AI & SUSTAINABILITY IN VET EDUCATION

Name of the Author/s: TR Takım 1

Title of WP: WP3 – AI-SAVES@ProjectHub

Activity Reference: 3.2 – Project template for AI-SAVES@ProjectHub

Logo:

DENİZ YILDIZLARI
MESLEKİ VE TEKNİK
ANADOLU LİSESİ
2006

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

İçindekiler

İçindekiler **2**

AI-SAVES@PROJECTHUB **3**

1. Proje Başlığı: 3
2. Proje Kategorisi: 3
3. Proje Açıklaması: 3
4. Anahtar Kelimeler: 3
5. Hedef Kitle: 3
6. Proje Hedefleri: 3
7. Proje Zaman Çizelgesi : 3
8. Sağlanan Kaynaklar: 4
9. Kullanılan Araçlar ve Teknolojiler: 4
10. Adım Adım Uygulama Planı: 4
11. Değerlendirme Kriterleri: 8
12. Mevcut Diller: 9
13. Katkıda Bulunanlar: 9
14. Geri Bildirim ve Sürekli İyileştirme: 9

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

AI-SAVES@PROJECTHUB

---

1. **Proje Başlığı:**

   Trafik Işığı Tanıma ve Kontrollü Araç

2. **Proje Kategorisi:**

   Robotikte Yapay Zeka

3. **Proje Açıklaması:**

Bu proje, Raspberry Pi 5 ve Raspberry Camera v2 kullanarak trafik ışıklarını tanıyabilen otonom bir araç geliştirmek amacıyla tasarlanmıştır. Proje, araca kırmızı ışıkta durmayı ve yeşil ışıkta ilerlemeyi öğreterek yapay zeka ve robotik alanındaki pratik kavramları tanıtacaktır.

4. **Anahtar Kelimeler:**

Yapay Zeka, Robotik, Raspberry Pi, Trafik Işığı Tanıma, Otonom Araç, Python

5. **Hedef Kitle:**

Hem Eğitimciler Hem de Öğrenciler

6. **Proje Hedefleri:**

- Yapay zeka ve robotik alanında pratik bilgiler geliştirmek.
- Görüntü işleme ve nesne tanıma (trafik ışıkları) uygulamalarını anlamak.
- Raspberry Pi ve motor kontrolü ile pratik becerileri geliştirmek.
- Yapay zekayı öğretim uygulamalarına entegre etme konusunda mesleki eğitim becerilerini geliştirmek.

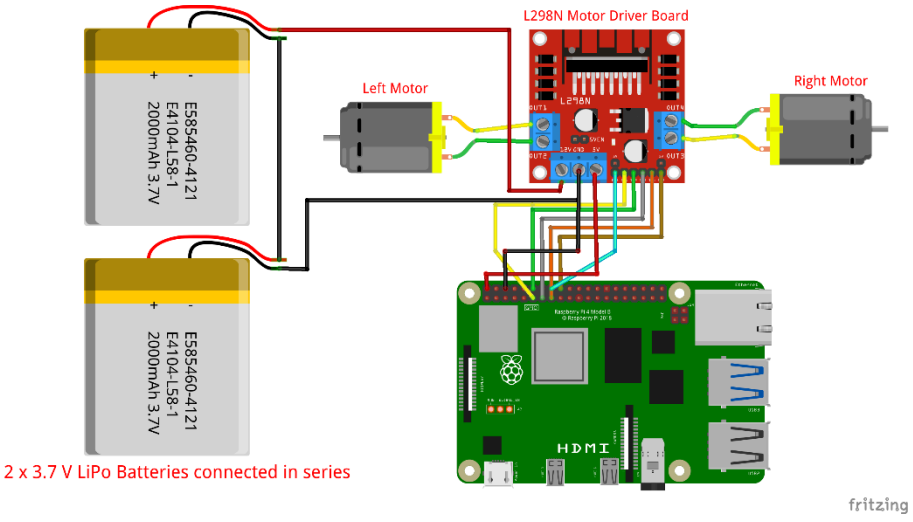
7. **Proje Zaman Çizelgesi :**
   - Başlangıç Tarihi: 15/12/2025
   - Bitiş Tarihi: 10/02/2026
   - Önemli Dönüm Noktaları:
     - Raspberry Pi Kurulumu ve Kamera Entegrasyonu: 15/12/2025
     - Motor ve Sürücü Entegrasyonu: 15/01/2026
     - Görüntü İşleme ve Trafik Işığı Tanıma Modeli Eğitimi: 20/01/2026
     - Araç Kontrol Algoritması Geliştirme: 25/01/2026
     - Proje Testleri ve Optimizasyonu: 01/02/2026

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

8. **Sağlanan Kaynaklar:**
    - Raspberry Pi 5 8GB RAM (Donanım)
    - Raspberry Camera v2 (Donanım)
    - İki DC motor, L298N motor sürücüsü, LIPO 3S 1000mAh, araba gövdesi, tekerlek malzemeleri (Donanım)
    - AI uygulamaları için Python programlama kılavuzları.
    - Ders planı: "Yapay Zeka ve Bilgisayar Görme Giriş" (PDF)
    - Proje rapor şablonu (DOCX).

9. **Kullanılan Araçlar ve Teknolojiler:**
    - Raspberry Pi 5 8GB RAM
    - Raspberry Camera v2
    - Python programlama dili
    - OpenCV library (görüntü işleme)
    - TensorFlow/Keras (makine öğrenme modeli)
    - Thonny IDE (python program yazımı arayüzü)
    - Sürüm kontrolü ve proje işbirliği için GitHub



10. **Adım Adım Uygulama Planı:**

    - **Step 1: Donanım Kurulumu ve Bağlantılar**
        - Raspberry Pi 5'i gerekli işletim sistemi (Raspberry Pi OS) ile kurun.

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

- Raspberry Camera v2'yi Raspberry Pi'ye bağlayın.
- Motorları ve L298N motor sürücüsünü Raspberry Pi'nin GPIO pinlerine bağlayın.
- LIPO 3S 1000mAh pili motor sürücüsüne bağlayın.
- Araba gövdesini monte edin ve motorları, tekerlekleri, Raspberry Pi'yi ve diğer bileşenleri takın.

- **Adım 2: Geliştirme Ortamı Kurulumu**
  - Raspberry Pi'ye Python ve gerekli kütüphaneleri (OpenCV, TensorFlow/Keras) yükleyin.
  - Thonny IDE gibi bir geliştirme ortamı kurun.

- **Adım 3: Trafik Işığı Tanıma Modelini Eğitme**
  - Trafik ışığı görüntülerinden oluşan mevcut bir veri setini toplayın veya kullanın.
  - Toplanan verileri kullanarak bir görüntü sınıflandırma modeli (ör. CNN tabanlı) eğitin. Model, kırmızı, yeşil ve diğer renkleri/durumları tanıyabilmelidir.
  - Modeli, Raspberry Pi ile entegrasyona uygun bir biçime dönüştürün (ör. TensorFlow Lite).

- **Adım 4: Araç Kontrol Algoritmasının Geliştirilmesi**
  - Raspberry Kameradan canlı video akışı elde edin.
  - Eğitimli modeli kullanarak video akışının her karesindeki trafik ışıklarını tanıyın.
  - Tanınan trafik ışığı rengine göre aracın hareketini kontrol etmek için Python kodu yazın:
  - Kırmızı ışık algılandığında motorları durdurun.
  - Yeşil ışık algılandığında motorları ileri doğru hareket ettirin.
  - Işık algılanmadığında veya belirsiz durumlarda varsayılan davranışı tanımlayın.

- **Adım 5: Entegrasyon ve Test**
  - Tüm donanım ve yazılım bileşenlerini entegre edin.
  - Aracın trafik ışıklarına doğru tepki verip vermediğini test edin.
  - Modelin ve kontrol algoritmasının performansını optimize etmek için farklı aydınlatma koşulları ve açılar altında testler gerçekleştirin.

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

- **Step 6: Proje İlerlemesi ve Değerlendirmesinin İzlenmesi**
  - Haftalık incelemelerle projenin ilerleyişini takip etme.
  - Proje sunumları aracılığıyla öğrencilerin kavrayışını değerlendirme.

Python Kod Örneği :

```
import cv2
import numpy as np
import RPi.GPIO as GPIO
import time
from tensorflow.keras.models import load_model # If using
Keras/TensorFlow model

# GPIO pin definitions (for motor driver L298N)
# These pin numbers should be adjusted according to your Raspberry
Pi's GPIO scheme
IN1 = 17
IN2 = 18
IN3 = 27
IN4 = 22
ENA = 23 # Enable A for Motor 1
ENB = 24 # Enable B for Motor 2

GPIO.setmode(GPIO.BCM)
GPIO.setup(IN1, GPIO.OUT)
GPIO.setup(IN2, GPIO.OUT)
GPIO.setup(IN3, GPIO.OUT)
GPIO.setup(IN4, GPIO.OUT)
GPIO.setup(ENA, GPIO.OUT)
GPIO.setup(ENB, GPIO.OUT)

# PWM settings (for speed control)
pwmA = GPIO.PWM(ENA, 100) # 100 Hz frequency
pwmB = GPIO.PWM(ENB, 100)
pwmA.start(0) # Motors stopped initially
pwmB.start(0)

# Motor control functions
def move_forward():
    GPIO.output(IN1, GPIO.HIGH)
    GPIO.output(IN2, GPIO.LOW)
    GPIO.output(IN3, GPIO.HIGH)
    GPIO.output(IN4, GPIO.LOW)
    pwmA.ChangeDutyCycle(60) # 60% speed
    pwmB.ChangeDutyCycle(60)
    print("Moving forward")

def stop_motors():
```

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

```
    GPIO.output(IN1, GPIO.LOW)
    GPIO.output(IN2, GPIO.LOW)
    GPIO.output(IN3, GPIO.LOW)
    GPIO.output(IN4, GPIO.LOW)
    pwmA.ChangeDutyCycle(0)
    pwmB.ChangeDutyCycle(0)
    print("Motors stopped")

# Load traffic light recognition model
# This part will vary depending on the file path and structure of your
trained model
# model = load_model('path/to/your/traffic_light_model.h5')
# LABELS = ['red', 'green', 'yellow', 'none'] # Output labels of your
model

def preprocess_frame(frame):
    # Preprocessing steps suitable for your model
    # Resizing, normalization, etc.
    resized_frame = cv2.resize(frame, (224, 224)) # Example size
    normalized_frame = resized_frame / 255.0
    return np.expand_dims(normalized_frame, axis=0)

def predict_traffic_light(frame):
    # Process the frame and get prediction from the model
    # processed_frame = preprocess_frame(frame)
    # predictions = model.predict(processed_frame)
    # predicted_class_index = np.argmax(predictions[0])
    # return LABELS[predicted_class_index]

    # Simple color detection (temporary placeholder for the model)
    hsv = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)

    # HSV color ranges for red light
    lower_red1 = np.array([0, 100, 100])
    upper_red1 = np.array([10, 255, 255])
    lower_red2 = np.array([160, 100, 100])
    upper_red2 = np.array([180, 255, 255])

    # HSV color range for green light
    lower_green = np.array([40, 40, 40])
    upper_green = np.array([80, 255, 255])

    mask_red1 = cv2.inRange(hsv, lower_red1, upper_red1)
    mask_red2 = cv2.inRange(hsv, lower_red2, upper_red2)
    mask_red = cv2.bitwise_or(mask_red1, mask_red2)

    mask_green = cv2.inRange(hsv, lower_green, upper_green)

    red_pixels = cv2.countNonZero(mask_red)
    green_pixels = cv2.countNonZero(mask_green)

    if red_pixels > 1000: # Adjust threshold value through
experimentation
```

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

```
        return "red"
    elif green_pixels > 1000: # Adjust threshold value through
experimentation
        return "green"
    else:
        return "none"


# Initialize camera
cap = cv2.VideoCapture(0) # 0 for default camera

if not cap.isOpened():
    print("Camera could not be opened!")
    exit()

try:
    while True:
        ret, frame = cap.read()
        if not ret:
            print("Frame could not be retrieved!")
            break

        # Resize the image (for processing speed)
        frame = cv2.resize(frame, (320, 240))

        # Traffic light detection
        traffic_light_status = predict_traffic_light(frame)

        if traffic_light_status == "red":
            stop_motors()
        elif traffic_light_status == "green":
            move_forward()
        else:
            # Stop if no light or ambiguous
            stop_motors()

        # Display the image on screen (optional, not needed if no GUI
on Raspberry Pi)
        cv2.imshow('Frame', frame)
        if cv2.waitKey(1) & 0xFF == ord('q'):
            break

        time.sleep(0.1) # Short delay to reduce processor load

except KeyboardInterrupt:
    print("Program terminated.")
finally:
    cap.release()
    cv2.destroyAllWindows()
    GPIO.cleanup()
```

11. Değerlendirme Kriterleri:

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

- **Öğrenci Geri Bildirimi:** Anketler aracılığıyla öğrenme süreciyle ilgili geri bildirim toplama.
- **Proje Başarı Ölçütleri:** AI uygulamaları kullanılarak robotik görevlerin başarıyla tamamlanması.
- **Beceri Değerlendirmesi:** Öğrencilerin yapay zeka programlama ve robotik montajındaki yeterliliklerini ölçme.

12. Mevcut Diller:

Türkçe.

13. Katkıda Bulunanlar:

Türkiye–Proje Takımı 1

- **Mehmet Emin Uğur** (Proje Lideri) – Proje geliştirme ve koordinasyondan sorumlu baş eğitimci.
- **Harun Yenişan** (Teknik Uzman) – Yapay zeka programlama ve robotik alanlarında uzmanlık sağladı.
- **Ali İmran Şahin** (Müfredat Tasarımcısı) – AI tabanlı el yazısı tanıma için ders planları geliştirdi.
- **Ahmet Karagöz** (Eğitim Materyali Geliştiricisi) – Projede kullanılan destekleyici belgeler ve kodlama materyalleri hazırlandı.

14. Geri Bildirim ve Sürekli İyileştirme:

- **Anketler:** Her aşamanın sonunda, öğrenciler ve eğitimciler projenin etkinliğini değerlendirmek için geri bildirim anketlerini dolduracaklar.
- **Proje incelemeleri:** Proje tamamlandıktan sonra, gelecekteki AI tabanlı projeler için olası iyileştirmeleri tartışmak üzere bir son değerlendirme toplantısı düzenlenecektir.
- **Veri analizleri:** Geri bildirim verileri, güçlü yönleri ve iyileştirilmesi gereken alanları belirlemek için analiz edilecek ve gelecekteki projelere bilgi sağlayacaktır.

Video Linki: https://youtu.be/r_i5fHAa3Vw?si=OwUi8XfqVfPDXKi3

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.