

Departamento de Sistemas e Computação – FURB
Curso de Sistemas de informação
Trabalho de Conclusão de Curso – 2016/2

DERMANOSTIC: um método para calcular o diâmetro e evolução de lesões cutâneas

Acadêmico: Felipe Darugna Fey
felipe_fey@hotmail.com

Orientador: Prof. Aurélio Faustino Hoppe
aurelio.hoppe@gmail.com

Grupo de Pesquisa em Computação
Gráfica, Processamento de Imagens e
Entretenimento Digital
<http://www.inf.furb.br/gcg>



Roteiro

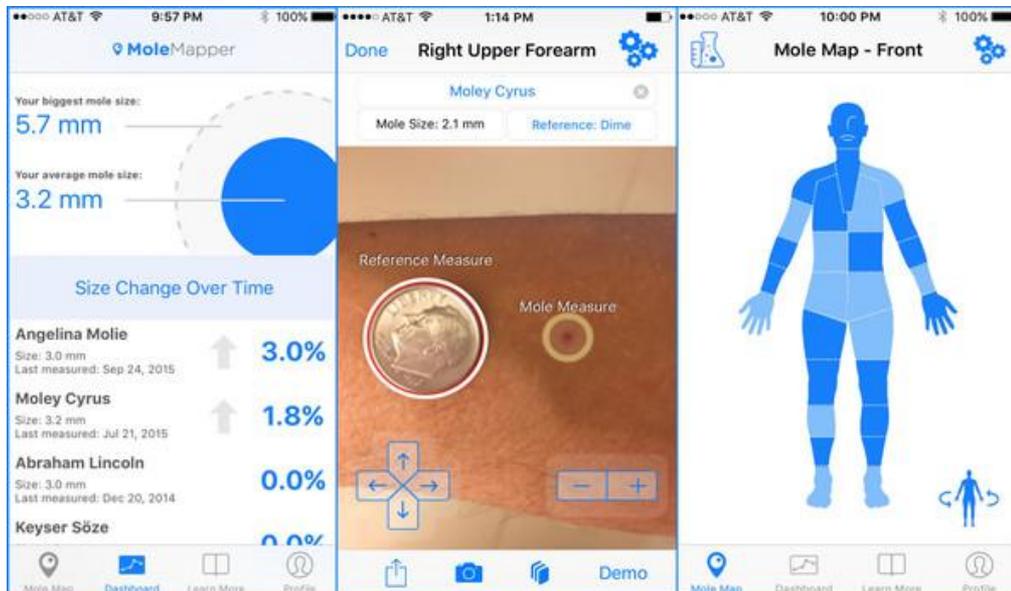
- Motivação
- Trabalhos correlatos
- Trabalho proposto
- Requisitos
- Desenvolvimento
- Testes
- Operacionalidade
- Conclusões
- Limitações
- Extensões
- Demonstração

Motivação

- Câncer configura-se como um problema de saúde pública mundial.
- Sua detecção precoce de extrema importância para definição do tratamento mais adequado.
- O Processamento Digital de Imagens é uma forma confiável de análise.

Trabalhos correlatos

Título: Automated analysis of melanocytic lesion
(PERUCH, 2015)

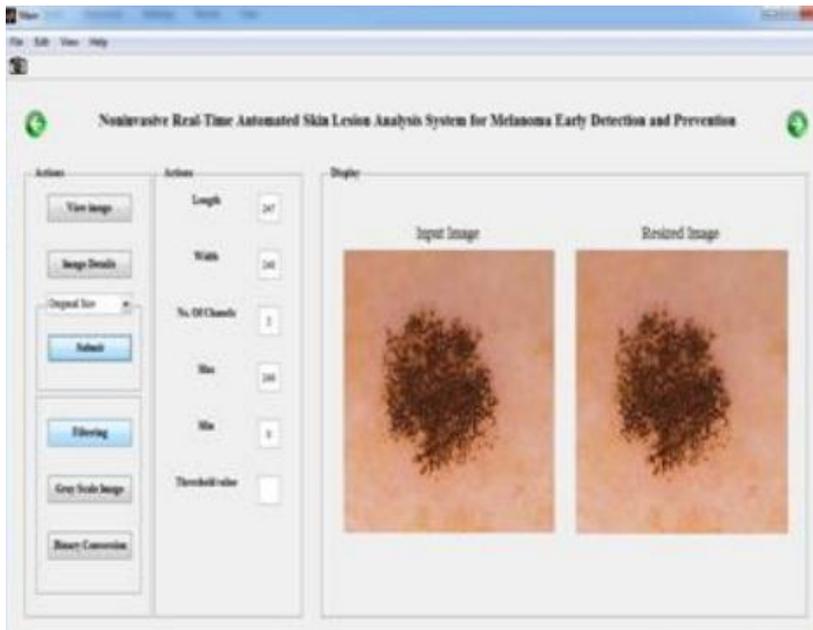


Objetivo:

criar uma solução chamada de “Mole Mapper” para ajudar dermatologistas durante o processo de diagnóstico de câncer de pele.

Trabalhos correlatos

Título: Melanoma Early Detection using Dual Classifier
(CHAITANYA e PAVANI, 2015)

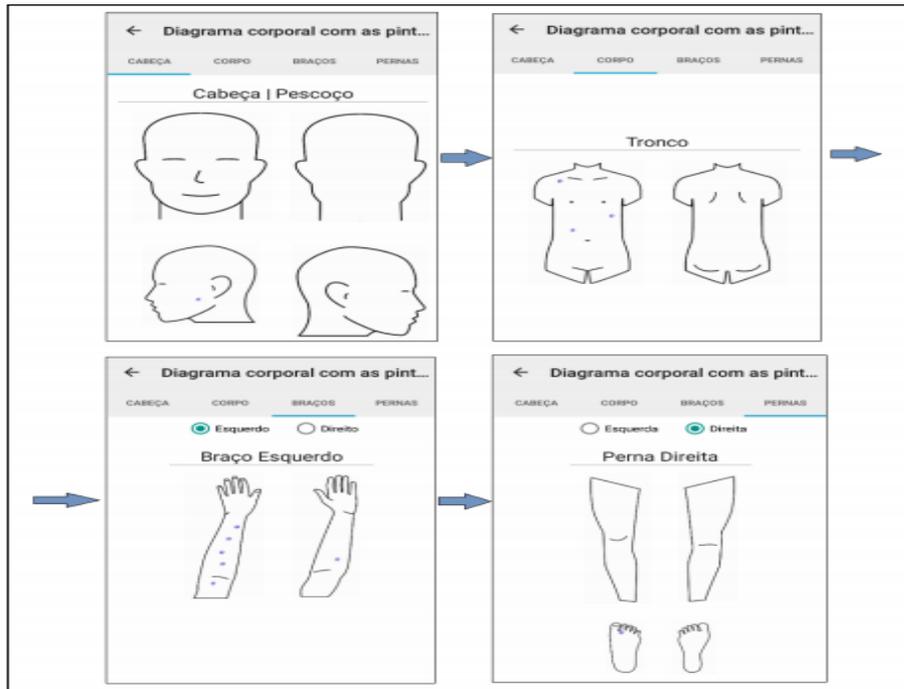


Objetivo:

O protótipo desenvolvido classifica a lesão como benigno, atípico ou melanoma.

Trabalhos correlatos

Título: Corpus Mapping: Um aplicativo para o mapeamento corporal de lesões cutâneas (LIMA, 2015)



Objetivo:

Desenvolver um aplicativo para a plataforma Android que realizasse o mapeamento corporal de lesões cutâneas

Tabela comparativa entre os trabalhos correlatos

Características	Trabalhos	Peruch (2015)	Chaitanya e Pavani (2015)	Lima (2015)
Plataforma		IOS	Desktop	Android
Cálculo de tamanho da lesão		Sim	Não	Não
Segmentação de lesão		Sim	Sim	Não
Mapeamento das lesões		Sim	Não	Sim
Classificação da lesão em lesional ou não lesional		Sim	Sim	Não

Trabalho proposto

Criar um método para calcular o tamanho e acompanhar a evolução de lesões cutâneas utilizando técnicas de processamento de imagens

Objetivos específicos:

- identificar a lesão cutânea e o gabarito a partir de imagens capturadas por um dispositivo móvel;
- estabelecer a proporcionalidade entre o gabarito e a lesão cutânea;
- apontar se houve crescimento da lesão ao longo do tempo.

Requisitos

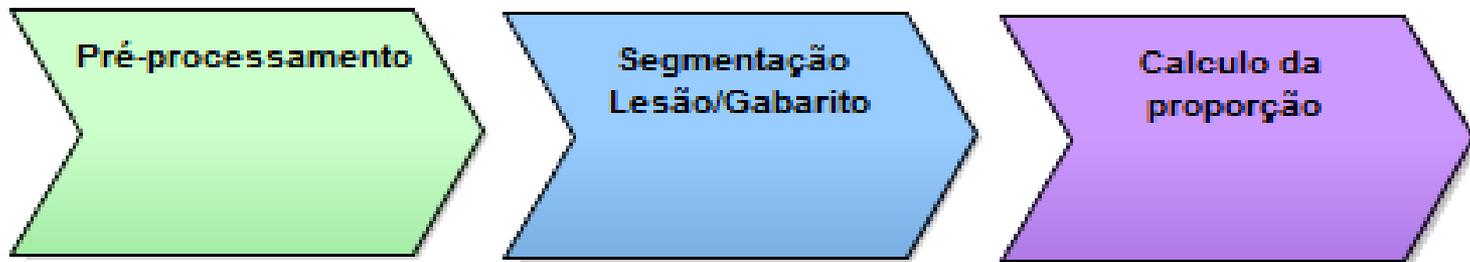
Requisitos funcionais:

- Permitir a captura de imagens utilizando a câmera do dispositivo Android
- Permitir o cadastro de lesões a partir de partes do corpo do usuário
- Realizar a segmentação do gabarito
- Realizar a segmentação da lesão
- Proporcionalizar o tamanho da lesão a partir do gabarito
- Criar um gráfico evolutivo da lesão em questão

Ferramentas utilizadas

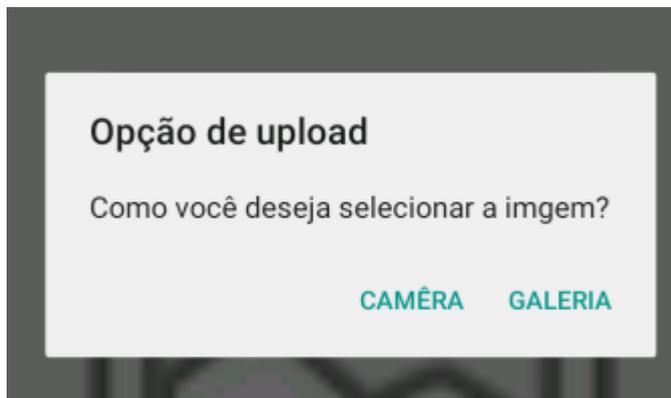
- linguagem C# (C-Sharp) na IDE Microsoft Visual Studio 2015
- EmguCV: *Wrapper* da biblioteca OpenCV para a linguagem C#.
- Xamarin
- SQLite

Fluxo geral da análise da lesão

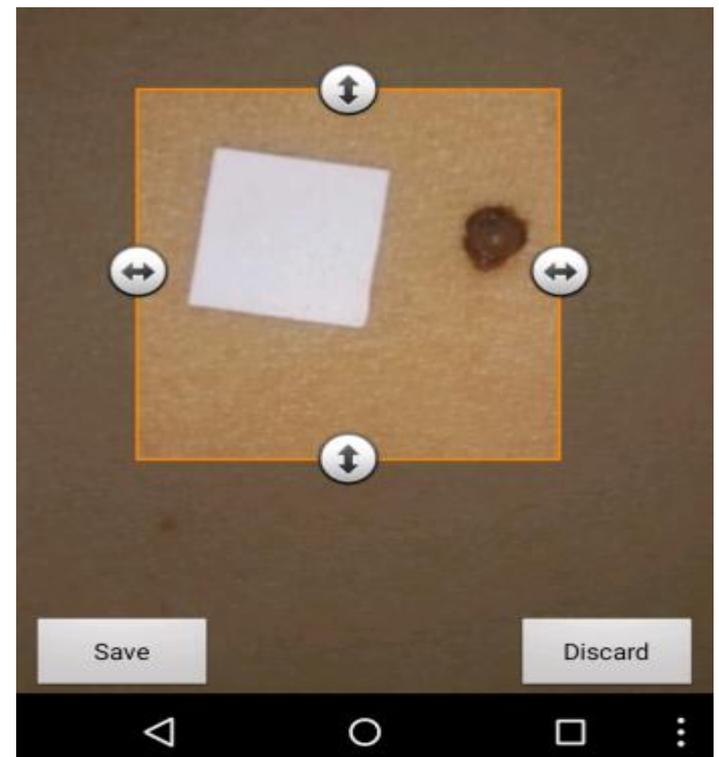


Pré-processamento – 1º passo

Carregar imagem



Cortar imagem



Pré-processamento – 2º passo



Imagem original

```
Mat image = new Mat();  
image.Bitmap = requestedPic;
```

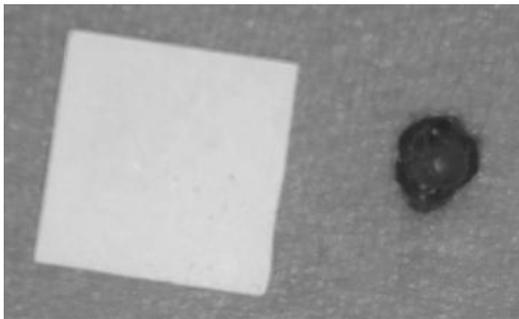


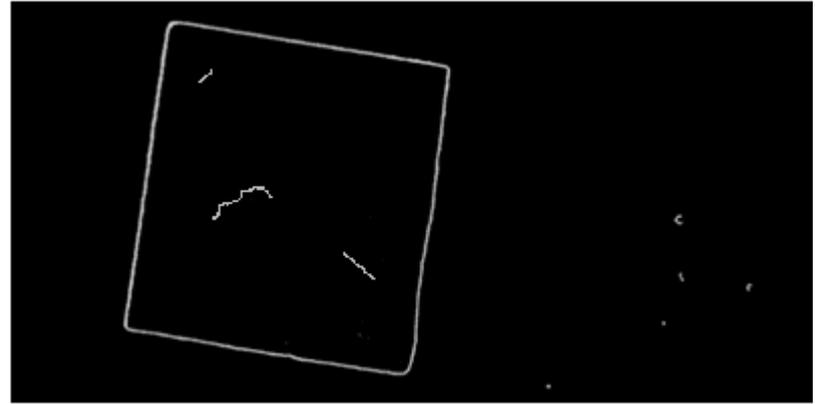
Imagem em escalas de cinza

```
Mat imageGray = new Mat(image.Size, DepthType.Cv8U,  
255);  
CvInvoke.CvtColor(image, imageGray,  
ColorConversion.Rgba2Gray);
```

Gabarito – 1º passo

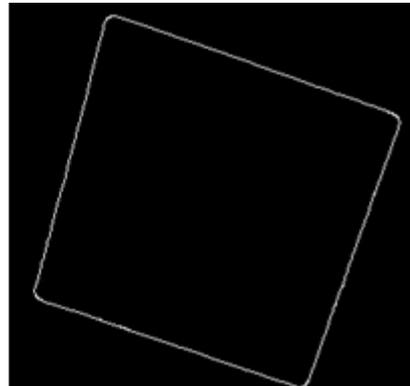
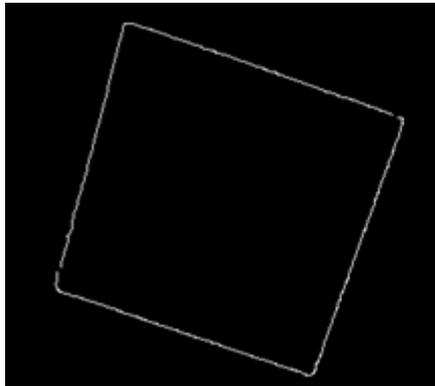
Filtro de Canny¹

```
CvInvoke.Canny(imageGray, binaryImg, 200, 90);
```



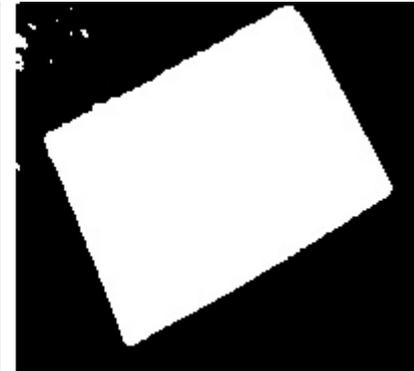
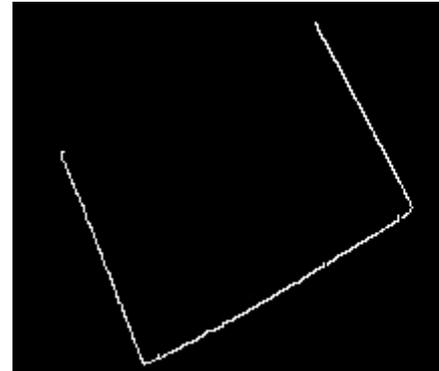
Borrão²

```
CvInvoke.MedianBlur(imageGray,  
binaryImg, 9);  
CvInvoke.Canny(binaryImg, binaryImg,  
200, 90);
```



Limiarização³

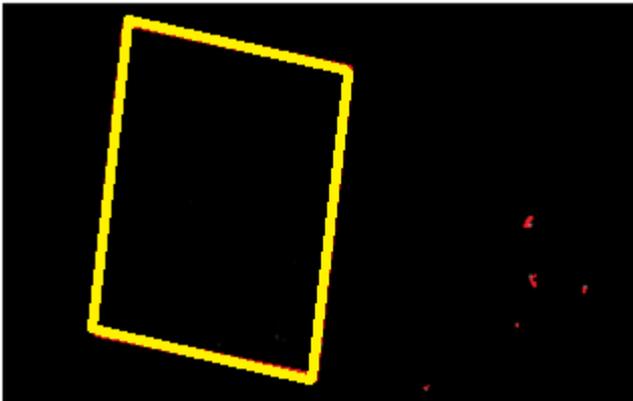
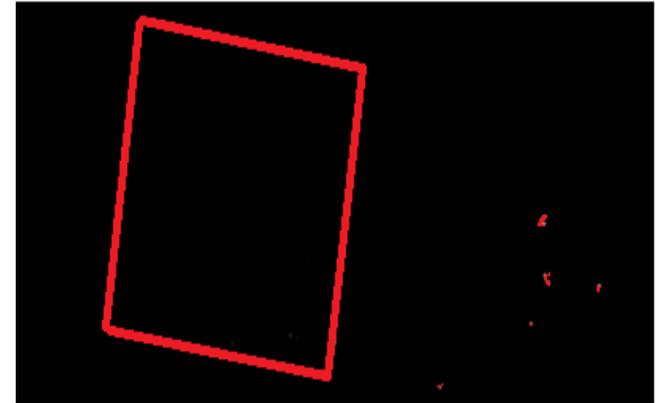
```
CvInvoke.Threshold(imageGray,  
binaryImg, 180, 255,  
Emgu.CV.CvEnum.ThresholdType.Binar  
y);
```



Gabarito – 2º passo

Contornos encontrados

```
VectorOfVectorOfPoint contours = new VectorOfVectorOfPoint();  
CvInvoke.FindContours(binaryImg, contours, null,  
    RetrType.External, ChainApproxMethod.ChainApproxSimple);
```



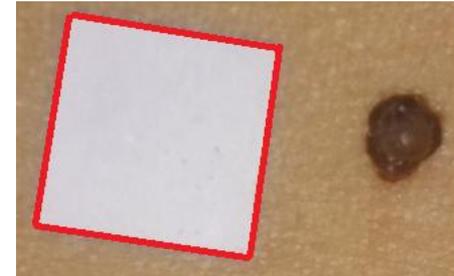
Processa contornos

```
VerifyQuadrangle(contours);  
ProcessQuadrangles(contours);
```

Gabarito – 3º passo

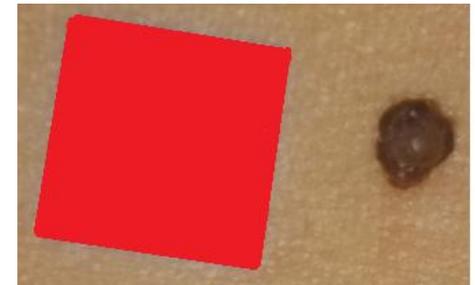
Perímetro

```
var quadranglePerimeter = CvInvoke.ArcLength(quadrangle, true);
```



Área

```
var quadrangleArea = CvInvoke.ContourArea(quadrangle, false);
```



Gabarito – 4º passo

Calculo comprimento

```
var rotatedRect = CvInvoke.MinAreaRect (quadrangle);  
var rect_points = rotatedRect.GetVertices ();
```



Lesão - 1º passo

Binarizar

```
CvInvoke.Threshold(imgDest, imgDest, _threshold, 255,  
Emgu.CV.CvEnum.ThresholdType.BinaryInv);
```



Lesão - 2º passo

Erosão e Dilatação

```
Erode(3,3);  
Dilate(12,12);  
Erode(12,12);  
Dilate(3,3);
```



Lesão - 3º passo

Contornos da lesão

```
var retContours = CvInvoke.FindContours(imgDest, contours, hierarchy,  
Emgu.CV.CvEnum.ReprType.External,  
Emgu.CV.CvEnum.ChainApproxMethod.ChainApproxSimple);  
var lesionContour = FindElementWithHiggestCircularity(retContours);  
Draw(lesionContour);
```



Lesão - 4º passo

Medidas da lesão

```
var rectMole = CvInvoke.BoundingRectangle(contours[contournIdx]);  
float moleW = rectMole.Width;  
float moleH = rectMole.Height;  
var moleArea = CvInvoke.ContourArea(contours[contournIdx]);  
var molePerimeter = CvInvoke.ArcLength(contours[contournIdx], false);
```



Calculo

```
float real_w_patch = 10; float real_h_patch = 10;  
float quadrangleWH = quadrangleLenght;  
var real_w_mole = moleW / quadrangleWH * real_w_patch;  
var real_h_mole = moleH / quadrangleWH * real_h_patch;  
var real_moleArea = (moleArea / quadrangleArea) * real_w_patch;  
var real_molePerimeter = (molePerimeter / quadranglePerimeter) * real_w_patch;
```

real_molePerimeter



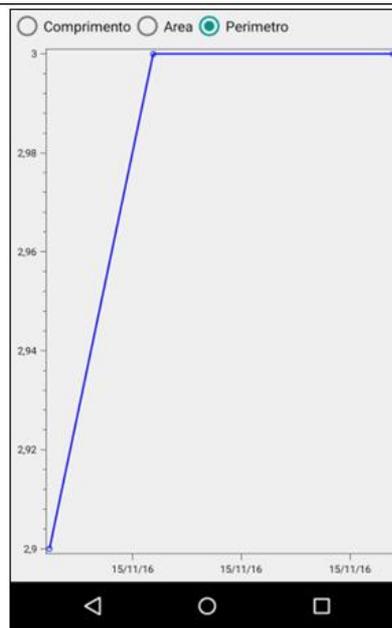
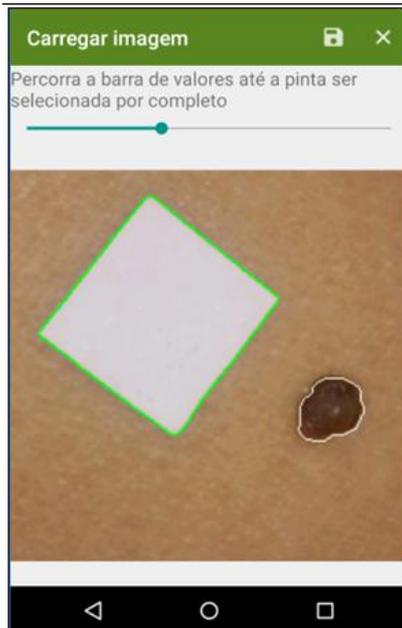
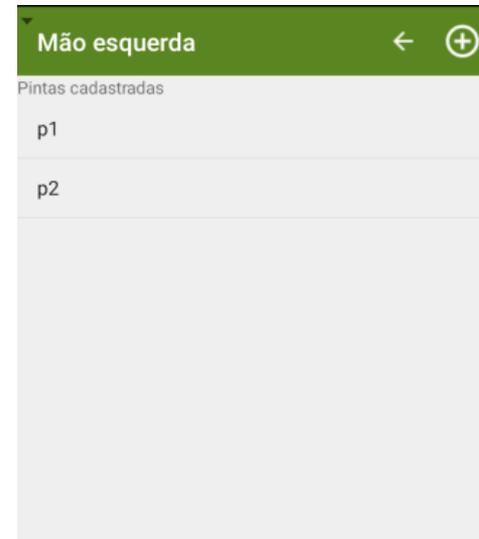
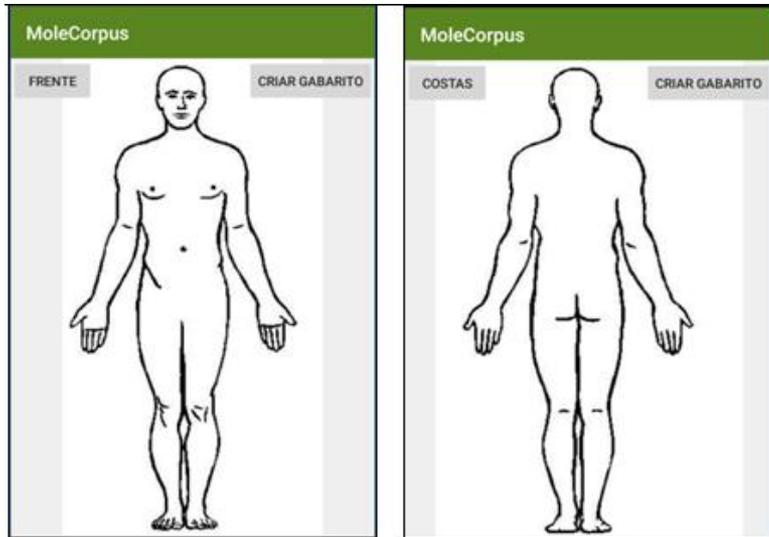
real_moleArea



real_h_mole



Operacionalidade do aplicativo



Testes

- Testes realizados com imagens capturadas em 10, 20 e 30 centímetros de distância da lesão e gabarito.
- Foram feitas 8 capturas de lesões em locais do corpo diferente com 1 usuário.
- Testes realizados pelo usuário.
- Testes realizados para área, perímetro e comprimento da lesão

Comprimento

Tabela 1 - Resultado dos dados obtidos no comprimento da lesão

Distância da captura	Braço direito/ Pinta 1	Coxa direita/ Pinta 1	Coxa direita/ Pinta 2	Ombro esquerdo/ Nuca	Ombro esquerdo/ Abaixo da nuca	Ombro esquerdo/ Parte esquerda	Peito / Pinta 1	Torso/ Meio
Tamanho real	4mm	6mm	5mm	6mm	5mm	6mm	6mm	5mm
10cm	-	5mm	5mm	6mm	5mm	6mm	6mm	4mm
20cm	4mm	6mm	6mm	6mm	5mm	6mm	5mm	4mm
30cm	5mm	6mm	5mm	6mm	5mm	6mm	6mm	5mm
40cm	ERRO	-	-	-	-	-	-	-

Imagens com medições erradas (comprimento)

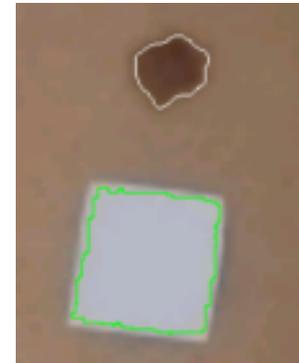
Coxa direita – pinta 1 (10cm)



Coxa direita – pinta 2 (10cm)



Braço direito – pinta 1 (20cm)



Peito – pinta 1 (20cm)



Torso - Meio (10cm)



Torso – Meio (20cm)



Área

Tabela 2 - Dados obtidos em relação a área da lesão

Distância da captura	Braço direito/ Pinta 1	Coxa direita/ Pinta 1	Coxa direita/ Pinta 2	Ombro esquerdo/ Nuca	Ombro esquerdo/ Abaixo da nuca	Ombro esquerdo/ Parte esquerda	Peito / Pinta 1	Torso / Meio
Tamanho real			1,98cm					
<u>10cm</u>	-	1,8cm	<u>2cm</u>	3,1cm	2,2cm	2,1cm	3,2cm	1,4cm
<u>20cm</u>	1,6cm	2,5cm	2,3cm	3,2cm	2,2cm	2,2cm	<u>3cm</u>	1,3cm
<u>30cm</u>	2,1cm	2,4cm	2,3cm	3,1cm	2,2cm	2,6cm	2,9cm	1,4cm
<u>40cm</u>	ERRO	-	-	-	-	-	-	-

Perimetro

Tabela 3 - Dados obtidos em relação ao perímetro da lesão

Distância da captura	Braço direito/ Pinta 1	Coxa direita/ Pinta 1	Coxa direita/ Pinta 2	Ombro esquerdo/ Nuca	Ombro esquerdo/ Abaixo da nuca	Ombro esquerdo/ Parte esquerda	Peito / Pinta 1	Torso/ Meio
10cm	-	4cm	4,2cm	5,5cm	4,5cm	5cm	4,8cm	3,6cm
20cm	3,8cm	4,6cm	4,5cm	5,1cm	4,4cm	5cm	4,5cm	3,4cm
30cm	4,1cm	4,7cm	4,7cm	5,1cm	4,2cm	5,1cm	4,6cm	3,5cm
40cm	ERRO	-	-	-	-	-	-	-

Conclusões

Motivos que corroboraram com a queda das taxas de acerto:

- Lesão e/ou gabarito não segmentados corretamente
- Ângulo da imagem
- Falta de nitidez/foco da imagem

Conclusões

- O protótipo obteve taxas de acerto satisfatórias em comprimento e perímetro, ambos de 75%. Já em questão de área, o método utilizado não se mostrou efetivo.
- O gabarito e a lesão foram segmentados corretamente em 100% dos casos, medindo-os em 10, 20 e 30cm de distância.
- Serve como base para outros trabalhos com foco na identificação de tamanho de elementos e segmentação de formas.

Limitações

- Deve-se cortar a parte de interesse para ter melhores resultados.
- Segmentação da lesão, na maioria das vezes, é feita manualmente pelo usuário.
- Funciona apenas na plataforma Android.

Extensões

- aprimorar a segmentação da lesão;
- identificar na representação do corpo o local exato da lesão;
- utilizar as outras classificações da regra ABCDE para tentar identificar um possível melanoma;
- aprimorar a captura da imagem para focalizar na lesão e no gabarito, assim facilitando o processo de segmentação.

Demonstração

Obrigado!