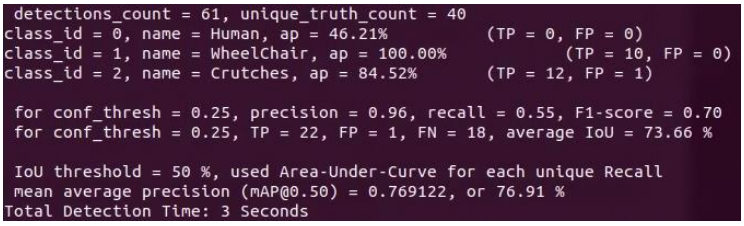


Figure 37: Wheelchair, Crutches & Human, 50% IoU

The figure 37 are the result for IoU threshold at 50%. Precision, recall and F1 scored 0.96, 0.55 and 0.70 respectively. The mean average precision is value at 76.91%

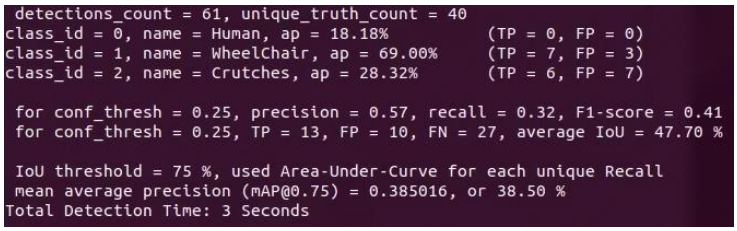


Figure 38: Wheelchair, Crutches & Human, 75% IoU

For IoU threshold 75%. The 3 classes have a precision, recall and F1 score of 0.57, 0.32 and 0.41 respectively. The mean average precision is at 38.50%

```
 detections_count = 61, unique_truth_count = 40
class_id = 0, name = Human, ap = 0.00%           (TP = 0, FP = 0)
class_id = 1, name = WheelChair, ap = 9.00%      (TP = 2, FP = 8)
class_id = 2, name = Crutches, ap = 0.00%        (TP = 0, FP = 13)

 for conf_thresh = 0.25, precision = 0.09, recall = 0.05, F1-score = 0.06
 for conf_thresh = 0.25, TP = 2, FP = 21, FN = 38, average IoU = 8.32 %

 IoU threshold = 90 %, used Area-Under-Curve for each unique Recall
 mean average precision (mAP@0.90) = 0.030000, or 3.00 %
Total Detection Time: 3 Seconds
```

Figure 39: Wheelchair, Crutches & Human, 90% IoU

Finally for 90% IoU threshold, the model has a score below 0.1 for precision, recall and F1. The mean average precision is at 3.00%

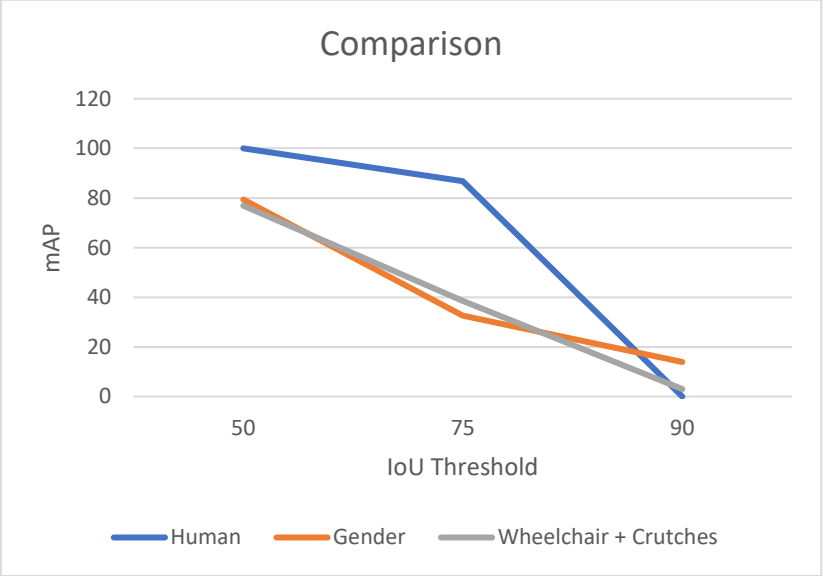


Figure 40: mAP vs IoU Comparison

Base on the 3 different model, the mAP can be plot across the IoU threshold. The figure 40 shows a trend where the threshold increases, the mean average precision score will drop. The IoU threshold are self-determine. Depending on the how accurate a class has to be, the threshold can be tweak for it. Base on this chart, 50% IoU give a satisfactory result, showing good accuracy. However, above 75% IoU the result drops. The possible reason could be insufficient dataset or diversity.

4.3 Angular position

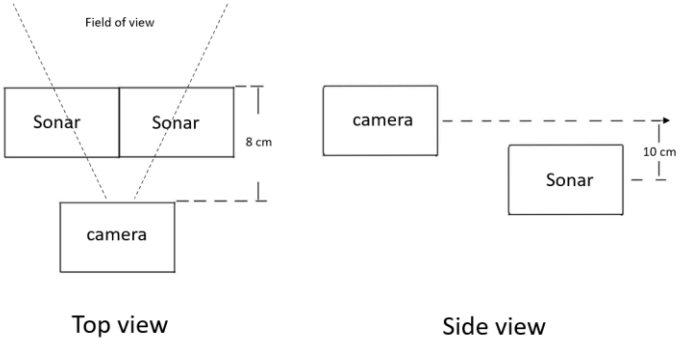
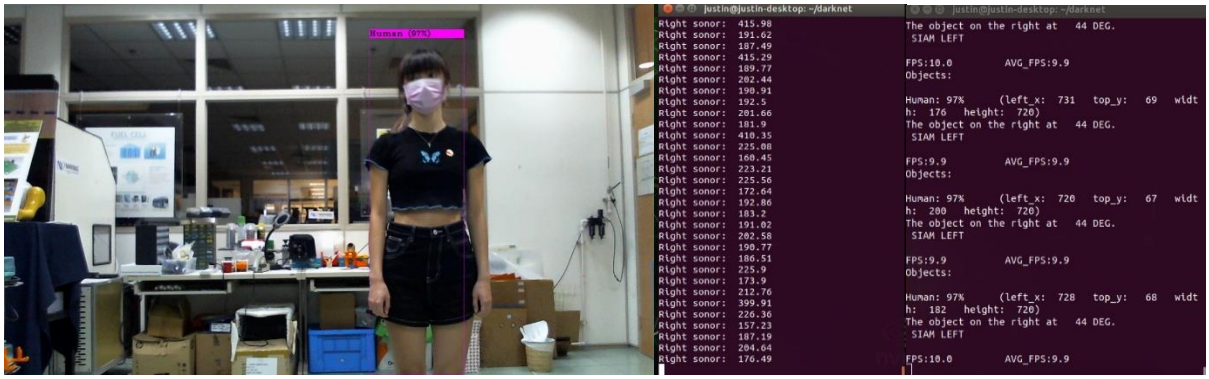


Figure 41: Layout for camera and sonar

The proof the working concept of angular position. The camera will determine the object is on the left or right base on the estimated angular position. After determining the position, left or right ultrasonic sensor will be activated and obtain the distance. The figure 41 shows the top and side view of the camera and ultrasonic sensor setup.

The distance from camera to 100cm, 200cm, and 300cm are mark on the floor. This point is for the reference for placing the object and determine the accuracy of ultrasonic sensor's reading.



at 200cm mark with ref to ?*Figure 42: Real-time detection and ultrasonic sensor*

Figures 42 show an example of the experiment's result, an object standing at 200cm mark at the right side of the image. Based on the camera input, it can be visually verified that the object is at right side of the image. Then a correct corresponding sonar is activated, taking the reading. The reminding result for this experiment will be in appendix.

4.4 Docking

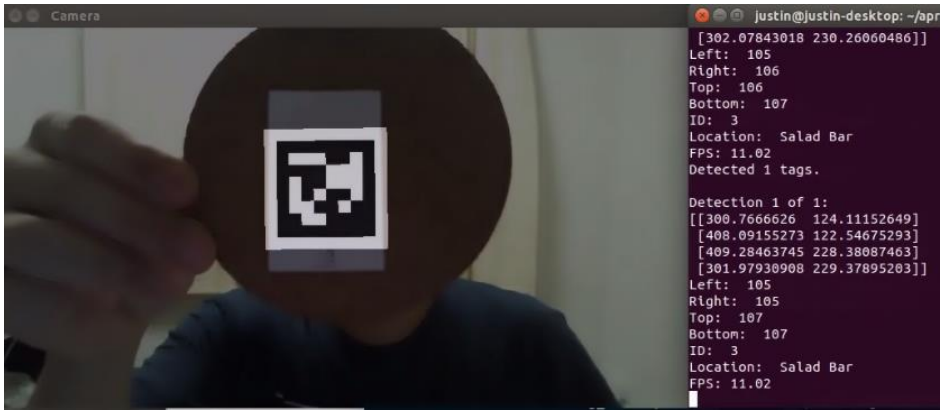


Figure 43: Apriltag centred

To verify the concept of far object appears smaller in an image. The Apriltag will be place on the cardboard to simulate the table position. The detection will calculate the 4 lines, which sum up as a boundary box. After calculating, the slanted direction of the tag can be predicted. In figure 43 the tag is held perpendicular to the camera, which is defined as aligned to the table. Left, right, top and bottom are the parameters for the boundary box with the tag.

Figure 44: Apriltag tilted upward

This figure 44 shows the tag slanted upwards. The top length appear far, therefore top length is smaller than bottom. Hence, it is possible to estimate the alignment base on the length of the boundary box. Figure 45, 46 and 47 shows a different slanted direction, and have similar outcome as figure 44.

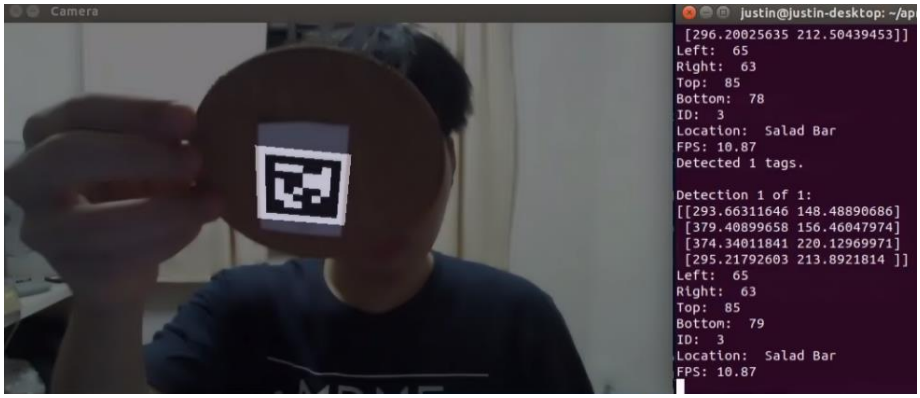


Figure 45: Apriltag tilted downward

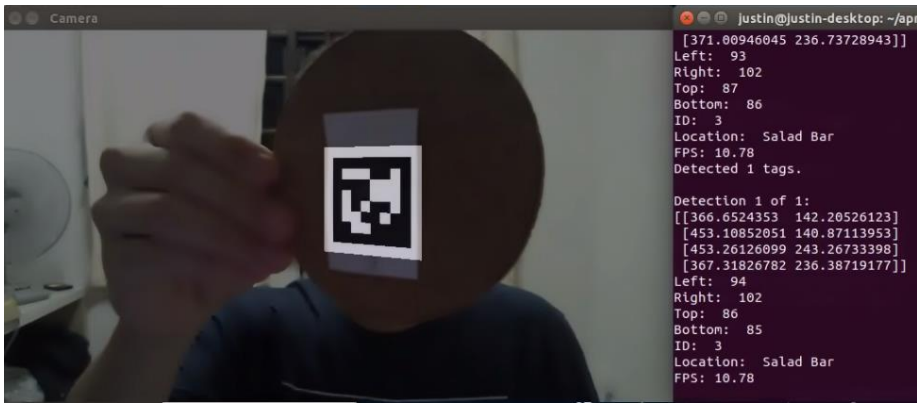


Figure 46: Apriltag tilted leftward

Figure 47: Apriltag tilted rightward

Chapter 5 Discussions

5.1 YOLO experiment

During the training phase, processing 100 image for a single class takes as long as 2 to 3 days. As compared to other training experiment, the hardware are more powerful. Jetson Nano having 128 cores is insufficient to train efficiently as compared to a RTX 2080 ti with 4352 cores.

However, it will not be very feasible to use all high end and powerful component for training and detection. As this will be overly expensive and consume too much power to be deem efficient enough. On the other hand, Nvidia Jetson have many models, featuring low power consumption and small formfactor. The experiments show the Jetson Nano have barely enough computing power to train and run real-time detection. A better solution could be upgrade to a higher model of Nvidia Jetson. Alternatively, the training phase can be done on a full fledge computer with all the bells and whistles.

The amount of image sent for training were certainly not enough. The rule of thumb was to include around 1000 image per class. However, the experiments show that it is possible to train and detect with a limit of 100 image per class. This does not mean the best solution but a proof of concept. The reason for such a low amount of image was due to the low computing power of Jetson Nano. The image quality does also affect the number of images used for training. If this factor were to ignore, the system will crash as there are insufficient video memory to process the large amount and HD images.

By using Nvidia Jetson Nano, the experiment was lengthy, and the trained models were not certain to be use in a simulated restaurant setting. 100 image per class is definitely not enough for actual simulation, more classes should be included into the model.

5.2 Model validation

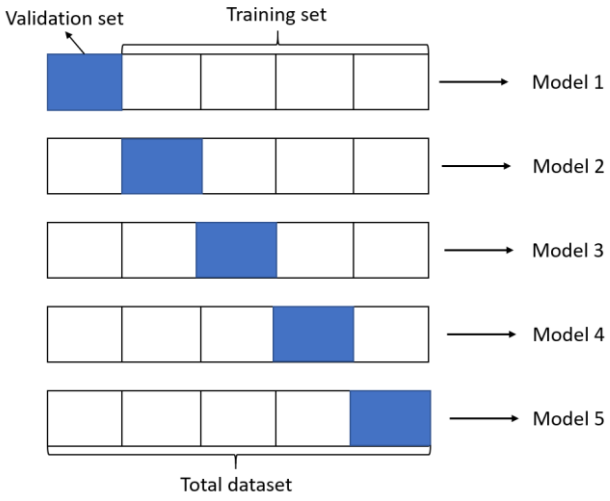


Figure 48: Cross validation

Further validation can be done if there is more computing power with large dataset. One of the techniques was to apply cross validation. The figure 48 shows an example of how cross validation were to be done. In this example 5 set of models were train, dataset was split into 1:4 ratio for validation and training. An important factor is the trained model should not have the image of the validation as it will defeat the purpose.

The goal of cross validation is to verify the model's prediction on new independent data, this will flag out any problem such as overfitting. Cross validation does not increase the accuracy of detection, but to verify if the model's accuracy is accurate.

5.3 Incorporating LiDAR

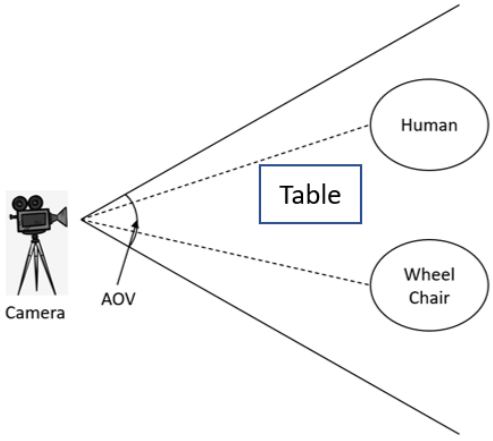


Figure 49: Overview of detection

In experiment for angular positioning of object, the object's angular position was derived with the help of the equation and the AOV of the camera. It then activates the corresponding ultrasonic sensor. Since ultrasonic sensor pings its estimate distance in a cone shape, readings may be not accurate as compared to LiDAR. Example of inaccurate reading, figure 49 shows a table in front of the human. By detecting human, the left ultrasonic senor will read the distance. However, due to the table blocking the line of sight for sonar, the reading will be lesser.

LiDAR uses point location instead of cone for ultrasonic sensor. Using a 360 LiDAR and the angular position given by the camera, the human distance can be estimate. Predicting human's trajectory will be possible, which will help in early warning system.

5.4 Point system for obstacle avoidance

The purpose of this project is to allow the robot to be more aware of it surround, giving faster or earlier respond before collision happens. Avoid stationary object are simple for current waiter robot, but nonstationary object such as human are harder. Humans' mobility is categorised into 2 category, difficult to move and no difficulties.

Factors such as wheelchair and crutches can straight rule out that the robot must avoid the human being. However, body posture, wrinkles, movement speed, and hair colour are some of the factors to determine the level of mobility of the human. With the context of Singapore's diners, the list of possible factors can be added into the point system with different weightage. The end goal of this point system is for the robot to determine which type of obstacle avoidance base on the point system, for example the robot's speed during avoidance.

Chapter 6 Conclusion

In conclusion, the experiment on YOLO algorithm was a proof of concept. The system was able to detect the different type of object at a level of confidences. With the help of YOLO's algorithm, the object angular position can aid in obstacle avoidance. Apriltag experiment shows better alternative for localisation and docking with lesser cost. Using the theory of far object appears small, the alignment to the table can be done. But it is not ready for commercial use and require further experiment and better hardware.

The project can further progress by porting over to a powerful Jetson such as AGX Xavier. On paper, Nvidia AGX Xavier has up to 20 times the performance and 10 times the energy efficiency of Nvidia Jetson Nano while maintaining small form factor and low power. The object detection can be implemented into the waiter robot through Robot Operating System (ROS) as the centralised system. Hence removing the need for extra processing unit.

References

[1] Haas Stacey, Kuehl Eric, R John, Moran, and Venkataraman Kumar, 'How the restaurant industry can thrive in the next normal | McKinsey', May 2020, Accessed: Jul. 14, 2021. [Online]. Available: https://www.mckinsey.com/industries/retail/our-insights/how-restaurants-can-thrive-in-the-next-normal#.

[2] Wu Leslie, 'The Rise Of Restaurant Robots Amidst Pandemic Measures', Sep. 27, 2020.

[3] Vella Heidi, 'Waiting on robots: Can a robot server really replace a human one? - Tech Wire Asia', Sep. 12, 2016.

[4] A. CHEONG, E. FOO, H. GAN, J. CHEN, and M. LAU, 'Development of a Robotics Waiter System for the food and beverage industry', no. October 2015, pp. 21–25, 2015, doi: 10.15224/978-1-63248-066-8-57.

[5] Ackerman Evan, 'Shockingly, Robots Are Really Bad at Waiting Tables - IEEE Spectrum', Apr. 12, 2016.

[6] N. Mishraa, D. Goyal, and A. D. Sharma, 'Issues in Existing Robotic Service in Restaurants and Hotels', *SSRN Electron. J.*, pp. 2016–2019, 2018, doi: 10.2139/ssrn.3166508.

[7] Len Caalderone, 'What is Machine Vision?', *Robotics Tomorrow*, Dec. 17, 2019. https://www.roboticstomorrow.com/article/2019/12/what-is-machine-vision/14548 (accessed Mar. 04, 2021).

[8] 'Beijing Haidilao hotpot: Good reasons to dine in on your China tour', *ChinaTours.com*, Feb. 02, 2021. https://www.chinatours.com/beijing-haidilao-hotpot/ (accessed Mar. 04, 2021).

[9] Ilija Mihajlovic, 'Everything You Ever Wanted To Know About Computer Vision', Apr. 26, 2019. https://towardsdatascience.com/everything-you-ever-wanted-to-know-about-computer-vision-heres-a-look-why-it-s-so-awesome-e8a58dfb641e (accessed Mar. 06, 2021).

[10] Grover Prince, 'Evolution of Object Detection and Localization Algorithms', Feb. 15, 2018. https://towardsdatascience.com/evolution-of-object-detection-and-localization-algorithms-e241021d8bad (accessed Mar. 06, 2021).

[11] Lentin Joseph, 'A Gentle Introduction to YOLO v4 for Object detection', Jan. 05, 2020. https://robocademy.com/2020/05/01/a-gentle-introduction-to-yolo-v4-for-object-detection-in-ubuntu-20-04/#Overview (accessed Mar. 06, 2021).

[12] 'Instance Segmentation with Deep Learning'. https://missinglink.ai/guides/neural-network-concepts/instance-segmentation-deep-learning/ (accessed Mar. 06, 2021).

[13] Zawadazi Jan, 'Convolutional Neural Networks For All', Feb. 07, 2018. https://medium.com/machine-learning-world/convolutional-neural-networks-for-all-part-ii-b4cb41d424fd (accessed Mar. 06, 2021).

[14] Amidi Afshine and Amidi Shervine, 'CS 230 - Convolutional Neural Networks Cheatsheet'. https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-convolutional-neural-networks (accessed Mar. 06, 2021).

[15] Gandhi Rohith, 'R-CNN, Fast R-CNN, Faster R-CNN, YOLO', Jul. 10, 2018. https://towardsdatascience.com/r-cnn-fast-r-cnn-faster-r-cnn-yolo-object-detection-algorithms-36d53571365e (accessed Mar. 06, 2021).

[16] Sharma Nikita, 'Introduction to basic object detection algorithms | by Nikita Sharma | Heartbeat', Dec. 18, 2019. https://heartbeat.fritz.ai/introduction-to-basic-object-detection-algorithms-b77295a95a63 (accessed Jul. 28, 2021).

[17] Hemanthhari, 'Object Detection using Single Shot MultiBox Detection (SSD) and Deep Neural Network (DNN) | by Hemanthhari2000 | featurepreneur | Medium', Apr. 06, 2021. https://medium.com/featurepreneur/object-detection-using-single-shot-

multibox-detection-ssd-and-opencvs-deep-neural-network-dnn-d983e9d52652 (accessed Jul. 28, 2021).
[18] C. Borngrund, U. Bodin, and F. Sandin, 'Machine vision for automation of earth-moving machines: Transfer learning experiments with YOLOv3', Lulea University of Technology, 2019.
[19] Nishad Garima, 'You Only Look Once(YOLO): Implementing YOLO in less than 30 lines of Python Code', Mar. 02, 2019. https://medium.com/analytics-vidhya/you-only-look-once-yolo-implementing-yolo-in-less-than-30-lines-of-python-code-97fb9835bfd2 (accessed Mar. 07, 2021).
[20] Tan Daryl, 'Depth Estimation: Basics and Intuition', Feb. 14, 2020. https://towardsdatascience.com/depth-estimation-1-basics-and-intuition-86f2c9538cd1 (accessed Mar. 07, 2021).
[21] Shah Shachi, 'Do we really need GPU for Deep Learning? - CPU vs GPU | by Shachi Shah | Medium', Mar. 26, 2018. https://medium.com/@shachishah.ce/do-we-really-need-gpu-for-deep-learning-47042c02efe2 (accessed Jul. 17, 2021).
[22] Brownlee Jason, 'How to Configure the Learning Rate When Training Deep Learning Neural Networks', Jan. 23, 2019. https://machinelearningmastery.com/learning-rate-for-deep-learning-neural-networks/ (accessed Jul. 26, 2021).
[23] Y. Qing-xiao, Y. Can, F. Zhuang, and Z. Yan-zheng, 'Research of the Localization of Restaurant Service Robot':, *https://doi.org/10.5772/9706*, vol. 7, no. 3, pp. 227–238, Sep. 2010, doi: 10.5772/9706.

Appendix

Appendix A: Gantt Chart

Project Activity	Capstone Project Timeline (Weeks)																								
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
	Dates																									
	1 Feb - 7 Feb	8 Feb - 14 Feb	15 Feb - 21 Feb	22 Feb - 28 Feb	1 Mar - 7 Mar	8 Mar - 14 Mar	15 Mar - 21 Mar	22 Mar - 28 Mar	29 Mar - 4 Apr	5 Apr - 11 Apr	12 Apr - 18 Apr	19 Apr - 25 Apr	26 Apr - 2 May	3 May - 9 May	10 May - 16 May	17 May - 23 May	24 May - 30 May	31 May - 6 Jun	7 Jun - 13 Jun	14 Jun - 20 Jun	21 Jun - 27 Jun	28 Jun - 4 Jul	5 Jul - 11 Jul	12 Jul - 18 Jul	19 Jul - 25 Jul	26 Jul - 1 Aug
Research, Learning, & Literature Review																										
1 Introduction to Capstone Project																										
2 ROS Learning																										
3 Initial purchase material																										
4 Literature Review & Research on Robotics																										
Orientation to Nvidia Jetson Nano (Familiarisation)																										
1 Mechanical & Electrical Setup																										
2 Software & ROS installation																										
3 Learning of Nvidia jetson AI Fundamentals																										
4 Test and Run basic code with Stereo camera																										
5 Troubleshooting																										
Coding and Testing Machine Vision																										
1 Learning of various object detection																										
2 Angular position base on camera detection																										
3 Object detection with localisation																										
4 Apriltag for localisation and docking																										
5 Apriltag for alignment technique																										
6 Dataset preparation																										
7 Understanding Machine Learning and train custom object																										
8 Evaluation of model and proof technique.																										
Capstone Assessment Components																										
1 Preparation for Capstone Poster																										
2 Preparation for Capstone Final Report																										
3 Preparation for Capstone Final Presentation																										

Appendix B: Risk Assessment school

Inventory of Work Activities			
Reference Number: (please refer to S&H RA Repository for next running number)		**Division**	**MDME**
Title	**A smart restaurant – is IIOT only for manufacturing?**		

Ref	Location	Process	Work Activity	Remarks
1	**SIT@NYP SR6C**	Setup of Nvidia Jetson	Wiring of various components	
2			Software installation and troubleshooting	
3		Code Development	Programming camera to detect object	
4			Map Generation	
5			Machine Learning for Object Detection	

*add more rows if necessary.

** Examples: Contents will be automatically deleted when new content is typed into the rows.

RISK ASSESSMENT							
Reference Number				RA Leader:	Quek Jun Liang, Justin	Approved by:	Dr Michael Lau
Title:	A smart restaurant – is IIOT only for manufacturing?			RA Team:		Signature:	
						Designation:	
Division:	MDME	Location:	SIT@NYP SR6C;				
Last Review Date:		Next Review Date:				Date	

	Hazard Identification			Risk Evaluation				Risk Control							
M	Activity	Hazard	Possible injury / ill-health	Existing risk controls	S	L	RPN	Additional controls	S	L	RPN	Implementation Person	Due date	Remarks
1) Setup of Nvidia Jetson															
1	Wiring of various components	Exposed Wires	Short Circuit	Use heat shrinks or wire cutters as a preventive measure	3	3	6	N.A.	-	-	-	N.A.	-	-
2	Software installation and troubleshooting	Sitting in front of computer screen for long hours	Neck Aches	Take a short break every 1-2 hours to stretch muscles	2	3	6	N.A.	-	-	-	N.A.	-	-
2) Code Development															
3	Programming camera to detect object	Sitting in front of computer screen for long hours	Neck Aches	Take a short break every 1-2 hours to stretch muscles	2	3	6	N.A.	-	-	-	N.A.	-	-
4	Map Generation	Sitting in front of computer screen for long hours	Neck Aches	Take a short break every 1-2 hours to stretch muscles	2	3	6	N.A.	-	-	-	N.A.	-	-
5	Machine learning for object detection	Sitting in front of computer screen for long hours	Neck Aches	Take a short break every 1-2 hours to stretch muscles	2	3	6	N.A.	-	-	-	N.A.	-	-

*add/ delete rows if necessary.

	Hazard Identification			Risk Evaluation				Risk Control						
M	Activity	Hazard	Possible injury / ill-health	Existing risk controls	S	L	RPN	Additional controls	S	L	RPN	Implementation Person	Due date	Remarks

** Examples: Contents will be automatically deleted when new content is typed into the rows.

Level	Severity	Description
1	Negligible	Not likely to cause injury or ill-health.
2	Minor	Injury or ill-health requiring first-aid only (includes minor cuts and bruises, irritation, ill-health with temporary discomfort).
3	Moderate	Injury or ill-health requiring medical treatment (includes lacerations, burns, sprains, minor fractures, dermatitis and work-related upper limb
4	Major	Serious injuries or life-threatening occupational diseases (includes amputations, major fractures, multiple injuries, occupational cancers, acute poisoning, disabilities and deafness).
5	Catastrophic	Death, fatal diseases or multiple major injuries.

Level	Likelihood	Description
1	Rare	Not expected to occur but still possible.
2	Remote	Not likely to occur under normal circumstances.
3	Occasional	Possible or known to occur.
4	Frequent	Common occurrence.
5	Almost certain	Continual or repeating experience.

Risk score	Acceptability of risk	Recommended actions
Low 1-3	Acceptable	No additional risk control measures may be needed. Frequent review and monitoring of hazards are required to ensure that the risk level assigned is accurate and does not increase over time.
Medium 4-12	Tolerable	A careful evaluation of the hazards should be carried out to ensure that the risk level is reduced to as low as reasonably practicable (ALARP) within a defined time period. Interim risk control measures, such as administrative controls, may be implemented while long term measures are being established. Management attention is required.
High 15-25	Not acceptable	High Risk level must be reduced to at least Medium Risk before work commences. There should not be any interim risk control measures and risk control measures should not be overly dependent on personal protective equipment. If practicable, the hazard should be eliminated before work commences. Management review is required before work commences.

Appendix C: Risk assessment WFH

Inventory of Work Activities

Reference Number: (please refer to S&H RA Repository for next running number)		**Division**	**MDME**
Title	**Waiter**		

Ref	Location	Process	Work Activity	Remarks
1	WFH	Setup of Nvidia Jetson	Wiring of various components	
2			Software installation and troubleshooting	
3		Code Development	Programming camera to detect object	
4			Machine Learning for object detection	
5			Differential child, adult, elderly, wheelchair, etc	

*add more rows if necessary.

** Examples: Contents will be automatically deleted when new content is typed into the rows.

RISK ASSESSMENT							
Reference Number				RA Leader:	Quek Jun Liang, Justin	Approved by:	Dr Michael Lau
Title:	Machine Vision – Recognising customer			RA Team:		Signature:	
						Designation:	
Division:	MDME	Location:	WFH				
Last Review Date:		Next Review Date:				Date	

	Hazard Identification			Risk Evaluation				Risk Control							
Ref	Activity	Hazard	Possible injury / ill-health	Existing risk controls	S	L	RPN	Additional controls	S	L	RPN	Implementation Person	Due date	Remarks
	1) Setup of Nvidia Jetson													
1	Wiring of various components	Exposed Wires	Short Circuit	Use heat shrinks or wire cutters as a preventive measure	3	3	6	N.A.	-	-	-	N.A.	-	-
2	Software installation and troubleshooting	Sitting in front of computer screen for long hours	Neck Aches	Take a short break every 1-2 hours to stretch muscles	2	3	6	N.A.	-	-	-	N.A.	-	-
	2) Code Development													
4	Machine Learning for object detection	Sitting in front of computer screen for long hours	Neck Aches	Take a short break every 1-2 hours to stretch muscles	2	3	6	N.A.	-	-	-	N.A.	-	-
5	Differential child, adult, elderly, wheelchair, etc	Sitting in front of computer screen for long hours	Neck Aches	Take a short break every 1-2 hours to stretch muscles	2	3	6	N.A.	-	-	-	N.A.	-	-

*add/ delete rows if necessary.

** Examples: Contents will be automatically deleted when new content is typed into the rows.

Level	Severity	Description
1	Negligible	Not likely to cause injury or ill-health.
2	Minor	Injury or ill-health requiring first-aid only (includes minor cuts and bruises, irritation, ill-health with temporary discomfort).
3	Moderate	Injury or ill-health requiring medical treatment (includes lacerations, burns, sprains, minor fractures, dermatitis and work-related upper limb disorders).
4	Major	Serious injuries or life-threatening occupational diseases (includes amputations, major fractures, multiple injuries, occupational cancers, acute poisoning, disabilities and deafness).
5	Catastrophic	Death, fatal diseases or multiple major injuries.

Level	Likelihood	Description
1	Rare	Not expected to occur but still possible.
2	Remote	Not likely to occur under normal circumstances.
3	Occasional	Possible or known to occur.
4	Frequent	Common occurrence.
5	Almost certain	Continual or repeating experience.

Risk score	**Acceptability of risk**	**Recommended actions**
Low 1-3	Acceptable	No additional risk control measures may be needed. Frequent review and monitoring of hazards are required to ensure that the risk level assigned is accurate and does not increase over time.
Medium 4-12	Tolerable	A careful evaluation of the hazards should be carried out to ensure that the risk level is reduced to as low as reasonably practicable (ALARP) within a defined time period. Interim risk control measures, such as administrative controls, may be implemented while long term measures are being established.
High 15-25	Not acceptable	High Risk level must be reduced to at least Medium Risk before work commences. There should not be any interim risk control measures and risk control measures should not be overly dependent on personal protective equipment. If practicable, the hazard should be eliminated before work commences. Management review is required before work commences.

Appendix D: Angular position with ultrasonic sensor reading

Orientation	Front						Back						Side					
Distance	100 cm		200 cm		300cm		100cm		200cm		300cm		100cm		200cm		300cm	
S/N	Left Sonar	Right Sonar	Left Sonar	Right Sonar	Left Sonar	Right Sonar	Left Sonar	Right Sonar	Left Sonar	Right Sonar	Left Sonar	Right Sonar	Left Sonar	Right Sonar	Left Sonar	Right Sonar	Left Sonar	Right Sonar
1	97.05	91.58	188.39	128.23	298.42	329.27	103.39	108.78	200.33	175	377.34	311.61	86.05	89.97	196.37	198.34	416.52	418.61
2	84.56	92.36	190.13	148.99	326.72	331.96	91.89	114.91	199.77	186.18	3079.17	170.62	87.47	92.2	128.31	237.04	474.6	418.61
3	79.62	86.61	189.61	143.43	314.43	318.73	90.59	91.69	192.33	184.51	372.93	204.54	89.71	90.88	223.58	419.49	416.55	457.84
4	93.62	78.46	192.22	222.33	318.67	332.2	102.1	115.42	202.14	179.37	365.56	226.87	83.73	86.33	175.07	422.94	392.1	421.61
5	97.84	84.7	181.19	159.05	322.9	317.85	106.31	115.06	191.73	180.08	379.15	162.8	81.36	92.13	164.5	191.92	423.7	468.3
6	98.7	85.97	177.8	149.19	319.66	331.68	105.34	113.64	200.19	181.21	3097.58	233.65	83.34	93.99	194.22	248.02	422.87	468.07
7	95.34	87.08	191.6	156.95	324.26	326.44	92.51	115.18	199.59	185.02	380.94	212.83	87.19	92.24	190.49	145.37	423.41	413.86
8	98.7	83.03	189.08	148.24	323.57	327.17	96.65	115.68	198.59	186.3	371.9	188.15	75.28	94.93	167.22	422.13	404.87	420.86
9	99.41	89.83	187.09	162.42	317.99	320.52	95.75	110.02	201.98	183.5	380.21	116.67	87.5	93.37	195.21	420.35	476.26	470.21
10	93.22	89.67	190.65	148.67	3120.8	331.51	100.83	110.99	202.33	184.46	3108.6	322	86.08	90.48	384.43	423.7	3111	462.66
11	95.43	88.85	171.57	102.16	322.65	328.71	103.62	97.77	171.89	177.31	381.31	158.11	87.96	92.77	364.31	420.62	421.62	422.81
12	100.51	90.42	190.68	226.54	3126.63	333.67	92.33	103.95	193.18	183.71	385.95	272.91	81.31	92.19	384.81	417.41	3114.02	468.28
13	85.78	90.69	193.48	159	320.69	416.27	171.69	108.38	198.02	166.24	391.19	216.27	86.91	93.57	386.59	470.1	423.36	464.45
14	91.79	91.31	149.19	147.91	320.21	330.07	93.68	113.66	189.67	185.55	374.6	178.24	88.15	94.29	379.26	423.85	426.06	422
15	79.82	91.17	191.34	154.31	325.91	333.43	102.16	103.76	202.79	181.1	379.06	205.62	86.89	91.66	385.42	178.94	420.12	469.62
16	85.59	91.31	190.76	149.69	324.67	330.58	106.67	100.5	202.61	185.58	3095.86	182.42	87.68	82.78	422.29	186.18	425.07	469.95
17	85.38	90.55	119.77	144.73	323.89	327.76	92.27	113.3	175.56	185.51	383.32	240.18	87.71	90.06	384.07	251.17	417.63	467.98
18	90.71	88.5	189.02	110.62	304.65	321.39	98.64	113.76	202.41	184.02	3095.04	181.9	83.25	90.57	473.46	192.71	406.77	3230.92
19	97.29	88.72	172.61	330.78	307	412.39	79.21	112.5	193.4	180.04	379.66	209.47	109.84	91.07	378.5	193.29	419.35	469.65
20	101.99	89.79	186.63	328.81	317.25	321.18	94.51	114.91	200.65	182.86	380.73	233.01	82.38	91.13	467.78	207.2	410.61	467.81
21	98.59	79.47	195.12	332.8	316.33	324.31	304.28	114.71	200.48	200.43	373.69	165.7	83.36	86.3	469.46	189.17	3117.78	471.25
22	97.81	89.12	189.77	197.26	322.34	279.91	96.72	115.85	190.99	178.42	374.84	282.99	65.02	88.46	378.86	423.78	420.05	467.82
23	98.85	89.54	189.95	105.99	320.54	307.8	106.87	106.85	193.4	186.16	379.38	149.58	84.78	93.13	384.5	467.3	473.54	421.46
24	96.8	89.43	188.27	196.42	319.64	331.19	104.93	115.38	199.56	185.61	3134.93	280.22	88.02	89.98	384.44	416.34	419.13	470.36
25	99.8	88.83	183.2	192.31	323.6	328.86	106.09	112.37	190.41	185.62	372.17	146.76	86.69	93.12	472.21	467.9	405.36	318.22
26	92.24	91.51	181.65	98.3	321.13	333.84	96	101.49	198.21	182.06	380.78	281.54	87.7	92.09	384.83	190.7	472.49	3194.63
27	92.19	89.66	192.8	193.4	3139.81	311.08	106.98	112.07	162.81	183.27	381.63	103.74	84.44	90.92	382.8	195.08	419.14	3209.36
28	92.75	89.1	192.62	121.98	317.05	328.69	84.87	78.69	199.78	179.25	371.94	273.29	86.01	91.19	377.36	195.71	468.73	420.93
29	86.16	91.52	188.98	182.79	320.95	329.74	92.8	118.05	184.72	3222.24	378.35	122.03	82.47	92.66	383.88	249.93	471.49	418.47
30	91.19	91.82	186.23	165.41	311.46	331.68	90.13	110.25	200.92	176.59	366.54	140.87	85.32	84.43	385.62	191.09	419.02	421.7
31	95.22	91.22	187.51	128.57	309.5	331.29	404.19	111.5	202.8	199.25	375.64	116.84	87.54	91.76	379.14	244.63	460.22	422.74
Avg	93.35	88.77	184.16	172.17	590.43	331.01	116.58	109.39	194.94	281.18	904.84	202.95	85.52	90.99	336.42	300.08	690.11	710.03
Upper Lim	102.69	97.65	202.57	189.39	649.47	364.11	128.24	120.33	214.44	309.29	995.32	223.24	94.07	100.09	370.06	330.09	759.12	781.04
Lower Lim	84.02	79.89	165.74	154.95	531.39	297.90	104.92	98.45	175.45	253.06	814.35	182.65	76.97	81.89	302.78	270.07	621.10	639.03

Notes:
Sonar direction indicates the object's direction relative to the camera.
Data reading in yellow stray >10% from the average reading

Appendix E: Angular position with ultrasonic picture

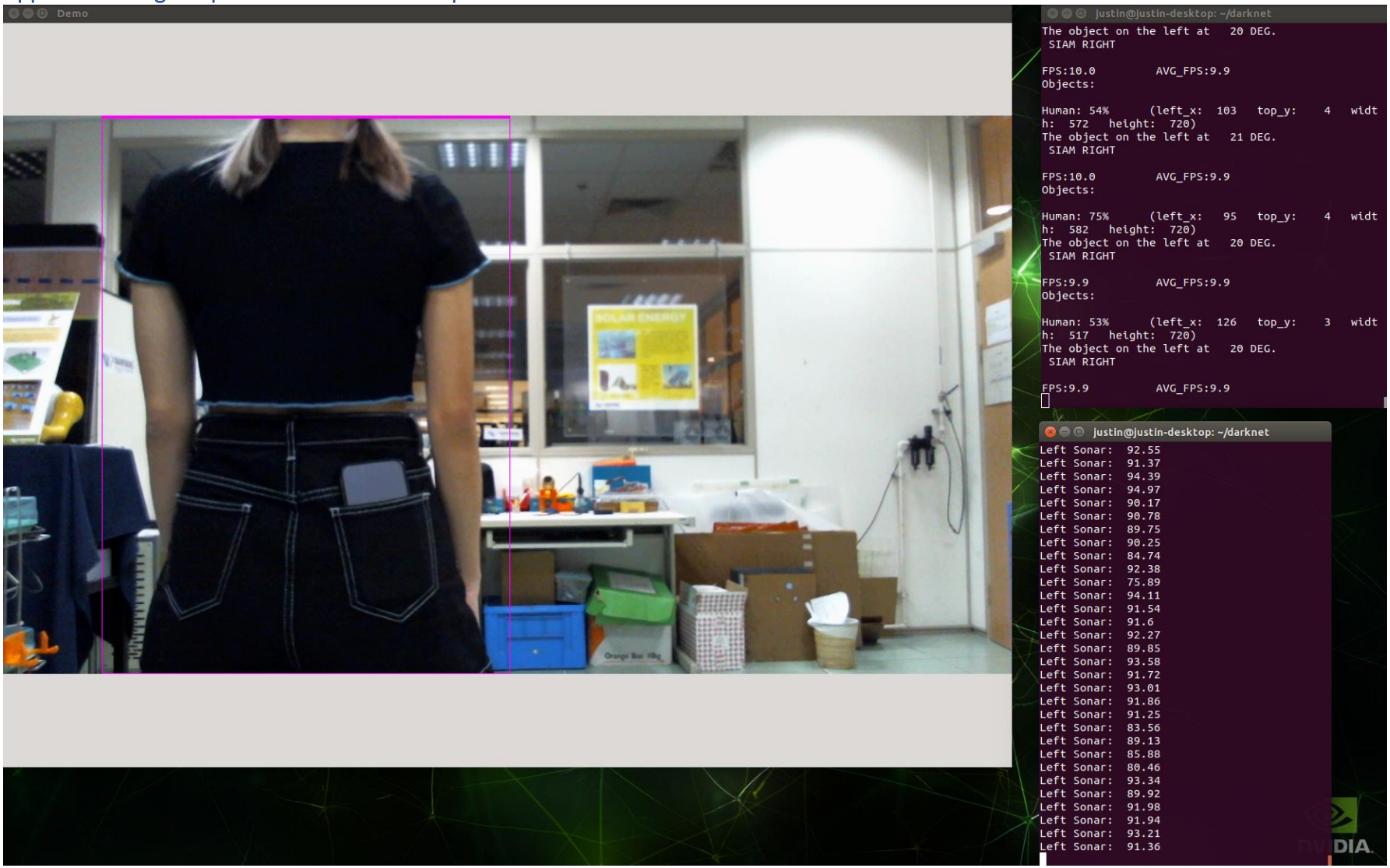


Figure 50: 100cm, left, face back

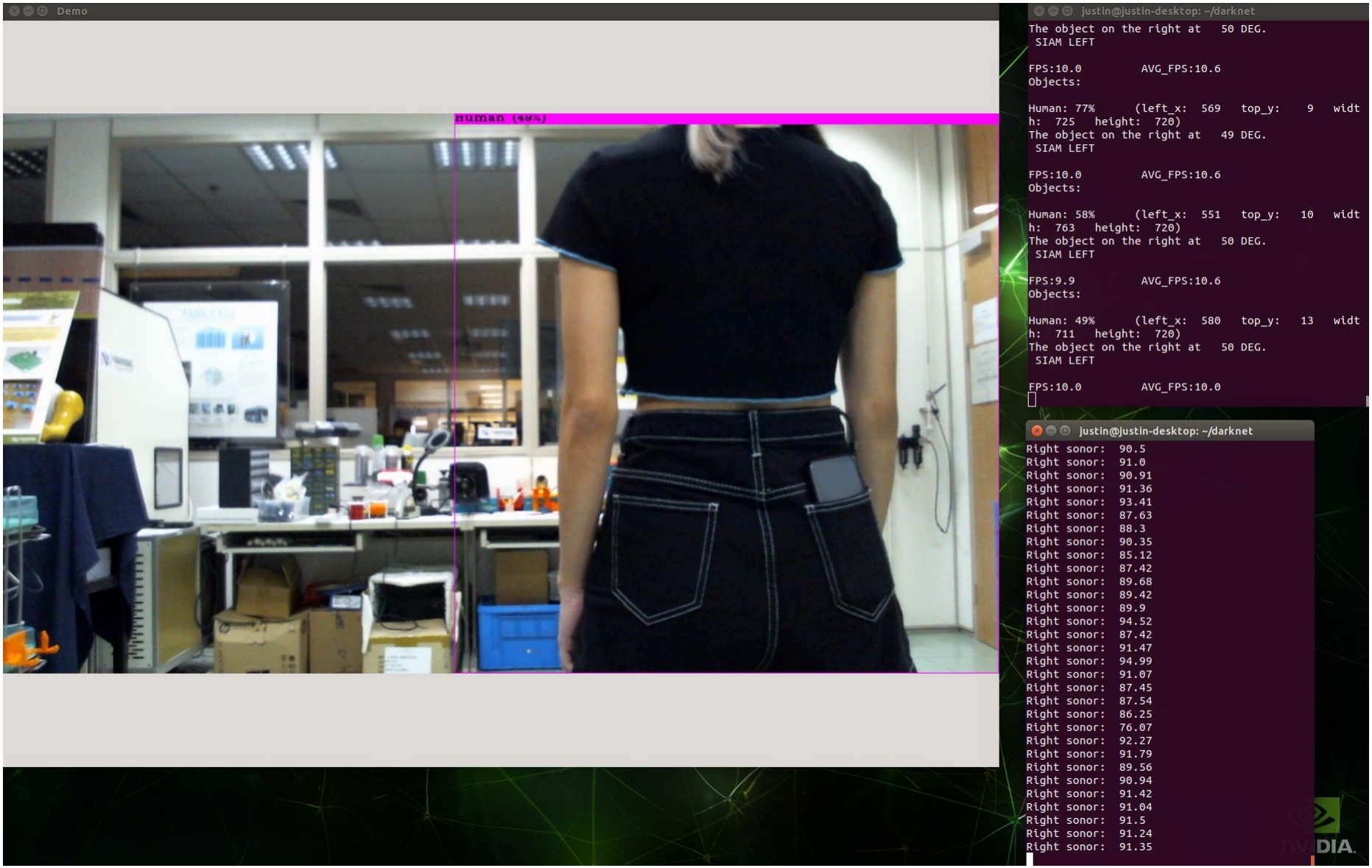


Figure 51: 100cm, right, face back

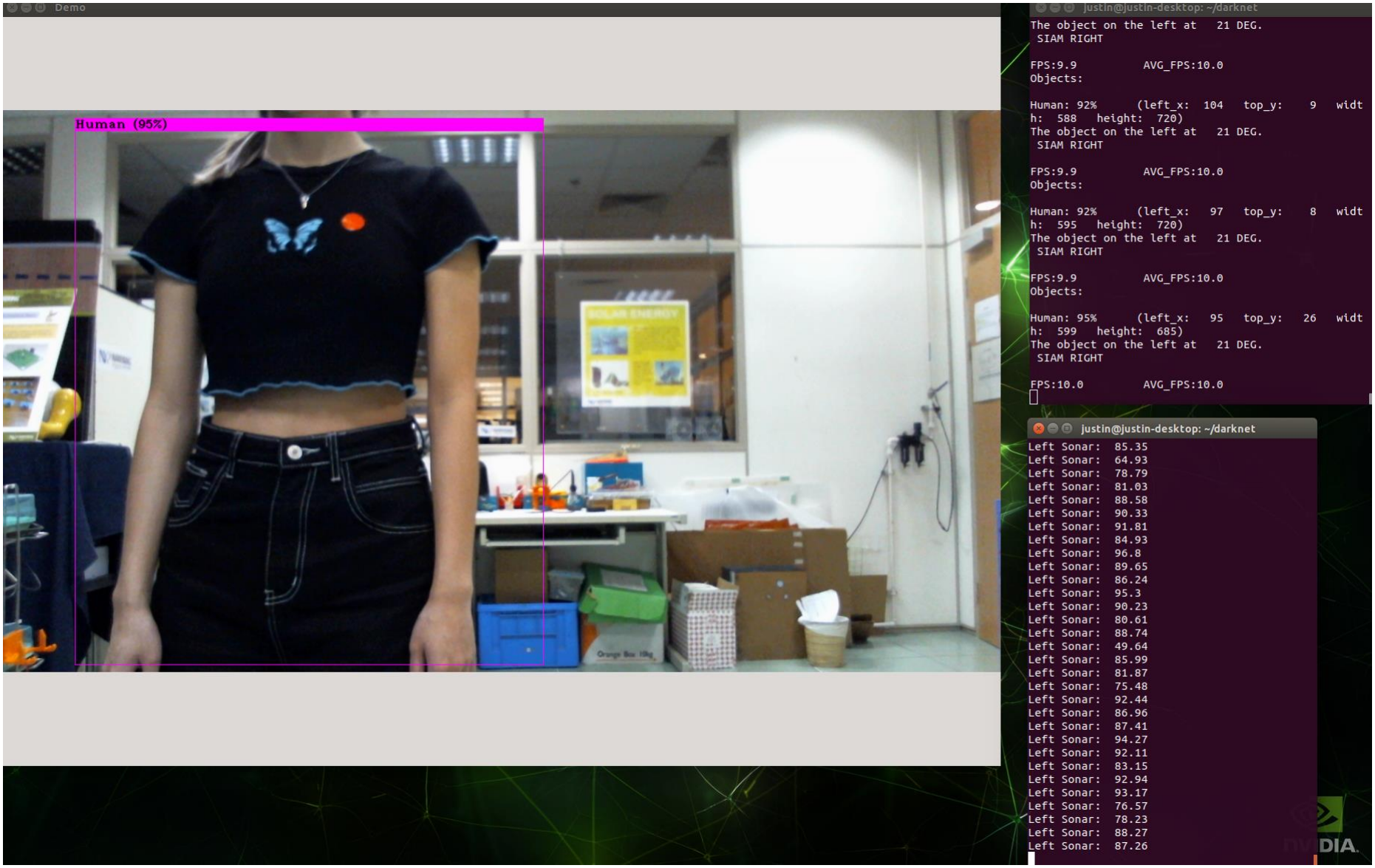


Figure 52: 100cm, left, face forward

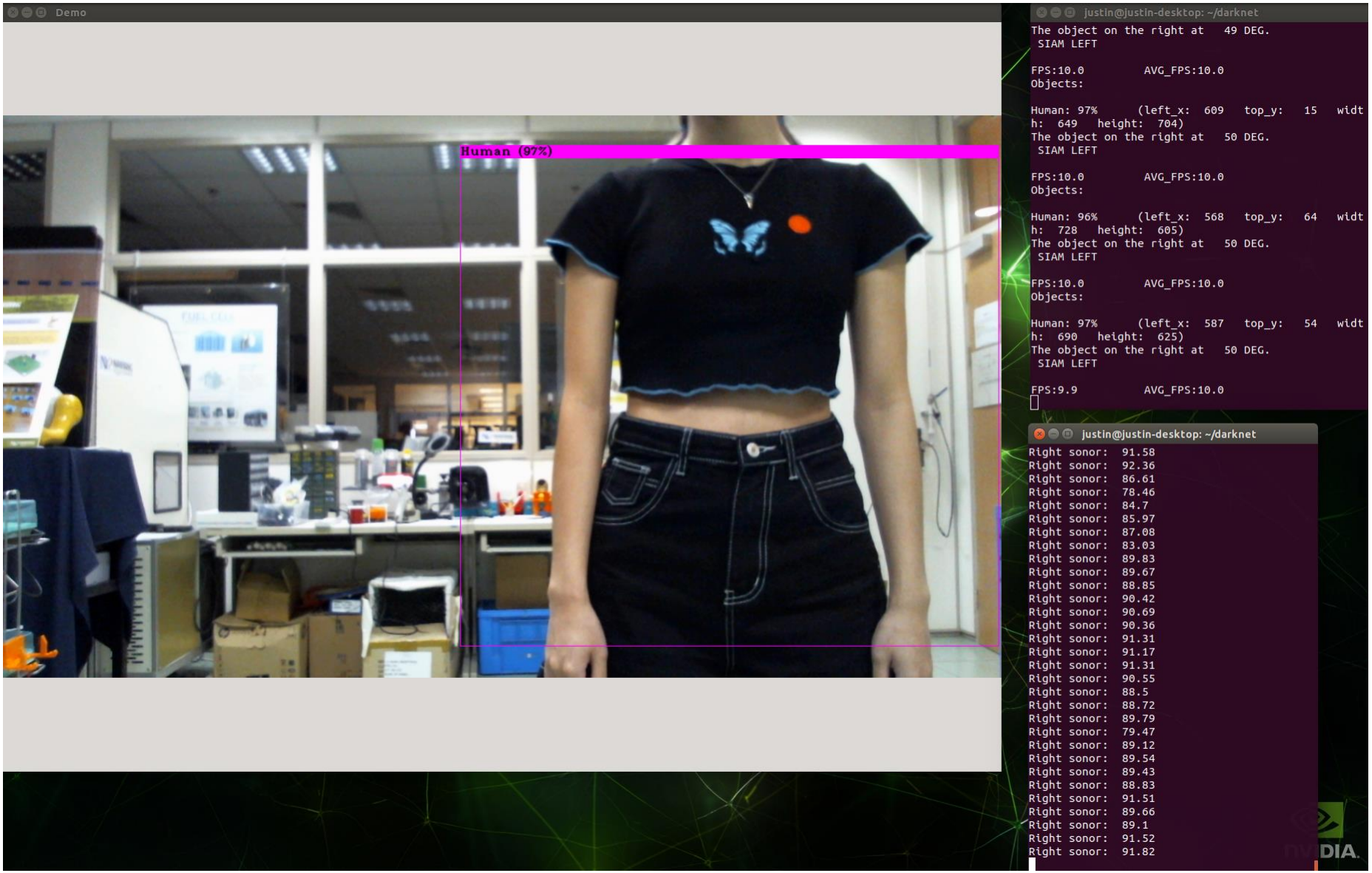


Figure 53: 100cm, right, face forward

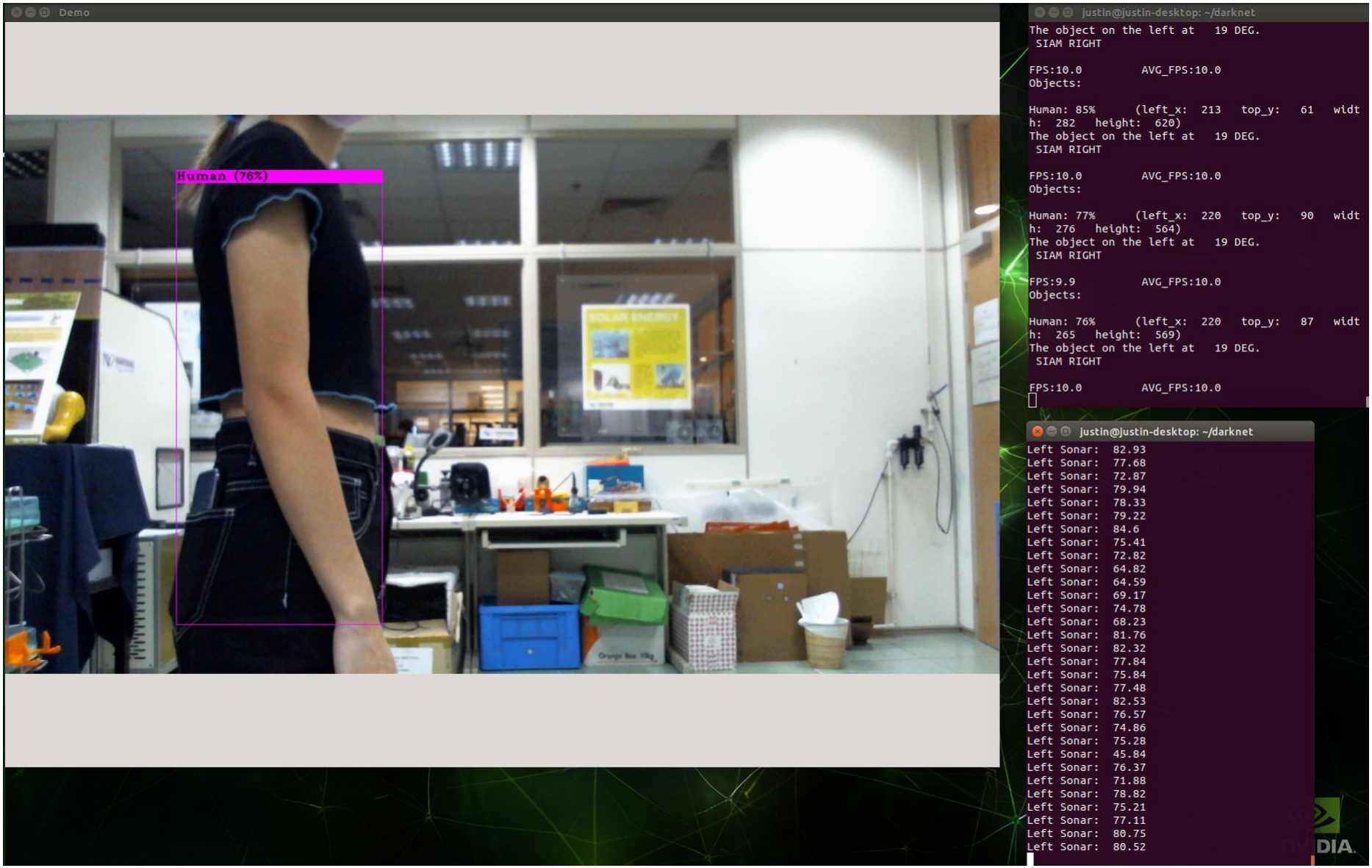


Figure 54: 100cm, left, face side

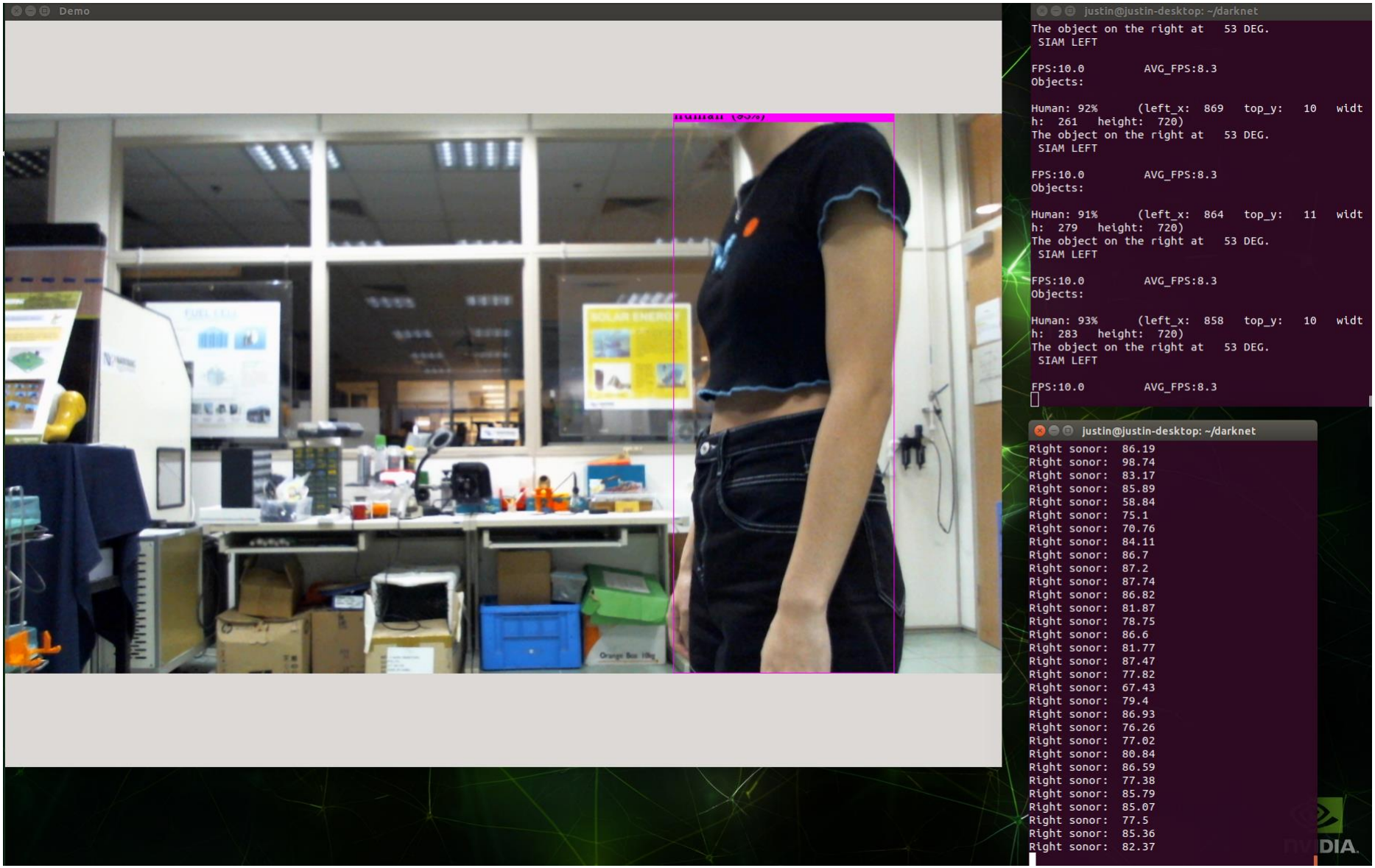


Figure 55:100cm, right, face side

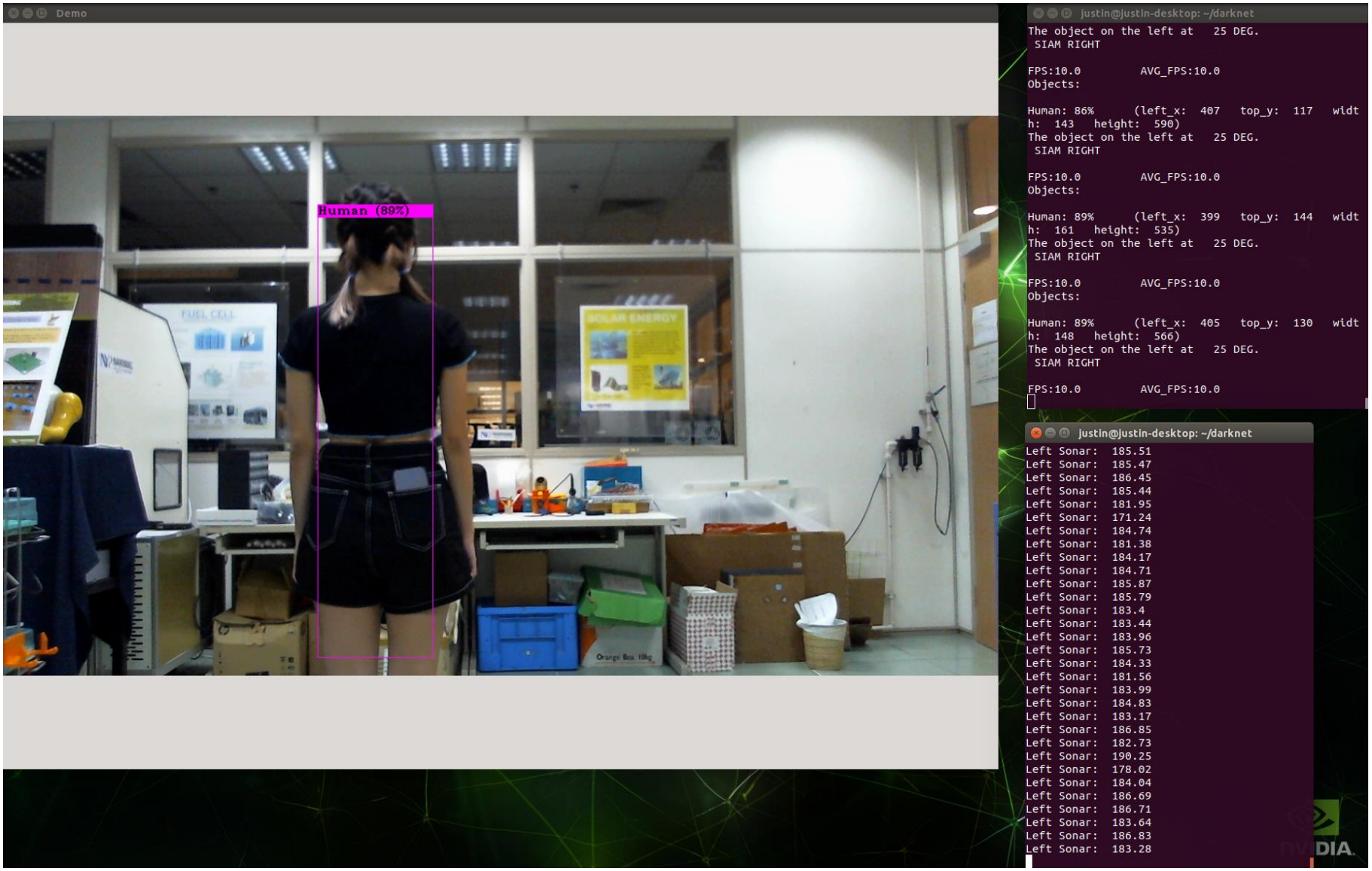


Figure 56: 200cm, left, face backwarad

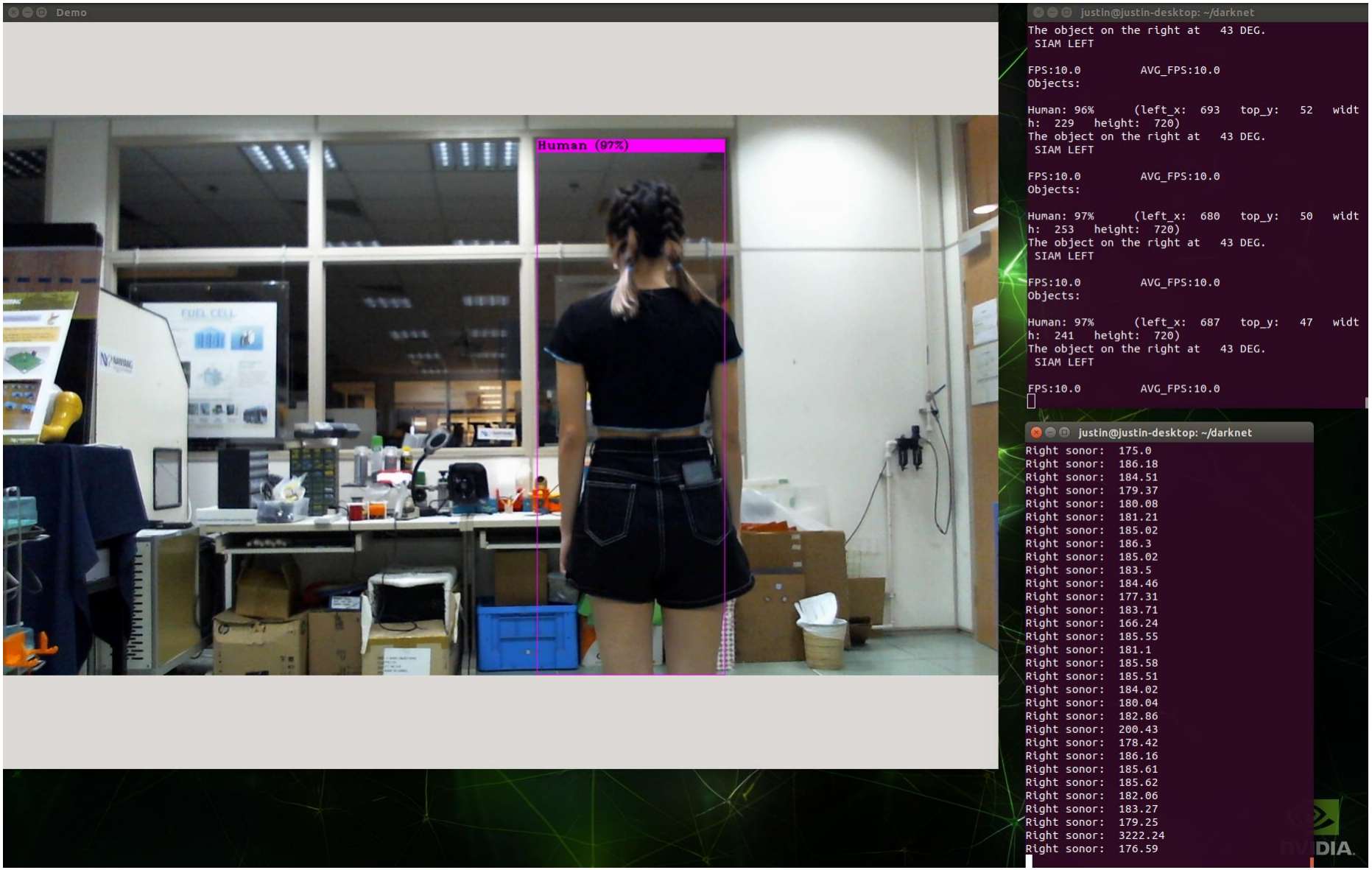


Figure 57: 200cm, right, face backwards

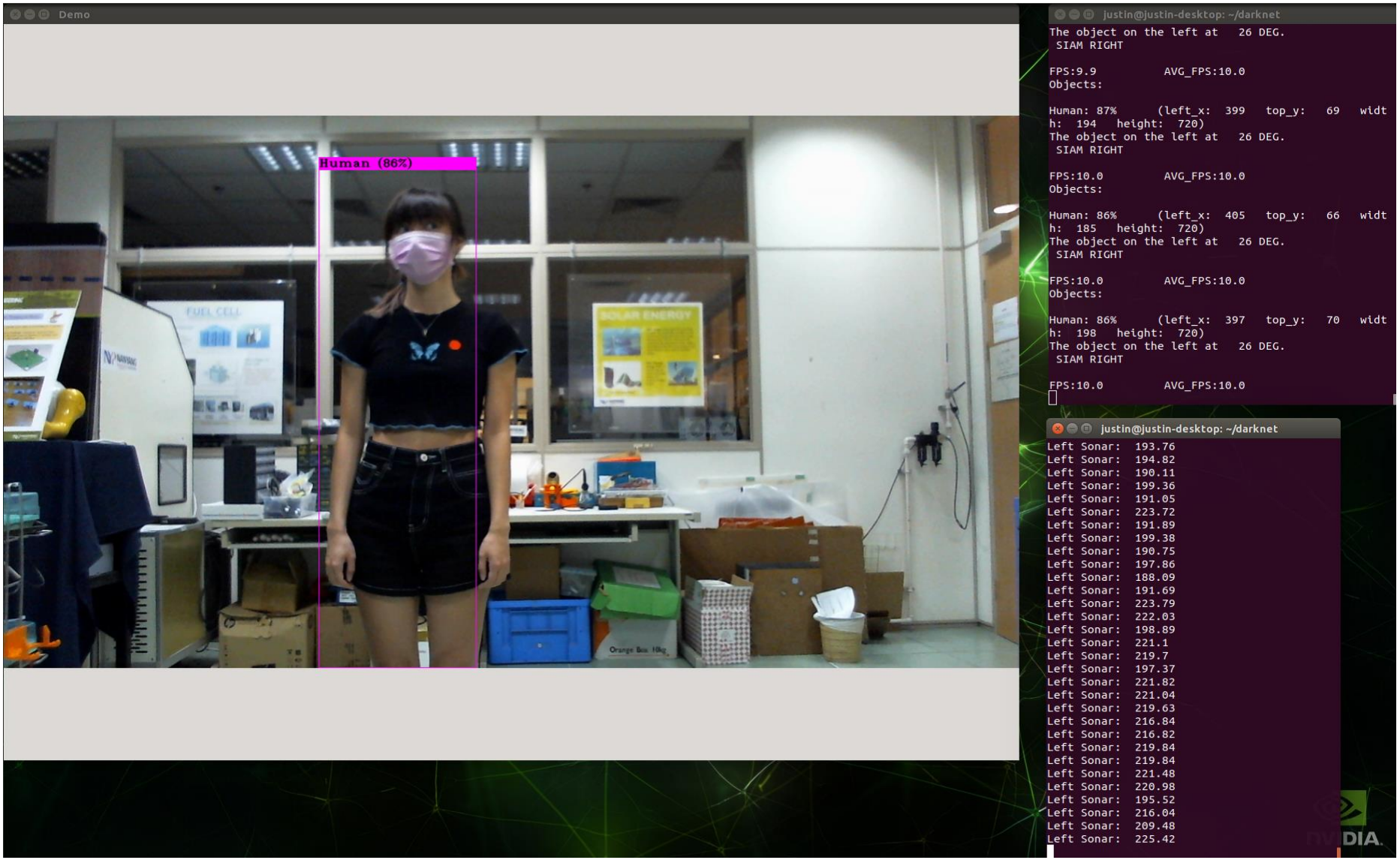


Figure 58: 200cm, left, face forward

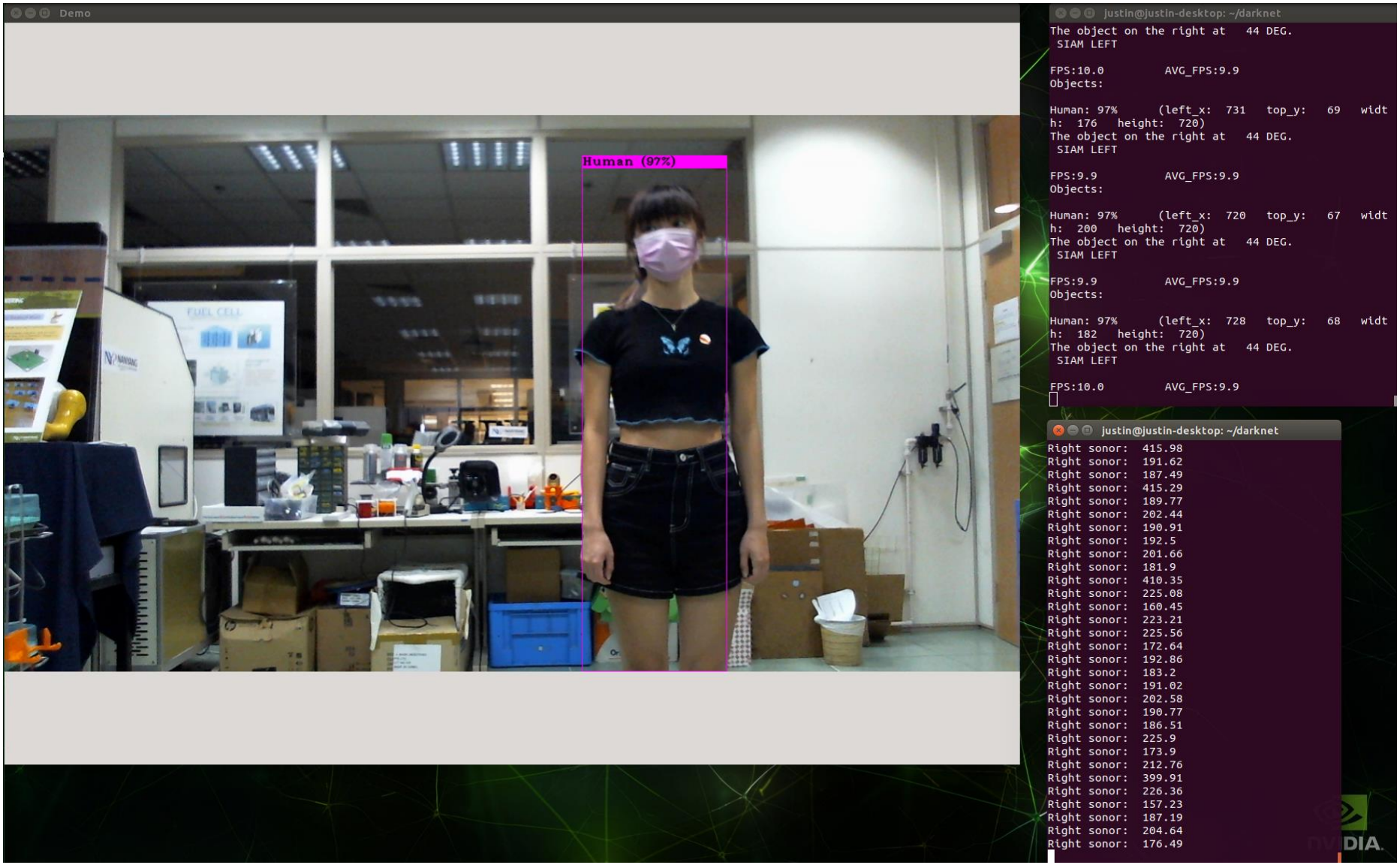


Figure 59: 200cm, right, face forward

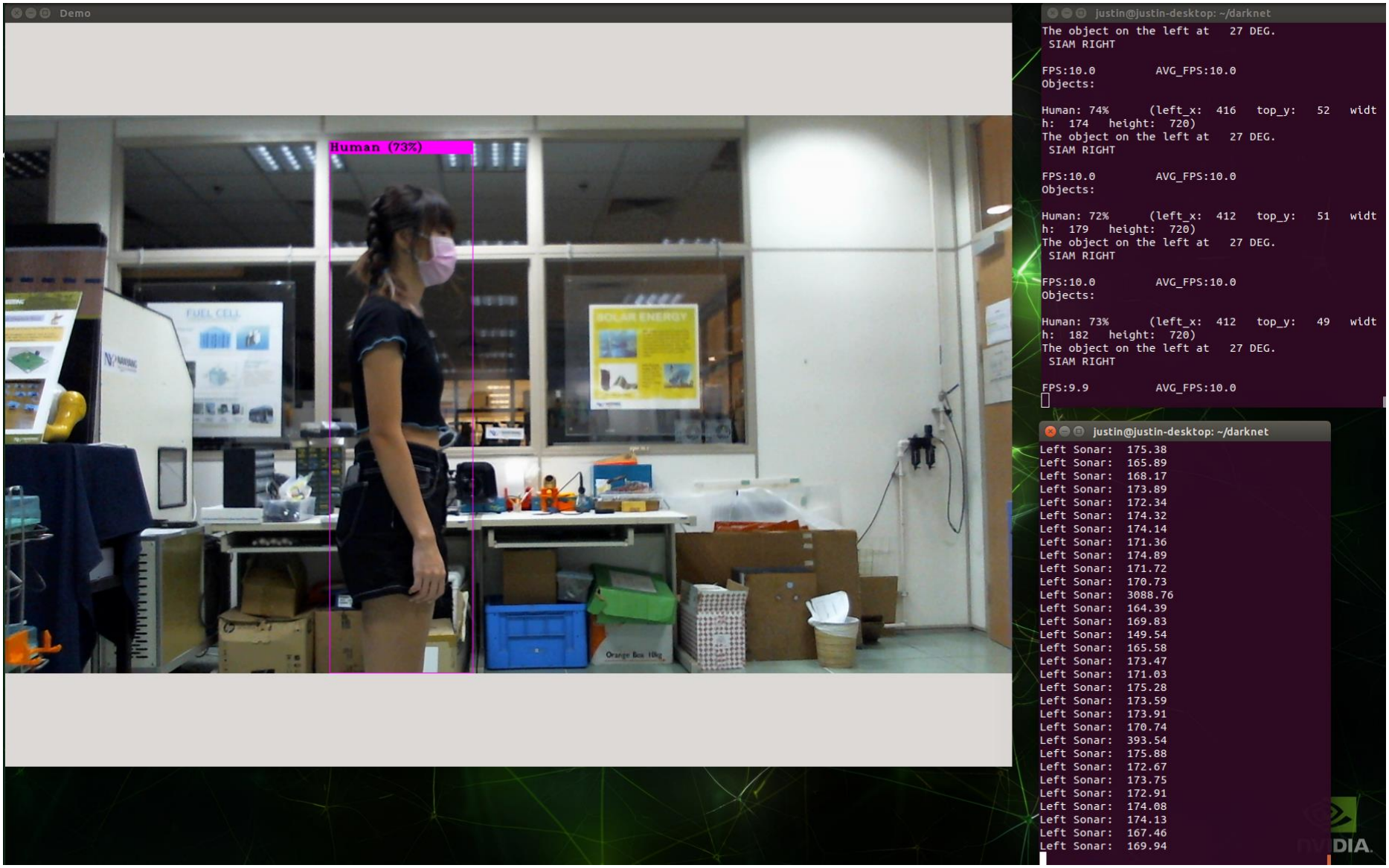


Figure 60:200cm, left, face side

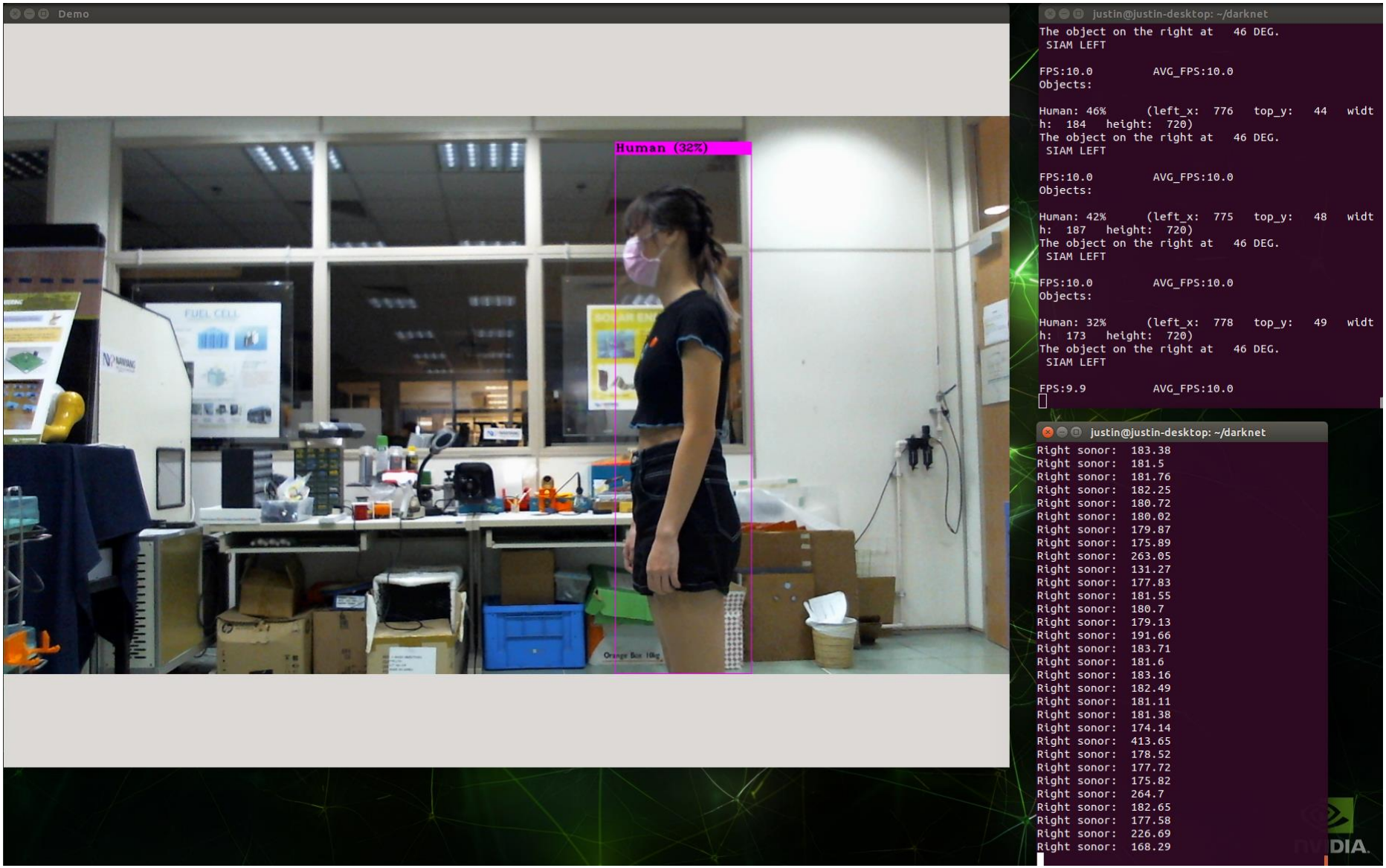


Figure 61: 200cm, right, face side

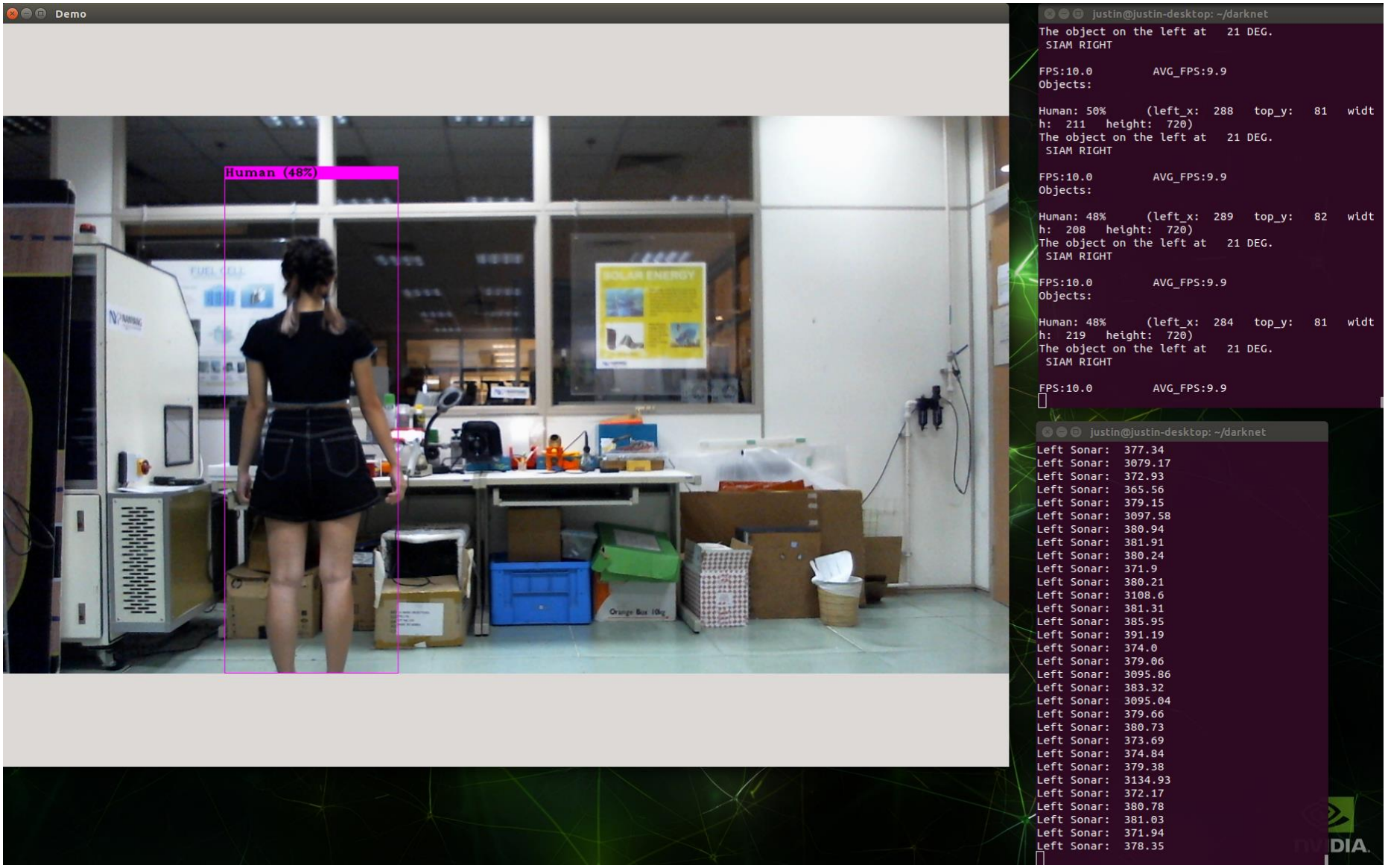


Figure 62: 300cm, left, face backward

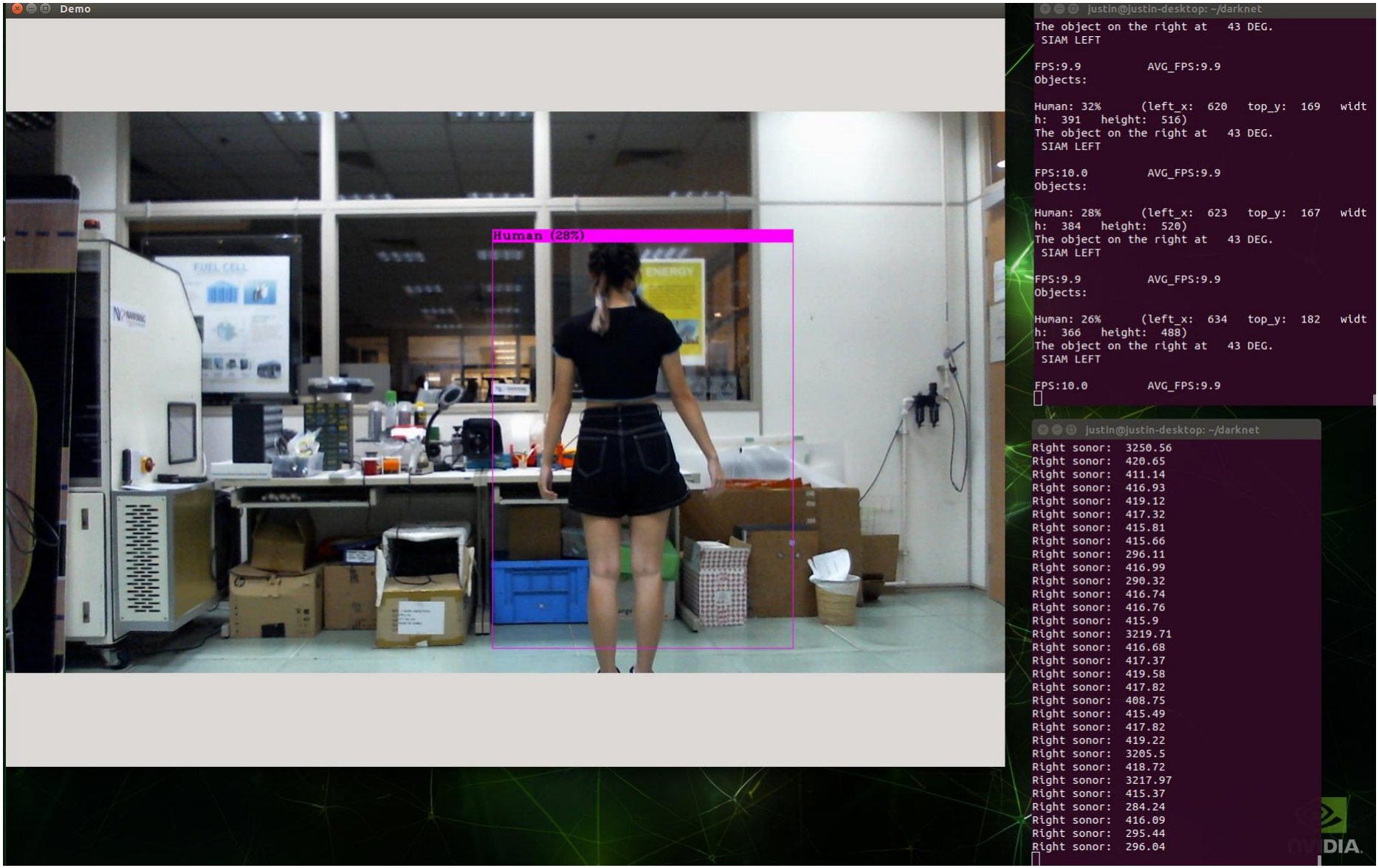


Figure 63: 300cm, right, face backward

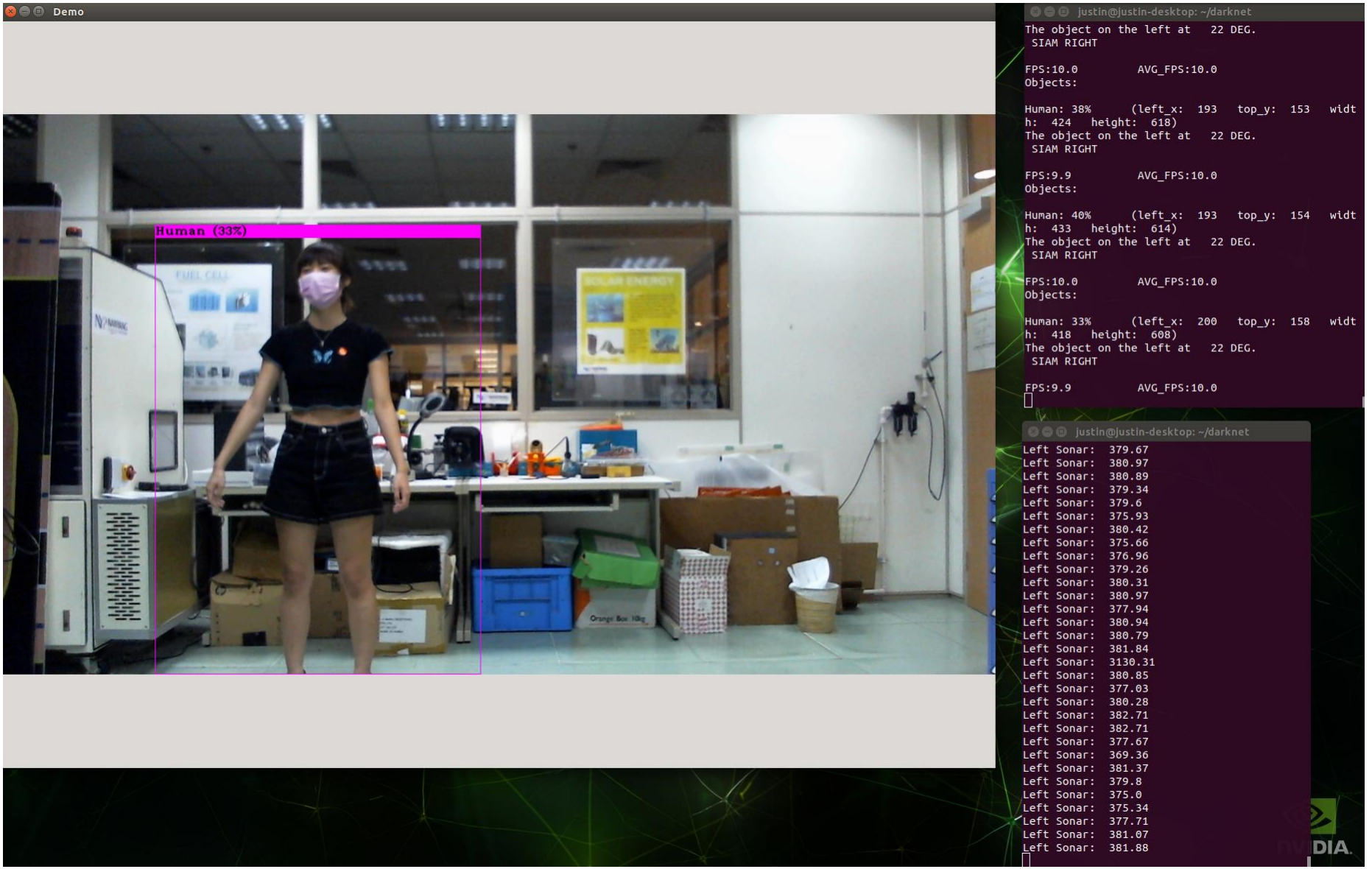


Figure 64: 300cm, left, face forward

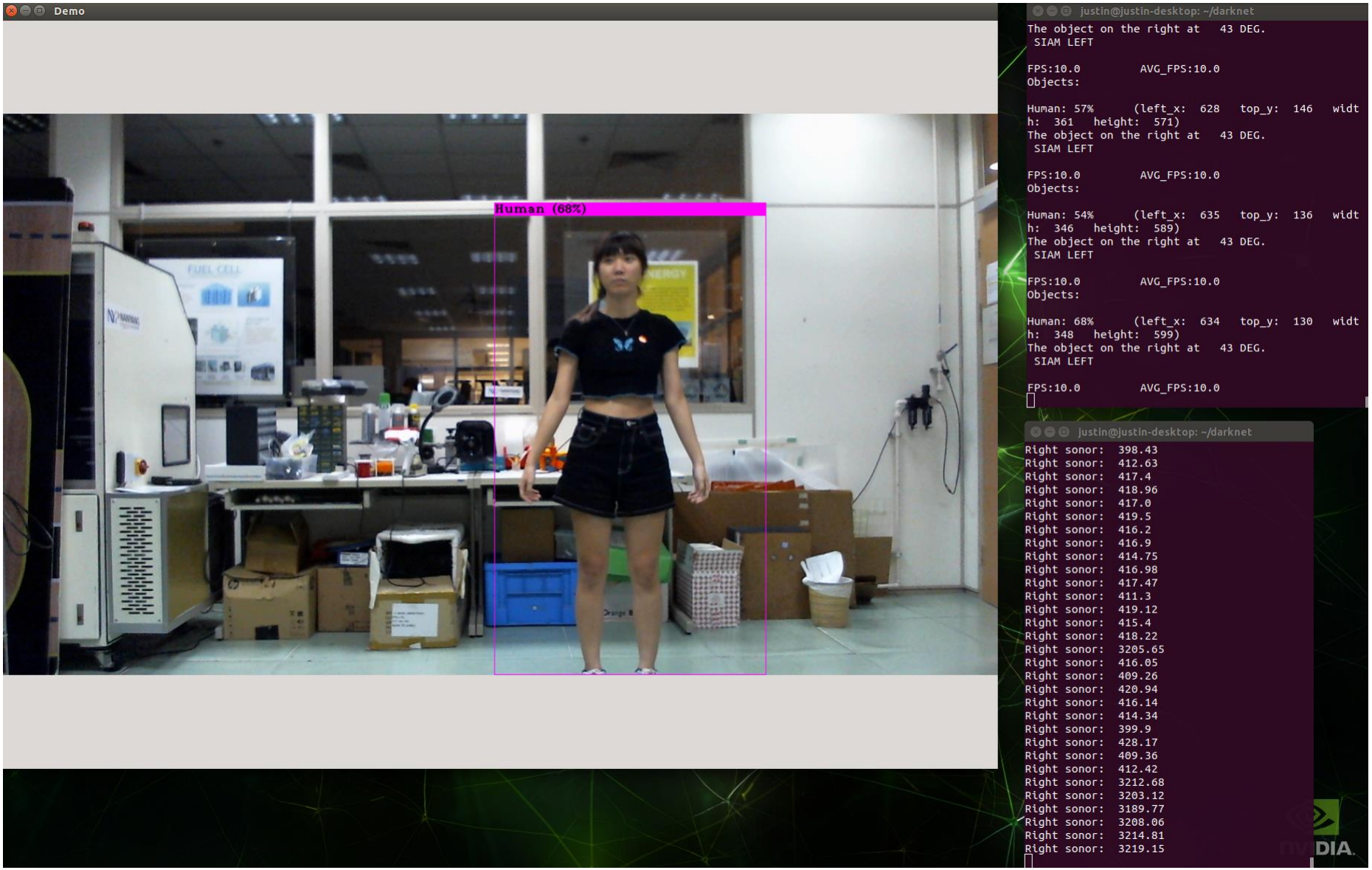


Figure 65: 300cm, right, face forward

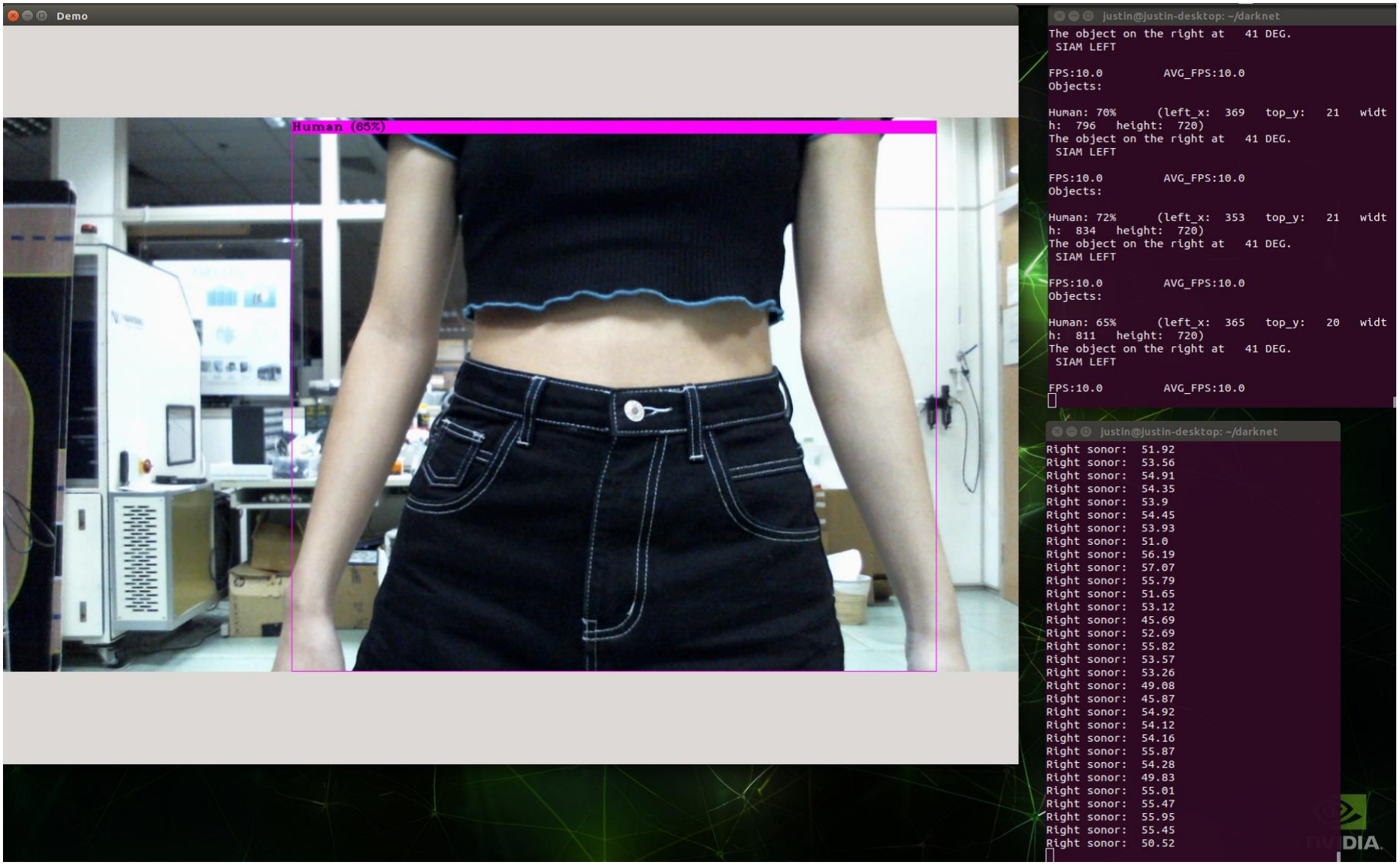


Figure 66: closest possible reading before no detection from camera

Appendix F: Example picture classification and detection

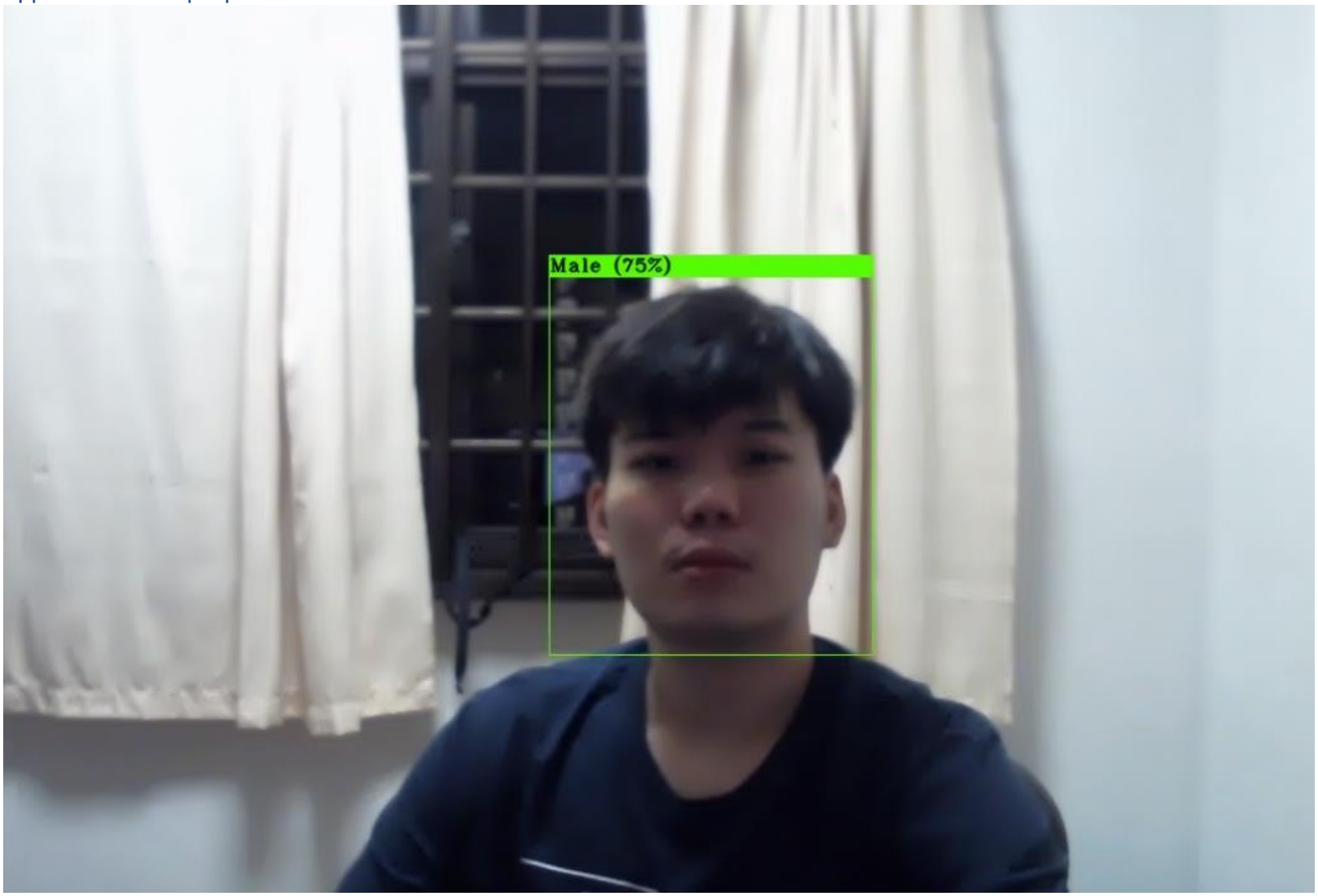


Figure 67: Gender detection example

Figure 68: Wheelchair detection example

Figure 69: Crutches detection example

Figure 70: Cructches detection exampe 2

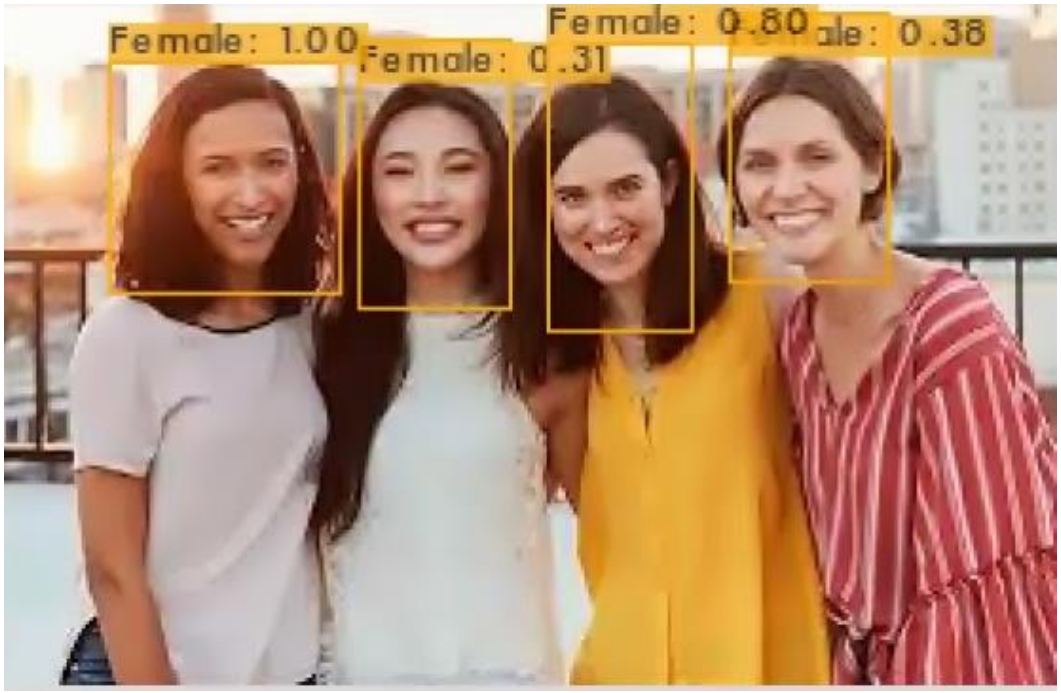


Figure 71: Gender detection example 2