**Universidade dos Açores**

Pós-Graduação em Programação, Robótica e Inteligência Artificial (PRIA)

Relatório de Integração de Aprendizagem Automática

Carro Robótico com Visão Computacional usando HuskyLens

Autores: João Pavão, Tiago Martins

Docente: Prof. Armando B. Mendes

Ponta Delgada, 23 de maio de 2025

# Resumo

Este relatório descreve o desenvolvimento de um projeto prático no âmbito da unidade curricular de Aprendizagem Automática, inserido na Pós-Graduação PRIA da Universidade dos Açores. O projeto consiste na construção de um carro robótico autónomo equipado com a câmara HuskyLens, capaz de realizar tarefas de reconhecimento visual e decisão em tempo real.

O relatório explora a ligação entre os componentes físicos do projeto (Arduino UNO R4 WiFi, ponte H L298P, HuskyLens) e os conceitos de Machine Learning lecionados na UC, como classificação supervisionada, KNN, redes neurais e visão computacional. São ainda apresentadas perspetivas futuras, incluindo a expansão com Raspberry Pi e a preparação para eventos como o AzoresBot.

**Palavras-chave:** Aprendizagem Automática, Visão Computacional, HuskyLens, Arduino, Robótica Educacional, IA Embarcada.

# Índice

[Resumo 2](#_Toc197901870)

[Índice 3](#_Toc197901871)

[1. Introdução 4](#_Toc197901872)

[2. Análise Detalhada da HuskyLens 5](#_Toc197901873)

[2.1 Características Físicas e Hardware 5](#_Toc197901874)

[2.2 Processador Kendryte K210 5](#_Toc197901875)

[2.3 Modos de Operação da HuskyLens 5](#_Toc197901876)

[2.4 Machine Learning e Inteligência Artificial Embutida 5](#_Toc197901877)

[3. Relação com a Aprendizagem Automática 7](#_Toc197901878)

[3.1 Tipos de aprendizagem 7](#_Toc197901879)

[3.2 Comparativo com os modelos da UC 7](#_Toc197901880)

[3.3 Frameworks e Processamento 7](#_Toc197901881)

[4. Contribuição do Projeto para a Aprendizagem 8](#_Toc197901882)

[5. Plano de Interação do Robô com a HuskyLens 9](#_Toc197901883)

[6. Expansão do Projeto com Raspberry Pi 10](#_Toc197901884)

[7. Bibliografia e Recursos 11](#_Toc197901885)

[7.1 Bibliografia 11](#_Toc197901886)

[7.2 Webgrafia 11](#_Toc197901887)

[7.3 Prompts utilizados (ChatGPT/OpenAI) 11](#_Toc197901888)

## 1. Introdução

O presente relatório visa documentar a integração de conceitos de Aprendizagem Automática (ML) num projeto prático de robótica baseado na câmara AI HuskyLens. O projeto consiste na construção de um carro robótico com capacidades de visão computacional, utilizando como microcontrolador o Arduino UNO R4 WiFi, uma ponte H L298P, motores DC, rodas, estrutura em 3D e possibilidade de integração de um sensor ultrassónico HC-SR04.

A HuskyLens é uma câmara de visão computacional com modelos de IA pré-treinados e permite a interação direta com algoritmos de classificação e reconhecimento de objetos, rostos, cores e linhas, sem necessidade de programação complexa.

## 2. Análise Detalhada da HuskyLens

### 2.1 Características Físicas e Hardware

- Sensor de imagem: OV2640 (2.0 Megapixeis)

- Interface: UART / I2C / SPI

- Display integrado: ecrã LCD para visualização em tempo real

- Alimentação: 3.3V ou 5V

- Microcontrolador principal: Kendryte K210

### 2.2 Processador Kendryte K210

* Chip especializado em IA embarcada, com foco em visão computacional e reconhecimento de áudio.
* Dual-core RISC-V 64-bit, cada núcleo com FPU de precisão dupla (IEEE754).
* Inclui KPU (Unidade de Processamento Neural) dedicada para executar redes neuronais convolucionais (CNNs).
* Capaz de realizar inferência em tempo real com modelos quantizados de até 5.9 MB (ex: YOLOv2-tiny).
* 8 MB de memória SRAM interna, sendo 2 MB exclusivos para a IA.
* Baixo consumo de energia e operação em larga faixa de temperatura (-40 °C a 125 °C).
* Integra aceleradores de FFT, AES e SHA256, facilitando aplicações com segurança e análise de sinais.

### 2.3 Modos de Operação da HuskyLens

A HuskyLens é uma câmara de visão computacional com algoritmos de IA integrados e interface interativa. Ela suporta múltiplos modos de funcionamento autónomos, prontos para uso:

* **Reconhecimento Facial**
  + Deteta e reconhece rostos humanos com alta precisão. Permite registar múltiplas identidades.
* **Reconhecimento de Objetos**
  + Permite treinar objetos por exemplo visual, associando imagens captadas a etiquetas (rótulos) definidas pelo utilizador.
* **Deteção de Linhas (Line Following)**
  + Ideal para carros robóticos; identifica linhas curvas ou retas para seguimento autónomo.
* **Reconhecimento de Cores**
  + Deteta e classifica cores específicas aprendidas pelo utilizador com um simples toque.
* **Leitura de Códigos QR e de Barras**
  + Capaz de ler códigos QR e barras 1D (como EAN-13), permitindo comunicação via imagem.
* **Deteção de Objetos Pré-treinados (Object Tracking)**
  + Usa modelos integrados (ex: Tiny YOLO) para detetar categorias genéricas (pessoa, carro, etc).
* **Aprendizagem por Exemplos (Supervised Learning)**
  + O utilizador pode mostrar imagens diferentes para cada categoria e a HuskyLens aprende com base nessas amostras visuais.

### 2.4 Machine Learning e Inteligência Artificial Embutida

A HuskyLens integra algoritmos de Machine Learning supervisionado e Deep Learning, sem necessidade de programação externa:

* Aprendizagem direta na câmara, usando apenas o botão físico ou interface serial (I2C/UART).
* Permite treinar modelos diretamente a partir da imagem da câmara, com feedback visual imediato.
* O processo é baseado em classificação supervisionada: a câmara associa uma imagem captada a um rótulo fornecido pelo utilizador.
* O reconhecimento é feito em tempo real, com feedback visual (retângulo, ID, confiança) no ecrã embutido.
* Possui modelos pré-treinados otimizados, mas com capacidade de reaprendizagem contínua baseada em novos exemplos fornecidos.

## 3. Relação com a Aprendizagem Automática

### 3.1 Tipos de aprendizagem

A HuskyLens utiliza predominantemente aprendizagem supervisionada, um tipo de Machine Learning onde o sistema aprende a partir de exemplos rotulados fornecidos pelo utilizador.

* O utilizador apresenta à câmara imagens associadas a rótulos (por exemplo, “Garrafa”, “Pessoa A”), e a HuskyLens utiliza essas amostras para aprender a classificar novas imagens com base na similaridade.
* Este tipo de aprendizagem está em sintonia com os conceitos trabalhados na disciplina, como a classificação supervisionada, e modelos como o KNN (K-Nearest Neighbors).
* Embora a HuskyLens utilize redes neuronais embarcadas, o funcionamento é análogo a um classificador supervisionado: reconhece padrões em imagens com base nos exemplos anteriores.
* Cada classe é aprendida interativamente, e a HuskyLens armazena os vetores de características para comparar com futuras imagens captadas em tempo real.

### 3.2 Comparativo com os modelos da UC

Durante a unidade curricular foram estudados vários algoritmos de Machine Learning, cuja aplicação pode ser observada ou inspirada nas funcionalidades do projeto com a HuskyLens e o processador Kendryte K210:

**KNN (K-Nearest Neighbors)**

Reflete a lógica de reconhecimento baseado em exemplos usada pela HuskyLens. Quando o utilizador fornece imagens rotuladas, a câmara classifica novas imagens comparando-as com as mais semelhantes, como num sistema KNN.

**Redes Neuronais**

A HuskyLens executa inferência com redes neuronais convolucionais (CNNs) otimizadas no processador Kendryte K210, através da KPU (Kendryte Processing Unit). Este componente foi desenvolvido para realizar inferência em tempo real com modelos quantizados, sem necessidade de computação externa.

**Classificação e Regressão**

O projeto base atual utiliza classificação supervisionada, mas pode evoluir para regressão supervisionada, por exemplo:

* Estimar a distância de um objeto com sensores como o HC-SR04.
* Associar valores contínuos a variáveis visuais detetadas pela câmara.

**Clustering (Agrupamento Não Supervisionado)**

Embora não implementado diretamente, futuras versões do projeto poderão incluir técnicas de agrupamento de imagens captadas, permitindo identificar padrões emergentes sem rótulos prévios, útil em análise de comportamento ou deteção de anomalias.

### 3.3 Frameworks e Processamento

Embora a HuskyLens execute os modelos de IA embarcada diretamente no dispositivo, também foi considerada a possibilidade de expandir o projeto com frameworks de Machine Learning e Visão Computacional. Esta abordagem permite maior controlo sobre o treino de modelos, análise de desempenho e experimentação com diferentes algoritmos.

**TensorFlow**

Permite construir e treinar redes neuronais personalizadas, que podem ser posteriormente quantizadas e convertidas para formatos compatíveis com dispositivos embarcados, como o K210 (via NNCase). É ideal para projetos que exigem maior flexibilidade e precisão no modelo.

**scikit-learn**

Oferece uma coleção rica de algoritmos clássicos de ML, como KNN, SVM, regressão e clustering. Pode ser usado para analisar os dados recolhidos pela HuskyLens ou sensores complementares (ex.: distância, temperatura), e construir classificadores externos.

**OpenCV**

Biblioteca essencial para processamento de imagem, usada para:

* Captura e pré-processamento de imagens
* Deteção de bordas, contornos e objetos
* Extração de características visuais para posterior classificação

Ao integrar estas ferramentas com dados captados pela HuskyLens (ou por uma webcam adicional), é possível realizar treino de modelos personalizados fora do dispositivo, com posterior implementação ou comparação com os resultados da IA embarcada.

## 4. Contribuição do Projeto para a Aprendizagem

- Exploração prática de \*\*aprendizagem supervisionada\*\*.

- Integração de \*\*IA embarcada\*\* em dispositivos reais.

- Reflexão sobre \*\*limitações de modelos pré-treinados\*\* vs. modelos personalizados.

- Estabelecimento de ligação entre teoria e aplicação através de hardware educacional.

## 5. Plano de Interação do Robô com a HuskyLens

A proposta prática para a demonstração do robô com HuskyLens, discutida com o Professor Mathias Funck (UC de Programação), inclui o desenvolvimento de \*\*tarefas interativas\*\* baseadas em visão computacional. Estas tarefas representam ações específicas do robô baseadas na interpretação da cena visual captada pela HuskyLens:

- \*\*Seguir uma linha detetada\*\* pela HuskyLens em modo line tracking.

- \*\*Parar ou mudar de direção\*\* ao reconhecer um objeto específico ou cor definida.

- \*\*Aproximar-se de um rosto ou recuar\*\* quando este sai do campo de visão.

- \*\*Ler códigos QR ou etiquetas visuais\*\* e reagir de forma autónoma.

Estas interações serão programadas com base em \*\*estruturas condicionais\*\* e permitirão combinar a programação do Arduino com as capacidades de reconhecimento do sensor.

## 6. Expansão do Projeto com Raspberry Pi

Está prevista a criação de um novo carro robótico com maior capacidade de processamento, utilizando \*\*Raspberry Pi 5\*\* como unidade central e câmaras Pi ou webcams USB comuns. Esta nova versão poderá constituir um projeto autónomo e independente, integrando as seguintes características:

- Treino e execução de modelos personalizados de ML com bibliotecas como \*\*TensorFlow\*\*, \*\*Keras\*\*, \*\*YOLO\*\* e \*\*MediaPipe\*\*.

- Reconhecimento de objetos, rostos, gestos ou caminhos usando visão computacional em tempo real.

- Integração de sensores adicionais e fusão sensorial.

- Navegação em ambientes estruturados (labirintos, trilhos, obstáculos).

- Comparação entre a performance da HuskyLens e dos modelos desenvolvidos no Raspberry Pi.

Este projeto paralelo poderá ser apresentado em futuras mostras científicas, com especial destaque para o \*\*desenvolvimento de modelos de ML controlados e programados inteiramente pelos alunos\*\*.

## 7. Bibliografia e Recursos

### 7.1 Bibliografia

- Apostilas da UC de Aprendizagem Automática – Prof. Armando B. Mendes

- Documentação técnica sobre Machine Learning (KNN, árvores de decisão, clustering, regressão)

- Livros e artigos científicos consultados sobre IA embarcada

### 7.2 Webgrafia

- [Wiki Oficial da HuskyLens - DFRobot]

(<https://wiki.dfrobot.com/HUSKYLENS_V1.0_SKU_SEN0305_SEN0336>)

- [Datasheet Kendryte K210]

(<https://cdn.hackaday.io/files/1654127076987008/kendryte_datasheet_20181011163248_en.pdf>)

- [HuskyLens WIKI Document.pdf]

(<https://raw.githubusercontent.com/DFRobot/Wiki/master/SEN0305/res/HuskyLens%20WIKI%20Document.pdf>)

### 7.3 Prompts utilizados (ChatGPT/OpenAI)

- 'Cria uma estrutura de relatório para um projeto com a HuskyLens relacionado com Machine Learning.'

- 'Explica o funcionamento do processador Kendryte K210.'

- 'Compara a aprendizagem supervisionada da HuskyLens com KNN.'

- 'Sugere planos de interação entre um robô Arduino e a HuskyLens para demonstração num evento.'

- 'Cria uma versão estendida do relatório destacando IA embarcada e futuras aplicações com Raspberry Pi.'