Department of Electrical and Computer Engineering

North South University

Senior Design Project

Semi-autonomous Wheelchair with Real-Time Object Detection Using YOLOv7

Submitted By : (Group#08)

Name	ID
Tanjerul Islam Tanim	-1811942043
Md Shahid Mazumder Sagor	-1521883643
Mushtari Mahapara -	-1620608042
Faysal Miah -	-1721758042

Faculty Advisor:

Dr. Shahnewaz Siddique

Assistant Professor

ECE Department

Fall 2022

Declaration

This is to declare that no part of this report or the project has been previously submitted elsewhere for the fulfillment of any other degree or program. Proper acknowledgement has been provided for any material that has been taken from previously published sources in the bibliography section of this report.

.....

Tanjerul Islam Tanim

ECE Department

North South University, Bangladesh

.....

Faysal Miah

ECE Department

North South University, Bangladesh

.....

Mushtari Mahapara

ECE Department

North South University, Bangladesh

.....

Md Shahid Mazumder Sagor

ECE Department

North South University, Bangladesh

Approval

The Senior Design Project entitled " Semi-autonomous Wheelchair with Real-Time Object Detection Using YOLOv7" by Tanjerul Islam Tanim(1811942043), Faysal Miah(1721758042), Mushtari Mahapara(1620608042), and Md Shahid Mazumder Sagor(1521883643), is approved in partial fulfillment of the requirement of the Degree of Bachelor of Science in Computer Science and Engineering on January 2, 2023 and has been accepted as satisfactory.

Supervisor's Signature

…….

Dr. Shahnewaz Siddique
Department of Electrical and Computer Engineering
North South University
Dhaka, Bangladesh.

Department Chair's Signature

……

Dr. Rajesh Palit
Department of Electrical and Computer Engineering
North South University
Dhaka, Bangladesh.

Acknowledgement

First of all, we wish to express our gratitude to the Almighty for giving us the strength to perform our responsibilities and complete the report. I would like to express my heartfelt gratitude to my instructor **DR. Shahnewaz Siddique** and our department chairman **Dr. Rajesh Palit** for providing me with the wonderful opportunity to work on this project "Semi-autonomous Wheelchair with Real-Time Object Detection Using YOLOv7". It aided me in conducting extensive research and learning a great deal about this subject.

Finally, I would like to express my gratitude to my seniors and friends for their assistance in completing this project within the time constraints.

ABSTRACT

Disability is part of being human. According to the World Health Organization(WHO) an estimated 1.3 billion people – about 16% of the global population – currently experience significant disability.

In our senior design project we introduce a Semi-Autonomous wheelchair for the disabled people. Autonomous wheelchair is a technology that is appropriate for people who have lost mobility due to brain injury or the loss of limbs but or loss of eyesight. The technology can also enhance safety for users who use ordinary joystick- controlled powered wheelchairs, by preventing collisions with walls, fixed objects, furniture and other people. This project is intended towards contributing to the helping hand of the paralyzed patients. This is a multifunctional wheelchair that helps the is a multifunctional wheelchair that helps the unable person to control it in ways like automatic obstacle detection control and remote control. The main features of the wheel chairs are: Automotive format - (detects obstacles by YOLOv7 and finds new path to move). Remote controlled feature - (It's controlled by any android device through bluetooth connectivity). Security assurance - (It can assure security of the disabled person by keeping record of the data of the moving paths through camera.)

TABLE OF CONTENTS

CHAPTER 1: INTRODUCTION

Autonomous wheelchair is a technology that is appropriate for people who have lost mobility due to brain injury or the loss of limbs, but who retain speech. The technology can also enhance safety for users who use ordinary joystick-controlled powered wheelchairs, by preventing collisions with walls, fixed objects, furniture and other people.

This project is intended towards contributing to the helping hand of the paralyzed patients. This is a multifunctional wheelchair that helps the unable person to control it in ways like- automatic obstacle detection control and remote control. Not only that, it has also been decorated with a heart monitoring system that gives a total result of the heart condition of the patients.
The main features of the wheel chairs are :

- Automotive format. (detects obstacles by YOLOv7 and finds new path to move)
- Remote controlled feature. (It's controlled by any android device through bluetooth connectivity.
- Security assurance - (It can assure security of the disabled person by keeping record of the data of the moving paths through camera.)

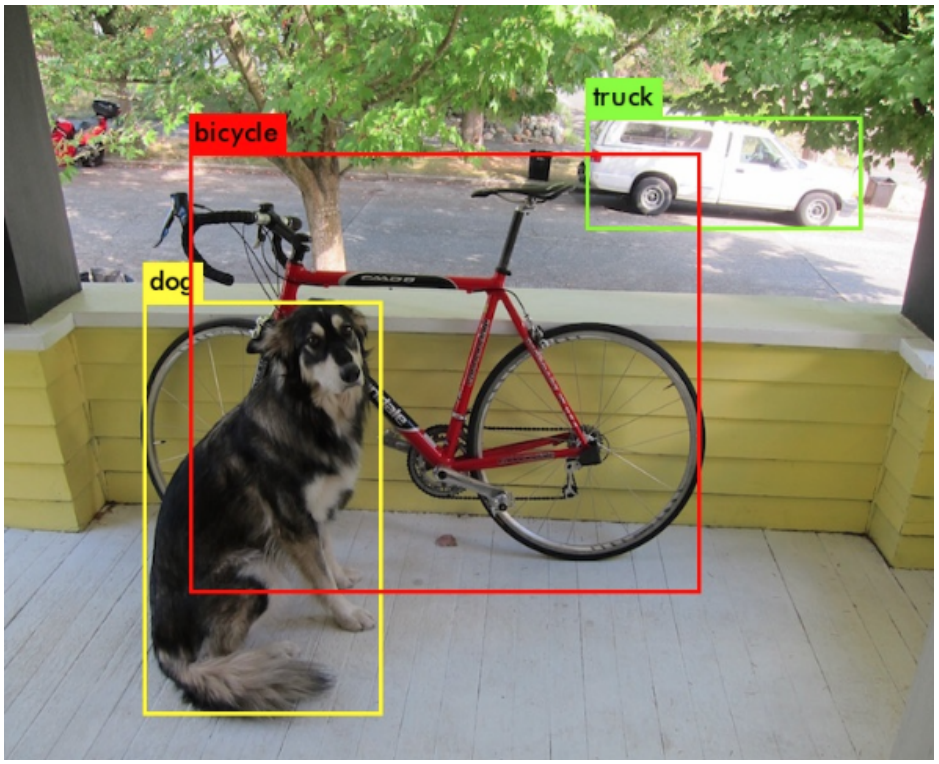


Figure: Real-time object detection

CHAPTER 2: LITERATURE REVIEW

We studied many research papers linked to our project issue and selected a few studies that were conducted on real-time object detection and semi autonomous wheelchair while working on this research project. We chose these papers because their working approach is similar to what we're doing in our research. We also derived several work-related concepts from there.

2.1 Literature Explanation on Semi Autonomous Wheelchair

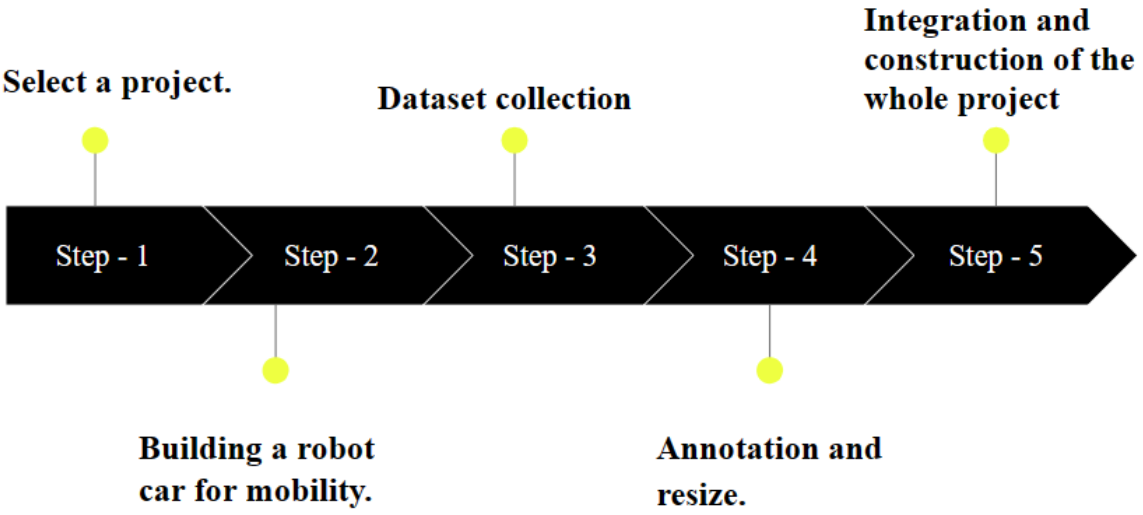
- A project has been done by Vishnu K. Narayanan and Anne Spalanzani which is similar to our project. Their project was about “A semi-autonomous framework for human-aware and user intention driven wheelchair mobility assistance [3]. They develop a semi-autonomous framework for assistive wheelchair navigation in human environments, which is driven by the intention of the wheelchair user. Safe and socially compliant motion provided by a user intention driven local motion planner is fused with user teleoperation in order to create such a system.
- Another project related to semi autonomous wheelchairs is “The autonomous mobile robot SENARIO: a sensor aided intelligent navigation system for powered wheelchairs [4] In this project a sensor-aided intelligent navigation system that provides high-level navigational aid to users of powered wheelchairs.

2.2 Literature Explanation Real-Time Object Detection

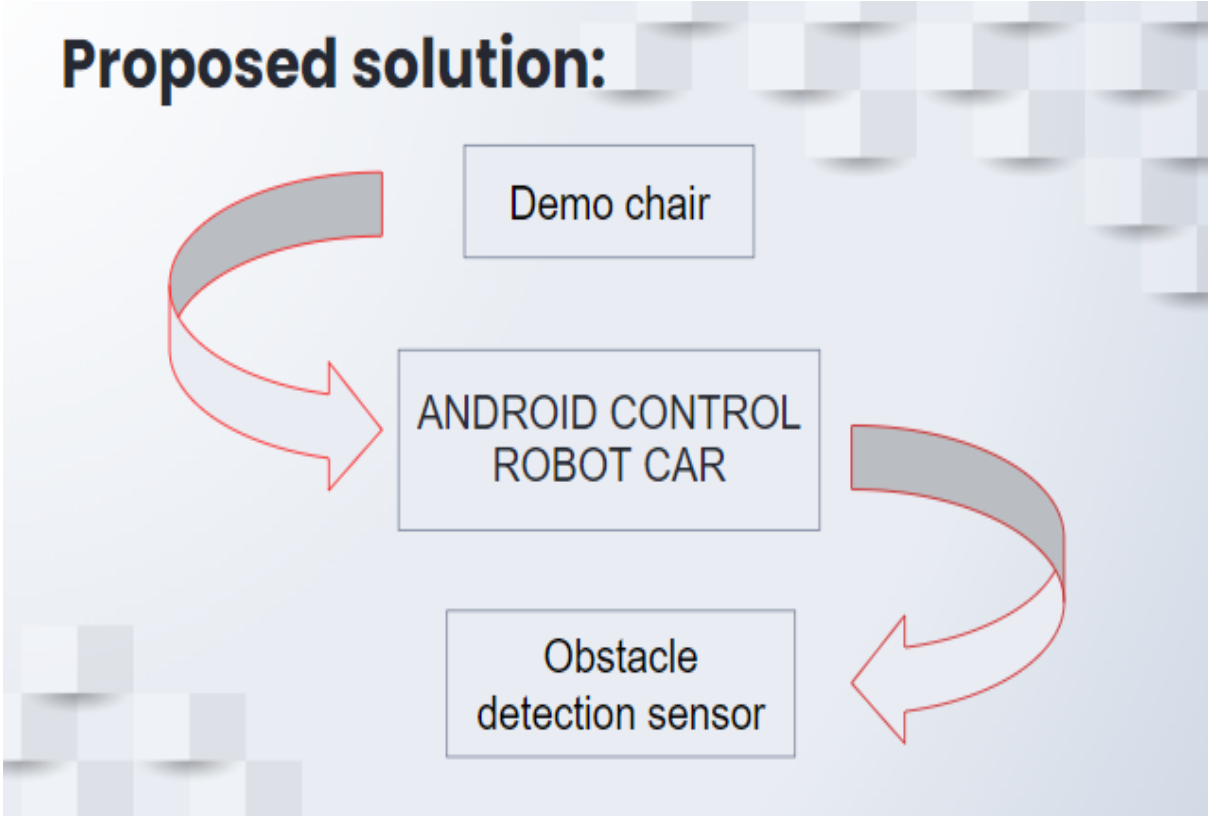
- A study has been done in [1] by **Upulie Handalage**, **Lakshini Kuganandamurthy** to detect objects. They used Convolutional Neural Networks for data pre-processing. Then, they used a YOLO detector for their model.
- Another research done in [2] by **Md. Bahar Ullah t**o detect objects in this computer vision with Deep Learning has been established and accomplished with time, primarily over one particular well-known algorithm named Convolutional Neural Networks (CNN). To generate a single step procedure including object detection and classification You Only Look Once (YOLO) was introduced.

CHAPTER 3: METHODOLOGY

After finalizing the project, we had to study the related works. First we work on building a robot car.Here robotics car is used for the purpose of electrically controlling the wheelchair.Switch Controlled robot car is a robot which is built using Microcontroller arduino Uno in which serial communication is done and programming is done in embedded C language. After constructing a robot car to give mobility to the wheelchair users we work on obstacle avoidance. We collected the dataset and started working on that. We found the dataset which was unlabeled. So, we annotated the dataset using Roboflow, Makesense.ai and LabelMe for the annotation. We annotated it into eight classes. The we resized it into 416 * 416 pixels. Firstly, we tried with some custom small dataset to train our model. After successfully running, we tried with the whole dataset. Then our model successfully detected obstacles. We also tried with Yolact++ and Mask RCNN. Finally, our model successfully detected obstacles from the video.



3.1 Workflow For Robot Car

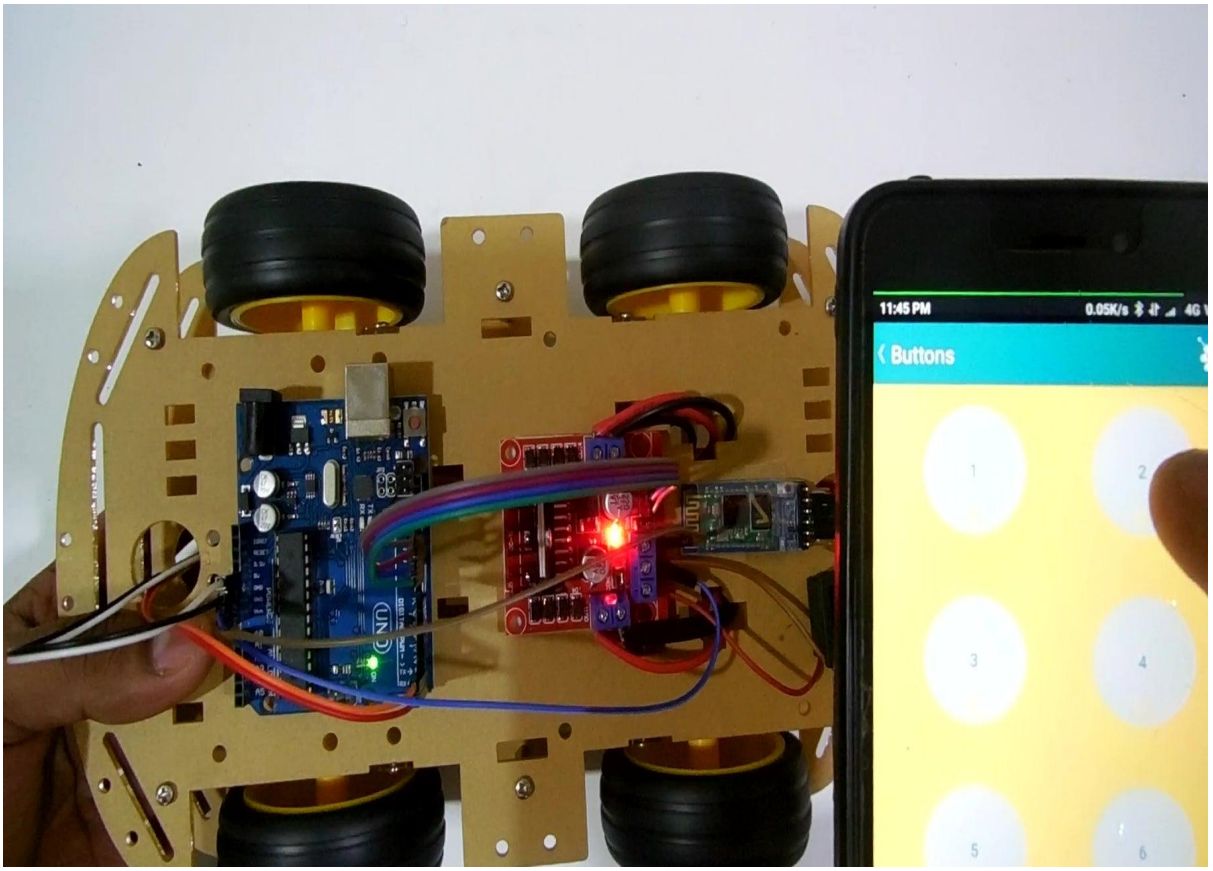


3.1.1 Demo Chair

- We make a demo chair by using a light weight PVC board.
- This demo chair will work as a wheelchair while demonstrating the project.

3.1.2 Android Control Robot Car

- We create a Bluetooth Controlled robot car which is built using Microcontroller in which serial communication is done via Bluetooth and programming is done in embedded C language.



3.1.3 Coding for Robot Car

This code is implemented in Arduino UNO. Programming is done in embedded C language.

```
 #include <AFMotor.h>
#include <Servo.h>
Servo servo;

AF_DCMotor LF(1);
AF_DCMotor LB(2);
AF_DCMotor RF(3);
AF_DCMotor RB(4);
#define trigPin  A0
#define echoPin  A1
```

```
void setup()
  {
    servo.attach(10);
    pinMode(trigPin, OUTPUT);          // Declare "trigPin" as "Output Pin".
    pinMode(echoPin, INPUT);           // Declare "echoPin" as "Input Pin".
    LF.setSpeed(200);
    LB.setSpeed(200);
    RF.setSpeed(200);
    RB.setSpeed(200);
  }

void loop()
  {
    float distance = 0.00;
    float RightDistance = 0.00;
    float LeftDistance = 0.00;
    distance = search();

    if(distance <= 30) // If obstacle found in 40 CM.
      {
         RobotStop(); //Robot Stop
         delay(100);
         servo.write(5);
         delay(300);
         RightDistance = search();
         delay(100);
         servo.write(180);
         delay(300);
```

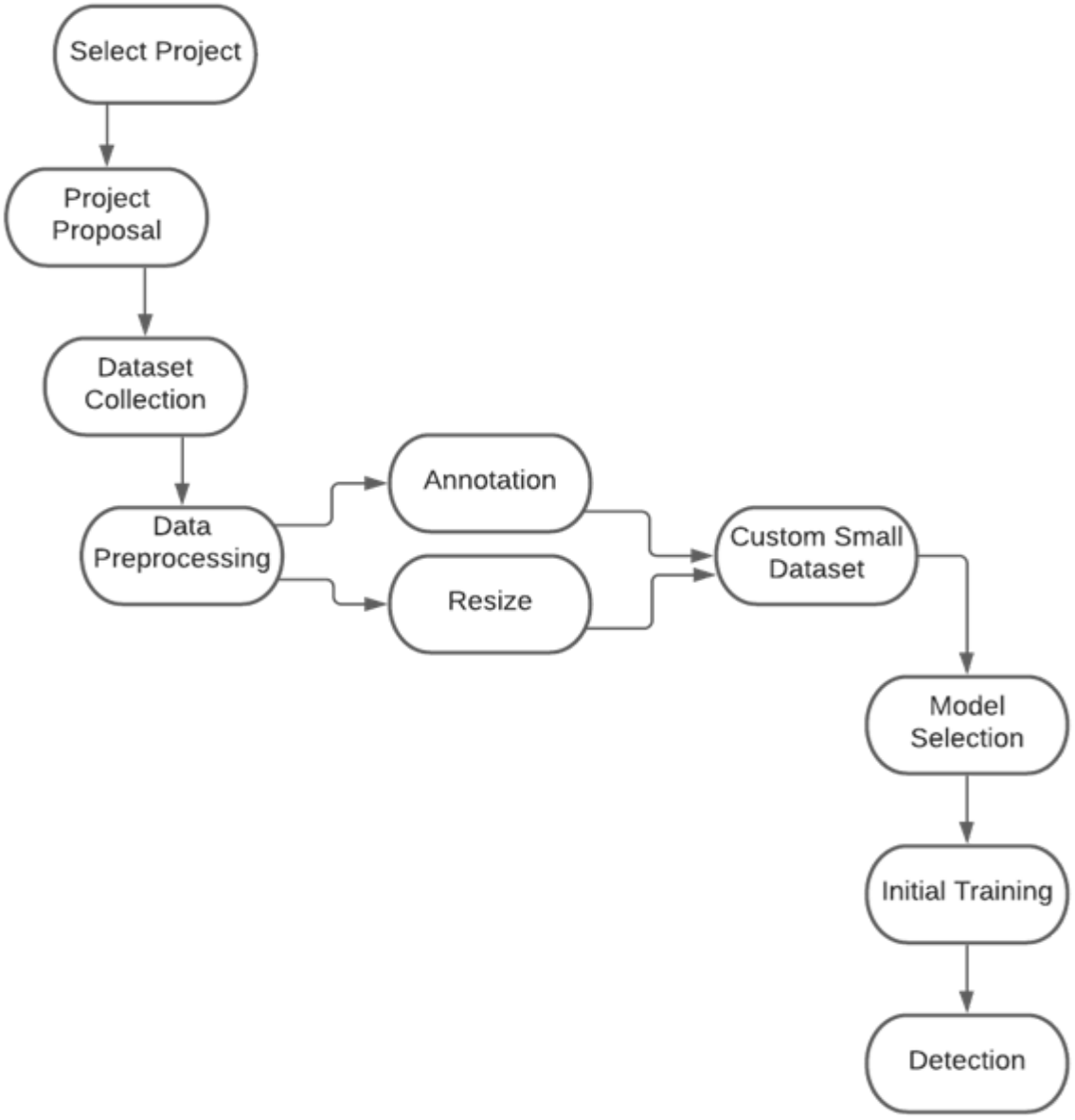
```
      LeftDistance = search();
      delay(100);
      servo.write(90);
      delay(300);
     if(LeftDistance > RightDistance)
      {
        RobotBackward();
        delay(100);
        RobotStop();
        delay(100);
        RobotLeft();
        delay(500);
        RobotStop();
        delay(100);
      }
     else
       {
        RobotBackward();
        delay(100);
        RobotStop();
        delay(100);
        RobotRight();
        delay(500);
        RobotStop();
        delay(100);
       }
   }
else
{
```

```
    RobotForward();
  }
}


float search(void)
  {
    float duration = 0.00;            // Float type variable declaration
    float CM = 0.00;
    digitalWrite(trigPin, LOW);       // Trig_pin output as OV (Logic Low-Level)
    delayMicroseconds(2);             // Delay for 2 us
    //Send 10us High Pulse to Ultrasonic Sonar Sensor "trigPin"
    digitalWrite(trigPin, HIGH);      // Trig_pin output as 5V (Logic High-Level)
    delayMicroseconds(10);            // Delay for 10 us
    digitalWrite(trigPin, LOW);       // Trig_pin output as OV (Logic Low-Level)
     duration = pulseIn(echoPin, HIGH); // Start counting time, upto again "echoPin" back to
Logical "High-Level" and putting the "time" into a variable called "duration"
    CM = (duration / 58.82); //Convert distance into CM.
    return CM;
  }

void RobotForward()
{
  LF.run(FORWARD);
  RF.run(FORWARD);
  LB.run(FORWARD);
  RB.run(FORWARD);
}
void RobotBackward()
```

```
{
  LF.run(BACKWARD);
  LB.run(BACKWARD);
  RF.run(BACKWARD);
  RB.run(BACKWARD);
}
void RobotLeft()
{
  LF.run(BACKWARD);
  LB.run(BACKWARD);
  RF.run(FORWARD);
  RB.run(FORWARD);
}
void RobotRight()
{
  LF.run(FORWARD);
  LB.run(FORWARD);
  RF.run(BACKWARD);
  RB.run(BACKWARD);
}
void RobotStop()
{
  LF.run(RELEASE);
  LB.run(RELEASE);
  RF.run(RELEASE);
  RB.run(RELEASE);
}
```

For Object Detection

3.2 Workflow For Object Detection



Firstly, we spend some time selecting a perfect model based on our interests. Then, we made a project proposal after reading some research papers related to our project. Then we collect a suitable dataset for our dataset. After collecting the dataset, we resize the images. Then, we create a small dataset and train our model with YOLOv7.

3.3 Dataset

The MS COCO dataset is a large-scale object detection, segmentation, and captioning dataset published by Microsoft. Machine Learning and Computer Vision engineers popularly use the COCO dataset for various computer vision projects. Understanding visual scenes is a primary goal of computer vision; it involves recognizing what objects are present, localizing the objects in 2D and 3D, determining the object's attributes, and characterizing the relationship between objects. Therefore, algorithms for object detection and object classification can be trained using the dataset.

COCO (Common Objects in Context)
[COCO - Common Objects in Context (cocodataset.org)]

Figure: Dataset Overview

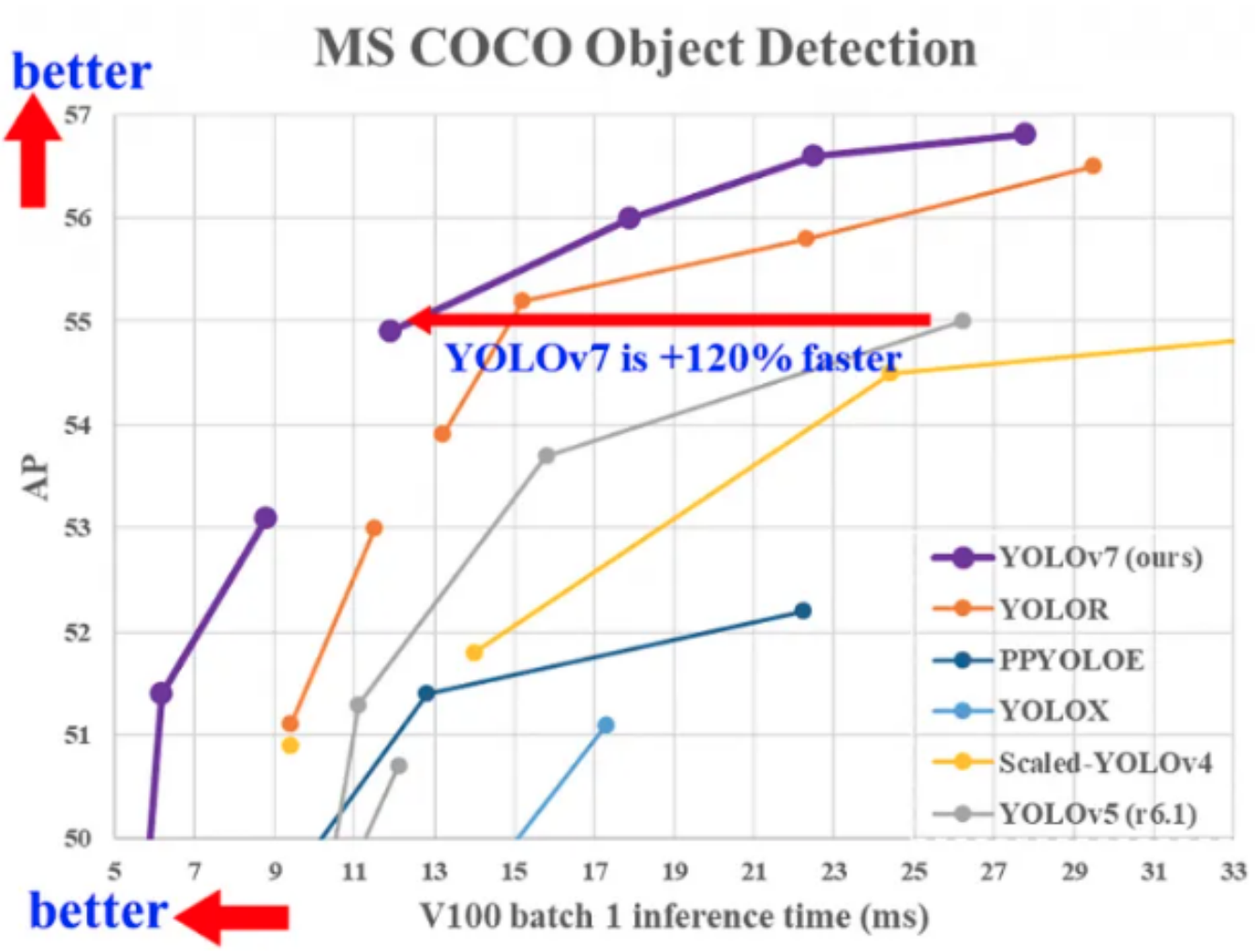


Figure: coco object detection using yolov7

3.4 Preprocessing

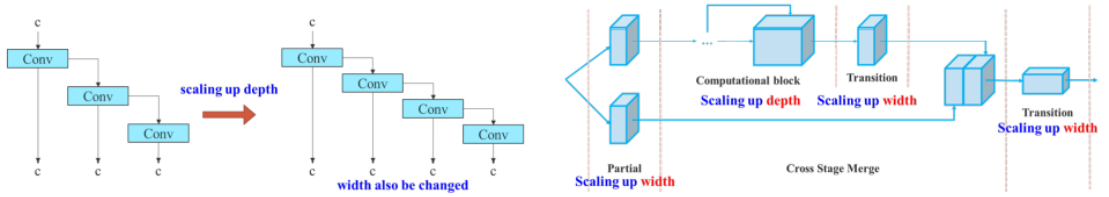
Here we use weights from the COCO dataset then resize it before applying the model.

3.5 Model Architecture

Since our project is based on object detection, we have tried 3 commonly known object detection models which are suitable for our work.

3.5.1 YOLOv7

The object detection method YOLO, which stands for "You Only Look Once," separates images into a grid structure. Each grid cell is in charge of detecting items within itself. Because of its speed and precision, YOLO is one of the most well-known object recognition algorithms. We first tried with a small dataset, then we trained our full dataset with YOLOv5. We also used Coco's pre-trained weight to get better accuracy. Then we saved our model after successfully running 100 epochs and then used the model to detect hazards from our custom video. Using our model, we successfully detect hazards from the videos.



(a) concatenation-based model (b) scaled-up concatenation-based model (c) compound scaling up depth and width for concatenation-based model

Figure 1: Model scaling for concatenation-based models. From (a) to (b), we observe that when depth scaling is performed on concatenation-based models, the output width of a computational block also increases. This phenomenon will cause the input width of the subsequent transmission layer to increase. Therefore, we propose (c), that is, when performing model scaling on concatenation based models, only the depth in a computational block needs to be scaled, and the remainder of the transmission layer is performed with corresponding width scaling.

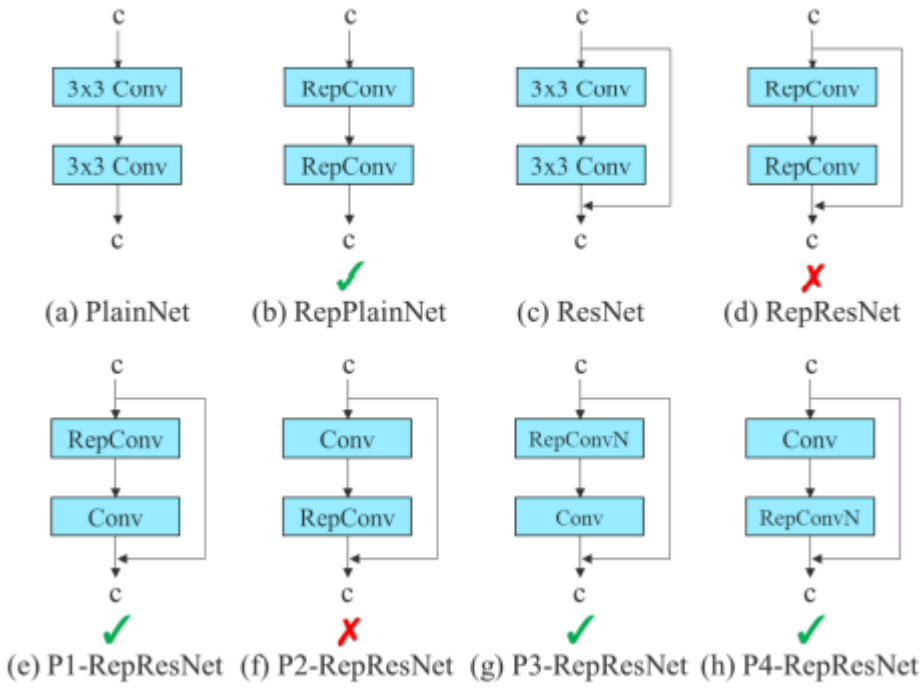


Figure 2: Planned re-parameterized model. In the proposed planned re-parameterized model, we found that a layer with residual or concatenation connections, its RepConv should not have identity connection. Under these circumstances, it can be replaced by RepConvN that contains no identity connections.

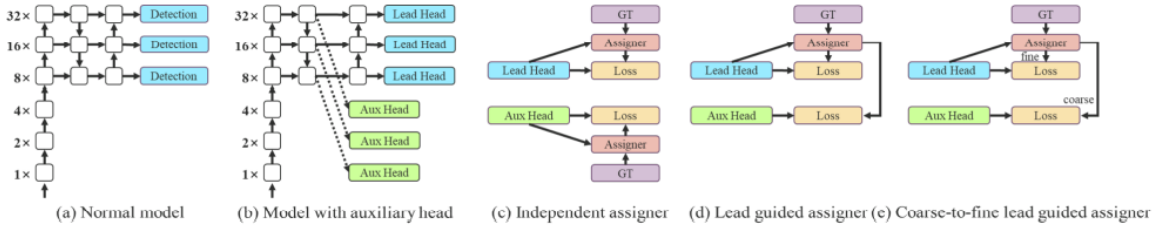


Figure 3: Coarse for auxiliary and fine for lead head label assigner. Compared with the normal model (a), the schema in (b) has an auxiliary head. Different from the usual independent label assigner (c), we propose (d) lead head guided label assigner and (e) coarse-to-fine lead head guided label assigner. The proposed label assigner is optimized by lead head prediction and the ground truth to get the labels of training lead head and auxiliary head at the same time. The detailed coarse-to-fine implementation method and constraint design details will be elaborated in Appendix.

3.5.2 Coding in Colab

```
# Webcam Helper Functions
# import dependencies
from IPython.display import display, Javascript, Image
from google.colab.output import eval_js
from google.colab.patches import cv2_imshow
from base64 import b64decode, b64encode
import PIL
import io
import html
# function to convert the JavaScript object into an OpenCV image
def js_to_image(js_reply):
  """
  Params:
        js_reply: JavaScript object containing image from webcam
  Returns:
        img: OpenCV BGR image
  """
  # decode base64 image
  image_bytes = b64decode(js_reply.split(',')[1])
  # convert bytes to numpy array
  jpg_as_np = np.frombuffer(image_bytes, dtype=np.uint8)
  # decode numpy array into OpenCV BGR image
  img = cv2.imdecode(jpg_as_np, flags=1)

  return img

# function to convert OpenCV Rectangle bounding box image into base64 byte string to be
overlayed on video stream
def bbox_to_bytes(bbox_array):
```

```
  """
  Params:
        bbox_array: Numpy array (pixels) containing rectangle to overlay on video stream.
  Returns:
       bytes: Base64 image byte string
  """
  # convert array into PIL image
  bbox_PIL = PIL.Image.fromarray(bbox_array, 'RGBA')
  iobuf = io.BytesIO()
  # format bbox into png for return
  bbox_PIL.save(iobuf, format='png')
  # format return string
  bbox_bytes = '{}'.format((str(b64encode(iobuf.getvalue()), 'utf-8')))

  return bbox_bytes


# JavaScript to properly create our live video stream using our webcam as input
def video_stream():
  js = Javascript('''
    var video;
    var div = null;
    var stream;
    var captureCanvas;
    var imgElement;
    var labelElement;

    var pendingResolve = null;
    var shutdown = false;
```

```
function removeDom() {
   stream.getVideoTracks()[0].stop();
   video.remove();
   div.remove();
   video = null;
   div = null;
   stream = null;
   imgElement = null;
   captureCanvas = null;
   labelElement = null;
}

function onAnimationFrame() {
  if (!shutdown) {
    window.requestAnimationFrame(onAnimationFrame);
  }
  if (pendingResolve) {
    var result = "";
    if (!shutdown) {
      captureCanvas.getContext('2d').drawImage(video, 0, 0, 640, 480);
      result = captureCanvas.toDataURL('image/jpeg', 0.8)
    }
    var lp = pendingResolve;
    pendingResolve = null;
    lp(result);
  }
}

async function createDom() {
  if (div !== null) {
```

```
  return stream;
}

div = document.createElement('div');
div.style.border = '2px solid black';
div.style.padding = '3px';
div.style.width = '100%';
div.style.maxWidth = '600px';
document.body.appendChild(div);

const modelOut = document.createElement('div');
modelOut.innerHTML = "<span>Status:</span>";
labelElement = document.createElement('span');
labelElement.innerText = 'No data';
labelElement.style.fontWeight = 'bold';
modelOut.appendChild(labelElement);
div.appendChild(modelOut);

video = document.createElement('video');
video.style.display = 'block';
video.width = div.clientWidth - 6;
video.setAttribute('playsinline', '');
video.onclick = () => { shutdown = true; };
stream = await navigator.mediaDevices.getUserMedia(
    {video: { facingMode: "environment"}});
div.appendChild(video);

imgElement = document.createElement('img');
imgElement.style.position = 'absolute';
imgElement.style.zIndex = 1;
```

```
  imgElement.onclick = () => { shutdown = true; };
  div.appendChild(imgElement);

  const instruction = document.createElement('div');
  instruction.innerHTML =
     '<span style="color: red; font-weight: bold;">' +
     'When finished, click here or on the video to stop this demo</span>';
  div.appendChild(instruction);
  instruction.onclick = () => { shutdown = true; };

  video.srcObject = stream;
  await video.play();

  captureCanvas = document.createElement('canvas');
  captureCanvas.width = 640; //video.videoWidth;
  captureCanvas.height = 480; //video.videoHeight;
  window.requestAnimationFrame(onAnimationFrame);

  return stream;
}
async function stream_frame(label, imgData) {
  if (shutdown) {
    removeDom();
    shutdown = false;
    return '';
  }

  var preCreate = Date.now();
  stream = await createDom();
```

```
    var preShow = Date.now();
    if (label != "") {
      labelElement.innerHTML = label;
    }

    if (imgData != "") {
      var videoRect = video.getClientRects()[0];
      imgElement.style.top = videoRect.top + "px";
      imgElement.style.left = videoRect.left + "px";
      imgElement.style.width = videoRect.width + "px";
      imgElement.style.height = videoRect.height + "px";
      imgElement.src = imgData;
    }

    var preCapture = Date.now();
    var result = await new Promise(function(resolve, reject) {
      pendingResolve = resolve;
    });
    shutdown = false;

    return {'create': preShow - preCreate,
            'show': preCapture - preShow,
            'capture': Date.now() - preCapture,
            'img': result};
  }
  '')

  display(js)

def video_frame(label, bbox):
```

```
  data = eval_js('stream_frame("{}", "{}")'.format(label, bbox))
  return data
```

#Start Streaming Video From Webcam

```
# start streaming video from webcam
video_stream()
# label for video
label_html = 'Capturing...'
# initialze bounding box to empty
bbox = ''
count = 0

with torch.no_grad():
  weights, imgsz = opt['weights'], (480,640)
  set_logging()
  device = select_device(opt['device'])
  half = device.type != 'cpu'
  model = attempt_load(weights, map_location=device)  # load FP32 model
  stride = int(model.stride.max())  # model stride

  if half:
    model.half()

  names = model.module.names if hasattr(model, 'module') else model.names
  colors = [[random.randint(0, 255) for _ in range(3)] for _ in names]
  if device.type != 'cpu':
    model(torch.zeros(1, 3, imgsz[0], imgsz[1]).to(device).type_as(next(model.parameters())))
  classes = None
  if opt['classes']:
    classes = []
```

```
    for class_name in opt['classes']:

      classes.append(names.index(class_name))

  if classes:

    classes = [i for i in range(len(names)) if i not in classes]

  while True:
    js_reply = video_frame(label_html, bbox)
    if not js_reply:
        break

    img0 = js_to_image(js_reply["img"])
    bbox_array = np.zeros([480,640,4], dtype=np.uint8)
    img = letterbox(img0, imgsz, stride=stride)[0]
    img = img[:, :, ::-1].transpose(2, 0, 1)  # BGR to RGB, to 3x416x416
    img = np.ascontiguousarray(img)
    img = torch.from_numpy(img).to(device)
    img = img.half() if half else img.float()  # uint8 to fp16/32
    img /= 255.0  # 0 - 255 to 0.0 - 1.0
    if img.ndimension() == 3:
      img = img.unsqueeze(0)

    # Inference
    t1 = time_synchronized()
    pred = model(img, augment= False)[0]

    # Apply NMS
    pred = non_max_suppression(pred, opt['conf-thres'], opt['iou-thres'], classes= classes,
agnostic= False)
```

```
t2 = time_synchronized()
for i, det in enumerate(pred):
  s = ''
  s += '%gx%g ' % img.shape[2:]  # print string
  gn = torch.tensor(img0.shape)[[1, 0, 1, 0]]
  if len(det):
    det[:, :4] = scale_coords(img.shape[2:], det[:, :4], img0.shape).round()

    for c in det[:, -1].unique():
      n = (det[:, -1] == c).sum()  # detections per class
      s += f"{n}  {names[int(c)]}{'s' * (n > 1)}, "  # add to string

    for *xyxy, conf, cls in reversed(det):

      label = f'{names[int(cls)]}  {conf:.2f}'
      plot_one_box(xyxy, bbox_array, label=label, color=colors[int(cls)], line_thickness=3)

bbox_array[:,:,3] = (bbox_array.max(axis = 2) > 0 ).astype(int) * 255
bbox_bytes = bbox_to_bytes(bbox_array)

bbox = bbox_bytes
```

CHAPTER 4: INTEGRATION WITH RASPBERRY PI

Raspberry Pi is a series of small single-board computers developed in the United Kingdom by the Raspberry Pi Foundation in association with Broadcom. The Raspberry Pi project originally leaned towards the promotion of teaching basic computer science in schools and in developing countries.

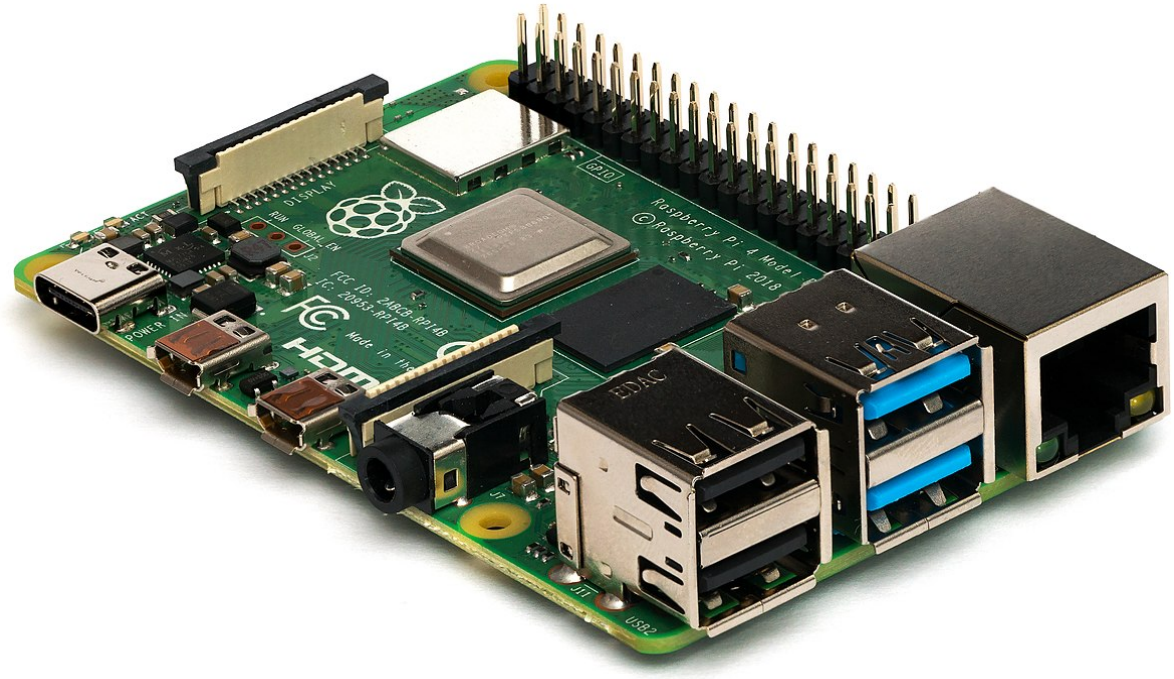


Image: Raspberry Pi

First we setup a webcam and install yolov7 in raspberry pi then enable the camera to get real time object input.

Image: Connection Raspberry Pi, Power Bank and Camera

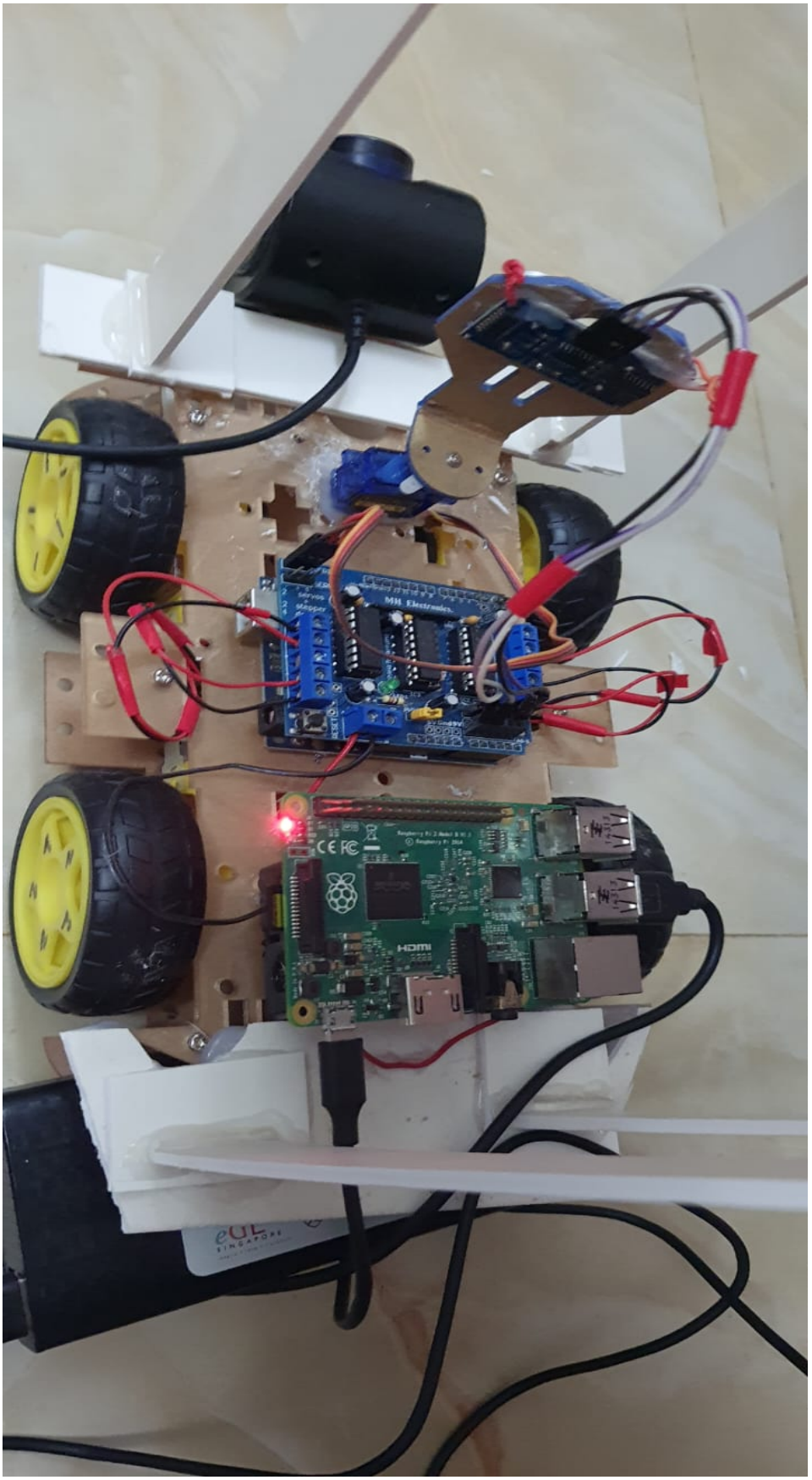


Image: Integration Raspberry Pi with Robot Car

CHAPTER 5: RESULT

5.1 Result Analysis

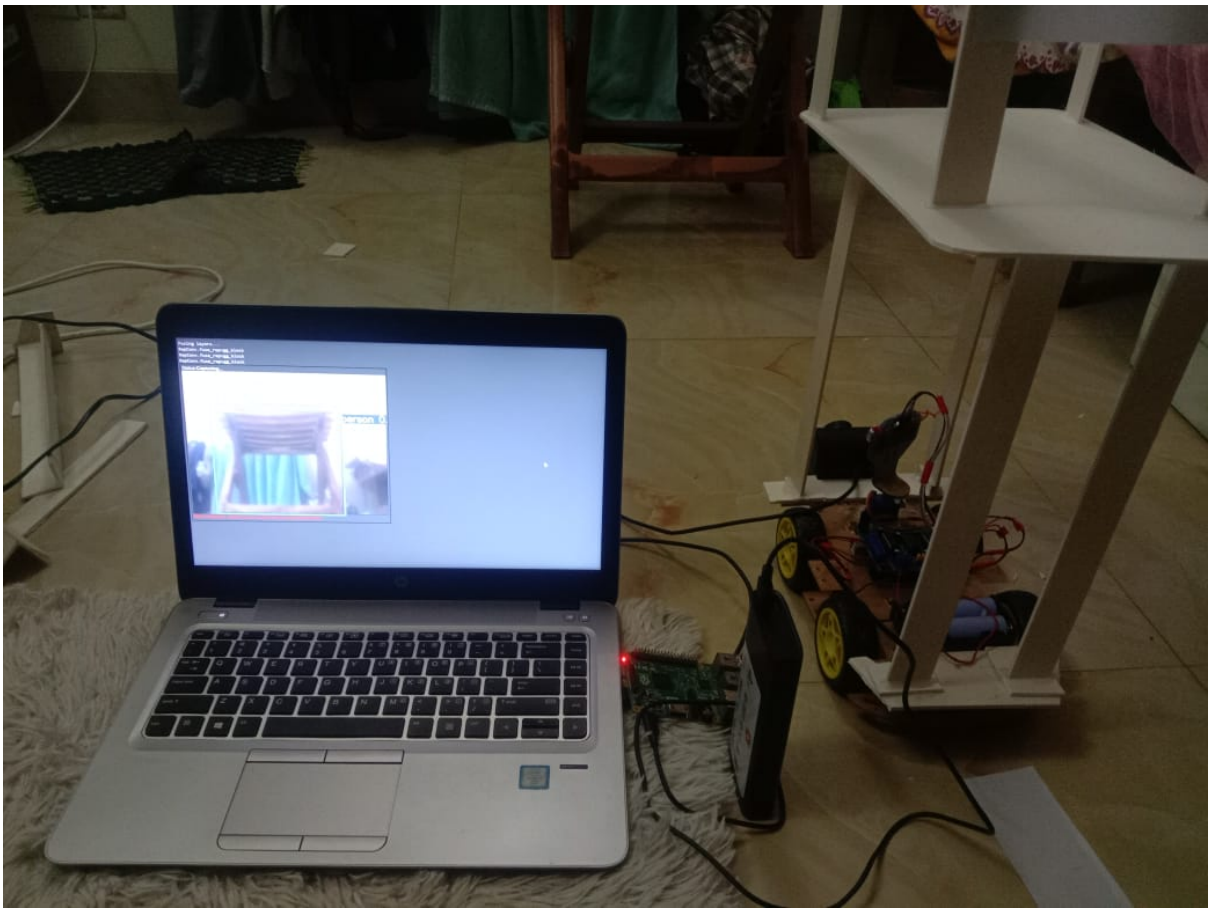


Image: Integration Raspberry Pi with Robot Car

Here we integrate raspberry pi and robot car and run the yolov7 model in raspberry pi we can see that the model works and detect the object also robot car move.

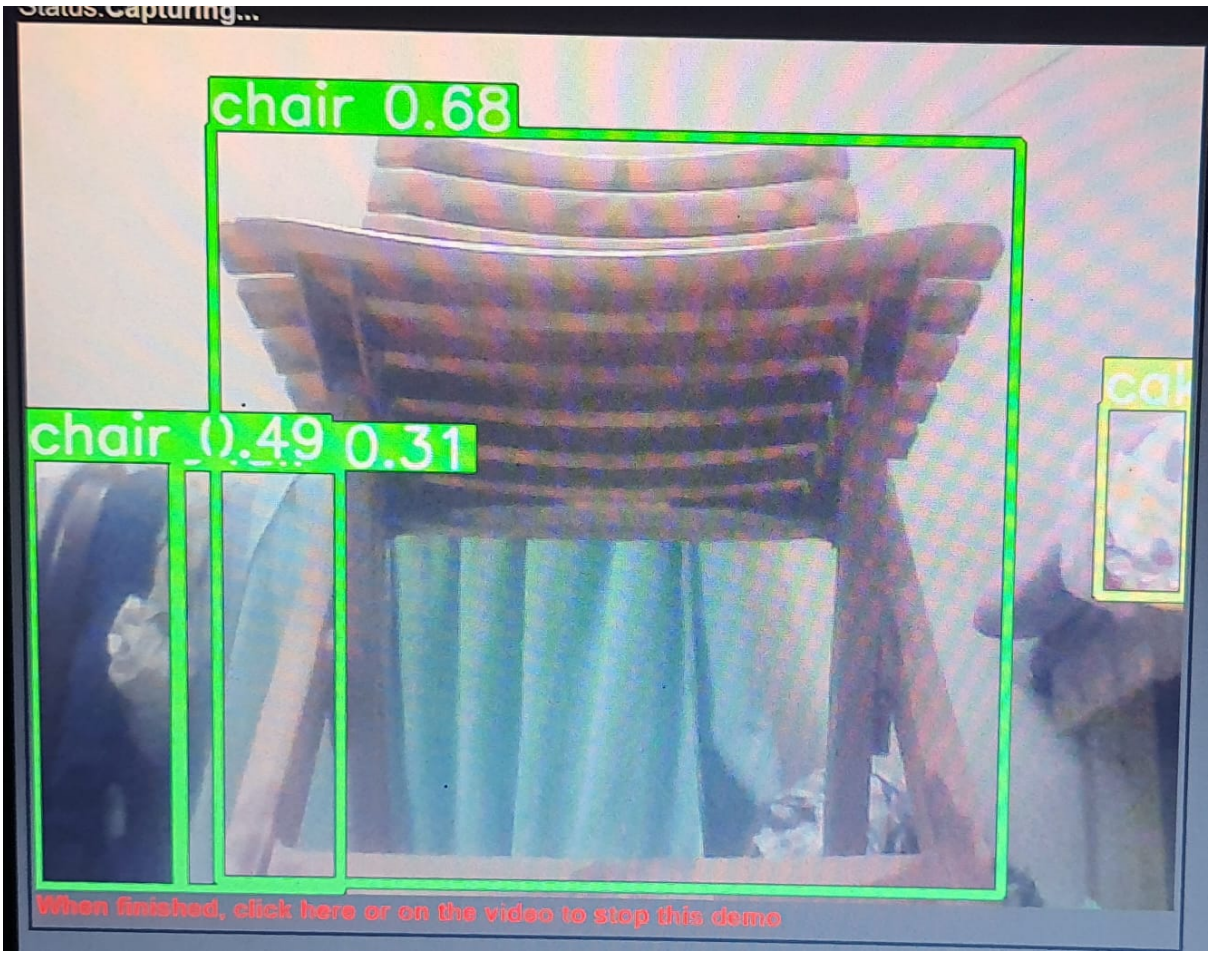


Image: Object Detection Using Raspberry Pi

Here we see that our model perfectly detects the object with good accuracy.

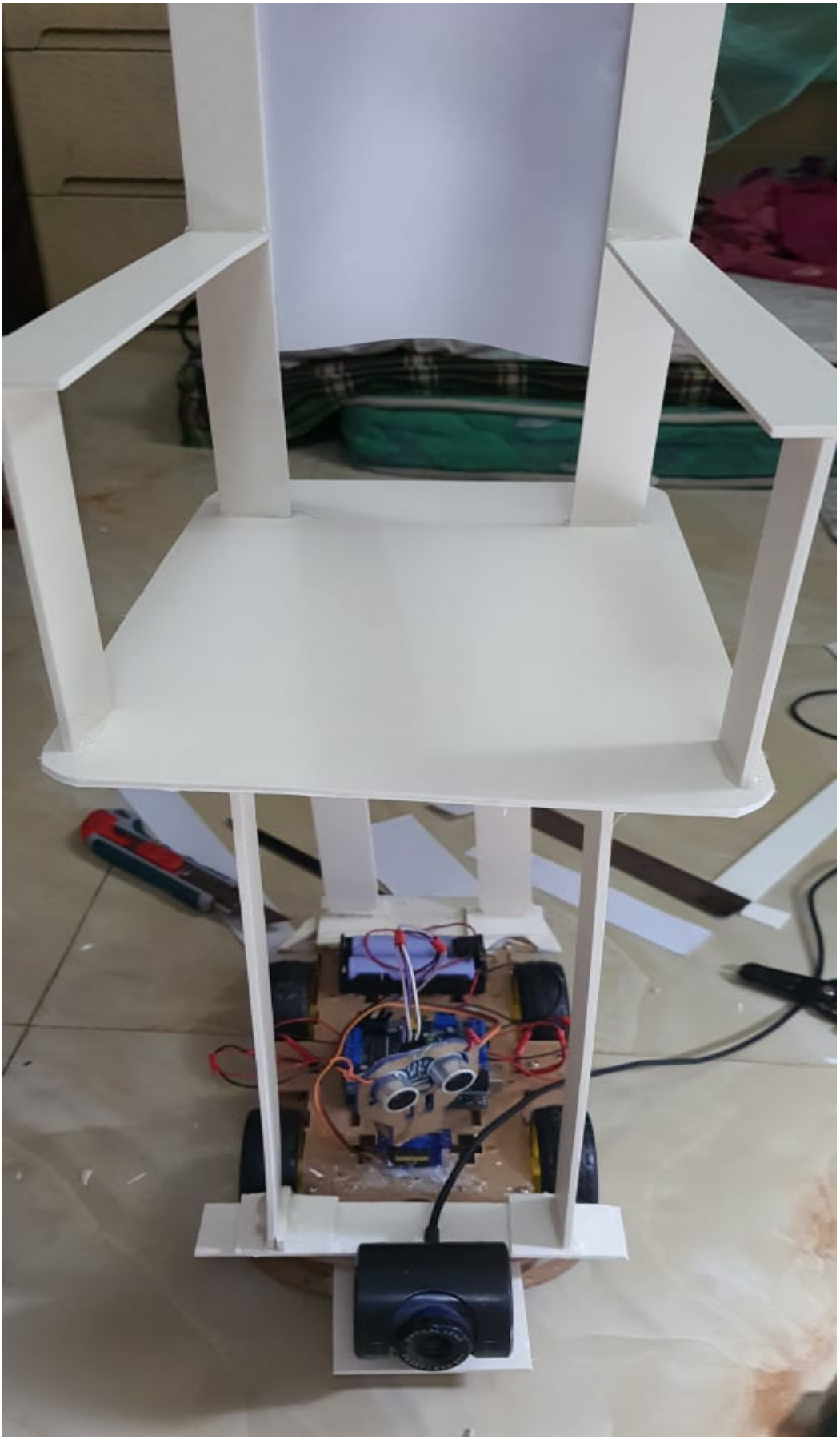


Image: Setup Prototype Chair

4.2 FINAL RESULTS

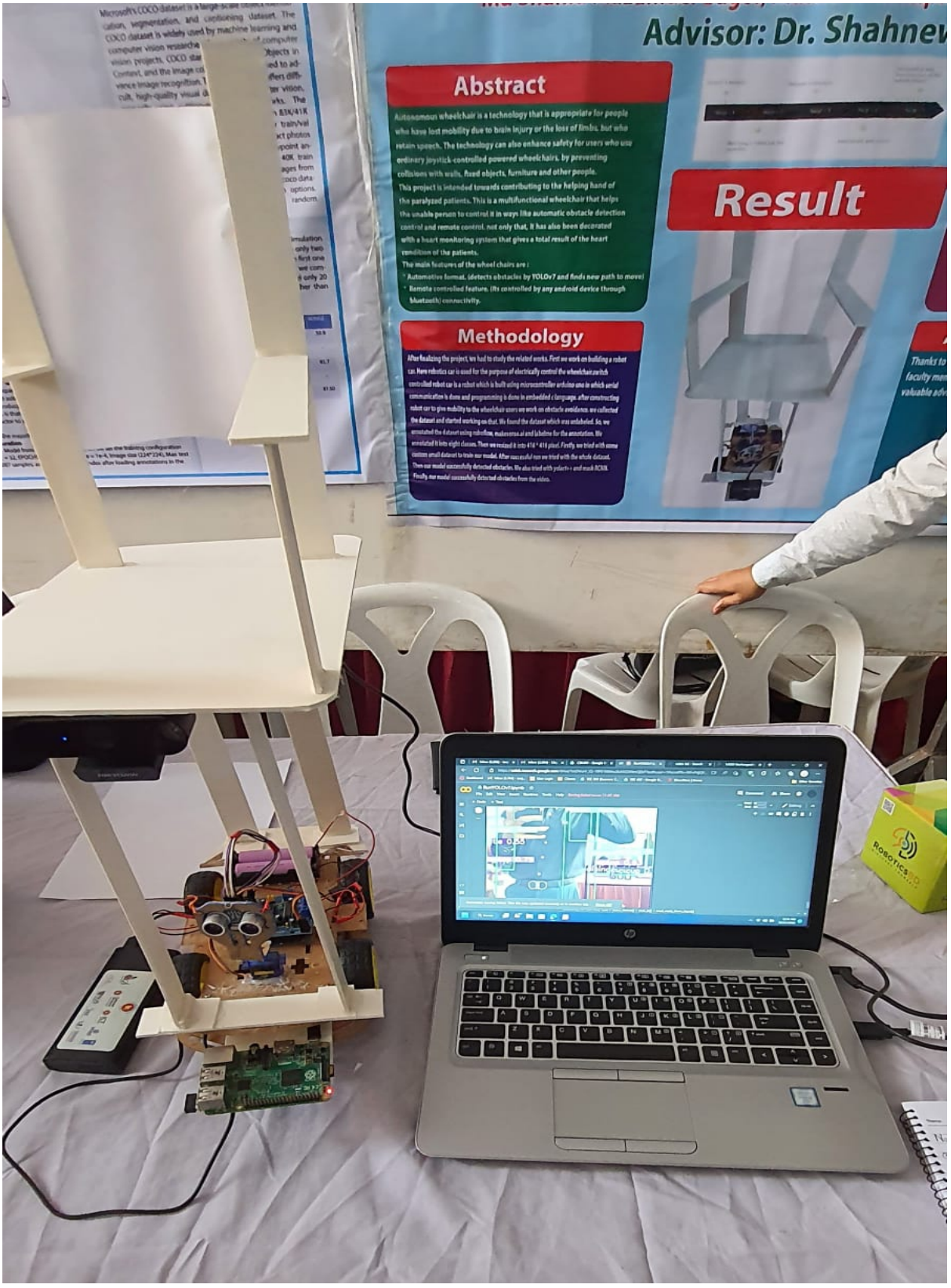


Image: Final Setup of The Project

CHAPTER 5: CONCLUSION

5.1 Discussion

This study presents a prototype version of a wheelchair that detects obstacles for autonomous movement; this could be a promising future for wheelchair technology. The design features simplicity and flexibility; it depends on the commercially available off-the-shelf components. Open-source components and software may enable the community to push this design further as new techniques and approaches become available. In our case, we build a prototype model, we rent a Raspberry Pi for 2000 BDT, bought a Robot Chassis Kit 1200 BDT, HC-05 Bluetooth Module for 345 BDT, Arduino Uno R3 1100 BDT, L298 H-Bridge Dual Motor 550 BDT, Soldering Kit, Board, Wire 1175 BDT, 3300Mah Lipo Battery 2000 BDT, 1080p Camera 1000 BDT, Power Bank 400 BDTand Glue Gun 150 BDT, and the total cost is 9370 BDT. Here we have used cameras to get front views that will help to detect if any object comes in front of the chair. If you find any object then the wheelchair automatically stops. If there is no way to go to the front side, then the system moves left or right and finds the way to go. Another benefit of this system is that it stores the data so if a desired person is lost, anyone can find him/her by analyzing the stored data. We have used 1MP cameras. More work on efficient model development for object detection needs to be done to ensure low power and robust operation. The object detection model that will run the wheelchair needs to be perfected. This study was limited to obstacle detection, wheelchair operating systems are required to introduce faster movement and automatic course correction.

5.2 Costing

Name of Components	Quantity	Product Price(BDT)
Robot car chassis Kit	1	1200
HC -05 Bluetooth Model	1	345
Arduino Uno R3	1	1,100
L298 H-Bridge Dual Motor	4	550
Soldering kit ,Board ,Wire	As per required	1,175
Raspberry Pi	1	2000
3300Mah lipo battery	1	2000
Glue gun	1	1000
Power Bank	1	400
1080p camera	1	500
Total		**10,270 BDT**

5.3 Summary

In our project we intended to contribute to the helping hand of the paralyzed patients. We wants to develop such a project which will extremely cost effective on the context of our countries economy.It can be used in larger scale in various hospitals and nursing homes having critical patients .It is very much convenient than always keeping human monitoring and effort.It keeps a good track of the heart conditions of the patients

5.4 Future Work

In this project we build a prototype wheelchair with AI technology that can detect objects and aware the disabled person. In future work we can convert it into a real wheelchair for large scale that will help disabled people and impact our social community.

A large human study is required to test the wheelchair with disabled individuals. Beside object avoidance, some other implementations can be done, like voice commands,

brain-computer control, etc., to support the user with low or no mobility of arms to reach the desired destination. Innovations like this mitigate the gap between technology and humanity, arguably the purpose of technology

REFERENCES

1. (PDF) Real-Time Object Detection Using YOLO: A Review..

   (PDF) Real-Time Object Detection Using YOLO: A Review (researchgate.net)

2. CPU Based YOLO: A Real Time Object Detection Algorithm | IEEE Conference Publication | IEEE Xplore.CPU Based YOLO: A Real Time Object Detection Algorithm | IEEE Conference Publication | IEEE Xplore

3. Author Index | IEEE Conference Publication | IEEE Xplore..

   Author Index | IEEE Conference Publication | IEEE Xplore

4. The autonomous mobile robot SENARIO: a sensor aided intelligent navigation system for powered wheelchairs | IEEE Journals & Magazine | IEEE Xplore..

   The autonomous mobile robot SENARIO: a sensor aided intelligent navigation system for powered wheelchairs | IEEE Journals & Magazine | IEEE Xplore