

## SISTEMAS SENSORIAIS

**MIEEC & MIEBM & MIEI & MIEF**

**2016-2017**

### **EvalForm - Avaliação automática de resultados**

#### **1 Descrição:**

Foi criada pelos docentes uma biblioteca (DLL) que terá de ser incluída no vosso projeto e que automatiza testes às várias funcionalidades implementadas nas aulas práticas, assim como, valida o reconhecimento de matrículas.

Para incluir esta biblioteca deverão seguir as regras indicadas neste documento, sendo que a principal preocupação deverá ser garantir que o cabeçalho das funções segue as regras apresentadas.

O resultado dos vossos métodos será comparado com a mesma operação implementada em OpenCV. A comparação será em termos de tempo de processamento, para ter uma base de comparação entre computadores com processadores diferentes e numa comparação pixel a pixel (pixéis do interior da imagem e dos limites exteriores) para avaliar se os resultados estão corretos.

Na comparação pixel a pixel é natural que tenham alguma diferença em algumas funções, nomeadamente de Rotação e Otsu (diferenças médias menores que 1 ou 2), para os restantes casos é possível reduzir as diferenças até zero usando algumas regras usadas pelo OpenCV nomeadamente as que se apresentam de seguida.

#### **Nota importante:**

Para garantir que os vossos resultados se assemelham o mais possível aos obtidos pelo OpenCV deverão arredondar os resultados antes de os guardar na imagem. Por exemplo no ajuste de brilho e contraste:

```
aux = dataPtr[0] * contrast + bright;  
if (aux < 0)  
    aux = 0;  
else if (aux > 255)  
    aux = 255;  
dataPtr[0] = (byte) Math.Round(aux); // arredondamento
```

Igualmente, sempre que precisarem de calcular o nível de cinzento deverão dividir por um valor real e arredondar o valor calculado:

```
gg = (byte) Math.Round(((int) blue + green + red) / 3.0);
```

## 2 Ficheiros

A EvalForm recorre a 3 ficheiros de texto contendo diferentes configurações:

- **SS\_description.txt** – contém os números de aluno e a descrição das principais funcionalidades que incluíram no vosso projeto. Nota: este ficheiro é mantido e gravado pela EvalForm;

- **SS\_files.csv** – contém a diretoria onde estão localizadas as imagens a testar e contém a lista das imagens que serão testadas pela EvalForm. Caso a imagem seja de uma matrícula terá informação relativa à localização da matrícula, localização dos caracteres e às letras nela contida. Neste ficheiro a localização da matrícula assim como dos caracteres diz respeito ao centroide de cada uma destas zonas.

- **SS\_operations.csv** – contém uma lista de todas as funções de processamento de imagem que serão testadas, contendo para cada uma breve descrição e o cabeçalho da função a implementar.

### 3 Preparação do projeto

- 1- Copiar para diretoria bin/debug os vários ficheiros fornecidos (conteudo do SS\_Eval Files.zip) (note que foi feita uma atualização da versão do EmguCV para a v2.9.0); Aceite a troca pelos novos ficheiros.
- 2- Editar o ficheiro SS\_files.csv e alterar o pathname para a diretoria onde estão as imagens de teste;
- 3- Adicionar a SS\_Eval.dll ao projeto como referência (encontra-se na diretoria bin/debug):

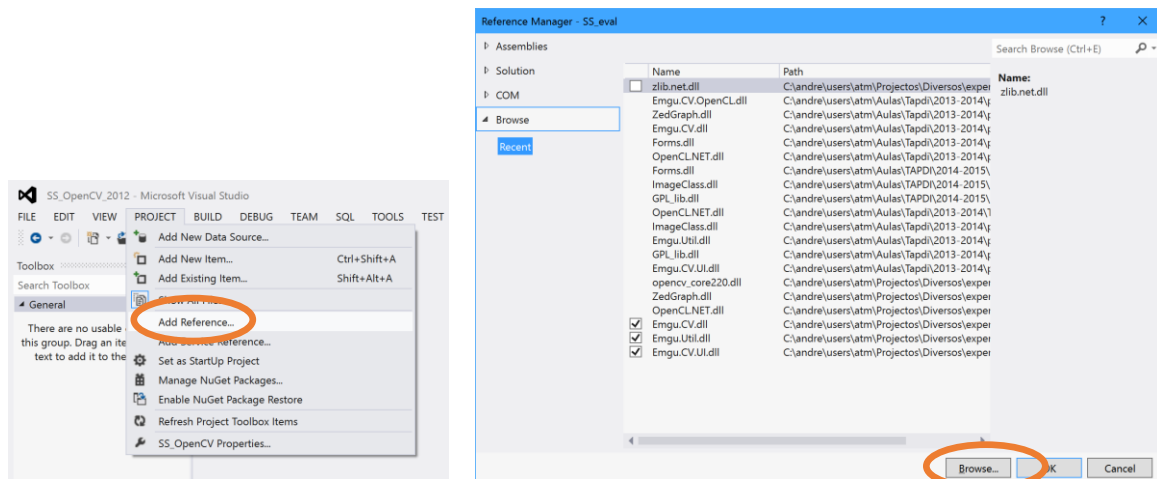


Figura 1 – Adicionar SS\_Eval.dll ao projeto.

- 4- Adicionar um menu na MainForm para chamar a EvalForm:

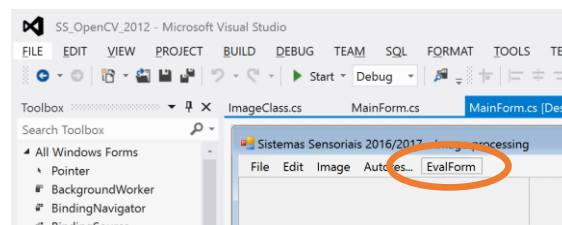


Figura 2 – Colocação do menu EvalForm na barra de menus.

- 5- No evento do menu EvalForm executar a janela:

```
new SS_OpenCV.EvalForm().ShowDialog();
```

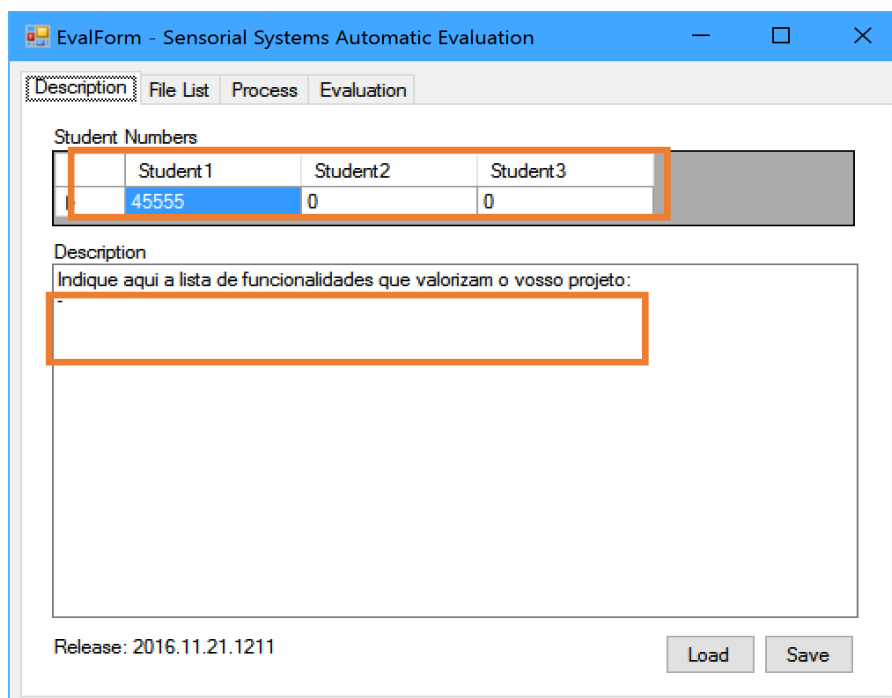
- 6- Alterar os cabeçalhos das funções do ImageClass de *internal* para *public* (sugestão fazer *find & replace*);
- 7- Fazer Rebuild do projeto (menu *Build->Rebuild*):

## 4 Funcionamento da Eval Form:

### 4.1 Painel 1 - Description

No primeiro painel (figura 1) devem colocar os vossos números de aluno e colocar na descrição uma lista de pontos contendo o que gostariam de chamar a atenção dos professores que forem avaliar o vosso projeto.

Aqui podem incluir os pontos extras que realizaram ou justificar alguma falha que o vosso programa possa ter.



The screenshot shows the 'EvalForm - Sensorial Systems Automatic Evaluation' window. The 'Description' tab is active. It features a table for 'Student Numbers' and a text area for 'Description'.

	Student1	Student2	Student3
	45555	0	0

The 'Description' text area contains the prompt: 'Indique aqui a lista de funcionalidades que valorizam o vosso projeto:'.

At the bottom of the window, there is a 'Release' label with the value '2016.11.21.1211' and two buttons: 'Load' and 'Save'.

Figura 3 – Primeiro painel de descrição.

## 4.2 Painei 2 – File List

No segundo painel (figura 4) é apresentada a lista de imagens que serão usadas para teste da aplicação e respetiva informação sobre as matrículas. Podem também ser incluídas imagens que não serão para fazer deteção de matrículas, apenas para testar as aulas práticas, neste caso a flag is\_LP está desmarcada.

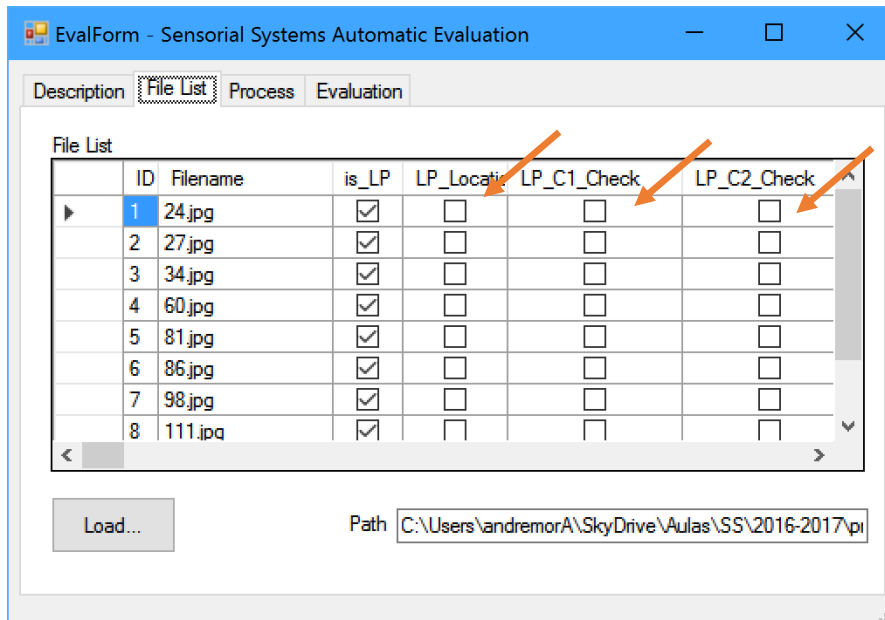
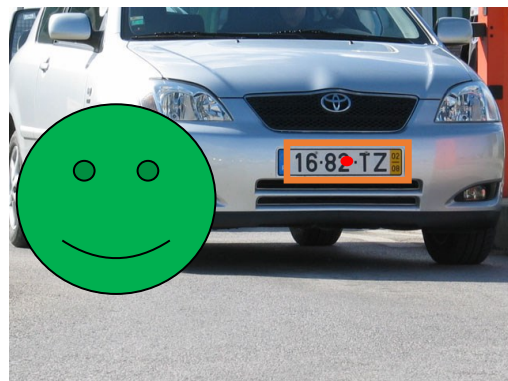


Figura 4 – Segundo painel contendo a lista de ficheiros.

Na Tabela File List as primeiras colunas (assinaladas na imagem) indicam se a matrícula e os carateres foram bem localizados. Já as restantes colunas tem informação sobre os carateres que estão na imagem e quais os resultados obtidos pela vossa aplicação.

A verificação da correta localização da matrícula e respetivos carateres faz por comparação dos retângulos contendo a matrícula e carateres detetados pela aplicação, com o ponto central definido manualmente no ficheiro SS\_Files.csv. Caso o ponto central esteja incluído dentro do retângulo será considerado como corretamente localizado (imagem direita).



### 4.3 Painel 3 – Process

Neste Painel (figura 5) são testadas as várias funções desenvolvidas nas aulas práticas. Os resultados obtidos pela ImageClass são comparados com a mesma operação implementada usando a biblioteca OpenCV.

A comparação será em tempo de execução e em diferença pixel a pixel entre as duas implementações. A diferença pixel a pixel é avaliada entre pixels do interior da imagem (coreDiff) e também entre os pixels dos bordos da imagem (borderDiff). Ambos estes indicadores são valores médios.

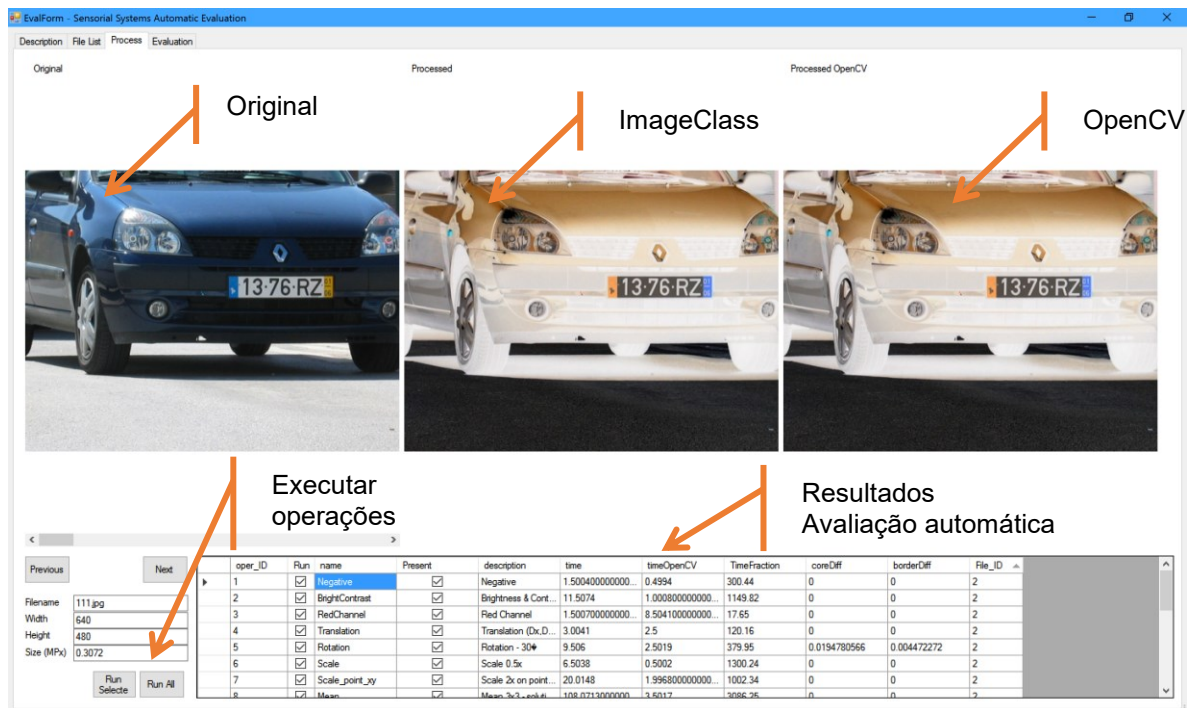


Figura 5 – Terceiro painel contendo os testes às imagens.

Descrição das colunas (todas ReadOnly exceto a coluna *Run*):

Oper_ID	Número interno para identificar a operação
Run	Serve para retirar ou adicionar algumas funções da lista de operações a executar
Name	Nome do procedimento
Present	Indica se o procedimento está implementado ou não no ImageClass. (output)
Description	Descrição da operação (com parametros)
Time	Tempo de execução do ImageClass
TimeOpenCV	Tempo de execução do OpenCV
TimeFraction	Valor da Relação: Time / TimeOpenCV
coreDiff	Diferença média entre os pixels do centro da imagem $x = [1; largura - 2]$ e $y = [1; altura - 2]$
borderDiff	Diferença média entre os pixels dos bordos da imagem $x = 1, x = largura - 1, y = 0$ e $y = altura - 1$
File_ID	Número interno para identificar o ficheiro

#### 4.4 Painel 4 – Evaluation

Este é um painel que será usado pelos docentes no momento da avaliação para compilar os resultados todos sob a forma de uma tabela e exportar para EXCEL.

Podem usar esta janela para executar as operações sobre todas as imagens premindo apenas um botão (*Run All Images*).

EvalForm - Sensorial Systems Automatic Evaluation

Description File List Process Evaluation

Process Interface Report Robustness

Comments

	Student 1	Student 2	Student 3	Filename
▶	45555	0	0	11.jpg
	45555	0	0	11.jpg
	45555	0	0	11.jpg
	45555	0	0	11.jpg
	45555	0	0	11.jpg
	45555	0	0	11.jpg
	45555	0	0	11.jpg

lpr\_bw.png

Run All Images Save Results

Figura 6 – Quarto painel usado para a avaliação final do projeto.

## 5 Operações – Funções ImageClass

1	<b>Negative</b>	Negativo de uma imagem
void Negative(Image<Bgr, byte> img) - recebe a imagem a alterar		
2	<b>BrightContrast</b>	Ajuste de brilho e contraste.
void BrightContrast(Image<Bgr, byte> img, int bright, double contrast) - recebe a imagem a alterar, o valor de brilho e de contraste		
3	<b>RedChannel</b>	Copia a componente Red para a Green e Blue.
void RedChannel(Image<Bgr, byte> img) - recebe a imagem a alterar		
4	<b>Translation</b>	Desloca a imagem de (Dx,Dy) =(-10 , -10), preenchendo as partes sem pixels a preto.
void Translation(Image<Bgr, byte> img, Image<Bgr, byte> imgCopy, int dx, int dy) - Recebe a imagem a alterar, uma cópia da imagem e o deslocamento em x e y.		
5	<b>Rotation</b>	Roda a imagem de 30º (angulo em radianos) , preenchendo as partes sem pixels a preto.
void Rotation(Image<Bgr, byte> img, Image<Bgr, byte> imgCopy, float angle) - Recebe a imagem a alterar, uma cópia da imagem e o angulo de rotação (radianos).		
6	<b>Scale</b>	Aplica um factor de escala de 0.5x centrado em (x,y) = (0,0), preenchendo as partes sem pixels a preto.
void Scale(Image<Bgr, byte> img, Image<Bgr, byte> imgCopy, float scaleFactor) - Recebe a imagem a alterar, uma cópia da imagem e o fator de escala.		
7	<b>Scale_point_xy</b>	Aplica um factor de escala de 2.2x centrado em (x,y) = (250,310), preenchendo as partes sem pixels a preto.
void Scale_point_xy(Image<Bgr, byte> img, Image<Bgr, byte> imgCopy, float scaleFactor,int centerX, int centerY) Recebe a imagem a alterar, uma cópia da imagem, o fator de escala e o centro (x, y).		
8	<b>Mean</b>	Filtro de média 3x3 – solução A (analisa 9 pixels)
void Mean(Image<Bgr, byte> img, Image<Bgr, byte> imgCopy) - Recebe a imagem a alterar e uma cópia da imagem		
9	<b>NonUniform</b>	Aplica um filtro não-uniforme à imagem, usando os coeficientes e pesos passados como parametro.
void NonUniform(Image<Bgr, byte> img, Image<Bgr, byte> imgCopy, float[,] matrix, float matrixWeight) - Recebe a imagem a alterar, uma cópia da imagem, uma matrix 3x3 contendo os coeficientes e o peso		
10	<b>Sobel</b>	Filtro de Sobel 3x3
void Sobel(Image<Bgr, byte> img, Image<Bgr, byte> imgCopy) - Recebe a imagem a alterar e uma cópia da imagem		



11	<b>Differentiation</b>	Filtro de diferenciação (2x2).
void Differentiation(Image<Bgr, byte> img, Image<Bgr, byte> imgCopy) - Recebe a imagem a alterar e uma cópia da imagem		
12	<b>Median</b>	Filtro de mediana a cores usando a avaliação 3D.
void Median(Image<Bgr, byte> img, Image<Bgr, byte> imgCopy) - Recebe a imagem a alterar e uma cópia da imagem		
13	<b>Histogram_Gray</b>	Calcula o histograma apenas da escala de cinzentos.
int[] Histogram_Gray(Emgu.CV.Image<Bgr, byte> img) - Recebe a imagem a analisar - Devolve o histograma		
14	<b>Histogram_RGB</b>	Calcula o histograma das componentes B, G e R
int[,] Histogram_RGB(Emgu.CV.Image<Bgr, byte> img) - Recebe a imagem a analisar - Devolve o histograma das 3 componentes B + G + R numa matrix [3, 256]		
15	<b>Histogram_All</b>	Calcula o histograma das componentes Gray, B, G e R
int[,] Histogram_All(Emgu.CV.Image<Bgr, byte> img) - Recebe a imagem a analisar - Devolve o histograma das 4 componentes GRAY + B + G + R numa matrix [4, 256]		
16	<b>ConvertToBW</b>	Binarização com valor de threshold recebido como parametro (valor = 157)
void ConvertToBW(Emgu.CV.Image<Bgr, byte> img, int threshold) - recebe a imagem a alterar e o valor de threshold		
17	<b>ConvertToBW_Otsu</b>	Binarização com valor de threshold calculado pelo método de Otsu
void ConvertToBW_Otsu(Emgu.CV.Image<Bgr, byte> img) - recebe a imagem a alterar		
18	<b>Mean_solutionB</b>	Filtro de média 3x3 – solução B (analisa 6 pixels)
void Mean_solutionB(Image<Bgr, byte> img, Image<Bgr, byte> imgCopy) - Recebe a imagem a alterar e uma cópia da imagem		
19	<b>Mean_solutionC</b>	Filtro de média 7x7 – solução C
void Mean_solutionC(Image<Bgr, byte> img, Image<Bgr, byte> imgCopy, int size) - Recebe a imagem a alterar, uma cópia da imagem e a dimensão do filtro		
20	<b>Roberts</b>	Filtro de Roberts
void Roberts(Image<Bgr, byte> img, Image<Bgr, byte> imgCopy) - Recebe a imagem a alterar e uma cópia da imagem		

40	<b>LP_Recognition</b>	Faz o reconhecimento total da matrícula , devolvendo a localização da matricula, carateres e o reconhecimento desses carateres.
<pre> public static void LP_Recognition(     Image&lt;Bgr, byte&gt; img, // imagem a alterar     Image&lt;Bgr, byte&gt; imgCopy, // cópia da imagem     out Rectangle LP_Location, // rectângulo(x,y,largura, altura) contendo a matricula     out Rectangle LP_Chr1, // rectângulo contendo o primeiro carater     out Rectangle LP_Chr2, // rectângulo contendo o segundo carater     out Rectangle LP_Chr3, // rectângulo contendo o terceiro carater     out Rectangle LP_Chr4, // rectângulo contendo o quarto carater     out Rectangle LP_Chr5, // rectângulo contendo o quinto carater     out Rectangle LP_Chr6, // rectângulo contendo o sexto carater     out string LP_C1, // valor do primeiro carater     out string LP_C2, // valor do segundo carater     out string LP_C3, // valor do terceiro carater     out string LP_C4, // valor do quarto carater     out string LP_C5, // valor do quinto carater     out string LP_C6, // valor do sexto carater     out string LP_Country // valor do País,     out string LP_Month // valor do mês da matricula,     out string LP_Year) // valor do ano da matrícula </pre>		
51	<b>Rotation_Bilinear</b>	Roda a imagem de 30º (angulo em radianos) , preenchendo as partes sem pixeis a preto. Usa Interpolação Bilinear.
<pre> void Rotation_Bilinear(Image&lt;Bgr, byte&gt; img, Image&lt;Bgr, byte&gt; imgCopy, float angle) </pre> <p>- Recebe a imagem a alterar, uma cópia da imagem e o angulo de rotação (radianos).</p>		
61	<b>Scale_Bilinear</b>	Aplica um factor de escala de 0.5x centrado em (x,y) = (0,0), preenchendo as partes sem pixeis a preto. Usa Interpolação Bilinear.
<pre> void Scale_Bilinear(Image&lt;Bgr, byte&gt; img, Image&lt;Bgr, byte&gt; imgCopy, float scaleFactor) </pre> <p>- Recebe a imagem a alterar, uma cópia da imagem e o fator de escala.</p>		
71	<b>Scale_point_xy_Bilinear</b>	Aplica um factor de escala de 2.2x centrado em (x,y) = (250,310), preenchendo as partes sem pixeis a preto. Usa Interpolação Bilinear.
<pre> void Scale_point_xy_Bilinear (Image&lt;Bgr, byte&gt; img, Image&lt;Bgr, byte&gt; imgCopy,     float scaleFactor,int centerX, int centerY) </pre> <p>Recebe a imagem a alterar, uma cópia da imagem, o fator de escala e o centro (x, y).</p>		