



Sistemas de Visão e Percepção Industrial

Trabalho Prático nº 2 - Abril/Maio 2023
Classificação de sinalética usando Matlab

1 Objetivo

Desenvolvimento de um programa em Matlab para fazer o processamento automático de imagens para detetar, classificar e contar o número e tipo de determinados símbolos em imagens fornecidas, de acordo com os parâmetros descritos. Serão dadas imagens exemplo para testar o programa, mas a avaliação será feita sobre outras imagens de natureza similar.

2 Descrição

As imagens que o programa deve processar consistem num fundo variável sobre o qual são colocadas diversas figuras de símbolos de determinados tipos e variedades. Os símbolos ocorrem num total de 12 tipos diferentes com 3 variedades para cada um. Usam-se indistintamente as palavras “símbolo” e “objeto”, e os tipos de símbolos (objetos) são os seguintes:

`bio, corrosive, electrical, explosive, infopoint, laser, litter, nophone,
nosmoking, toxic, wc, wifi`

A figura seguinte ilustra a lista completa de símbolos que podem surgir nas imagens:



Figura 1: Lista de todos os objetos que podem surgir nas imagens a processar

Os objetos podem estar em orientações e escalas variáveis na imagem, e poderá haver ligeiras flutuações de cor ao longo das diversas imagens. Também haverá objetos sobre o bordo da imagem. Não haverá objetos em contacto entre si. O fundo das imagens será variável e as possibilidades serão dadas a conhecer posteriormente com vários exemplos. Alguns objetos poderão estar envolvidos por uma moldura **retangular** (caixilho) de cor clara, mas isso não representa nenhum novo objeto, e a existência de moldura não altera o tipo do objeto. As dimensões da moldura serão variáveis. Os objetos sobre o bordo da imagem nunca terão moldura. A figura seguinte ilustra alguns exemplos de objetos com moldura em fundo preto. Pela sua natureza, alguns objetos podem ter eles próprios um bordo integrado no próprio



Figura 2: Exemplos de objetos representados com moldura

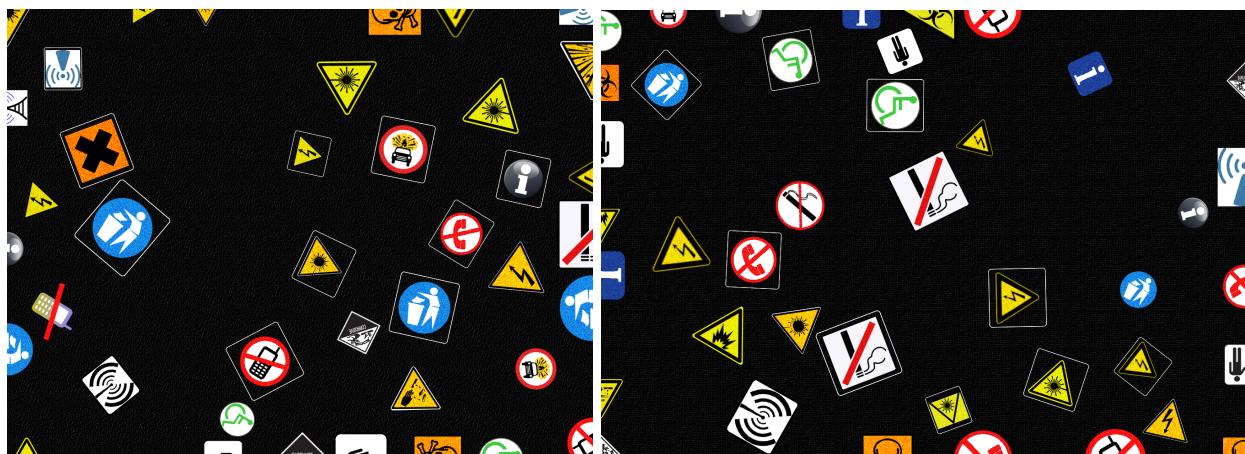
símbolo, mas nunca será simultaneamente de cor clara e forma retangular, e não é considerado objeto com moldura. A Figura 2 ilustra alguns objetos que têm uma moldura.

Quando executado, o programa a desenvolver pelo aluno deve ler um conjunto de imagens fornecidas com nomes do género `svpi2023_TP2_img_MMM_NN.png` onde NN poderá variar de 01 até 99; esse número NN deverá ser detetado automaticamente pelo programa do aluno, como descrito adiante. MMM será um número de três dígitos que designa a sequência de imagens em análise e também deverá ser detetado automaticamente a partir no nome do ficheiro.

O programa a desenvolver pelo aluno deve analisar as imagens da sequência, uma por uma, e gerar uma tabela de estatísticas com tantas linhas quantas as imagens da sequência, e onde em cada linha se indicam as contagens de objetos detetados. Esta tabela será sujeita a avaliação como descrito mais adiante, e deve ser escrita pelo programa do aluno num ficheiro com nome `tp2_nnnnnnn.txt` onde nnnnnn é o número mecanográfico do aluno. Nesse ficheiro, as respostas para cada imagem devem aparecer numa linha e separadas por vírgulas. Por exemplo, para o aluno número 999999, o código Matlab do trabalho deve gerar o ficheiro `tp2_999999.txt` que deverá conter os seguintes valores separados por vírgulas:

- Número mecanográfico do aluno; [NumMec]
- Número da sequência; [NumSeq]
- Número da imagem na sequência; [NumImg]
- Número total de objetos que tocam ou intersetam o bordo da imagem; [ObjBord]
- Número total de objetos que não intersetam o bordo da imagem; [ObjOK]
- Número total de objetos que tem um caixilho retangular; [ObjFrame]
- Lista de objetos por tipo: de entre os objetos que não intersetam o bordo, indicar, pela ordem apresentada anteriormente (alfabética), quantos objetos há de cada um dos 12 tipos. Se um dado tipo não tiver objetos na imagem deverá constar na tabela de resultados com valor 0.

Seja um caso hipotético com duas imagens a analisar com nomes `svpi2023_TP2_img_181_03.png` e `svpi2023_TP2_img_181_10.png`, como se ilustra:



Nesse caso, para o aluno de número mecanográfico 999999, o ficheiro `tp2_999999.txt` a gerar deveria ter o seguinte conteúdo e formato (duas linhas):

999999,181,3,20,20,10,0,2,3,2,1,3,2,3,0,1,1,2
999999,181,10,20,21,9,0,0,5,1,2,3,2,1,3,0,3,1

O significado destas linhas comprehende-se melhor pela análise da seguinte tabela reportada às duas imagens.

NumMec	NumSeq	NumImg	ObjBord	ObjOK	ObjFrame	bio	corrosive	electrical	explosive	infopoint	laser	litter	nophone	nosmoking	toxic	wc	wifi
999999	181	3	20	20	10	0	2	3	2	1	3	2	3	0	1	1	2
999999	181	10	20	21	9	0	0	5	1	2	3	2	1	3	0	3	1

Para gravar o ficheiro `tp2_nnnnnn.txt`, recomenda-se o uso da função `writematrix()` do Matlab, embora haja outras que se possam usar (como `fwrite()` ou outras).

2.1 Variantes do problema e opção de dificuldade

Para os estudantes que o pretendem, é possível resolver o problema de forma parcial com a consequente redução da classificação máxima. Assim, é possível optar por sequências de imagens onde é garantido não surgirem certas variedades de objetos, simplificando a resolução do problema:

Opção 1 (dificuldade maior) – Todos os objetos listados podem surgir nas imagens.

Opção 2 (dificuldade intermédia) – Só surgem objetos das variedades 1 e 2 de cada tipo.

Opção 3 (dificuldade menor) – Só surgem objetos da variedade 1 de cada tipo.

Classificação máxima atingível:

Opção 1 – 20 valores

Opção 2 – 17 valores

Opção 3 – 14 valores

As sequências (conjuntos) de imagens serão diferentes consoante a opção de dificuldade. As sequências distinguem-se pelo seu número **MMM** no nome genérico `svpi2023_TP2_img_MMM_NN.png`.

- Sequências com **MMM** terminado em 3 (exemplo, 503) são sequências de opção 3.
- Sequências com **MMM** terminado em 2 (exemplo, 302) são sequências de opção 2.
- Sequências com **MMM** terminado em 1 (exemplo, 411) são sequências de opção 1.

Para efeitos de avaliação do trabalho, a nota máxima será determinada pelo grau de dificuldade ditada pelo número da sequência em análise. Ou seja, as sequências terminadas em 1 são avaliadas com a cotação máxima quando estiverem todas certas, e as sequências terminadas em 2 ou 3 serão avaliadas com cotação mais reduzida, mesmo que estejam completamente certas. Recomenda-se portanto que os alunos façam as suas opções com esta consciência.

N.B. Os alunos devem desenvolver o trabalho analisando apenas as imagens das opções 1, 2 ou 3 e **NÃO** todas as imagens para além das que correspondem à opção escolhida porque isso poderá resultar numa classificação final inesperada dado que as imagens das opções 2 e 3 serão classificadas de forma reduzida face às imagens da opção 1, conforme explicado!

De seguida, apresentam-se exemplos de imagens e soluções respeitantes às opções 2 e 3 de dificuldade.

3 Exemplo de resultado para a opção 2 de dificuldade

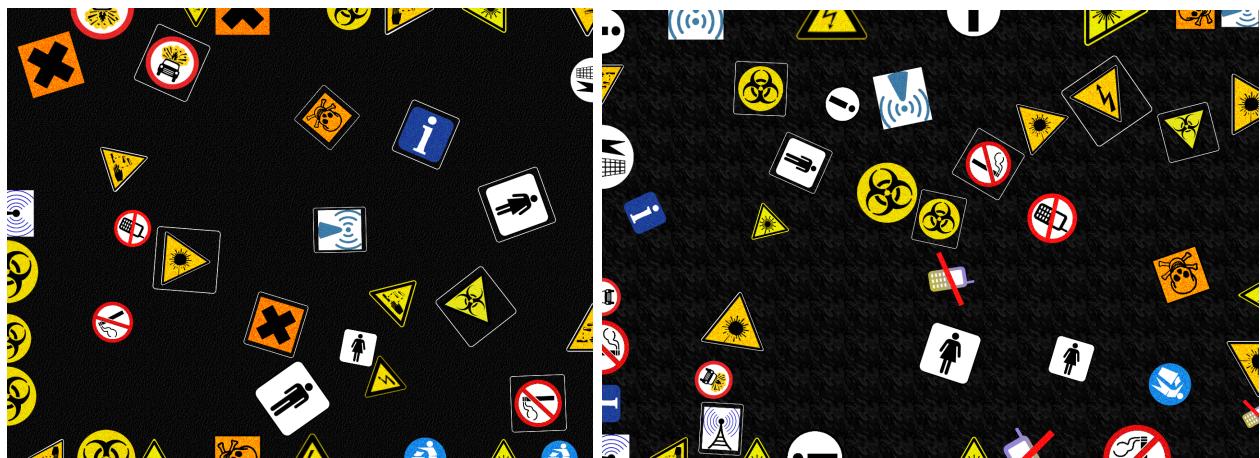
Foram usadas as imagens: svpi2023_TP2_img_182_03.png e svpi2023_TP2_img_182_05.png.

Resultados:

999999,182,3,21,17,9,1,2,1,1,1,1,0,1,2,3,3,1

999999,182,5,23,21,7,4,0,1,1,2,3,1,2,1,1,3,2

NumMec	NumSeq	NumImg	ObjBord	ObjOK	ObjFrame	bio	corrosive	electrical	explosive	infopoint	laser	litter	nophone	nosmoking	toxic	wc	wifi
999999	182	3	21	17	9	1	2	1	1	1	1	0	1	2	3	3	1
999999	182	5	23	21	7	4	0	1	1	2	3	1	2	1	1	3	2



4 Exemplo de resultado para a opção 3 de dificuldade

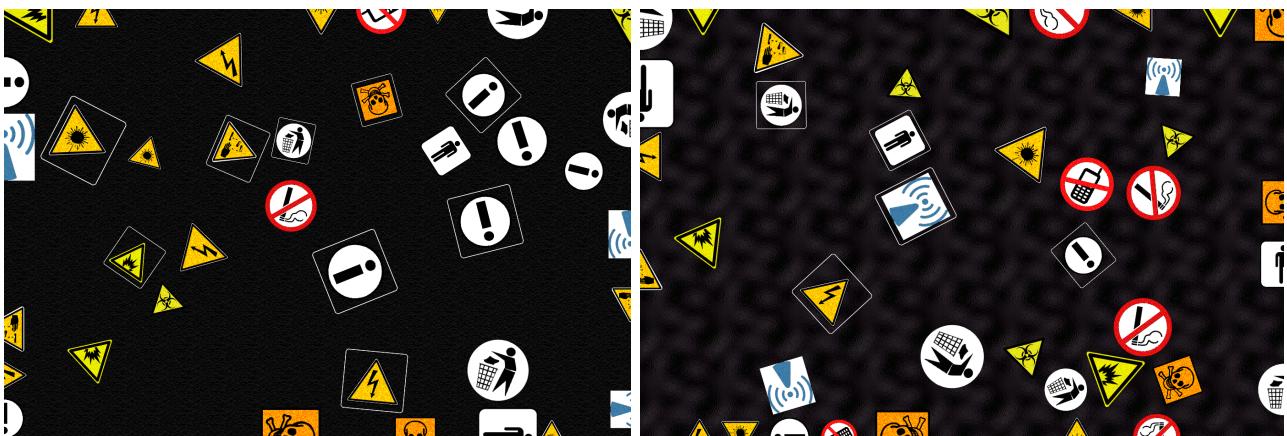
Foram usadas as imagens: svpi2023_TP2_img_203_02.png e svpi2023_TP2_img_203_03.png.

Resultados:

999999,203,2,20,19,9,1,1,3,2,5,2,2,0,1,1,1,0

999999,203,3,20,20,5,3,1,1,2,1,1,3,1,2,1,1,3

NumMec	NumSeq	NumImg	ObjBord	ObjOK	ObjFrame	bio	corrosive	electrical	explosive	infopoint	laser	litter	nophone	nosmoking	toxic	wc	wifi
999999	203	2	20	19	9	1	1	3	2	5	2	2	0	1	1	1	0
999999	203	3	20	20	5	3	1	1	2	1	1	3	1	2	1	1	3



5 Sugestões para a metodologia

O trabalho tem múltiplas formas de ser resolvido, mas os passos principais recomendados são:

- Criar uma base dos objetos de referência (a partir das imagens de referência fornecidas).
- Identificar e estabelecer os descritores relevantes para distinguir os diversos objetos.
- Nas imagens a processar separar os objetos do fundo (binarização, deteção de contornos, ou outros, são técnicas esperadas para o fazer).
- Eliminar os objetos em contacto com o bordo da imagem.
- Obter os descritores dos objetos restantes (na representação em máscara binária e/ou na representação completa a cores).
- Classificar os objetos por comparação de descritores com os objetos de referência mediante critérios de distância, ou outros.
- Contar as ocorrências de cada classe/tipo de objetos e atualizar o ficheiro de resultados.

6 Formato do ficheiro com o trabalho do aluno (.m file)

O trabalho de cada aluno deve ficar contido num único ficheiro Matlab do tipo .m e que terá o formato de função (*function*), devendo retornar um único valor que é o numero mecanográfico do aluno.

Se o aluno desenvolver funções auxiliares necessárias ao trabalho, elas deverão ficar incluídas nesse mesmo ficheiro, na parte final, depois da função principal. O mesmo se aplica se forem criadas bases de descritores ou parâmetros de referência a usar na classificação. Esses dados devem ficar definidos de forma estática em variáveis internas ao programa, e NÃO podem ser calculadas durante a execução a partir de quaisquer imagens de referência, ou carregadas a partir de qualquer ficheiro de variáveis previamente gerado pelo Matlab (como os ficheiros *.mat).

7 Procedimento para entrega do trabalho

O aluno deve executar o comando `SVPI_ProcessTP2_2023` (fornecido on-line) usando como argumento o nome do seu ficheiro Matlab. Por exemplo, se o programa do aluno se designar `tp2_999999.m` então, dentro da pasta que o contém, deve executar o comando:

```
SVPI_ProcessTP2_2023('tp2_999999.m')
```

Se o programa do aluno executar sem erros, será criado o ficheiro `cod_svp2023_tp2_nnnnnn.vsz` (com `nnnnnn=999999` neste exemplo). Este é o ficheiro que deve ser entregue no E-learning e que é um ficheiro de arquivo que inclui o código desenvolvido pelo aluno. Se o ficheiro de respostas `tp2_nnnnnn.txt` não for gerado pelo programa do aluno, então o comando `SVPI_ProcessTP2_2023` aborta em erro. Só o ficheiro com extensão `*.vsz` é aceite para entrega no E-learning e nenhum outro será considerado.

Resumo das condições necessárias para se conseguir a geração do ficheiro a entregar `cod_svp2023_tp2_nnnnnn.vsz`:

- O programa do aluno não pode ter erros de execução.
- A pasta onde está a ser executado não pode ter nenhum ficheiro `svpi2023_TP2_img_*_*.png`
- Os ficheiros de imagem `svpi2023_TP2_img_*_*.png` têm de estar na pasta anterior (`../`)
- O programa do aluno (formato de função Matlab) tem de devolver o número mecanográfico.
- O programa do aluno tem de gerar o ficheiro `tp2_nnnnnn.txt` onde `nnnnnn` é o número mecanográfico.

8 Avaliação

A avaliação é baseada em dois parâmetros principais:

1. Conformidade dos resultados do trabalho de acordo com imagens de avaliação.
2. Apreciação do código fornecido.

A conformidade dos resultados é feita essencialmente com base na taxa de resultados corretos em comparação com os resultados reais resultantes de uma análise correta às imagens de avaliação. A apreciação do código fornecido poderá contemplar a eficiência geral, nomeadamente no tempo de execução. Programas com tempos de execução demasiado longos (mais de 30 segundos, em média, por imagem de uma sequência) poderão ser penalizados, ou mesmo excluídos de avaliação. O código entregue pelos alunos será sujeito a um sistema de verificação de plágio. Todos os trabalhos que apresentarem um índice de plágio maior do que um dado limite são passíveis de análise e tratamento específicos, conforme comunicado nas aulas.

9 Observações e recomendações sobre o trabalho e o Matlab

- Os trabalhos serão avaliados na versão R2022b do Matlab.
- O programa a executar deve-se chamar `tp2_nnnnnn.m` (onde `nnnnnn` é o numero mecanográfico do aluno)
- O programa deve procurar as imagens na pasta anterior à sua. (`../`)
- O programa deve correr de forma não interativa, ou seja, não deve ter interface gráfica nem deve esperar entrada de dados do utilizador.
- O programa deverá ser capaz de determinar quantas imagens a pasta tem, para as poder abrir e analisar.

NB. Para designar nomes de pastas (diretórios) não usar a barra do tipo *backslash* "\ ou barra invertida (como é usual em Microsoft Windows), mas usar a barra do tipo *slash* "/". Por exemplo, num programa em Matlab usar "../" e não "..\" para designar a pasta anterior. Em Matlab ambas são aceitáveis mas terão significados diferentes conforme o sistema operativo e poderá interferir no processo de execução durante a avaliação.

NB. A pasta de imagens terá sempre três grupos (sequências) de imagens, uma por cada opção de dificuldade (1, 2, 3). O número de imagens por sequência é variável. Cabe ao programa de cada aluno escolher qual a sequência (opção de dificuldade) que quer processar. Se por algum motivo o programa do aluno processar mais do que uma das três sequências (opção de dificuldade), a classificação final será obtida por média das classificações parciais obtidas nos respetivos escalões dentro de cada opção.

Os fundos das imagens serão variáveis, como nos exemplos dados no enunciado e outros exemplos disponibilizados na página do e-learning, onde poderá haver diversas texturas.

O programa deve gerar, na pasta corrente, o ficheiro `tp2_nnnnnn.txt`, (onde `nnnnnn` é o número mecanográfico do aluno) nos moldes descritos anteriormente.

As imagens a processar estarão na pasta anterior à pasta que contém o programa. Por exemplo, se o programa do aluno estiver numa pasta de nome parcial "as_minhas_pastas_locais/svpi/tp2/" então as imagens estarão em "as_minhas_pastas_locais/svpi/".

Ou seja, para tornar o programa independente do nome da pasta onde se encontre, ele deve procurar as imagens na pasta anterior que se representa pela expressão "...".

Para obter uma lista de ficheiros, para depois os poder processar, sugere-se o uso da função `dir()` do Matlab para listar o conteúdo de uma pasta. Neste caso, e para obter a lista de **todas** imagens que estão na pasta anterior, pode-se fazer algo como:

```
lista=dir('..\\svpi2023_TP2_img_*.png');
```

Porém, como se pretende resolver o problema para uma **única** opção de dificuldade, para o estudante restringir a listagem de ficheiros para a opção 1 de dificuldade, deverá fazer algo como:

```
lista1=dir('..\\svpi2023_TP2_img_*1_*.png');
```

Neste caso, a variável `lista1` conterá uma listagem de todos os ficheiros que obedecem à regra do nome das imagens para a opção 1 de dificuldade, conforme descrito no enunciado. De igual modo, se o estudante optar por outra opção de dificuldade, como por exemplo a opção 2, poderá usar o seguinte comando:

```
lista2=dir('../svpi2023_TP2_img_*2_*.png');
```

Dos casos anteriores, a variável `lista`, `lista1`, `lista2`, etc. será uma estrutura em Matlab (*structure*) que conterá toda a informação sobre os ficheiros. Por exemplo:

```
size(lista1,1) -> dá o número de ficheiros encontrados pelo comando dir.  
lista1(3).name -> dá o nome do terceiro ficheiro da lista1.
```

Os estudantes escolhem as imagens que pretendem processar de acordo com o grau de dificuldade e nas condições descritas no enunciado. Podem, portanto, optar pela sequência de dificuldade 1, 2 ou 3. Qualquer que seja a sequência escolhida, todas as imagens da sequência devem ser processadas e não apenas um subconjunto parcial da sequência.

Para a geração de resultados, além da contabilização dos objetos, como descrito no enunciado, é necessário incluir o número da sequência e o número da imagem na sequência. Todos estes números se extraem do próprio nome do ficheiro da imagem. Basta observar que o nome do ficheiro é um *array* de caracteres sempre com o mesmo comprimento e, portanto, as indicações do número de sequência e do número de imagem estão sempre na mesma posição.