AI Programming & Self-Driving Car

Acknowledgement

- Cytron
- RaspberryPi.org

Hello. I’m Daniel Vong.

Ts. Daniel Vong

BEng (Hons), BSc, CEng, MIET, MCP

Co-Founder & CTO, Wangi Lai PLT

Hello. I’m Tong En.

Lim Tong En

AI Engineering Intern, Wangi Lai PLT

EEE Student in University of Southampton Malaysia

AI Programming & Self-Driving Car

Unit Outline

- Session 1: Introduction to Hardware
- Session 2: Introduction to Python
- Session 3: Computer Vision & Object Recognition
- Session 4: Assembly & Integration of Software and Hardware
- Session 5: Challenge & QnA

AI Programming & Self-Driving Car

Session 3

Learning objectives

- In this session you will learn about:
  - Introduction to Computer Vision
  - Object Recognition Concept
  - Intro to Yolo model
  - Set up & start computer vision
  - Using RPiCam to detect object with `yolov11s` model

Session 3: Computer Vision & Object Recognition

Day 2

Divided into 3 parts

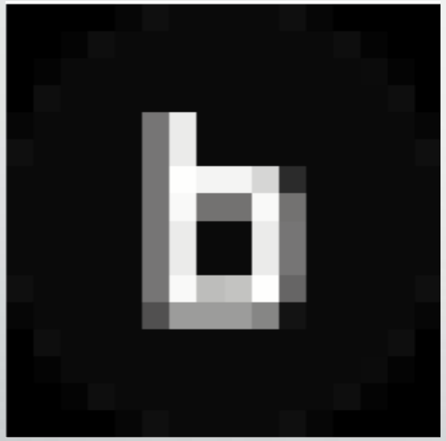
- This session is divided to:
  - Introduction to Computer Vision
  - Switching to Pi Camera
  - Introduction to Lane Keeping Detection

Introduction to Computer Vision

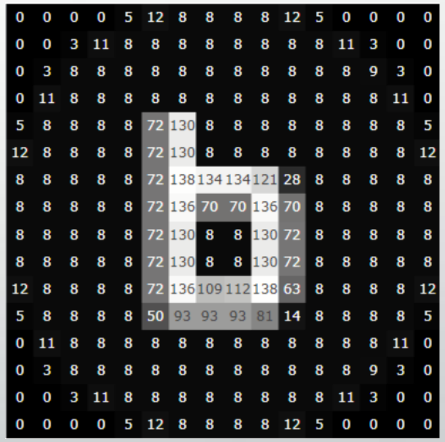
Session 3-1

Computer Vision

- Computer "sees" and understand images the way humans do. However, there are differences:



What human sees:



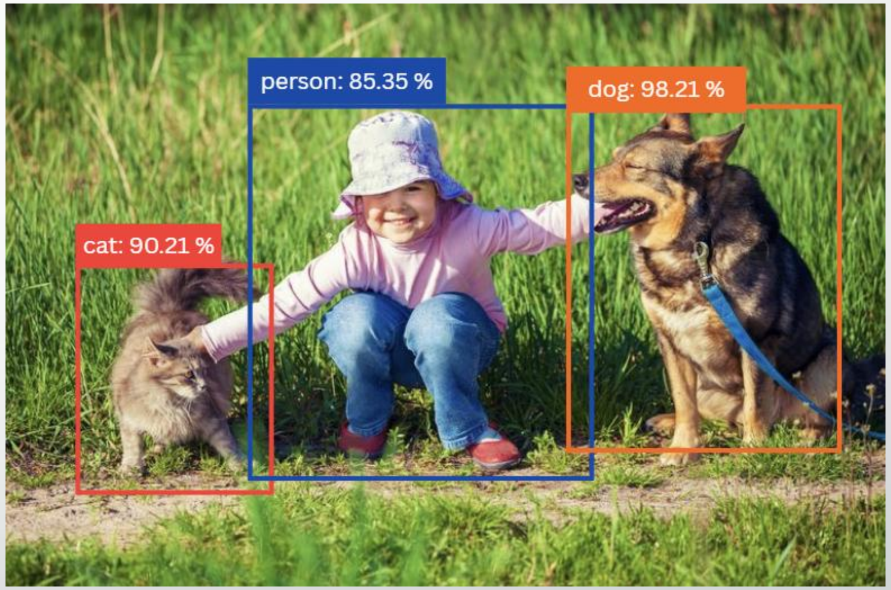
0	0	0	0	5	12	8	8	8	8	12	5	0	0	0	0
0	0	3	11	8	8	8	8	8	8	8	8	11	3	0	0
0	3	8	8	8	8	8	8	8	8	8	8	8	9	3	0
0	11	8	8	8	8	8	8	8	8	8	8	8	8	11	0
5	8	8	8	8	72	130	8	8	8	8	8	8	8	8	5
12	8	8	8	8	72	130	8	8	8	8	8	8	8	8	12
8	8	8	8	8	72	138	134	134	121	28	8	8	8	8	8
8	8	8	8	8	72	136	70	70	136	70	8	8	8	8	8
8	8	8	8	8	72	130	8	8	130	72	8	8	8	8	8
8	8	8	8	8	72	130	8	8	130	72	8	8	8	8	8
12	8	8	8	8	72	136	109	112	138	63	8	8	8	8	12
5	8	8	8	8	50	93	93	93	81	14	8	8	8	8	5
0	11	8	8	8	8	8	8	8	8	8	8	8	8	11	0
0	3	8	8	8	8	8	8	8	8	8	8	8	9	3	0
0	0	3	11	8	8	8	8	8	8	8	8	11	3	0	0
0	0	0	0	5	12	8	8	8	8	12	5	0	0	0	0

What computer sees:

Pre-trained Vision Models

- Yolo
- R-CNN
- RetinaNet
- EfficientDet
- Single Shot MultiBox Detector

What is Object Detection?

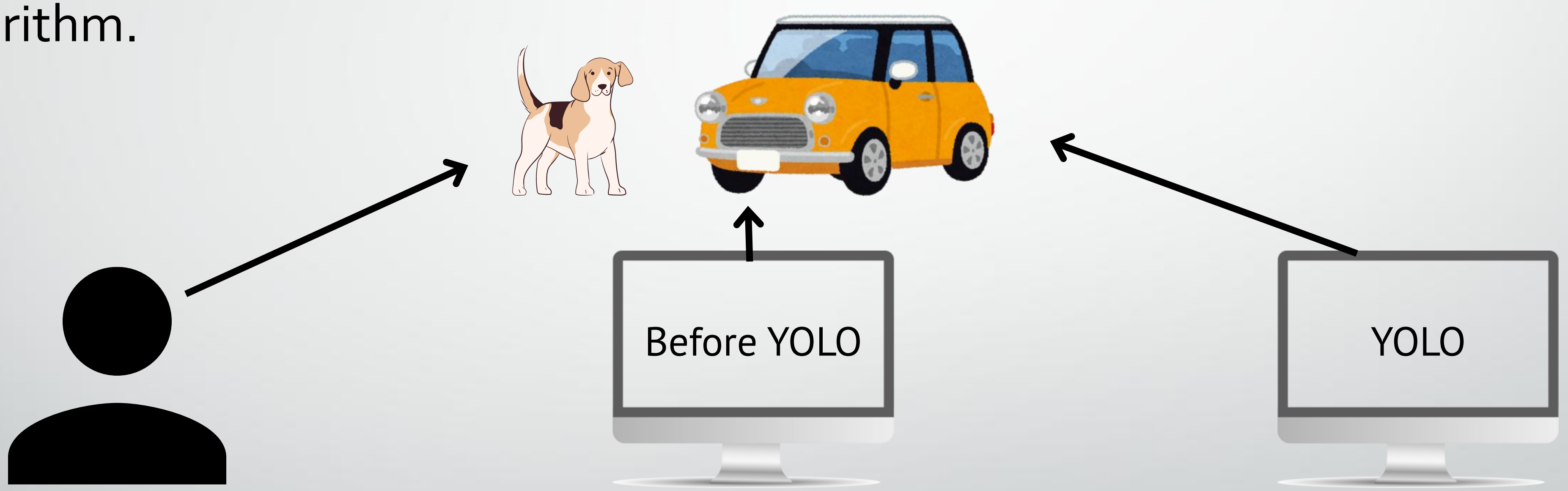


- Person
- Cat
- Dog

- Class Prediction
- Probability Score
- Bounding Box

What is YOLO?

Also known as "You Only Look Once". A popular real-time object detection algorithm.



"There's a dog and a car."

Look at picture multiple times

"Is that a cat? No, this is a dog. No, this is a dog and car…"

Look at picture once

"There's a dog and a car."

Get Started with Object Detection

Training Phase

1. Image tagging
2. Model Training

Detection Phase

1. Use model for detection

YOLO Training Phase (1)

- Image tagging
  - Manually tag classes and bounding boxes
  - Serve as ground truth for classes prediction
  - Example Tool: Label Studio

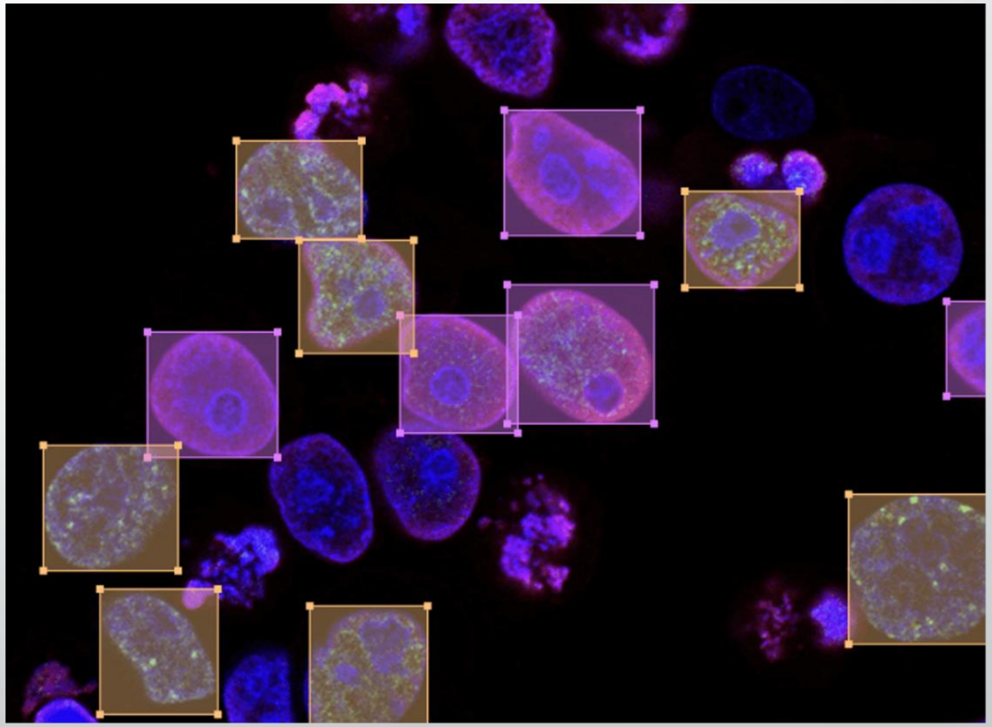


Image from Label Studio (labelstud.io)

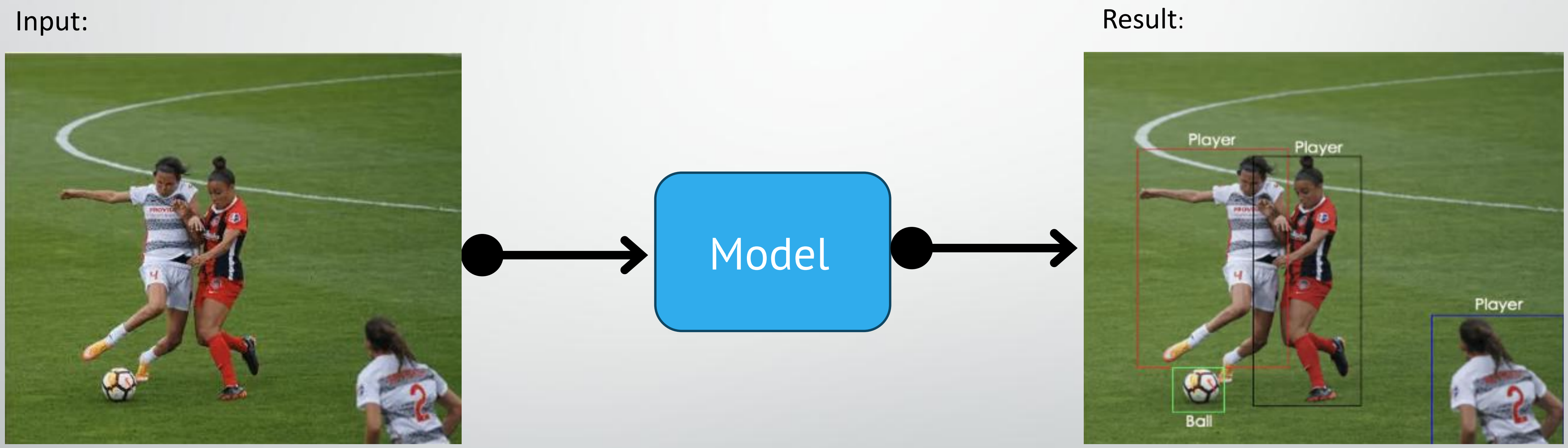
YOLO Training Phase (2)

- Model training and evaluation
  - Pre-trained models loaded to train custom dataset.
    - model = YOLO('yolov8n.pt') *# nano version - fastest*
    - *model = YOLO* ('yolov8s.pt') *# small version - good balance*
    - *model = YOLO* ('yolov8m.pt') *# medium version - more accurate but slower*

YOLO Training Phase (3)

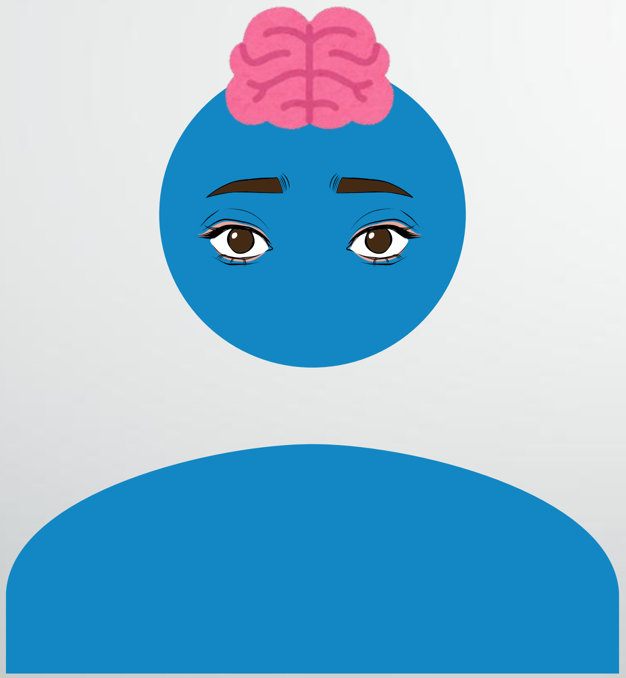
- Model Performance quantifies by:
  - Precision: how many of the detected objects were correct
  - Recall: how many of the actual objects were successfully detected
  - Mean average precision (mAP): combines both to give an overall score (higher is better)

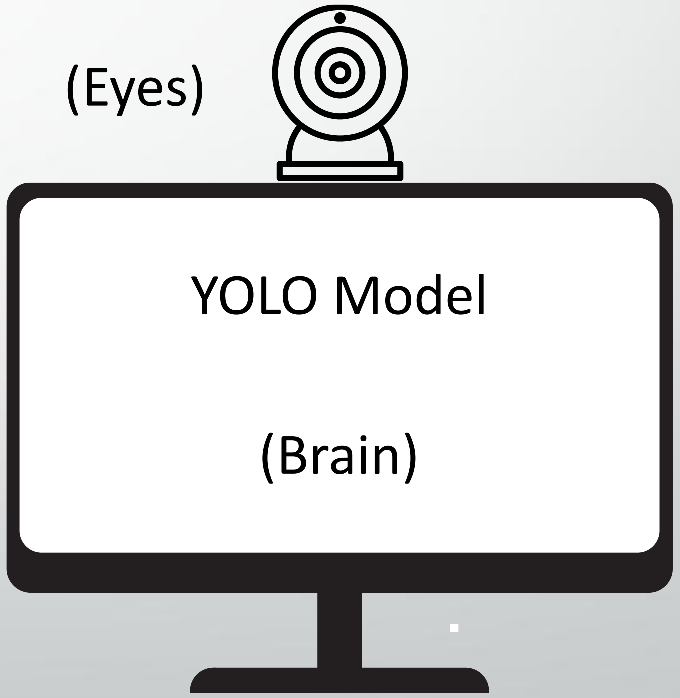
Detection Phase



What is Model?

A digital brain that is trained to recognize any object of interest in images.





Knowledge check (1)

1. What is the main goal of computer vision in robotics?
   a) To enable the robot to see and understand images or video
   b) To allow the robot to hear and speak
   c) To store large amounts of data

2. What does the YOLO model do in a computer vision pipeline?
   a) Compresses image files
   b) Connects to the internet
   c) Detects and classifies objects in real time

Knowledge check (2)

3. You're training a model to detect cats and dogs. You have 1,000 photos, but you notice that some images labeled as 'cat' actually show dogs, and some 'dog' images actually show cats. What will happen to your model, and how do you fix this?
   a) The model will automatically fix the wrong labels during training.
   b) The model will learn to classify cats as dogs and dogs as cats in some cases.
   c) The model will ignore the wrongly labeled images.

Knowledge check (3)

4. You want to train a model to recognize 3 types of fruits: apples, bananas, and oranges. You have 900 apple photos, 50 banana photos, and 50 orange photos. What's wrong with this dataset, and what should you do before training?
   a) The model will perform equally well on all three fruits
   b) The model will be very good at detecting apples but poor at detecting bananas and oranges.
   c) The model will automatically balance the dataset during training

Exercise

Explore Object Detection

https://colab.research.google.com/github/Eagleshot/CustomYOLOModel/blob/main/yolo.ipynb#scrollTo=6_ZfqZ6Id2Th

Summary (Part 1)

- In this part 1, we took our first steps into the exciting world of computer vision:
  - We learned that computer vision enables machines to “see”
  - We discussed how YOLO is different

Switching to Pi Camera

Session 3-2

`scp` to Raspberry Pi (1)

```
scp [local_file_path]
[username]@[hostname_or_ip]:[destination_path]
```

`scp` to Raspberry Pi (2)

- Example 1: Copy `file.txt` to Pi’s home directory
```
scp file.txt raspi@raspberrypi.local:~
```

- `file.txt`: your local file
- `pi`: username on the Pi
- `raspberrypi.local`: Pi's hostname (or IP like 192.168.1.25)
- `~`: target is Pi’s home directory

`scp` to Raspberry Pi (3)

- Example 2: Copy folder to Desktop on Pi
  ```
  scp -r MyFolder raspi@raspberrypi.local:~/Desktop
  ```
- `-r`: recursive copy for folders
- Destination must exist (`~/Desktop/`)

`scp` to Raspberry Pi (4)

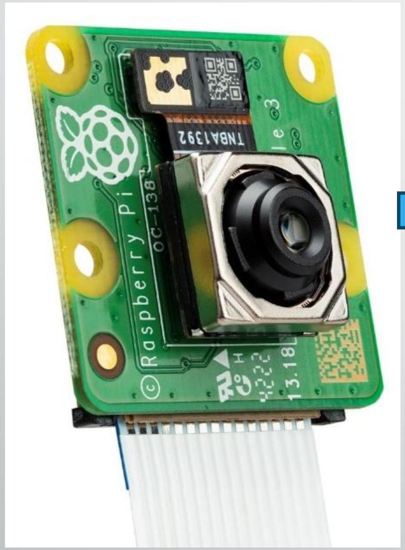
- If using the terminal is not your thing and you need a GUI tool:
    - Use WinSCP: https://winscp.net
    - Use FileZilla: https://filezilla-project.org/

BONUS: `scp` from Raspberry Pi

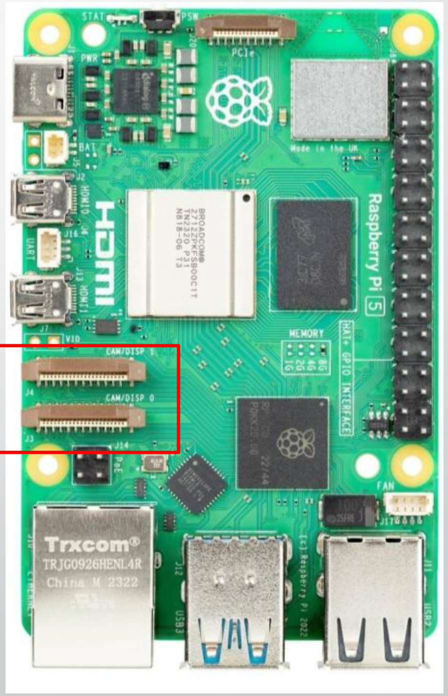
- To pull a file from Pi to your machine:

```
scp pi@raspberrypi.local:/home/pi/myfile.txt .
```

Architecture PiCam and Raspberry Pi



Pi Camera Module



Pi Camera Ribbon input

Get started with PiCam (1)

- Create and activate virtual environment.

```
# Create virtual environment
python -m venv my_env # my_env is name of my venv.
# Activate virtual environment
my_env/Scripts/activate.bat
```

Get started with PiCam (2)

- Install all libraries and dependencies.

```
bash requirements.sh
```

- Enable all connections and Edit firmware config text file.

```
sudo raspi-config
sudo nano /boot/firmware/config.txt
```

Get started with PiCam (3)

- Save and Exit with Ctrl+X then Y.
- Reboot with:

```
sudo reboot
```

- Create a new python file.
- Run the python file.

Test your Pi Camera (1)

- Run the below command in terminal:

```
sudo libcamera-hello
```

Test your Pi Camera (2)

- Integration in code:

```
import cv2
from camera import Camera # Import all required libraries

picam2 = Camera() # Initialize Camera Object

picam2.start_now() # Start and set up Camera Object

while True:
    frame = picam2.capture_frame() # Capture and store frame from the camera

    cv2.imshow("Camera", frame) # Show frame from the camera with help from cv2

    # Exit the program if q is pressed

    if cv2.waitKey(1) == ord("q"):

        break

picam2.stop_capture() # Stop Camera

cv2.destroyAllWindows() # Close all the created windows
```

Test your Pi Camera (3)

- Find more about pi camera here (https://datasheets.raspberrypi.com/camera/picamera2-manual.pdf).

Summary (Part 2)

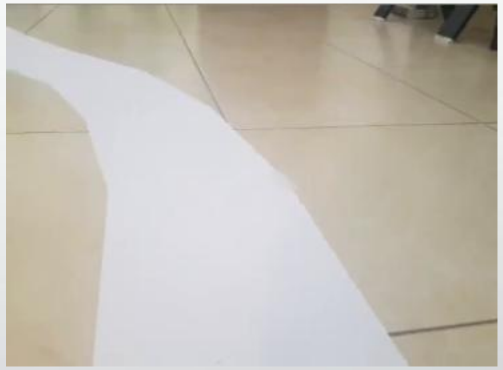
- In this part 2, we took our second step into the exciting world of computer vision:
  - We `scp` our local files to Raspberry Pi
  - We connected the Raspberry Pi Camera to our Raspberry Pi and capture images

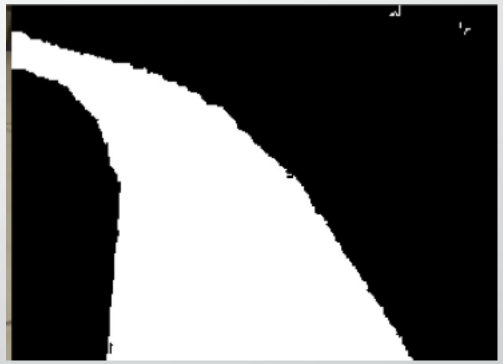
Introduction to Lane Keeping Detection

Session 3-3

How to detect lanes?

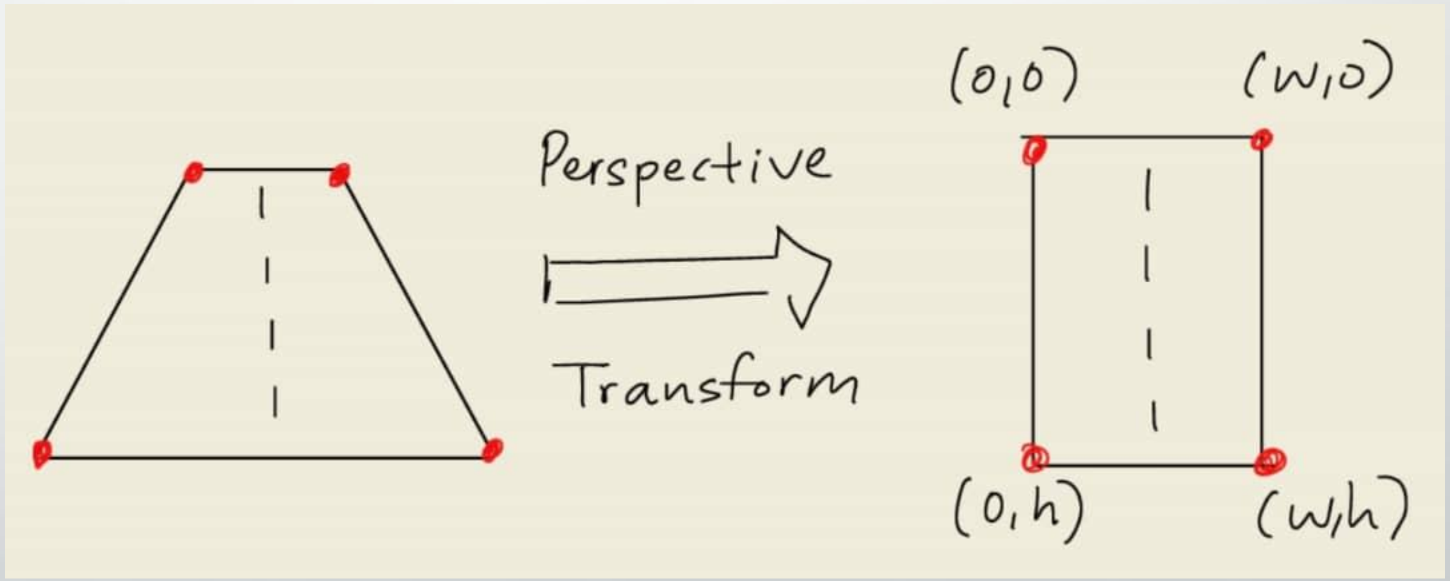
- Extract our road of interest:
  - Create a mask
  - Convert to HSV format
  - Hue, Saturation and Value





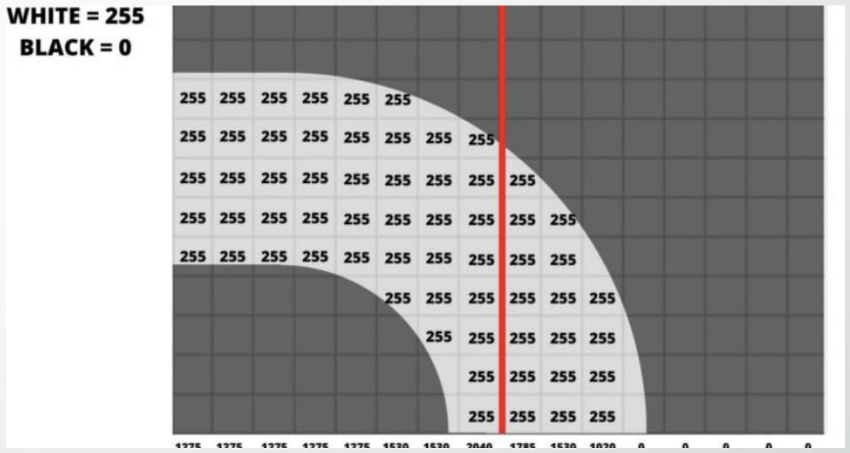
How to detect lanes?

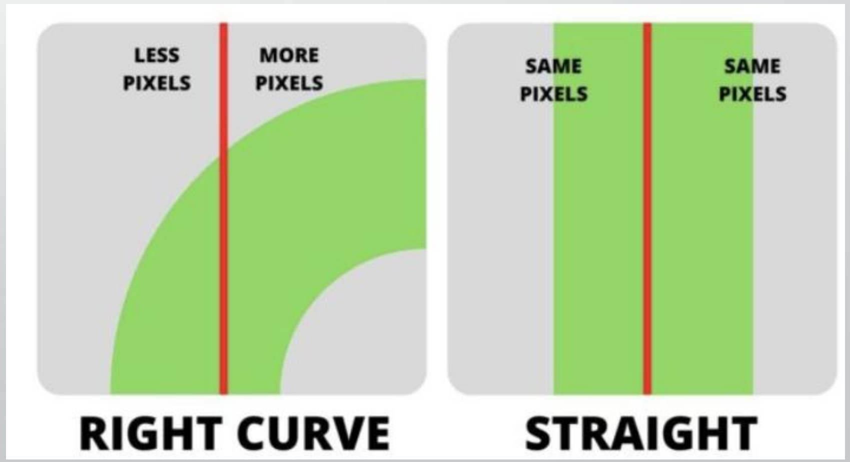
- Change Perspective (Bird Eye View)
  - Crop the lane of interest
  - Efficient direction detection



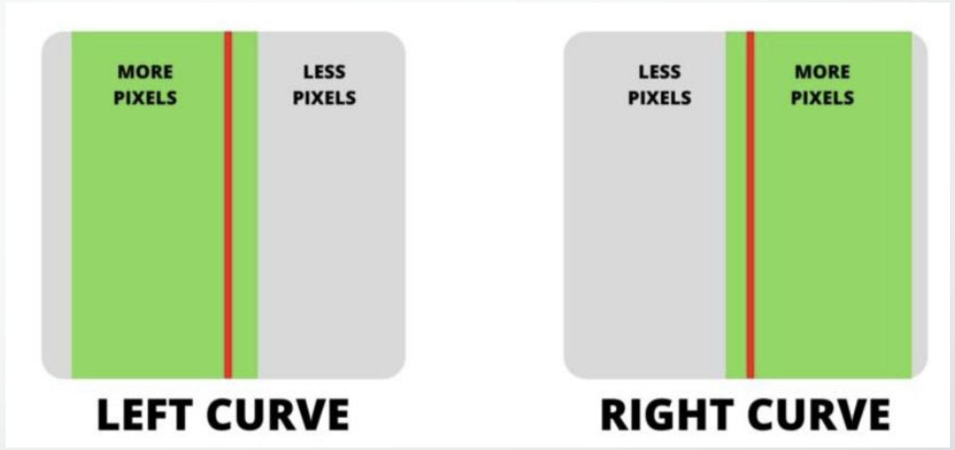
Finding Direction

- Pixel Summation
  - Checks for total pixel value per column
  - Compare the column pixel values
  - Highest pixel value area == curving direction

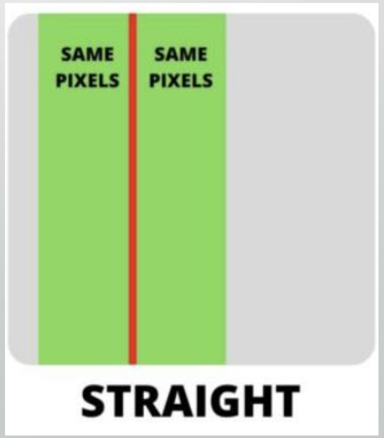




Variable Lane Center

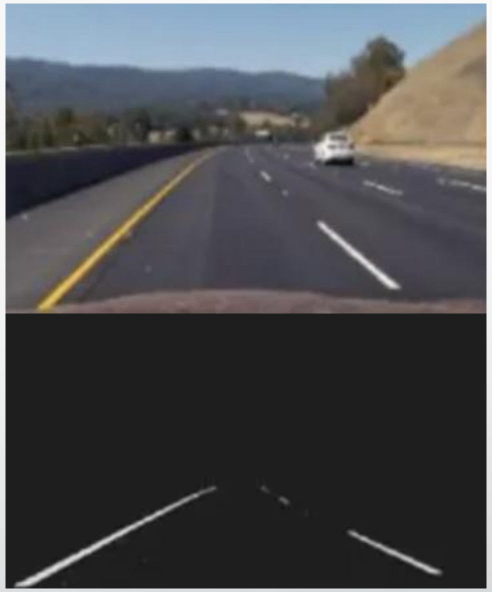


- Bias Driving Direction
  - Not driving at center of lane
  - Mistaken as curving lanes
  - Find the average point for lane of interest

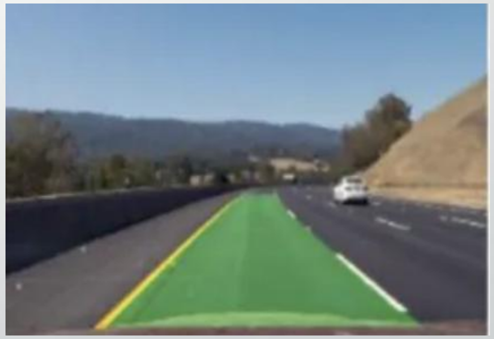


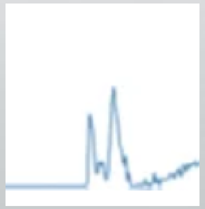
Optimizing Detection

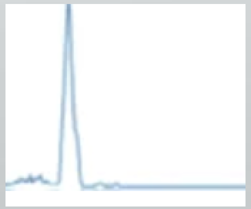
- Histogram Detection
  - Representation of pixel intensities distribution
  - Eliminating low intensity value pixels (noise)



Output:







Knowledge check

1. What does histogram detection help with in lane detection?
   a) It shows where the lane lines are by counting bright pixels in each column.
   b) It detects traffic signs.
   c) It changes the color of the lane lines.

2. What is a common challenge when detecting lanes on real roads?
   a) The lanes are always perfectly straight.
   b) Lighting changes, shadows, and occlusions by other vehicles.
   c) Roads have no lanes.
   d) Cameras cannot capture images at night.

Summary (Part 3)

- In this session, we took our third steps into the exciting world of computer vision:
  - We discussed how Self-Driving Car detects lanes