AI & SUSTAINABILITY IN VET EDUCATION

Nome do(s) autor(es): Sucessos Criativos, Lda

Título do WP: WP3 – IA-SAVES@ProjectHub

Referência de Atividade: 3.2 – Modelo de projeto para IA-SAVES@ProjectHub

Logótipo:

Co-funded by the European Union

Este projeto foi financiado com o apoio da Comissão Europeia. Esta comunicação reflete apenas as opiniões do autor, e a Comissão não pode ser responsabilizada por qualquer uso que possa ser feito da informação nela contida.

Índice

1. Título do Projeto: 4
2. Categoria do Projeto: 4
3. Descrição do Projeto: 4
4. Palavras-chave: 4
5. Público-alvo: 4
6. Objetivos do Projeto: 4
7. Cronologia do Projeto: 5
8. Recursos Fornecidos: 5
9. Ferramentas e Tecnologias Utilizadas: 5
10. Plano de Implementação Passo a Passo: 6
11. Critérios de Avaliação: 13
12. Línguas Disponíveis: 13
13. Colaboradores: 13
14. Feedback e Melhoria Contínua: 13

Co-funded by the European Union

Este projeto foi financiado com o apoio da Comissão Europeia. Esta comunicação reflete apenas as opiniões do autor, e a Comissão não pode ser responsabilizada por qualquer uso que possa ser feito da informação nela contida.

1. Título do Projeto:

Veículo Autónomo Sustentável de Deteção de Resíduos

---

2. Categoria do Projeto:

IA em Robótica

---

3. Descrição do Projeto:

Este projeto apresenta o desenvolvimento de um veículo inteligente autónomo muito básico, capaz de conduzir continuamente usando lógica de evitação de obstáculos baseada em deteção ultrassónica de distância, aplicando ao mesmo tempo que a deteção de objetos por inteligência artificial para identificar resíduos no ambiente — especificamente copos e garrafas de plástico. O robô navega de forma independente controlando dois motores DC, evitando colisões através de um sensor de ultrassom.

Quando a câmara de bordo deteta resíduos (garrafas), o veículo pausa o movimento e aciona uma notificação enviando uma mensagem do WhatsApp através de um pedido HTTP, simulando um sistema inteligente de alerta de recolha de lixo. Após enviar a notificação e esperar brevemente, o veículo retoma a sua rotina de condução autónoma. O objetivo do sistema é demonstrar como a IA e a robótica podem ser combinadas para melhorar a monitorização de resíduos em espaços públicos, promover a consciencialização sobre a sustentabilidade e incentivar a aplicação prática da automação impulsionada por IA.

---

4. Palavras-chave:

Robô Autónomo, Deteção de Objetos, Deteção Inteligente de Resíduos, Raspberry Pi

---

5. Público-alvo:

Tanto Educadores como Alunos

---

6. Objetivos do Projeto:

Co-funded by the European Union

Este projeto foi financiado com o apoio da Comissão Europeia. Esta comunicação reflete apenas as opiniões do autor, e a Comissão não pode ser responsabilizada por qualquer uso que possa ser feito da informação nela contida.

- Constrói e configura um carro robótico totalmente autónomo usando o Raspberry Pi.
- Implemente a prevenção de obstáculos baseada em ultrassons com controlo de movimento em tempo real.
- Aplique deteção de objetos com o YOLO para identificar garrafas e copos.
- Integre notificações do WhatsApp via pedido HTTP.
- Compreenda a ligação entre o reconhecimento de IA, o movimento robótico e os sistemas de comunicação.
- Explore aplicações focadas na sustentabilidade da IA e automação.

---

7. Cronologia do Projeto:

Com os recursos fornecidos, este projeto deverá demorar aproximadamente 7-8 horas ou menos a ser concluído de forma calma.

---

8. Recursos Fornecidos:

- Base de código Python contendo controlo motor, deteção de objetos, medição de distância e lógica de alertas do WhatsApp
- Vídeo de demonstração do robô a detetar resíduos (garrafas) e a enviar notificações pelo WhatsApp
- Diagrama de cablagem para motores, sensor ultrassónico e câmara

---

9. Ferramentas e Tecnologias Utilizadas:

- Raspberry Pi 5
- Módulo da Câmara Raspberry Pi 4 v2
- Linguagem de programação Python
- OpenCV para entrada de câmara e processamento de imagem
- Modelo de Deteção de Objetos Pré-treinado (YOLO11-Nano)
- Sensor de distância ultrassónico
- Driver de motor DC duplo (L298N) + rodas, etc.
- API CallMeBot para notificações do WhatsApp via pedido HTTP

Co-funded by the European Union

Este projeto foi financiado com o apoio da Comissão Europeia. Esta comunicação reflete apenas as opiniões do autor, e a Comissão não pode ser responsabilizada por qualquer uso que possa ser feito da informação nela contida.

- (Opcionalmente) Power bank para substituir a fonte de alimentação do Raspberry Pi

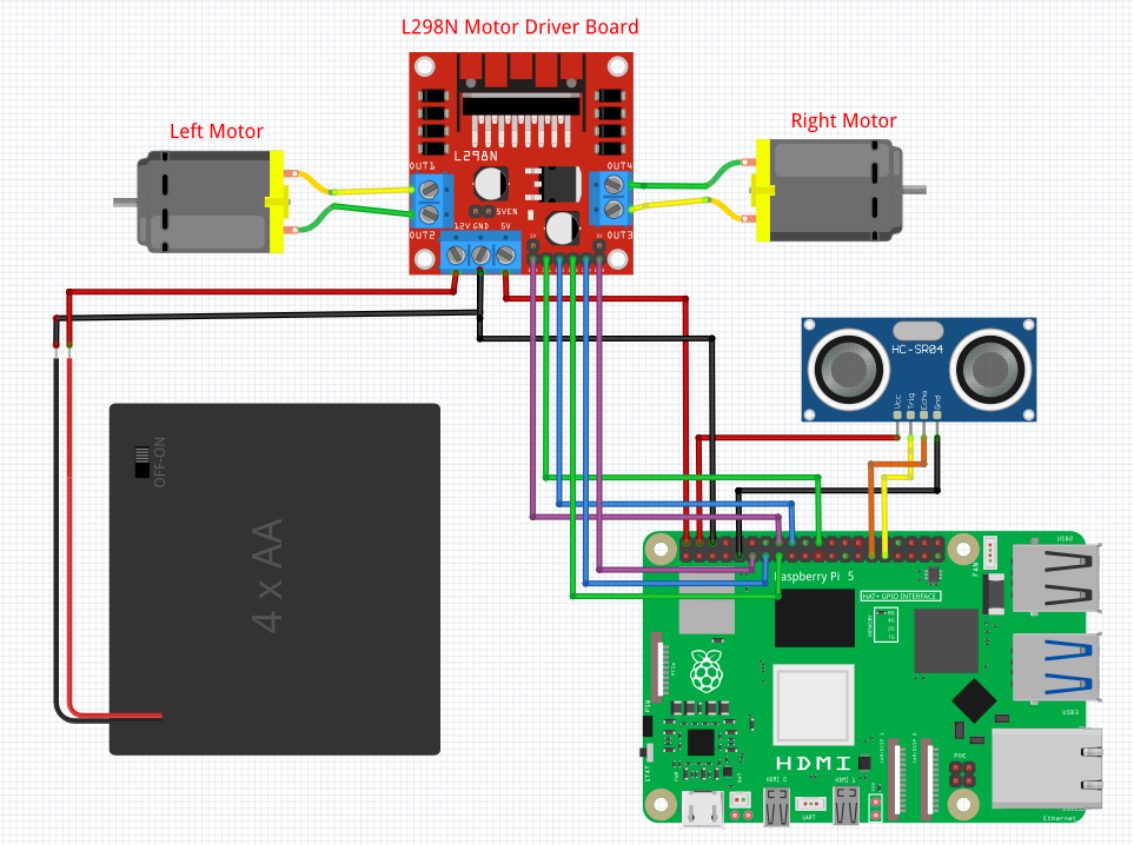
---

10. Plano de Implementação Passo a Passo:

Forneça um guia detalhado passo a passo sobre como o projeto será implementado. Isto ajudará os utilizadores que desejem replicar o projeto ou seguir o processo.

1. **Preparação de Hardware:** Monte os motores no chassis do carro e ligue-os ao driver do motor. Liga o Raspberry Pi, o módulo da câmara, o driver do motor e o sensor ultrassónico de acordo com o diagrama de cablagem. Ao ligar o driver do motor, não se esqueça de remover todos os jumpers da placa do driver do motor.

   Opcional: Durante os testes, podes alimentar o Raspberry Pi com um PowerBank.



2. **Configuração de Software:** Instale o software necessário no Raspberry Pi, incluindo bibliotecas Raspbian OS e Python, incluindo OpenCV, Picamera2, Ultralytics YOLO, Requests API e bibliotecas GPIO.

Co-funded by the European Union

Este projeto foi financiado com o apoio da Comissão Europeia. Esta comunicação reflete apenas as opiniões do autor, e a Comissão não pode ser responsabilizada por qualquer uso que possa ser feito da informação nela contida.

3. **Configuração da API do CallMeBot:** Siga o guia de configuração do CallMeBot (https://www.callmebot.com/blog/free-api-whatsapp-messages/) e guarde a sua chave API.

4. **Execute o código:** Copie o código fornecido abaixo e guarde-o no seu Raspberry Pi. Antes de executar o código, substitua o valor de "APIKEY" na sua chave API do passo 3 e substitua o valor de "PHONE_NUMBER" pelo número de telefone registado no passo 3 (este valor deve conter o seu código de país).

```
Tempo de importação

pedidos de importação

importar RPi.GPIO como GPIO

Importado CV2

de picamera2 importado Picamera2

da importação de ultralytics YOLO



# ===== CONFIGURAÇÃO =====

APIKEY = "1111111" # Substituir pela sua chave API real

PHONE_NUMBER = "+1123456789" # O seu número de telefone real com código de país (exemplo para Portugal: "+351911234567")



# -------------------- PINOS SENSORES --------------------

TRIG = 6

ECHO = 5



GPIO.setmode(GPIO. BCM)

GPIO.setup(TRIG, GPIO. FORA)

GPIO.setup(ECHO, GPIO. DENTRO)



# -------------------- PIN MOTOR --------------------

IN1 = 25
```

Co-funded by the European Union

Este projeto foi financiado com o apoio da Comissão Europeia. Esta comunicação reflete apenas as opiniões do autor, e a Comissão não pode ser responsabilizada por qualquer uso que possa ser feito da informação nela contida.

```
IN2 = 24

IN3 = 22

IN4 = 27

ENA = 23

ENB = 17


GPIO.setup(IN1, GPIO. FORA)

GPIO.setup(IN2, GPIO. FORA)

GPIO.setup(IN3, GPIO. FORA)

GPIO.setup(IN4, GPIO. FORA)

GPIO.setup(ENA, GPIO. FORA)

GPIO.setup(ENB, GPIO. FORA)


pwmA = GPIO. PWM(ENA, 100)

pwmB = GPIO. PWM(ENB, 100)

pwmA.start(0)

pwmB.start(0)


# -------------------- FUNÇÕES MOTORAS --------------------

def move_forward():

    GPIO.output(IN1, GPIO. ALTO)

    GPIO.output(IN2, GPIO. BAIXO)

    GPIO.output(IN3, GPIO. ALTO)

    GPIO.output(IN4, GPIO. BAIXO)

    pwmA.MudançaDeCicloDeServiço(60)

    pwmB.MutyDutyCycle(60)

    print ("A avançar")
```

Co-funded by the European Union

Este projeto foi financiado com o apoio da Comissão Europeia. Esta comunicação reflete apenas as opiniões do autor, e a Comissão não pode ser responsabilizada por qualquer uso que possa ser feito da informação nela contida.

```
def stop_motors():
    GPIO.output(IN1, GPIO. BAIXO)
    GPIO.output(IN2, GPIO. BAIXO)
    GPIO.output(IN3, GPIO. BAIXO)
    GPIO.output(IN4, GPIO. BAIXO)
    pwmA.ChangeDutyCycle(0)
    pwmB.ChangeDutyCycle(0)
    print ("Motores parados")


def turn_right():
    GPIO.output(IN1, GPIO. ALTO)
    GPIO.output(IN2, GPIO. BAIXO)
    GPIO.output(IN3, GPIO. BAIXO)
    GPIO.output(IN4, GPIO. ALTO)
    pwmA.MudançaDeCicloDeServiço(60)
    pwmB.MutyDutyCycle(60)
    print ("Virar à direita")
    Tempo.Sono(1.5)
    stop_motors()


# -------------------- SENSOR DE DISTÂNCIA --------------------
def measure_distance():
    GPIO.output(TRIG, Falso)
    tempo.sono(0.0002)
    GPIO.output(TRIG, Verdadeiro)
    tempo.sono(0.00001)
    GPIO.output(TRIG, Falso)
```

Co-funded by the European Union

Este projeto foi financiado com o apoio da Comissão Europeia. Esta comunicação reflete apenas as opiniões do autor, e a Comissão não pode ser responsabilizada por qualquer uso que possa ser feito da informação nela contida.

```
    enquanto GPIO.input(ECHO) == 0:
        pulse_start = tempo.tempo()
    enquanto GPIO.input(ECHO) == 1:
        pulse_end = tempo.tempo()

    retorno ((pulse_end - pulse_start) * 34300) / 2

# -------------------- CÂMARA + YOLO --------------------
picam2 = Picamera2()
picam2.preview_configuration.main.size = (640, 400)
picam2.preview_configuration.main.format = "RGB888"
picam2.preview_configuration.align()
picam2.configure("pré-visualização")
picam2.start()

modelo = YOLO("yolo11n.pt")

# Rastreamento do tempo de recarga
last_detected = {"objecto": Nenhum, "tempo": 0}
DETECTION_COOLDOWN = 30 # segundos

# -------------------- ENVIAR NOTIFICAÇÕES NO WHATSAPP --------------------
def send_notification(object_name):
    Experimente:
        url                                                                    =
f"https://api.callmebot.com/whatsapp.php?phone={PHONE_NUMBER}&text=Object+F
ound:+{object_name}&apikey={APIKEY}"
        requests.get(url, timeout=5)
        print(f"Notificação enviada: {object_name}")
```

Co-funded by the European Union

Este projeto foi financiado com o apoio da Comissão Europeia. Esta comunicação reflete apenas as opiniões do autor, e a Comissão não pode ser responsabilizada por qualquer uso que possa ser feito da informação nela contida.

```
    exceto exceção como e:
        print("Falhou no envio de notificação:", e)


# -------------------- LOOP PRINCIPAL --------------------
Experimente:
    move_forward()


    embora Verdadeiro:
        # 1 — verificação de distância
        distância = measure_distance()
        Se a distância < 50:
            stop_motors()
            print ("Obstáculo detetado")
            turn_right()
            move_forward()
            continuar


        #2 — deteção de objetos
        frame = picam2.capture_array()
        resultados = modelo(quadro)


        para resultados em resultados:
            Para a caixa em Result.Boxes:
                label = result.names[int(box.cls[0])].lower()


                se etiquetar em ("garrafa", "chávena"):
                    obj = label.upper()
                    agora = tempo.tempo()
```

Co-funded by the European Union

Este projeto foi financiado com o apoio da Comissão Europeia. Esta comunicação reflete apenas as opiniões do autor, e a Comissão não pode ser responsabilizada por qualquer uso que possa ser feito da informação nela contida.

```
                    cooldown_active = (
                        last_detected["objeto"] == obj e
                        (agora       -       last_detected["tempo"])       <
DETECTION_COOLDOWN
                    )

                    Se não cooldown_active:
                        print(f"Found: {obj}")
                        last_detected["objeto"] = obj
                        last_detected["tempo"] = agora

                        stop_motors()
                        send_notification(obj)
                        print ("À espera de 5 segundos...")
                        tempo.sono(5)

                        turn_right()
                        move_forward()

                    Pausa

exceto o KeyboardInterrupt:
    print ("Programa terminado")

Finalmente:
    stop_motors()
    GPIO.cleanup()
```

Co-funded by the European Union

Este projeto foi financiado com o apoio da Comissão Europeia. Esta comunicação reflete apenas as opiniões do autor, e a Comissão não pode ser responsabilizada por qualquer uso que possa ser feito da informação nela contida.

```
    cv2.destroyAllWindows()

    picam2.stop()
```

---

11. Critérios de Avaliação:

Explique como será medido o sucesso do projeto. Isto pode incluir feedback qualitativo dos participantes, indicadores quantitativos de desempenho ou resultados específicos do projeto.

- **Funcionalidade do Sistema:** Capacidade de replicar o projeto, incluindo Condução Autónoma Básica, Deteção de Copos/Garrafas e Notificação por WhatsApp.
- **Criatividade:** Sugestões de melhorias (novas funcionalidades, novos objetos detetados, etc.). Se for sugerido e tentado (com sucesso ou fracasso), deve ser mais recompensado.
- **Otimizações:** Otimizações voluntárias para o código com sucesso ou fracasso, especialmente em Condução Autónoma e Deteção de Objetos.

---

12. Línguas Disponíveis:

Inglês

---

13. Colaboradores:

- **Fátima Pais, Luís Soares** – Designers e Facilitadores de Projetos

---

14. Feedback e Melhoria Contínua:

Descreva como o feedback será recolhido dos participantes e como esse feedback será usado para melhorar futuros projetos ou iterações deste projeto.

- **Inquérito Pós-Implementação:** Após a implementação do projeto, o feedback será recolhido através de inquéritos direcionados aos participantes envolvidos no processo de testes e observação.

Co-funded by the European Union

Este projeto foi financiado com o apoio da Comissão Europeia. Esta comunicação reflete apenas as opiniões do autor, e a Comissão não pode ser responsabilizada por qualquer uso que possa ser feito da informação nela contida.

- **Sessões de Reflexão da Equipa:** A equipa do projeto organizará sessões internas de revisão para discutir os desafios enfrentados, o que funcionou bem e as áreas que precisam de aperfeiçoamento.

- **Avaliação Técnica:** Observações relativas à precisão da deteção, estabilidade do sistema e durabilidade do hardware serão documentadas para melhorar versões futuras.

- **Atualizações Iterativas:** Com base no feedback e desempenho, serão implementadas atualizações ao código do software e ao design do sistema para melhorar a fiabilidade e a eficácia.

Co-funded by the European Union

Este projeto foi financiado com o apoio da Comissão Europeia. Esta comunicação reflete apenas as opiniões do autor, e a Comissão não pode ser responsabilizada por qualquer uso que possa ser feito da informação nela contida.