



Facultatea de Autovehicule Rutiere, Mecatronică și Mecanică




Titlul lucrării: Dezvoltarea unui sistem de recunoaștere a semnelor de circulație și urmărire a liniei mediane folosind Inteligență Artificială

Autor: Dulceanu Andrei-Alin

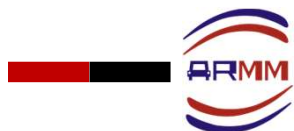
Coordonator: Ș.L. dr. ing. Alin PLEȘA



 /Facultatea.ARMM

 /utcluj.ro

 Universitatea Tehnică din Cluj-Napoca



Facultatea de Autovehicule Rutiere, Mecatronică și Mecanică



Cuprins

Scopul Lucrării

Componente Hardware

Arhitectură Software

Sistemul de recunoaștere al semnelor

Sistemul de detectare al liniei mediane

Rezultate experimentale

Concluzii



www.utcluj.ro



Scopul lucrării

Dezvoltarea unui sistem de recunoaștere a semnelor rutiere și a liniei mediane a șoselei folosind sisteme embedded și algoritmi avansați

Domeniul temei

- Autonomous Driving
- Advanced Driver-Assistance Systems (ADAS)

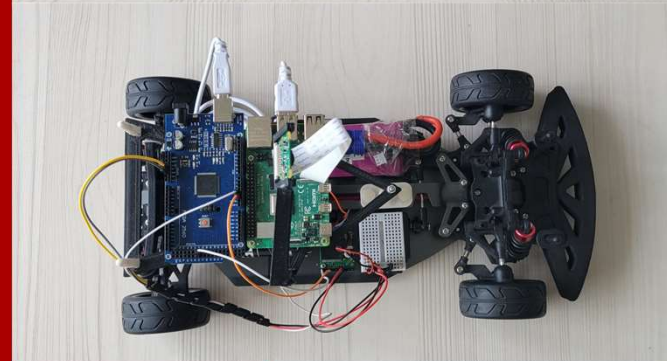
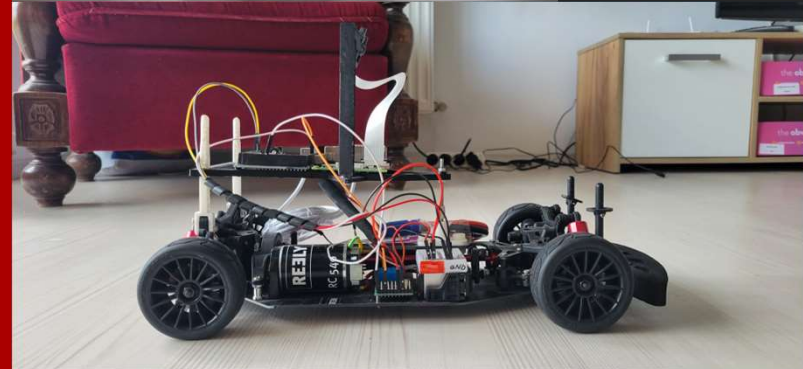


Fig. 1,2,3 Mașină electrică

Componente hardware

Motor CC

Servomotor

Motor Driver
VNH5019

Raspberry Pi
4B

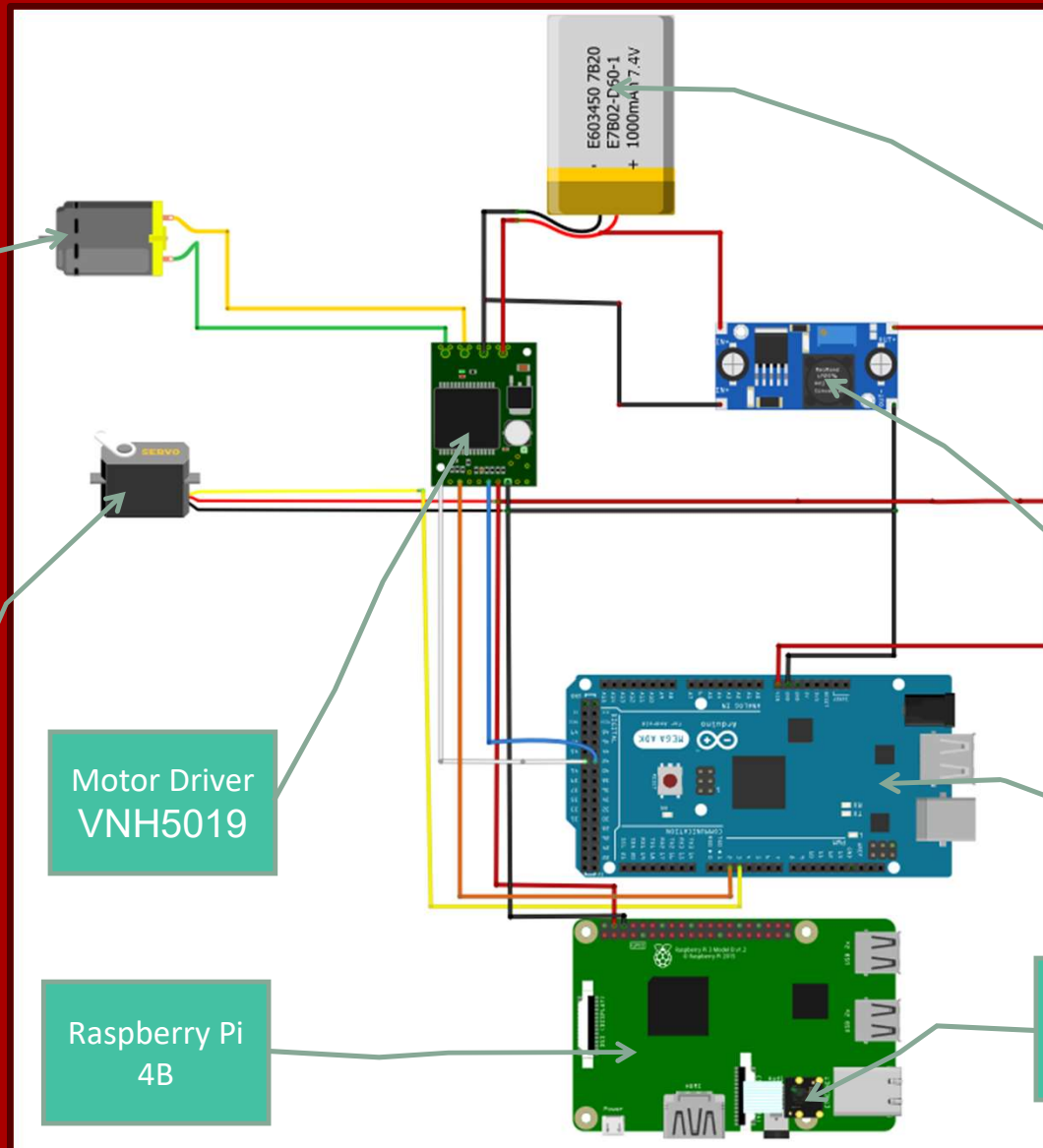
E603450 7820
E7802-D50-1
1000mAh 7.4V

Acumulator
7.4V

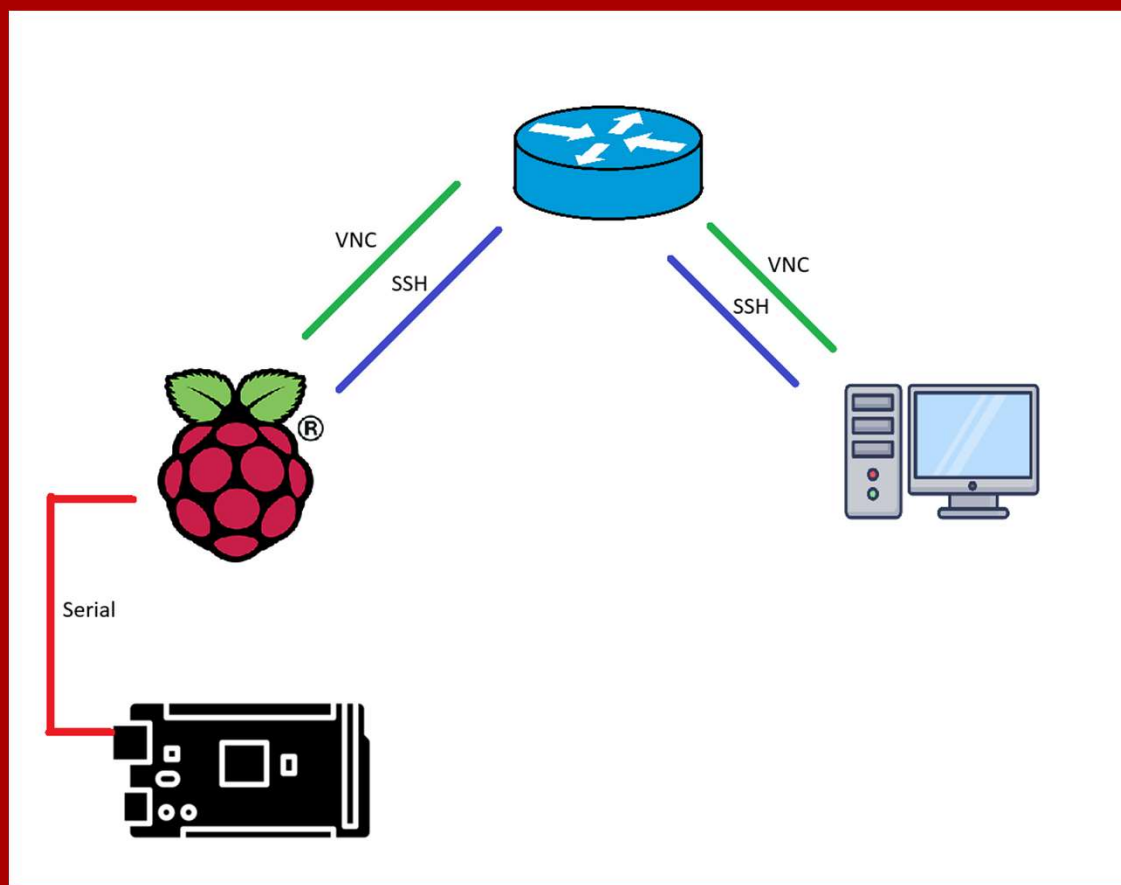
Buck Converter
LM2596

Arduino
MEGA 2560

Pi Camera V2



Rețelistică

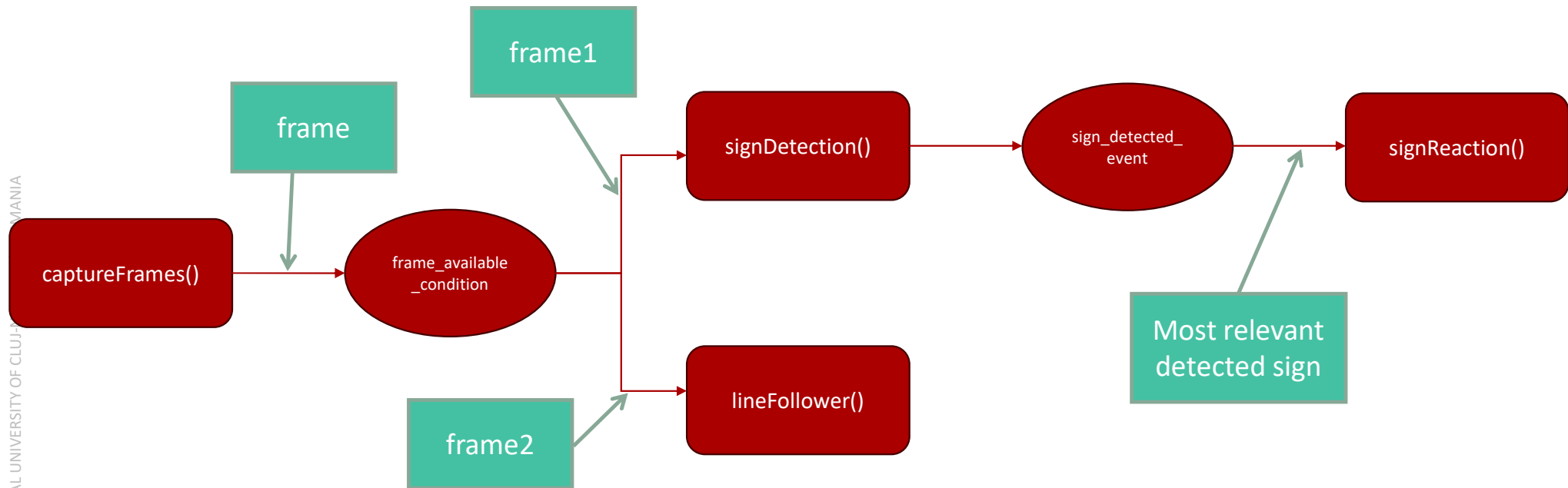


- Comunicare serial (115200 baud/s)
- Comunicare SSH
- Comunicare VNC



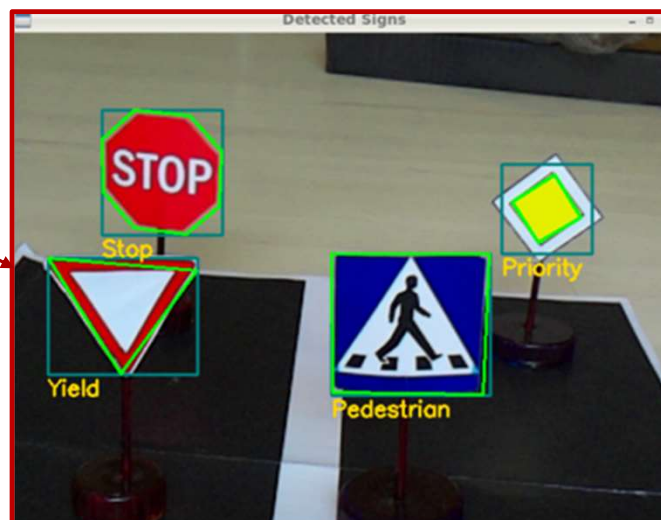
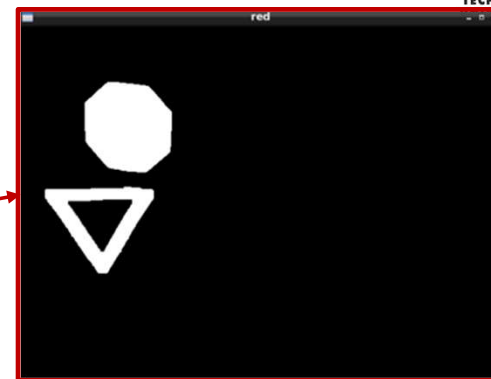
Arhitectură software și paradigme de programare

- Pipe and Filter
- OOP
- Multithreading



Recunoaștere de semne

- Segmentare
- Detectare de contururi și simplificarea acestora
- Categorizare
- Clasificare

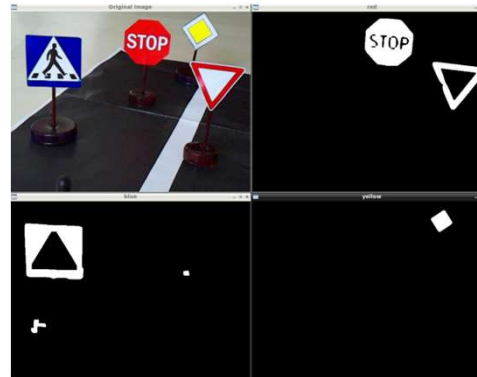


Segmentare

- Estompare Gaussiană
- Frame-ul se convertește din BGR în HSV
- 3 imagini binare
- Operații morfologice (dilate, erode)



HSV



Cele trei măști binare



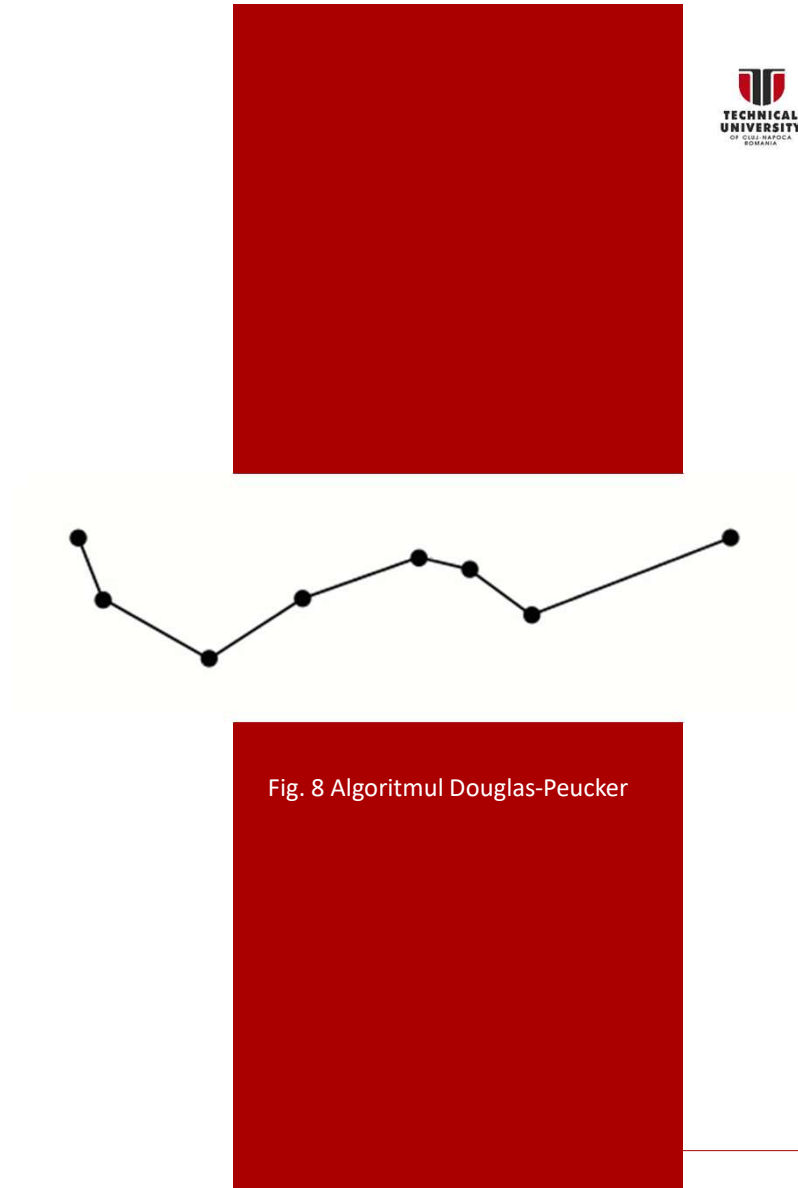
Estompare Gaussiană





Detectare de contururi

- Algoritmul Suzuki-Abe
- Algoritmul Douglas-Peucker



Categorizare

Extragerea unor zone de interes în funcție de numărul de puncte ce formează conturul, culoare și arie:

- redTriangles
- pedestrianSign
- priorityRoad
- stopSign

```

6      redTriangles = (3, "red", "red triangle")
7      stopSign = (8, "red", "stop sign")
8      pedestrianSign = (4, "blue", "pedestrian sign")
9      mandatory = ("circle", "blue", "mandatory sign")
10     priorityRoad = (4, "yellow", "priority sign")
11     prohibitory = ("circle", "red", "prohibitory sign")
12
13     self.categories = (redTriangles, stopSign, pedestrianSign, mandatory, priorityRoad, prohibitory)
  
```

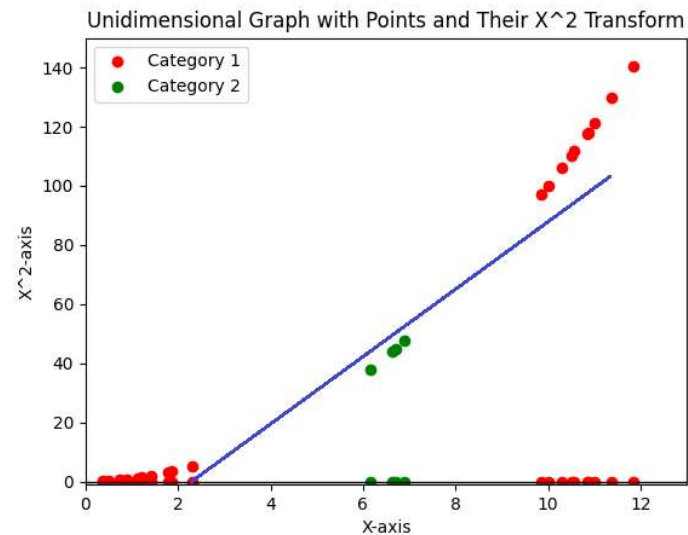
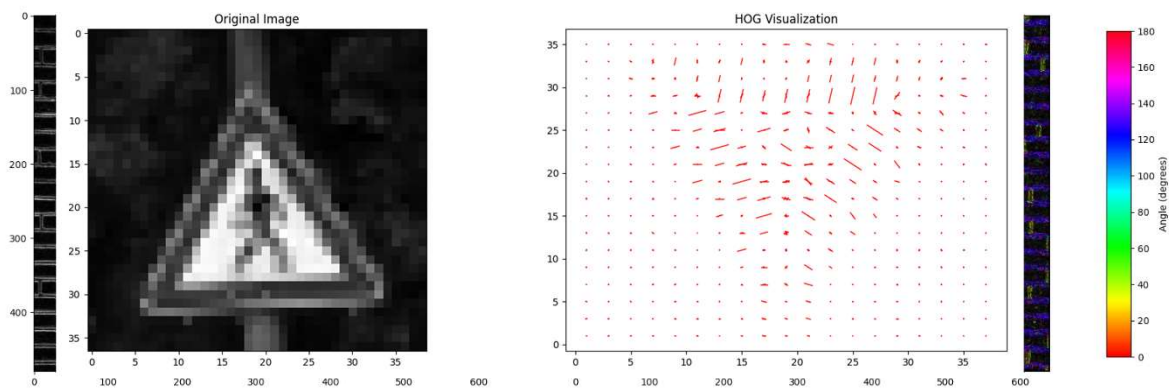


Clasificare

GTSRB (German Traffic Sign Recognition Benchmark)

HOG (Histogram of Oriented Gradients)

SVM (Support Vector Machine)





Line Follower

Etapele sunt următoarele:

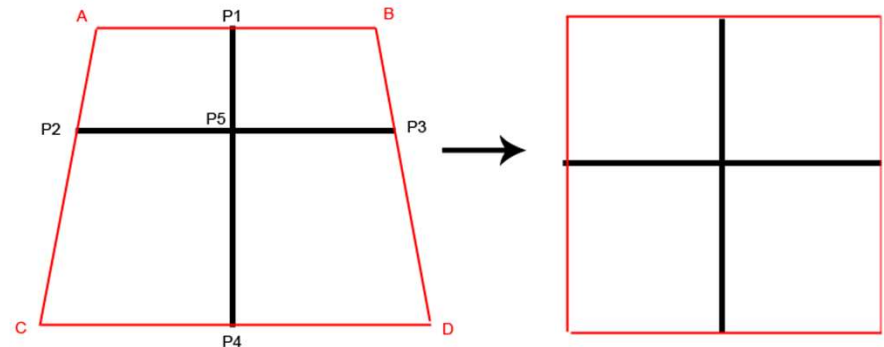
- Generarea unei vederi de sus a străzii
- Segmentare
- Detectarea liniei folosind algoritmul lui Hough
- Conversia liniei în comanda servomotorului

Crearea vederii de sus

Pentru fiecare punct (x, y) din trapezoid corespunde un punct (u, v) din vederea de sus iar relația dintre ele este următoarea :

$$\begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & 1 \end{bmatrix} \cdot \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix}, \text{ unde}$$

$$H = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & 1 \end{bmatrix} \text{ Este matricea de transformare}$$



Segmentare



- Convertirea imaginii din BGR in grayscale
- Aplicarea unei limite de luminozitate pentru a crea o imagine binara
- Scheletizarea liniei obținute până când aceasta ajunge la o grosime de aproximativ 1 pixel

Algoritmul lui Hough

Presupune conversia din sistemul cartezian în cel polar sub forma unei curbe sinusoidale al fiecarui punct din imagine folosind formula:

$$\rho = x * \cos \theta + y * \sin \theta$$

Se formează un grafic ρ/θ în care fiecarui punct din sistemul cartezian îi aparține o curbă sinusoidală

Punctele în care curbele se intersectează cel mai des reprezintă liniile din spatiul cartezian

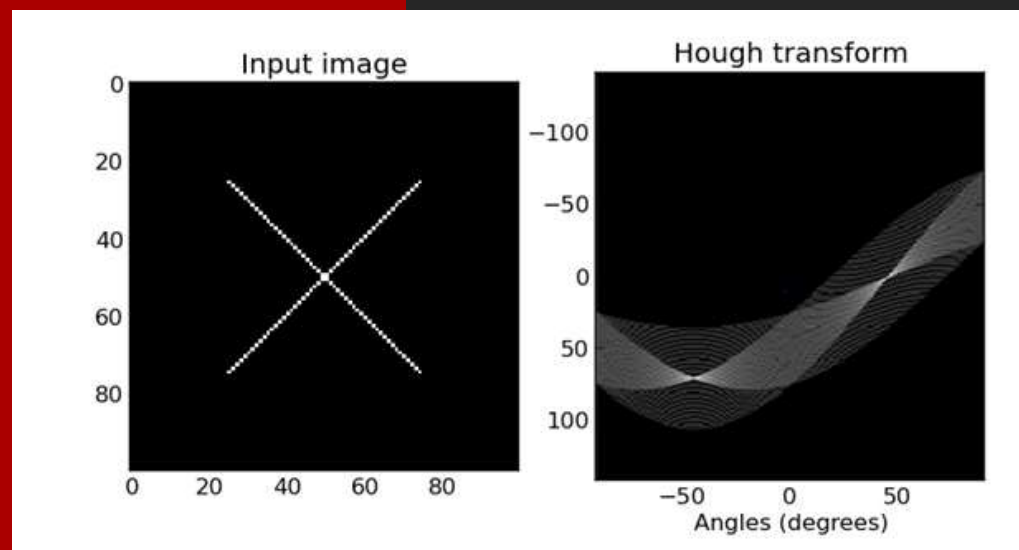


Fig. 9 Algoritmul lui Hough



Conversia liniei în comanda servomotorului

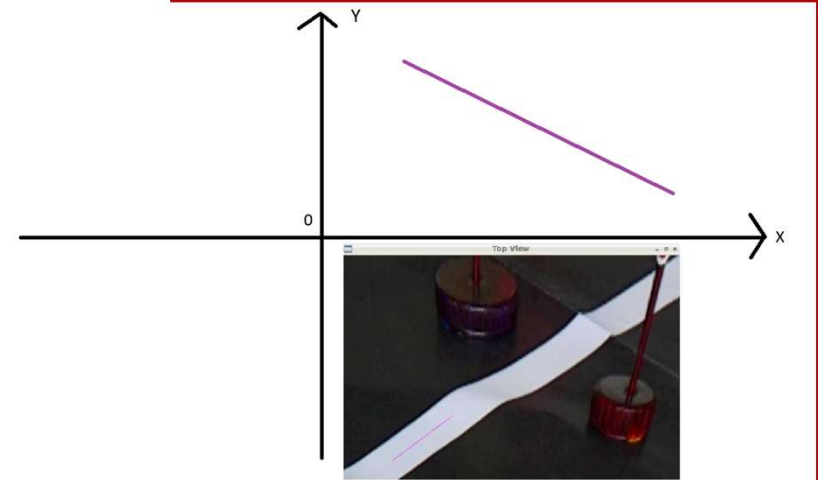
$$m = \frac{y_1 - y_2}{x_2 - x_1}$$

Unghiul față de axa verticală se obține folosind:

$$\alpha = \frac{\pi}{2} - \tan^{-1}(m)$$

Unghiul maxim este $\frac{\pi}{6}$ care corespunde valorii de 50 in arduino, deci:

$$poziție_{servo} = \alpha * \frac{50}{\frac{\pi}{6}} = \frac{300 * \alpha}{\pi}$$



Rezultate experimentale

- 18 semne detectate
- Urmărirea liniei mediane



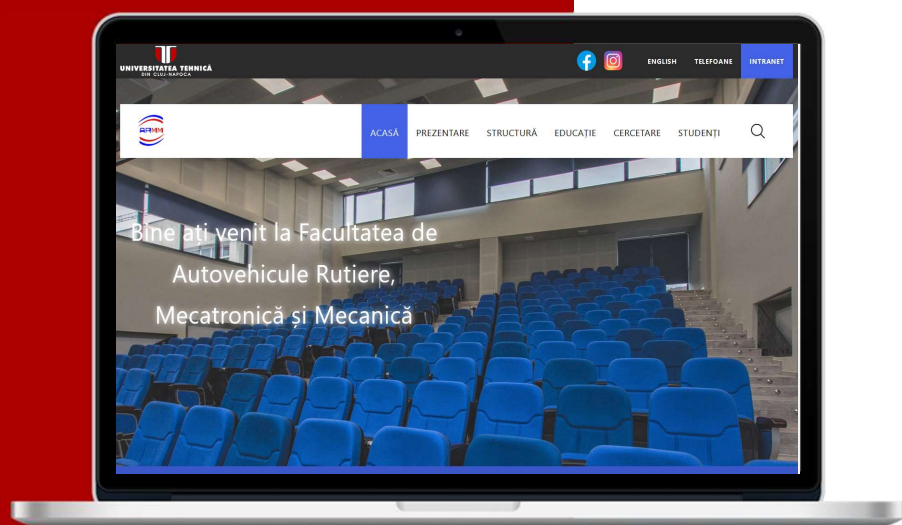
Concluzii

- Am reușit prin acest proiect să demonstrez faptul că nu este necesar un echipament puternic pentru a obține funcționalitatea unui sistem de recunoaștere a semnelor de circulație
- Am reușit să conectez într-un mod sinergic mai multe componente electronice (raspberry pi 4B, pi camera, Arduino mega 2560, servomotor, motor de curent continuu)
- Am reușit să creez un sistem software complex folosind tehnici paradigme moderne de programare
- Am reușit să optimizez acest sistem pentru a putea rula în timp real
- Am reușit să implementez aceste sisteme pe un model mecanic real






Facultatea de Autovehicule Rutiere, Mecatronică și Mecanică



Vă mulțumesc!

www.armm.utcluj.ro

 /Facultatea.ARMM

 /utcluj.ro

 Universitatea Tehnică din Cluj-Napoca