

## 2.º Projeto Laboratorial

Contagem de Veículos em Imagens de Vídeo

**Ricardo Faria**

*No. Estudante 51589*

**Bruno Pereira**

*No. Estudante 51811*

**Prof. Eng.º Pedro Mendes Jorge**

**Prof. Eng.º João Costa**

*Docentes Responsáveis*

Instituto Superior de Engenharia de Lisboa

Departamento de Engenharia Informática

Licenciatura em Engenharia Informática e Multimédia

7 de Dezembro de 2025

# Conteúdo

<i>Lista de Figuras</i>	ii
<i>Lista de Tabelas</i>	iii
<i>Glossário</i>	iii
<i>Síglas</i>	iii
<b>1 Introdução</b>	<b>1</b>
1.1 Objetivos do trabalho . . . . .	2
<b>2 Metodologia</b>	<b>3</b>
2.1 Arquitetura do Sistema . . . . .	3
2.2 Cálculo do fundo ( <i>Background</i> ) . . . . .	3
2.3 Detecção de veículos . . . . .	4
2.4 Cálculo de velocidade . . . . .	4
2.5 Pipeline principal (count) . . . . .	4
<b>3 Resultados</b>	<b>5</b>
3.1 <i>Background</i> . . . . .	5
3.2 <i>Region of Interest</i> . . . . .	5
3.3 Detecção e contagem . . . . .	6
3.4 Cálculo da Velocidade . . . . .	6
<b>4 Conclusões</b>	<b>7</b>

# Lista de Figuras

1.1	Momentos do vídeo de detecção e contagem de carros . . . . .	1
3.1	Background obtido. . . . .	5
3.2	Região de interesse (ROI). . . . .	6
3.4	Cálculo da velocidade dos carros. . . . .	6

# Lista de Tabelas

2.1	Resumo da arquitetura do sistema. . . . .	3
-----	---	---

# Siglas

**OpenCV** *Open Source Computer Vision.* (p. 2)

**PIV** Processamento de Imagem e Visão. (p. 1, 2)

**TL2** Trabalho Laboratorial 2. (p. 2)

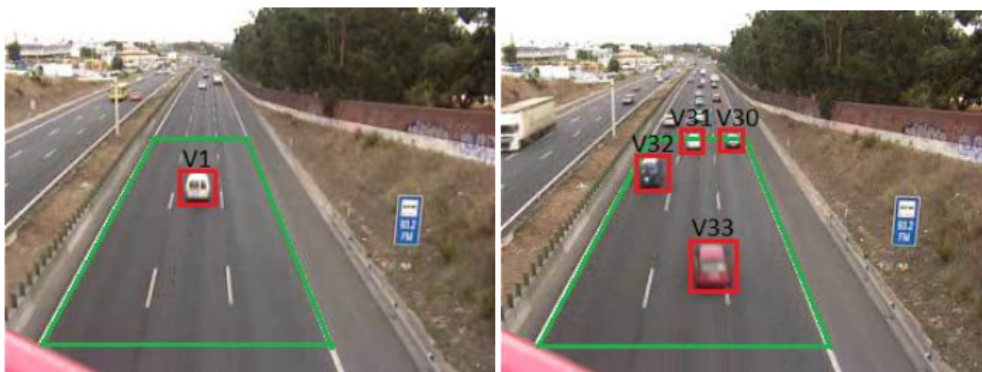
**UC** Unidade Curricular. (p. 1)

# Capítulo 1

## Introdução

A Unidade Curricular de **Processamento de Imagem e Visão (PIV)** avalia os conhecimentos adquiridos pelos alunos por meio de duas componentes de igual peso: *i)* a componente teórica, correspondente a um exame escrito; *ii)* e a componente laboratorial, que consiste em parte no desenvolvimento e documentação de dois projetos de grupo – entre os quais se inclui o presente documento.

Pretende-se desenvolver um algoritmo capaz de detectar e contar o número de veículos que circulam numa via de autoestrada. A **Figura 1.1** ilustra exemplos da saída que o algoritmo poderá apresentar.



**Figura 1.1:** Momentos do vídeo de deteção e contagem de carros

## 1.1 Objetivos do trabalho

[2.º Trabalho Laboratorial - T2A1sem2526.pdf](#) (Moodle de Processamento de Imagem e Visão 2025/2026).

O objetivo principal do TL2 é:

1. Desenvolver um algoritmo capaz de **detetar e contar** o número de veículos que circulam numa via de autoestrada.
2. Opcionalmente, o algoritmo poderá também **estimar a velocidade** dos veículos e a **velocidade média** em cada via da autoestrada.

Para efeitos de visualização, o objetivo também inclui:

1. Sobrepor **Bounding Boxes** à sequência original, uma para cada veículo detetado.
2. Incluir um **identificador** que representa o número de ordem do veículo.

Todo o código deve ser desenvolvido em **Python/OpenCV**.

## Capítulo 2

# Metodologia

No seguimento da apresentação do contexto de trabalho ([Capítulo 1](#)), serão detalhados os métodos propostos para a contagem de veículos em imagens de vídeo.

Para o efeito, foi desenvolvida uma sequência de funções modular, composta pelas seguintes etapas de processamento:

1. **Cálculo do fundo** (background) por média de frames.
2. **Deteção de veículos** por subtração de fundo, binarização e análise de contornos.  
Cálculo de velocidade com base no deslocamento do centróide.
3. **Rastreamento e contagem** com atribuição de IDs persistentes.
4. **Visualização** dos resultados em vídeo.

O presente relatório descreve a arquitetura do sistema, as opções de implementação tomadas em cada método, bem como a avaliação experimental realizada com base no vídeo de teste.

### 2.1 Arquitetura do Sistema

Inseridos no ficheiro `src/main.ipynb`, encontram-se 4 funções principais.

**Tabela 2.1:** *Resumo da arquitetura do sistema.*

Função	Descrição
<code>background()</code>	Calcula a imagem de fundo média a partir de uma amostra de frames.
<code>detect()</code>	Deteta veículos no frame atual com recurso à subtração de fundo e operações morfológicas.
<code>calc_speed()</code>	Calcula a velocidade (km/h) com base no deslocamento do centróide.
<code>count()</code>	Função principal que orquestra o processamento do vídeo, rastreamento e anotação.

### 2.2 Cálculo do fundo (*Background*)

A função `background()` calcula uma imagem de fundo estática como a média aritmética de até 300 frames amostrados a cada 3 frames. Este método reduz a influência de



veículos em movimento no fundo calculado.

---

```
1 def background(video_path, max_frames=300, step=3):
```

---

## 2.3 Detecção de veículos

A função `detect()` realiza: 1) Suavização Gaussiana do *frame* e do fundo; 2) Subtração absoluta entre *frame* e fundo; 3) Binarização com método de Otsu; 4) Aplicação de máscara ROI (*region of interest*) para restringir a análise; 5) Operações morfológicas (fechamento e abertura) para limpeza; 6) Extração de contornos e filtragem por área e *aspect ratio*.

---

```
1 def detect(frame, background_blur, mask_roi):
```

---

Filtros aplicados:

- **Área mínima:** 800 px<sup>2</sup>
- **Área máxima:** 10 000 px<sup>2</sup>
- **Aspect ratio:** entre 0.3 e 3.0

## 2.4 Cálculo de velocidade

A função `calc_speed()` estima a velocidade em km/h com base no deslocamento do centroide (pixels), FPS do vídeo e fator de conversão `meters_per_pixel` (padrão: 0.25 m/px).

$$v = \Delta p \times mpp \times fps \times 3.6$$

## 2.5 Pipeline principal (count)

A função `count()` integra todas as etapas:

1. Gera fundo e máscara ROI.
2. Processa cada *frame* com `detect()`.
3. Rastreia veículos por proximidade de centróides.
4. Calcula velocidade média móvel (últimas 3 medidas).
5. Desenha *bounding boxes*, IDs e velocidades.
6. Produz vídeo de saída.

## Capítulo 3

# Resultados

Neste capítulo, procede-se à análise da performance do sistema de visão computacional, com base nos resultados obtidos no vídeo de teste. Esta análise visa não só validar a eficácia da metodologia proposta, mas também identificar as falhas e as fontes de erro no algoritmo automático de contagem de veículos em imagens de vídeo.

O sistema foi testado com o vídeo `AutoEstrada.avi`, utilizando a ROI definida pelos pontos:  $[100, 70]$ ,  $[155, 70]$ ,  $[230, 240]$ ,  $[5, 240]$

### 3.1 *Background*

Na figura **Figura 3.1**, encontra-se o *background* calculado a partir do vídeo recebido. Esta imagem será utilizada como referência para a deteção de movimento, ou seja, qualquer objeto que surja para além deste exemplo será considerado um veículo em movimento.



**Figura 3.1:** *Background obtido.*

### 3.2 *Region of Interest*

Posteriormente ao cálculo dos *frames* médios para comparação, delimitamos a área de análise do vídeo conforme pedido no enunciado.

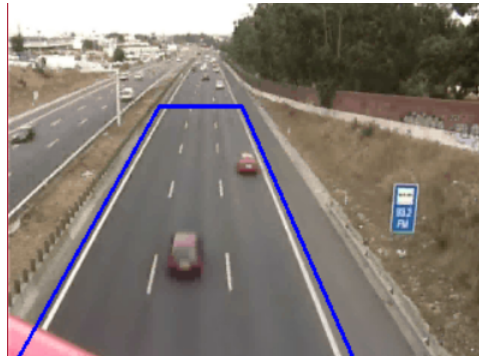
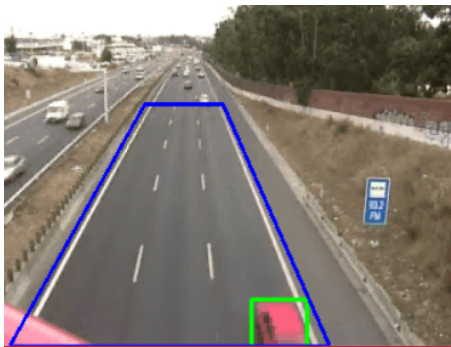


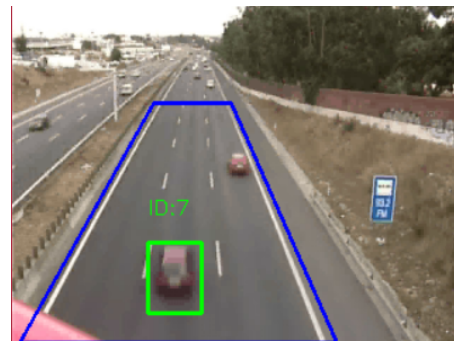
Figura 3.2: Região de interesse (ROI).

### 3.3 Detecção e contagem

Seguidamente, é efetuada a deteção do movimento dos carros, rastreada pelas *bounding boxes* que os acompanham. Assim, torna-se também possível efetuar a contagem.



(a) Detecção de veículos.



(b) Contagem de veículos.

### 3.4 Cálculo da Velocidade

Por fim, principalmente através do movimento do centro da *bounding box*, *frame a frame*, é calculada a velocidade de cada um dos carros.

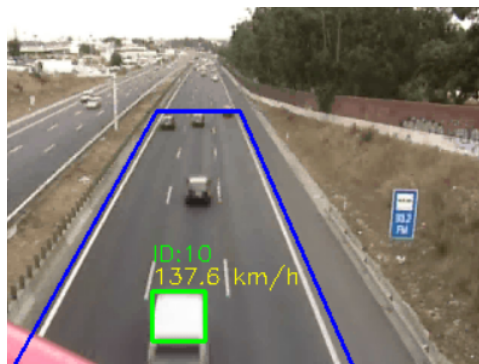


Figura 3.4: Cálculo da velocidade dos carros.

## Capítulo 4

# Conclusões

O projeto culminou na implementação bem-sucedida de um sistema de visão computacional que cumpre os requisitos de detecção, rastreamento e estimativa de velocidade de veículos em vídeos de autoestrada. A arquitetura adotada, baseada na subtração de fundo, processamento morfológico e rastreamento de centróides, demonstrou ser funcional e estruturalmente correta.

Apesar da funcionalidade central, o desempenho global do sistema foi comprometido por limitações na fase inicial de segmentação e extração de contornos.

A precisão deficiente das *bounding boxes* geradas introduziu instabilidade, resultando em falsos positivos, o que afeta diretamente a fiabilidade da contagem. Esta instabilidade refletiu-se significativamente na estimativa de velocidade, onde os valores calculados se apresentaram frequentemente exagerados. Esta discrepância é atribuída tanto a uma calibração inadequada do fator de conversão pixel-para-metros quanto à instabilidade na detecção dos centroides entre frames. Adicionalmente, o objetivo opcional de cálculo da velocidade média por via não foi cumprido.

