Self- Driving Car using Computer Vision

Saransh Bhalla

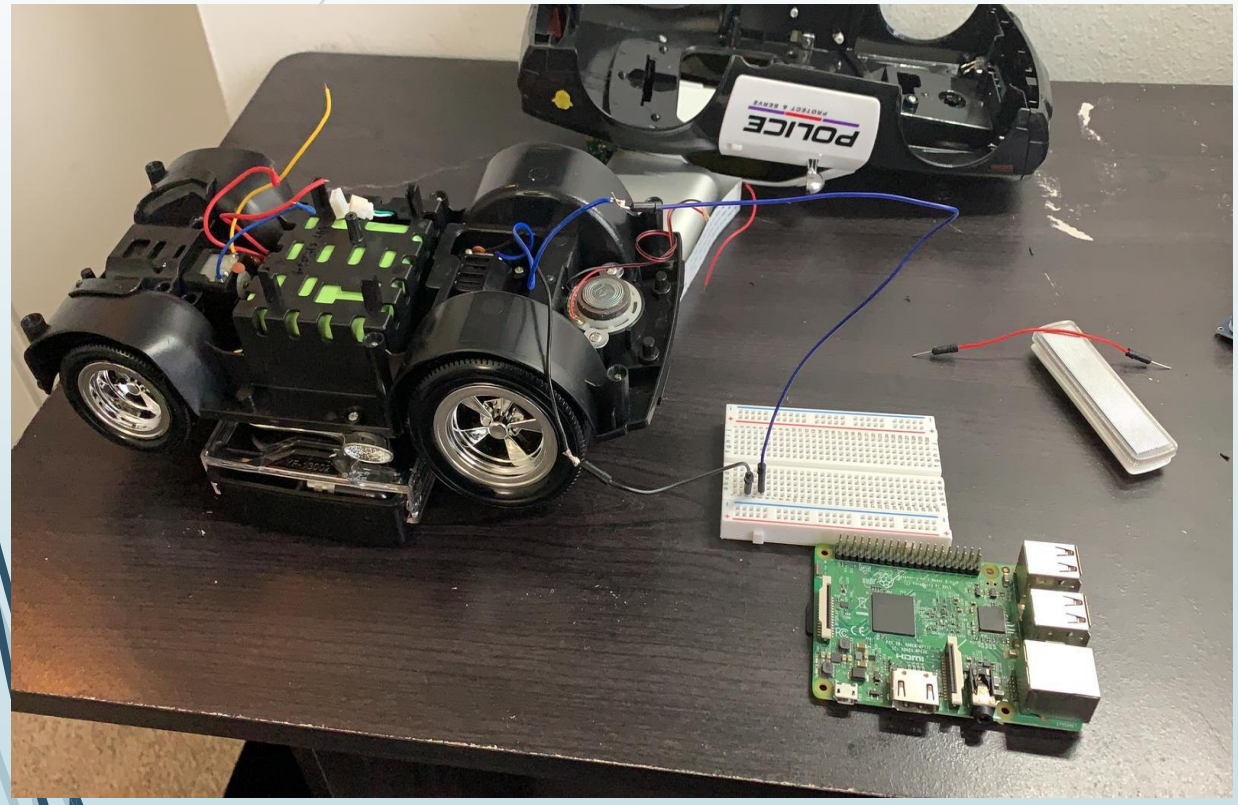
Bini Elsa Paul

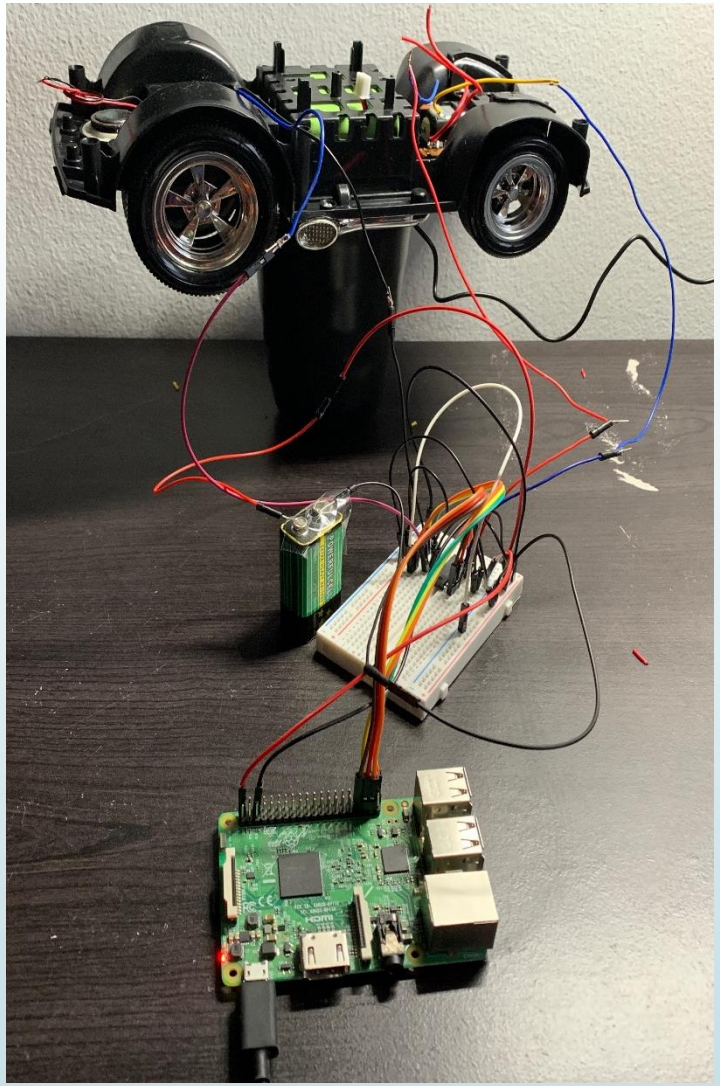
Over View

- Hardware
  - Remote controlled car
  - Raspberry pi
  - Pi camera
  - Bread board
  - Motor
  - Speed controller
  - Power supply
  - IC

- Software
  - Opencv
    - Edge detection techniques
    - Neural Network
  - python

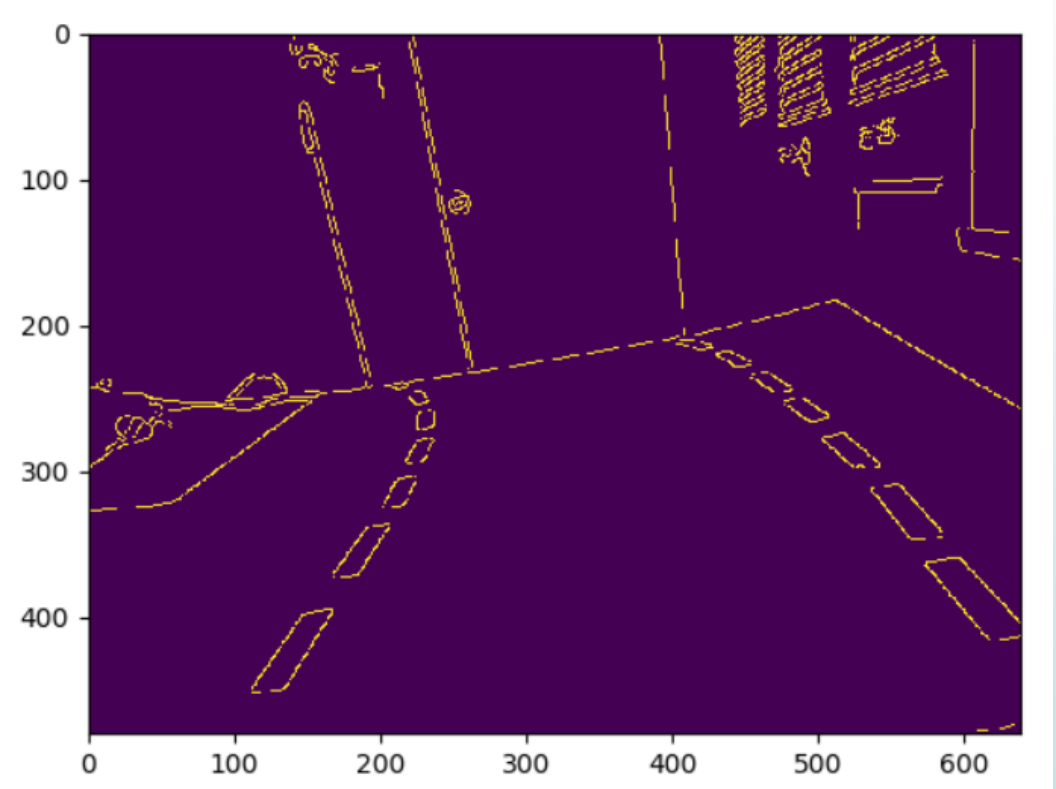
Hardware





OpenCV – Edge detection

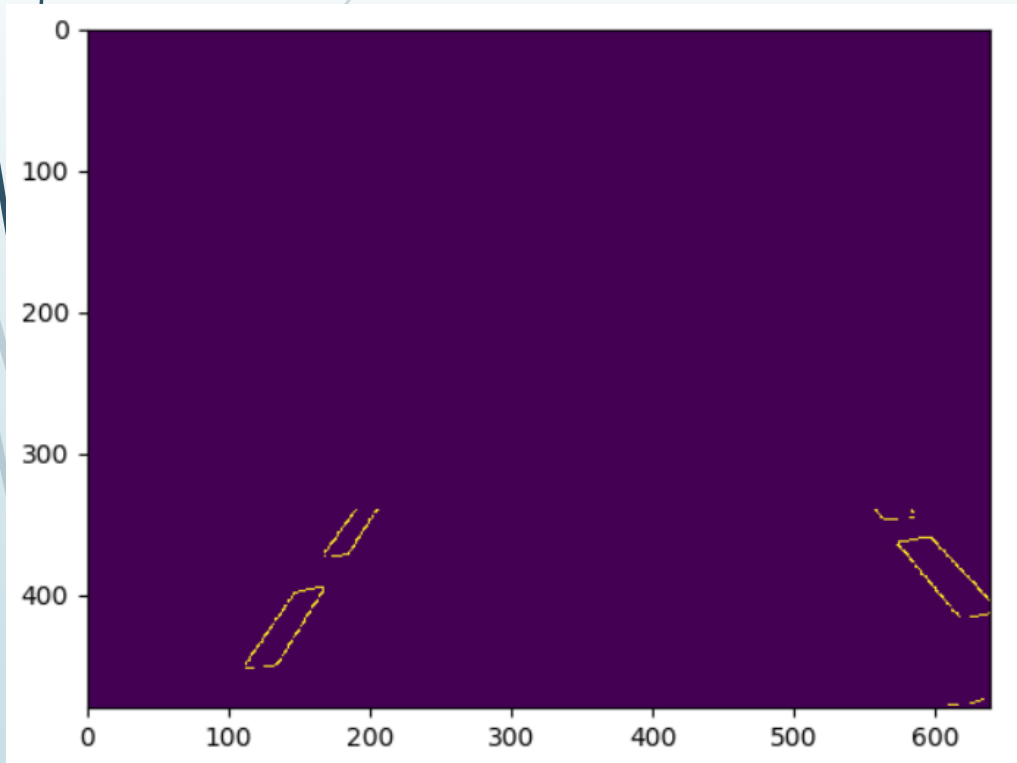
1. RBG to Gray
2. GaussianBlur
3. cv2.Canny



Initial Frame

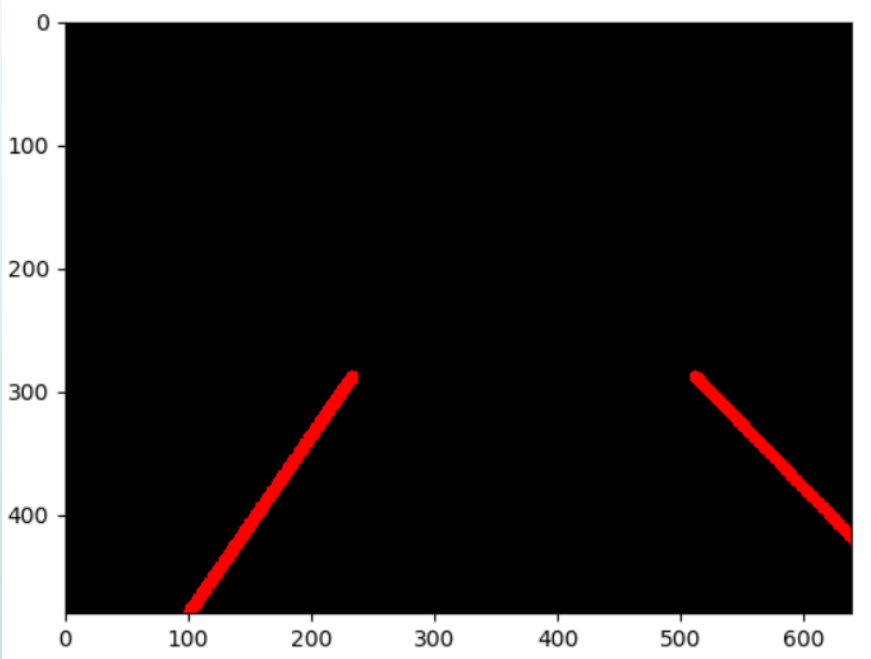
Canny Image

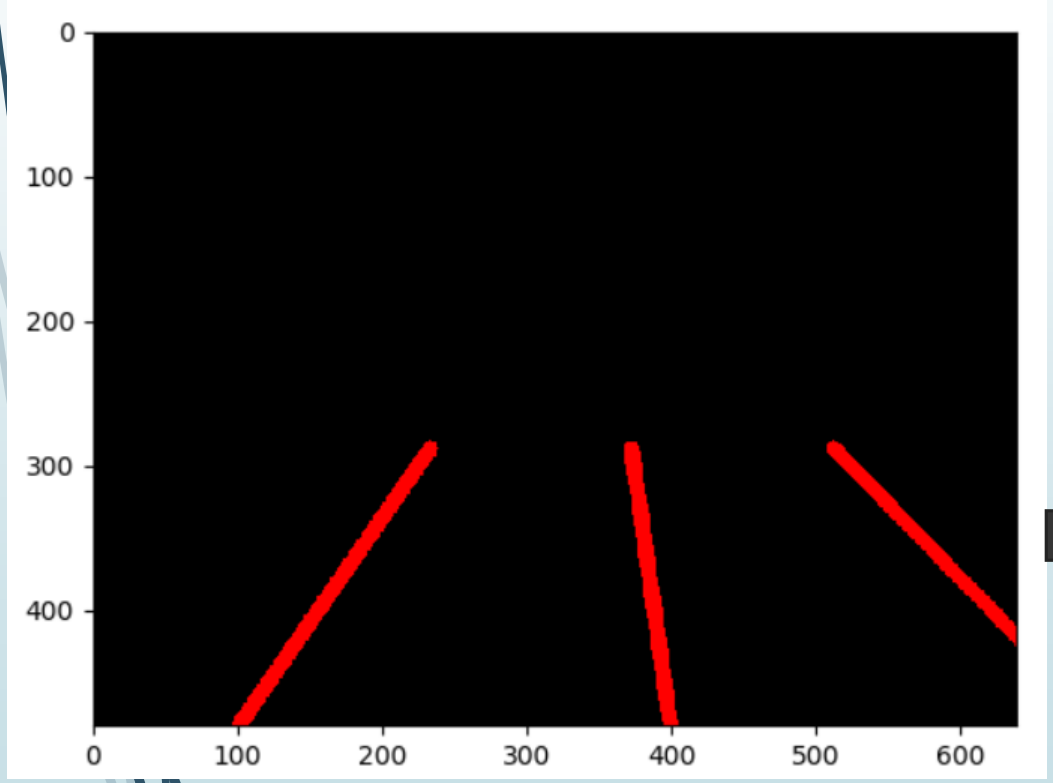
(100,height),(width,height),(width,300),(100,300)



Cropped canny image

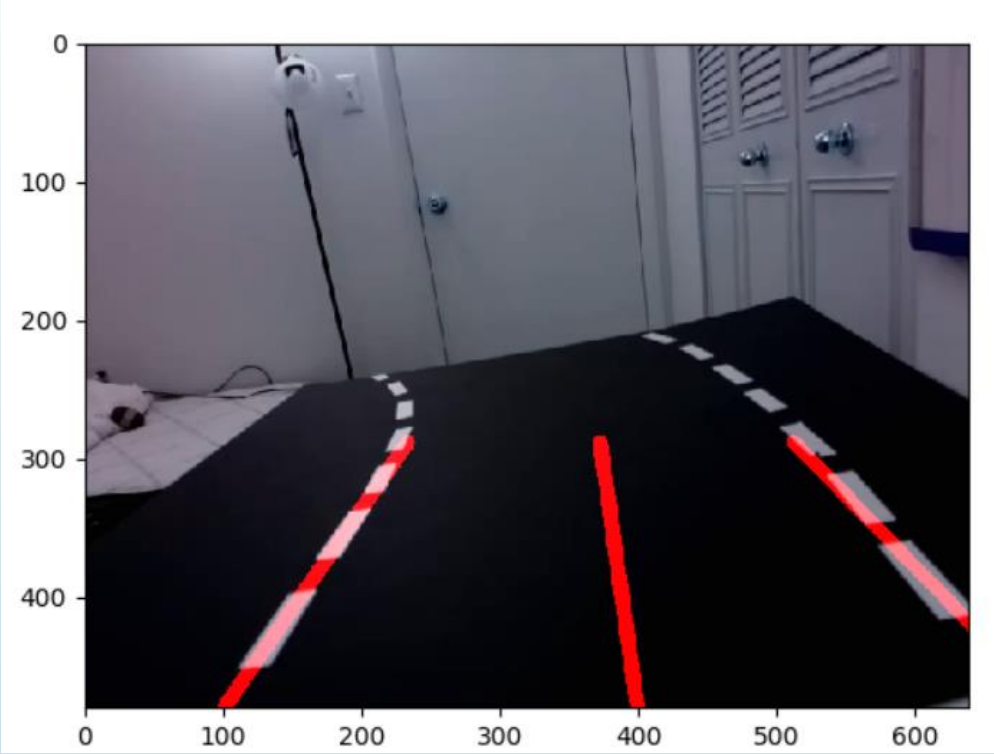
1. Get lines using cv2.HoughLinesP
2. Separate left and right lanes

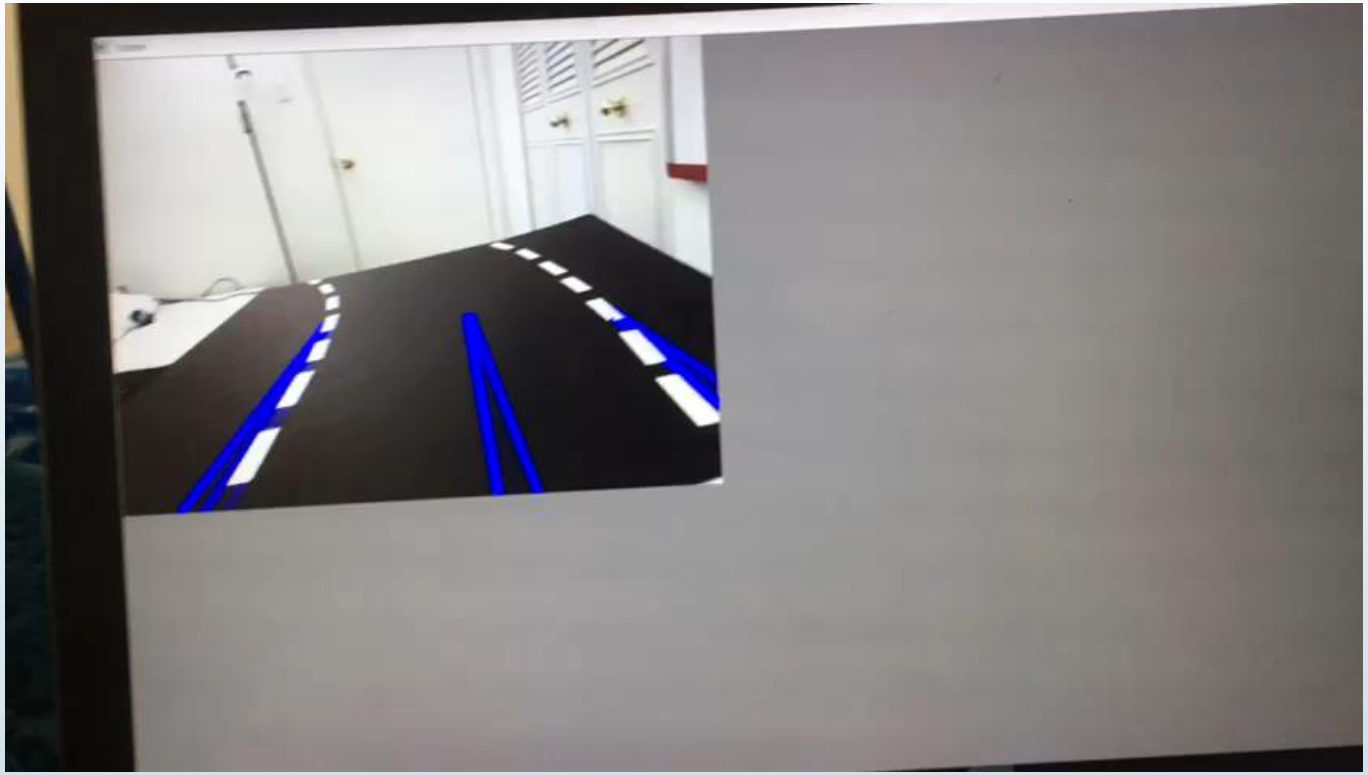




Left lane, right lane and mid line

Combined image





CV2 Neural Network

- Collecting training data
- Training and generating the model
- Testing

Collecting Training data

- Run the car using keyboard control in different backgrounds
- Get the video from pi camera
- Convert to grayscale image :- cv2.COLOR_RGB2GRAY
- Discard the upper half (surroundings) of the image
- Store the pixels in stack (2D => 1D)
- Save as numpy format (.npz) with two attributes
  - Training data :- 1D array
  - Training Labels:- Key pressed
- Keys used (training labels)
  - Left arrow :- 0
  - Right arrow :- 1
  - Up arrow :- 2
- Saves the frames only when the keys pressed to avoid noises

The Neural Network

- cv2.ml.ANN_MLP_create()
- layer_sizes (input size, 32, 3) :- input_size = 120 * 320
- TrainMethod :- cv2.ml.ANN_MLP_BACKPROP
- ActivationFunction :- cv2.ml.ANN_MLP_SIGMOID_SYM
- Save the model

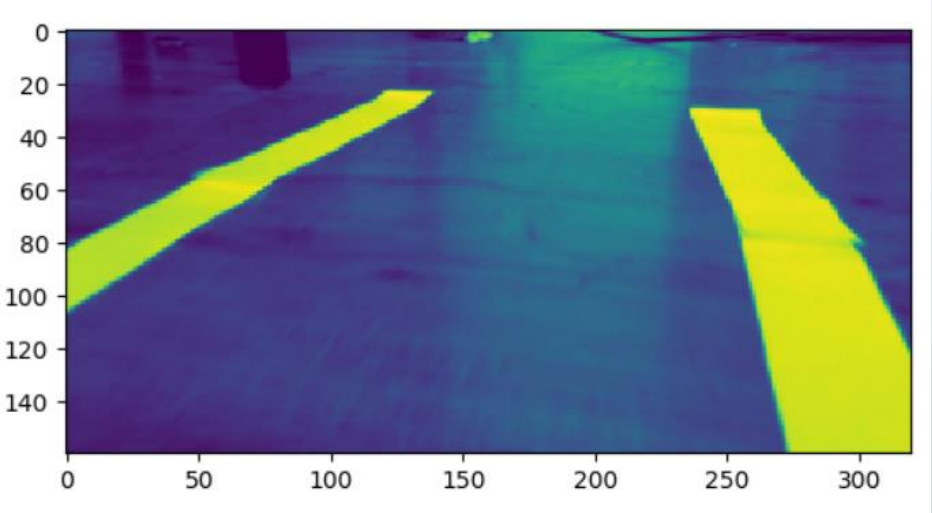
```
Image array shape:  (1167, 51200)
Label array shape:  (1167, 4)
Loading data duration: 2.85s
Before neural network
after nn
after nn1
Training ...
(1050, 51200)
(1050, 4)
Training duration: 779.77s
after nn2
Train accuracy:  69.14%
Validation accuracy:  60.68%
Model saved to: 'saved_model/nn_model.xml'
```

Testing

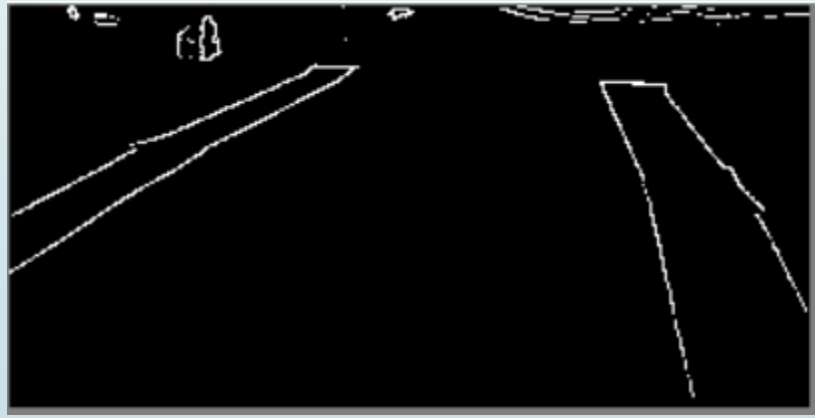
- Get the model to the pi
- Get the video
- Do the preprocessing (RBG2Gray and take half of the image)
- Give to the model
- Get the prediction (either 0, 1 , 2)
- Change the direction of the car accordingly

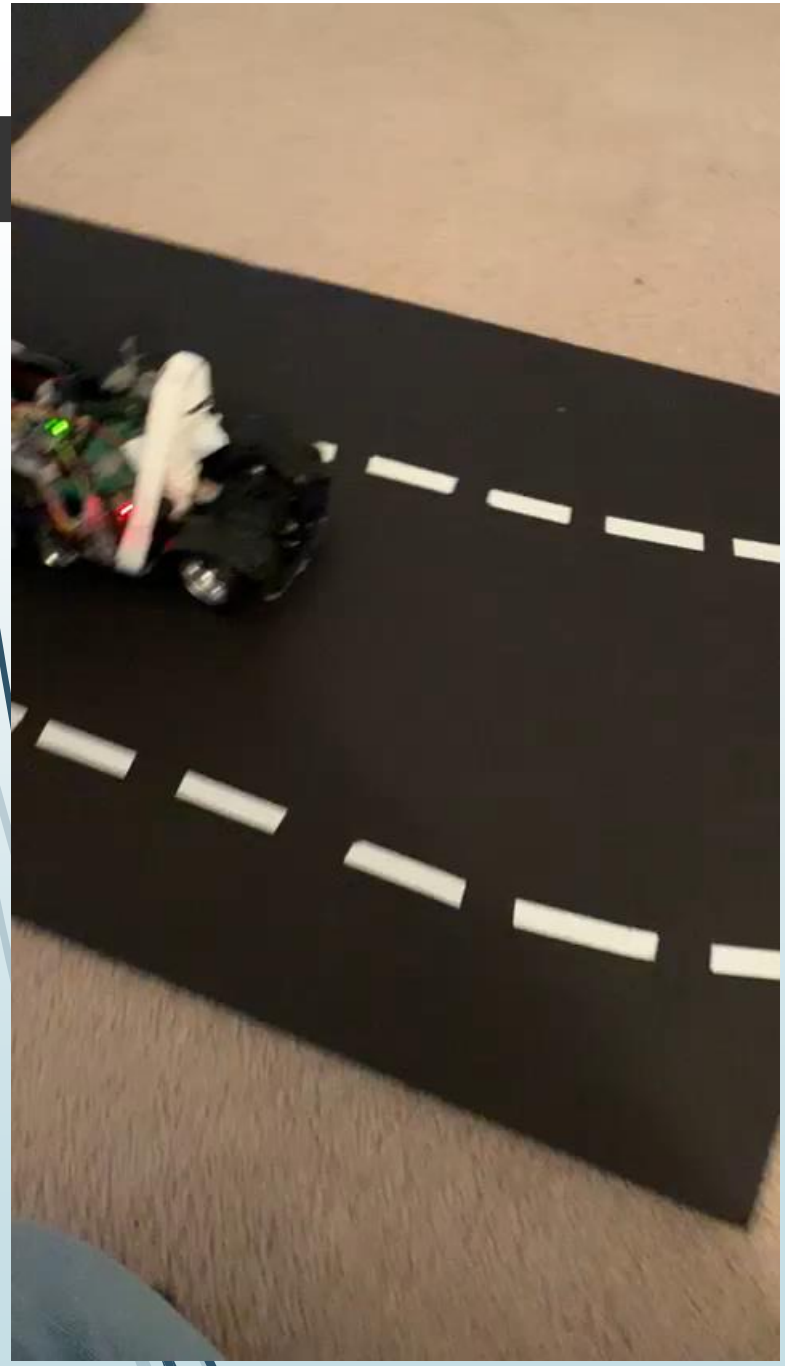
The improvement

- Do an edge detection before training



cv2.threshold(threshed, 0, 255, cv2.THRESH_BINARY_INV | cv2.THRESH_OTSU)





Demo