Universidade de Aveiro
Dep. de Engenharia Mecânica

Sistemas de Visão e Percepção Industrial

Trabalho Prático nº 1 - Março/Abril 2022

Deteção e análise de imagens de peças de jogos

Objetivo

Desenvolvimento de um programa em Matlab para fazer análise de imagens que contêm peças de dominó, faces de dados e cartas de jogar, bem como outros objetos sem significado. O programa deve ser capaz de interpretar imagens fornecidas e de gerar os resultados pedidos conforme descrito adiante. Serão dadas imagens exemplo para permitir o desenvolvimento, mas as imagens usadas para obter os resultados de avaliação serão novas.

Natureza das imagens

As imagens a processar terão um número variável de objetos dos diversos tipos, com eventual ruído ou textura na zona do fundo. Os objetos estarão delimitados por caixilhos retangulares ou quadrados. A figura 1 ilustra um exemplo de imagem a processar.

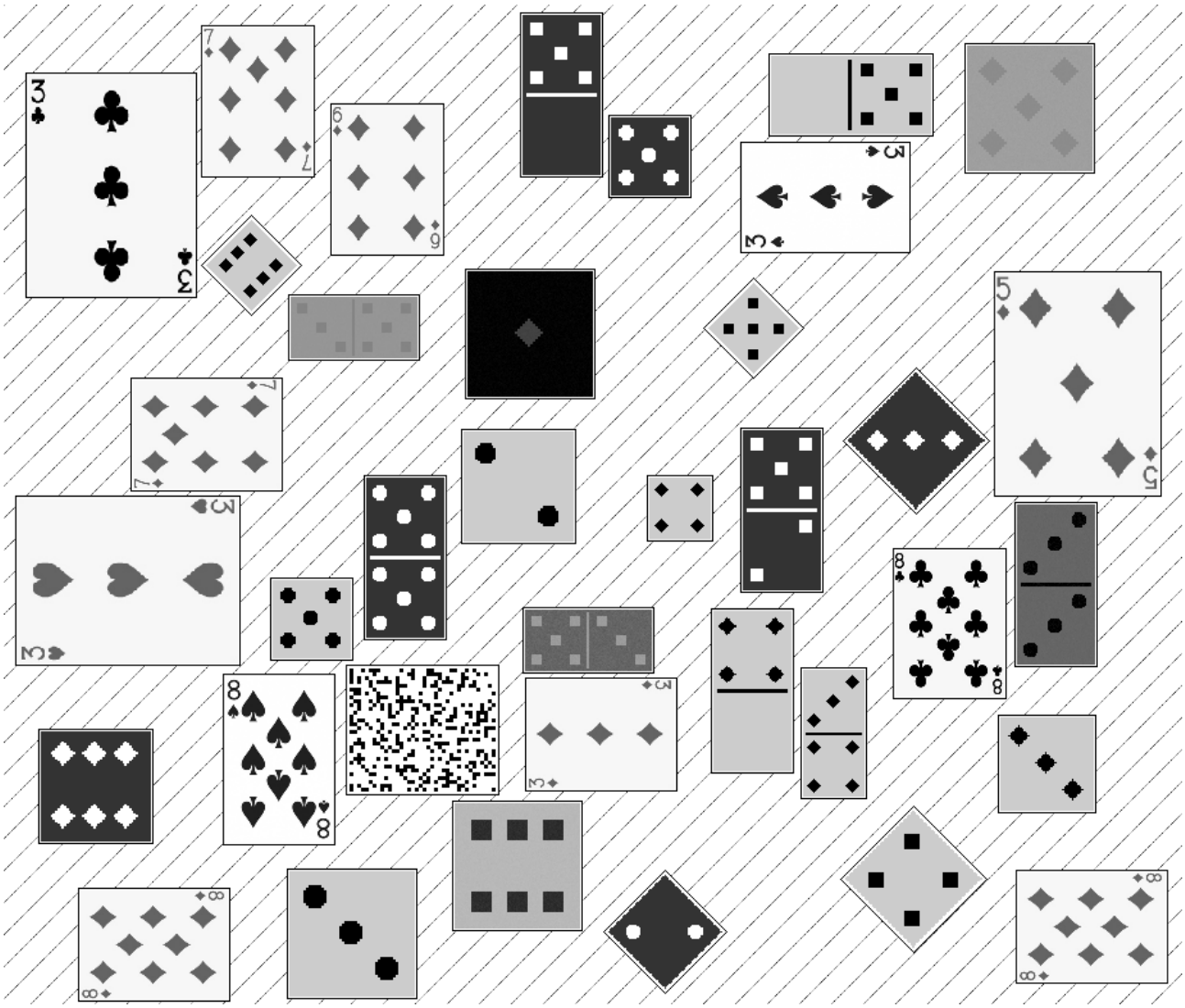


Figura 1: Exemplo de uma imagem com dominós, faces de dados, cartas de jogar e outros sem significado.

As características gerais das imagens são as seguintes:

- As imagens serão em níveis de cinzento.
- As imagens conterão 4 tipos de entidades: peças de dominó, faces de dados, cartas de jogo, e outras figuras retangulares ou quadradas sem significado.
- Os objetos poderão existir em diversas orientações como descrito adiante.
- Os objetos existirão em escalas variáveis.
- Os objetos poderão aparecer com fundo escuro ou fundo claro.
- O número total de objetos, e a informação que contêm, será variável em cada imagem.
- Os objetos estarão envolvidos por um caixilho preto.
- Haverá sempre espaço branco, de pelo menos um pixel na horizontal ou na vertical, junto ao caixilho da parte de dentro.
- O fundo da imagem, fora dos caixilhos, poderá conter ruído/textura de intensidade variável.

Cada região é caracterizada por duas partes: fundo e símbolos, que serão distinguíveis pela sua intensidade de nível de cinzento, embora possa haver ruído (variações de intensidade), e não esteja definido à partida qual é o mais escuro. É feita exceção nas cartas de jogar onde o fundo será sempre mais claro do que os símbolos.

Parâmetros a detetar em cada imagem

Em cada imagem, os principais parâmetros a detetar pelo programa do aluno são os seguintes:

- Número de peças de dominó.
- Número de faces de dados de jogo.
- Número de cartas de jogo.
- Número de peças de dominó numa dada orientação.
- Número de faces de dados numa dada orientação.
- Número de peças de dominó com igual número de pintas em cada lado (duplas).
- Números totais acumulados de pintas das peças de dominó e dos dados.
- Número de cartas de jogo de certos naipes.
- *String* com os dígitos do número de pintas de todas as cartas ordenados de forma crescente.

Adiante especificam-se exatamente quais são as grandezas a obter da análise da imagem.

Peças de dominó

As peças de dominós terão duas partes com uma linha a servir de separador central. Cada parte poderá ter entre 0 e 6 marcas (pintas) que poderão ter formas de quadrado, círculo ou de losango. As peças poderão surgir em posição vertical ou horizontal. Ilustram-se algumas figuras de dominós:

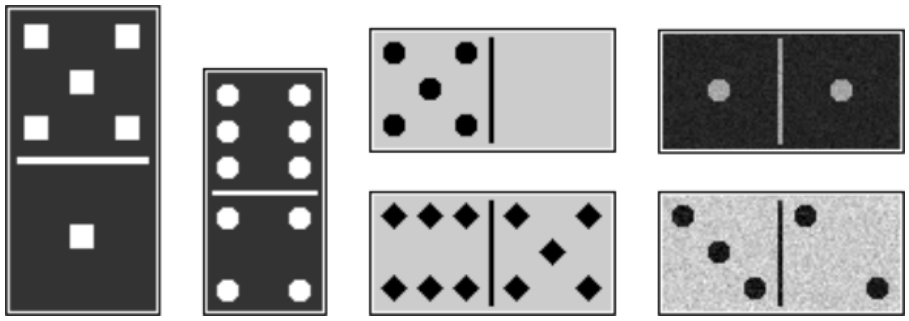


Figura 2: Exemplos de peças de dominó.

Faces de dados de jogo

As faces de dados terão entre 1 e 6 marcas (pintas), que poderão ter formas de quadrado, círculo ou de losango. As faces do dado serão quadradas e poderão estar em duas orientações principais: orientação normal (0°) e orientação a 45°. Ilustram-se algumas representações de faces de dados:

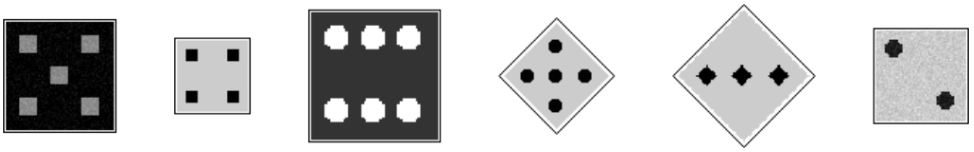


Figura 3: Exemplos de faces de dados.

Cartas de jogo

As cartas de jogo poderão ser dos 4 naipes possíveis (Copas, Espadas, Ouros, Paus) e o número facial variará de 1 (ases - com letra A), até 9. As cartas serão retangulares e também poderão surgir na vertical ou horizontal.

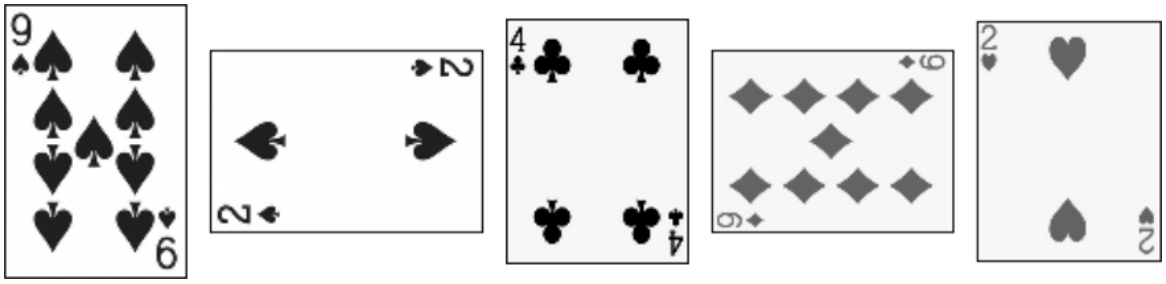


Figura 4: Exemplos de cartas de jogo.

Objetos sem significado

A última categoria de entidades são os objetos sem significado. Podem ser quadrados ou retangulares e em escalas variadas, mas nunca representarão nenhuma versão válida dos tipos anteriores (dominó, face de dado, carta de jogar). Como não têm significado, não será preciso detetar nenhuma das suas características. Seguem alguns exemplos:

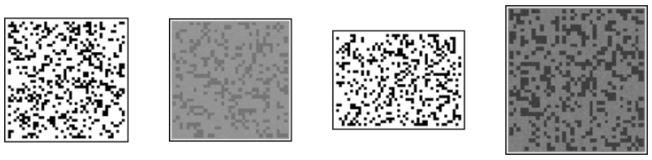


Figura 5: Exemplos de objetos sem significado.

Gerais

As posições relativas das marcações nos dominós, dados e cartas (pintas, símbolos do naipe) preservam-se nos objetos onde são definidas, embora a sua geometria possa ser ligeiramente deformada pela escala e/ou rotação. As letras ou números das cartas de jogar poderão sofrer grandes deformações na geometria e intensidade de alguns *pixels* conforme a escala e a rotação. A dimensão do símbolo central nos ases poderá ser proporcionalmente maior do que os símbolos das restantes cartas do naipe.

Algumas sugestões de técnicas para realizar o trabalho

Indicam-se de seguida algumas sugestões de técnicas possíveis para abordar o problema, mas são meramente indicações que não têm de ser seguidas, e nem sequer são obrigatórias para a solução.

- Detetar os cantos delimitadores dos caixilhos dos objetos com filtros de convolução ou outras técnicas.
- Isolar as regiões das peças a partir das caixas individuais de cada objeto;
  - pode-se fazer uma análise por linha e por coluna a partir dos cantos detetados para obter os limites dos caixilhos dos objetos na horizontal ou vertical.
  - para os objetos rodados 45° podem-se procurar os padrões das linhas do caixilho que estarão na diagonal.
- Detetar se é peça de dominó, face de dado, carta de jogo ou objeto sem significado.
- Detetar a orientação do objeto e eventualmente reorientá-lo para o processar.
- Fazer os testes de verificação da validade do objeto e outras eventuais propriedades.

Formato do ficheiro de resultados

Quando executado, o programa a desenvolver pelo aluno deve ler um conjunto de imagens fornecidas com nomes do género `svpi2022_TP1_img_MMM_NN.png` onde `NN` poderá variar de 01 até 99; esse número 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 os parâmetros pedidos. Esta tabela será sujeita a avaliação como descrito mais adiante, e deve ser escrita pelo programa do aluno num ficheiro com nome `tp1_nnnnnn.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 `tp1_999999.txt` que deverá conter os seguintes campos separados por vírgulas:

`NumMec`	Número mecanográfico do aluno.
`NumSeq`	Número da sequência da imagem (Cf. nome do ficheiro de imagem).
`NumImg`	Número da imagem na sequência (Cf. nome do ficheiro de imagem).
`tDom`	Número total de peças de dominó.
`tDice`	Número total de faces de dados.
`tCard`	Número total de cartas de jogo.
`RD0`	Número de dominós na orientação horizontal: 0°.
`RF0`	Número de faces de dados na orientação normal: 0° (por distinção com os rodados 45°).
`tDuplas`	Número de peças duplas de dominós.
`PntDom`	Número total de pintas acumuladas de todos os dominós.
`PntDad`	Número total de pintas acumuladas de todos os dados.
`CopOuros`	Número total de cartas dos naipes de copas e ouros: ♥ + ♦
`EspPaus`	Número total de cartas dos naipes de espadas e paus: ♠ + ♣
`Ouros`	Número total de cartas apenas do naipe de ouros: ♦
`StringPT`	*String* com os dígitos do número de pintas de todas as cartas ordenados de forma crescente.

Todos os campos são numéricos à exceção do campo `StringPT` que terá de ser representado como *string* em virtude da sua natureza.

Um parâmetro não verificado na imagem deve constar na tabela com valor 0. Por exemplo, se não houver na imagem nenhum objeto do tipo carta de jogar do naipe ouros, o parâmetro `Ouros` para essa imagem terá o valor 0.

Exemplo de resultados

Como exemplo concreto, tomem-se as duas imagens da figura 6 que se pretendem analisar e cujos ficheiros têm nomes: `svpi2022_TP1_img_160_11.png` e `svpi2022_TP1_img_160_12.png`.

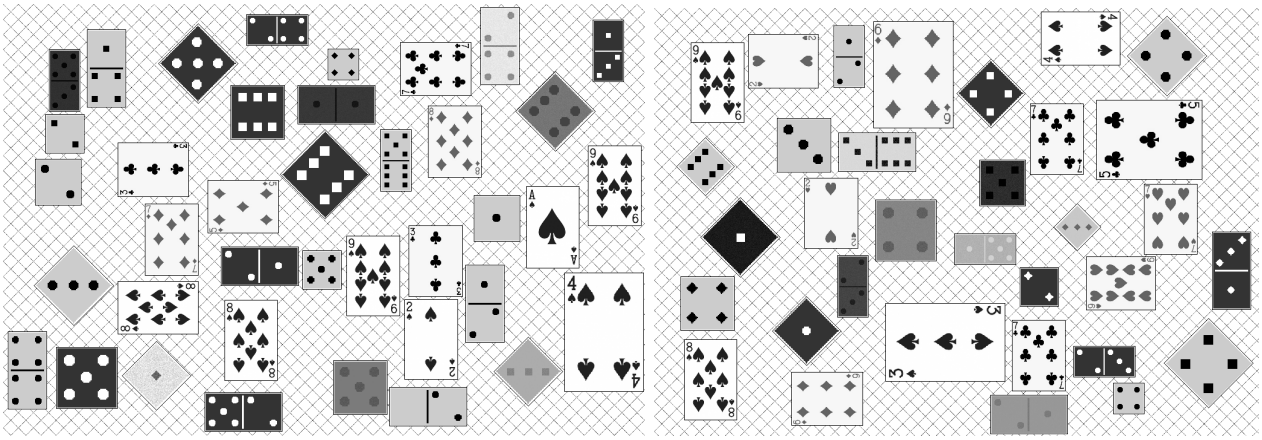


Figura 6: Exemplos de duas imagens a processar: sequência 160, imagens 11 e 12.

Se o número mecanográfico do aluno fosse 999999, o ficheiro de resultados `tp1_999999.txt` para esta sequência de duas imagens teria o seguinte conteúdo:

```
999999,160,11,12,14,13,5,8,2,65,54,3,10,3,1233457788899
999999,160,12,7,13,13,4,6,0,35,45,6,7,2,2234566777899
```

Para mais fácil referência e verificação, seguem os valores numa tabela com indicação do nome dos campos.

														
NumMec	NumSeq	NumImg	tDom	tDice	tCard	RD0	RF0	tDuplas	PntDom	PntDad	CopOuros	EspPaus	Ouros	StringPT
999999	160	11	12	14	13	5	8	2	65	54	3	10	3	1233457788899
999999	160	12	7	13	13	4	6	0	35	45	6	7	2	2234566777899

O ficheiro de resultados pode ser escrito com recurso a funções como `csvwrite()`, `dlmwrite()`, `writetable()`, entre outras.

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.

Procedimento para entrega do trabalho

O aluno deve executar o comando `SVPI_ProcessTP1_2022` (fornecido on-line) usando como argumento o nome do seu ficheiro Matlab. Por exemplo, se o programa do aluno se designar `tp1_999999.m` então, dentro da pasta que o contém, deve executar na linha de comando do Matlab o seguinte:

```
SVPI_ProcessTP1_2022('tp1_999999.m')
```

Se o programa do aluno executar sem erros, será criado o ficheiro `cod_svpi2022_tp1_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 `tp1_nnnnnn.txt` não for gerado pelo programa do aluno, então o comando `SVPI_ProcessTP1_2022`

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_svpi2022_tp1_nnnnn.vsz`:

- O programa do aluno não pode ter erros de execução.
- A pasta onde está a ser executado não pode ter nenhum ficheiro `svpi2022_TP1_img_*_*.png`
- Os ficheiros de imagem `svpi2022_TP1_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 `tp1_nnnnnnn.txt` onde `nnnnnn` é o número mecanográfico.

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 (prevê-se o uso do MOSS). Todos os trabalhos que apresentarem um índice de plágio maior do que um dado limite são passíveis de análise e tratamento específicos, incorrendo em risco de penalizações ou anulação do trabalho!

Neste trabalho espera-se que o aluno recorra às técnicas e ferramentas estudadas até à data, em especial: filtros lineares e de mediana, análise de histogramas, binarização e deteção de arestas, bem como os conceitos práticos de indexação e combinação e manipulação de imagens e matrizes em Matlab, e o recurso a máscaras. Algumas técnicas de morfologia binária poderão ser úteis em alguns problemas, mas não é forçoso usa-las.

Observações e recomendações sobre o trabalho e o Matlab

- Em caso de incertezas de versões de Matlab, o trabalho é avaliado na versão do Matlab disponível nos computadores das salas de aula (versão oficial da UA).
- O programa a executar deve-se chamar `tp1_nnnnnn.m` (onde `nnnnnn` é o numero mecanográfico)
- 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.

O programa deve gerar, na pasta corrente, o ficheiro `TP1_nnnnnnn.txt`, (onde `nnnnnnn` 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/tp1/`" então as imagens estarão em "`as_minhas_pastas_locais/svpi/`".

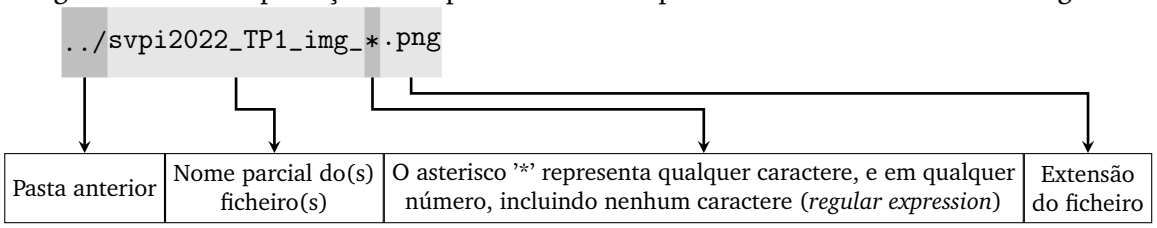
Ou seja, para tornar o programa independente do nome da pasta onde se encontre, ele deve procurar as imagens na pasta anterior. Assim recomenda-se que essa pasta seja adicionada ao `path` do Matlab, e isso pode-se fazer recorrendo ao seguinte comando a colocar no início do programa:

```
addpath('../');
```

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:

```
listaF=dir('../svpi2022_TP1_img_*.png');
```

O significado e interpretação da expressão anterior podem ser entendidos com o seguinte diagrama:



Neste caso, a variável `listaF` conterá uma listagem de todos os ficheiros de imagens. A variável será uma estrutura em Matlab (*structure*) que conterá toda a informação sobre os ficheiros. Por exemplo:

```
size(listaF,1) -> dá o número de ficheiros encontrados pelo comando dir.
listaF(1).name -> dá o nome do primeiro ficheiro da lista.
```

Uma vez que a pasta anterior (onde estarão as imagens) já foi colocado no `path`, um comando como `A=imread(listaF(1).name)` não deverá ter problemas em encontrar o ficheiro.

Para a geração de resultados de avaliação, além da contabilização dos parâmetros dos objetos presentes na imagem, como descrito no enunciado, e como o enunciado também refere, é necessário incluir o número da sequência e o número da imagem na sequência, que se extraem do próprio nome do ficheiro da imagem. Basta pensar 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:

Ferramentas auxiliares

Para os estudantes que o pretendam, é possível fazer uso de um comando disponibilizado na página Moodle da unidade curricular, e que se designa: `vs_getsubimages`.

Este comando implementa uma função que aceita uma matriz que contem uma das imagens a analisar, e devolve um '`cell array`' com todas as imagens dos objetos já extraídos, e separados da imagem original. Deste modo, os estudantes que optarem por esta solução não precisam de escrever o código para fazer essa deteção e separação. Porém, e como isso representa uma parte importante do trabalho, os programas entregues que usarem esta função/comando terão um **desconto** na avaliação de até **3 valores** (na escala de 0 a 20). O excerto de código abaixo ilustra um uso desta função:

```
close all
fName='sequencias/seq200/svpi2022_TP1_img_200_01.png';
Z=im2double(imread( fName ));
regions=vs_getsubimages(Z); %extract all regions
N=numel(regions);
SS=ceil(sqrt(N));
for k=1:N
    subplot( SS, SS, k)
    imshow( regions{k} )
end
```

Para esta facilidade poder ser usada, basta que o ficheiro `vs_getsubimages.p` esteja na pasta corrente do trabalho. No entanto, ele pode estar presente na pasta corrente e não ser usado!