

Departamento de Eletrónica, Telecomunicações e Informática

Linguagens Formais e Autómatos

Ano Letivo 2018/19

2º Ano, 2º semestre

Trabalho Prático: ImageManipulator

Turma P1

Alexey Kononov, nº mec: 89227

(alexeykononov@ua.pt)

Rodrigo Santos, nº mec: 89180

(rodrigo.l.silva.santos@ua.pt)

Daniel Lopes, nº mec: 87881

(daniel.vnl@ua.pt)

Gonçalo Freitas, nº mec: 85101

(g.freitas@ua.pt)

Nuno Félix, nº mec: 80330

(nunomfelix@ua.pt)

● Descrição da Linguagem:

A linguagem ImageManipulator é uma linguagem para manipulação, através da escrita de código, de imagens dos tipos: .png e .jpeg/.jpg. Para tal são utilizados alguns métodos e classes existentes na biblioteca *OpenCV* para java, logo para utilização correta da linguagem é necessário ter a biblioteca de *OpenCV* instalada no computador. Através da linguagem ImageManipulator o utilizador consegue entre outros, rodar imagens num ângulo que pretende, aplicar filtros de cores e guardar os seus resultados em novas imagens.

● Instruções:

1. LoadImage

Modo de utilização : **LoadImage ID = "PATH" ;**
ou LoadImage = ID;

Instrução para guardar a imagem localizada em PATH *ou* simplesmente guardar numa nova variável a imagem que esteja noutra variável ID. Caso a variável ID já exista é lançado um erro (*variable ID already exists*), caso PATH não exista ou não pertença aos tipos de imagens suportados, é mostrado o erro respetivo (*Failed to Load image/Invalid format*).

2. Rotate

Modo de utilização : **ID.rotate -> INT;**

Método para rodar a imagem guardada na variável ID INT graus. INT tem de ser um valor inteiro positivo. Caso ID não exista é lançado um erro (*variable ID not declared*).

3. Save

Modo de utilização : **Save ID -> "PATH";**

Instrução que permite guardar a imagem em PATH. Se ID da imagem não existir lança erro (*variable ID not declared*).

4. Filter

Modo de utilização : **ID.filter -> FILTER;**

Instrução que permite ao utilizador aplicar um filtro de cor FILTER á imagem guardada na variável ID; caso a variável não exista é apresentada uma mensagem de erro (*variable ID not declared*); caso o filtro desejado não exista é apresentada uma mensagem de erro (*Invalid FILTER*).

Os filters disponiveis são: red, blue, gray e green.

5. Add

Modo de utilização : **Add ID = ID1 + ID2;**

Instrução que permite ao utilizador adicionar duas imagens lado a lado numa só imagem. Existem várias maneiras de se utilizar esta instrução, podemos por exemplo:

- **Add imagem = imagem1 + imagem2;**

Sendo imagem1 e imagem2 previamente guardadas nas variáveis.

- **Add imagem = "PATH" + imagem1;**

Sendo PATH o caminho da imagem, e imagem1 uma imagem guardada na variável

- **Add imagem = "PATH" + "PATH";**

Pode diretamente guardada na variável imagem as duas imagens carregadas pelo "PATH".

Caso as variáveis/caminhos fornecidos não existam serão impressos os erros (*variable ID not declared* ou *failed to load image*).

6. Show

Modo de utilização : **Show ID;**

Instrução que permite visualizar a imagem que esteja guardada na variável ID caso a variável não exista é apresentada uma mensagem de erro (*variable ID not declared*).

7. FaceDetect

Modo de utilização : **FaceDetect ID;**

Instrução que deteta as caras de pessoas presentes numa imagem ID, e cria um retângulo verde em torno das mesmas; caso a variável desejada não exista é lançado um erro (*variable ID not declared*).

8. Zoom

Modo de utilização : **ID.zoom -> INT;**

Instrução que amplia a imagem ID, com valor INT de [0,100]%; caso valor de INT não esteja dentro dos limites será reportado um ao utilizador (*Invalid Zoom amount*).

9. Crop

Modo de utilização : **Crop ID -> (INT, INT) (INT, INT);**

Instrução para cortar uma imagem ID, a partir de um ponto inicial (INT, INT) até o ponto o desejado (INT, INT); caso os pontos não se encontrem dentro das dimensões da imagem será apresentado um erro a referindo isso (*out of bounds point (INT, INT)*) e assinalando com ‘ * ’ o valor do ponto no qual ocorreu a violação da dimensão.

10. Details

Modo de utilização : **ID.details;**

Instrução que imprime no terminal as dimensões da imagem ID; caso a variável ID não esteja declarada é impresso o erro (*variable ID not declared*).

11. Brightness

Modo de utilização : **ID.brightness -> INT;**

Instrução que permite alterar o brilho de uma imagem guardada em ID, com um fator de INT pertencente ao intervalo [0,100];

caso a variável desejada não exista é lançado um erro (*variable ID not declared*) caso valor de INT não pertença ao intervalo permitido ou não seja um valor inteiro positivo será mostrado o erro respetivo (*invalid number*).

12. gamma

Modo de utilização : **ID.gamma -> INT;**

Instrução que permite alterar a luminosidade de uma imagem guardada em ID, com um fator de INT; caso a variável desejada não exista é lançado um erro (*variable ID not declared*) caso valor de INT não seja um valor inteiro positivo será mostrado o erro respetivo (*invalid number*).

13. Smoothing

Modo de utilização : **ID.smoothing -> INT;**

Instrução para melhorar a qualidade de uma imagem ID, suavizando-a com um fator de INT; caso a variável desejada não exista é lançado um erro (*variable ID not declared*) caso valor de INT não seja um valor inteiro positivo será mostrado o erro respetivo (*invalid number*).

14. histogram

Modo de utilização : **ID.histogram;**

Instrução que mostra numa imagem o histograma de cores de uma imagem armazenada em ID; caso esta não exista é apresentado o erro (*variable ID not declared*).

15. Match

Modo de utilização : **ID.match ID -> ID;**

ou **ID.match ID;**

Instrução que indica colocando um retângulo preto na imagem base se uma outra imagem se encontra algures na imagem base; caso estas imagens e/ou variáveis não se encontrem definidas ou não existam é lançado o respetivo erro.

Exemplos de Programas

```
// comentário1
/# comentário2 #/

/# Exemplo 1 -----#/
LoadImage i1 = "southpark.png";    // inicializar i1
i1.filter -> gray;                  // aplicar o filtro gray a i1
i1.rotate -> 90;                    // rodar i1 90º
i1.brightness -> 0;                 // aumentar o brilho de i1
i1.gamma -> 1;                     // aumentar a gamma de i1
Show i1;                           // Mostrar i1;
/#-----#/

/# Exemplo 2 -----#/
LoadImage i1 = "southpark.png";    // inicializar i1
LoadImage i2 = i1;                  // inicializar i2 com a imagem
                                   // de i1
i2.filter -> red;                    // aplicar o filtro red a i2
i2.histogram;                       // apresentar o histograma de
                                   // cores de i2
i1.details;                         // imprimir as dimensões de i1
i2.details;                         // imprimir as dimensões de i2
/#-----#/

/# Exemplo 3 -----#/
LoadImage i3 = "pessoas.jpg";
FaceDetect i3;                      // deteta caras
i3.zoom->50;                         // fazer 50% de zoom em i3;
i3.smoothing->10;                   // Suaviza imagem
Show i3;                           // mostrar imagem
/#-----#/
```

```

/# Exemplo 4 -----#/
LoadImage i1 = "image.png";
LoadImage i4 = "southpark.png";
Add i5 = i1 + i4;           // Juntar i1 com i4 e
                           // guardar em i5

i5.filter -> gray;
Show i5;
Save i5 -> "output.png";    // Guardar imagem
/#-----#/

```

```

/# Exemplo 5 -----#/
LoadImage i3 = "pessoas.png";
LoadImage i2 = "pessoas_cut.jpg";
i3.match.i2 -> i3;          //guardar em i3 o resultado
                           //de match

Show i3;                   //mostrar imagem
/#-----#/

```

```

/# Exemplo 6 -----#/
LoadImage i3 = "pessoas.jpg";
i3.crop = (70,100) (100,100); //guardar em i3 o resultado
                           //de match

Show i3;                   //mostrar imagem
/#-----#/

```


Exemplos de Erros:

```
LoadImage img = "image.png";
```

```
LoadImage img1 = "southpark.png";
```

```
LoadImage img4 = img10;
```

```
Show img7;
```

```
Add img3 = i + img;
```

```
Show img;
```

```
img3.filter -> azul;
```

```
img3.smoothing -> 1;
```

```
im.rotate -> 90;
```

```
Show img3;
```

```
Save img->"out.png";
```

- **Contribuição dos autores**

No desenvolvimento do trabalho todos os autores contribuíram igualmente para a realização do projeto em todas as suas componentes. Cada elemento realizou 20% do trabalho total.

Auto Avaliações: Rodrigo Santos -> 20%
Daniel Lopes -> 20%
Gonçalo Freitas -> 20%
Alexey Kononov -> 20%
Nuno Félix -> 20%