Software de Detecção de Estrabismo para Dispositivos Móveis

Renato Martins Lorenzi

Orientador: Marcel Hugo

FURB
UNIVERSIDADE DE BLUMENAU

Roteiro

- Introdução
- Objetivos
- Fundamentação Teórica
- Requisitos
- Especificação
- Implementação
- Operacionalidade da Implementação
- Resultados e Discussões
- Conclusões e Sugestões

Introdução

- Estrabismo é uma anomalia dos olhos que faz com que eles percam o paralelismo entre si
- 2% das crianças no mundo sofrem com algum tipo desta patologia (VAUGHAN; ASBURY, 1990, p. 216)
- 2,5% de ambliopias nas crianças, na maioria das vezes é causadas por um estrabismo não tratado (DÓRIA, 2006)

Introdução

- O processamento de imagens digitais, especialmente na medicina, tem evoluído ao longo dos anos (DUNCAN; AYACHE, 2000)
- Aumento da capacidade de processamento dos computadores
- Aumento da área de pesquisa de processamento de imagens

Introdução

- Desenvolver um aplicativo para dispositivos móveis, que serve para efetuar uma triagem
- Uso em ambiente comum, por pessoas leigas no assunto

Objetivos (Geral)

- O objetivo deste trabalho é o desenvolvimento de um aplicativo para dispositivos móveis que usam a plataforma Android, para realizar a detecção do estrabismo utilizando-se de técnicas de processamento de imagem e visão computacional, alinhado com as técnicas utilizadas por oftalmologistas para detecção desta patologia.

Objetivos (Específicos)

a) capturar as imagens utilizando a câmera do dispositivo móvel;

b) determinar a técnica de detecção de estrabismo e de processamento de imagem mais adequadas às limitações tecnológicas de um dispositivo móvel;

c) realizar a detecção do estrabismo através das imagens capturadas;

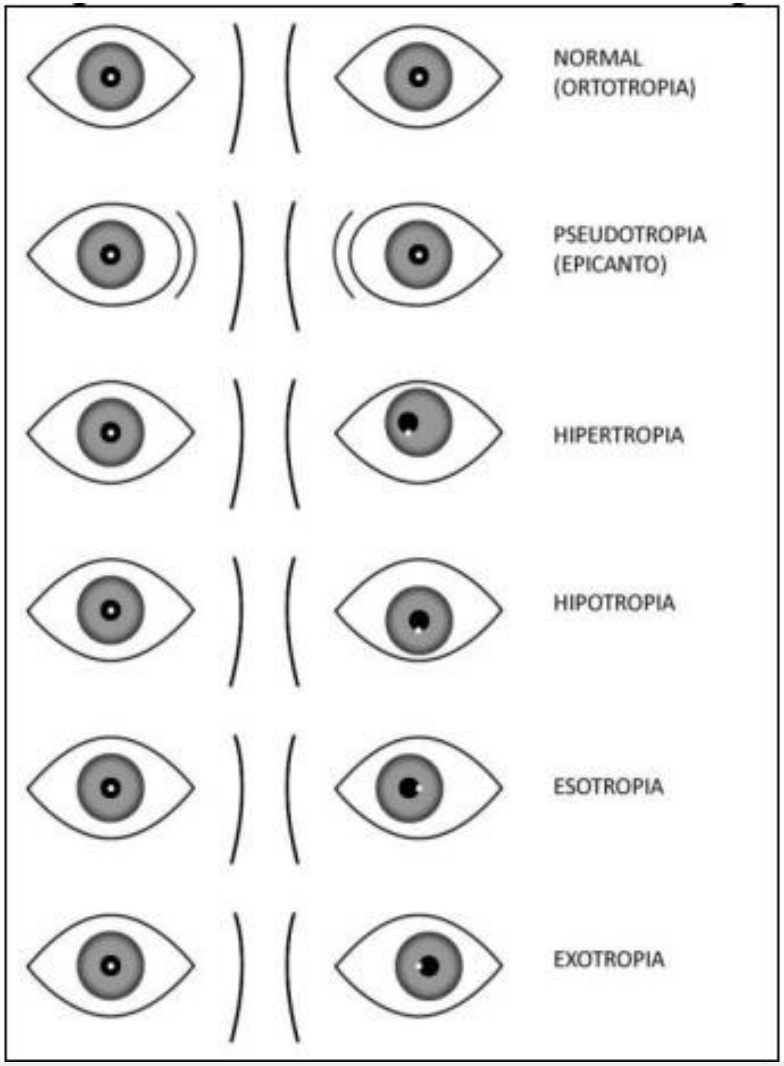
d) apresentar o resultado da detecção informando o tipo de desvio.

Fundamentação Teórica
Estrabismo

- Dois objetos diferentes projetados, serão vistos superpostos e as diferenças impedirão a fusão em uma única imagem
- Em uma pessoa com estrabismo qualquer um dos olhos pode estar desalinhado

Fonte: Brum (2014).

Fundamentação Teórica
Método de Hirshberg



Fonte: Silva, Ferreira e Pinto (2013, p. 12).

Fundamentação Teórica
Processamento de imagens

- Processo de receber de entrada uma imagem e retornar uma imagem modificada ou atributos da imagem
- Uma imagem digital é uma função bidimensional de intensidade de luz f (x, y)

Fundamentação Teórica
Transformada de Hough

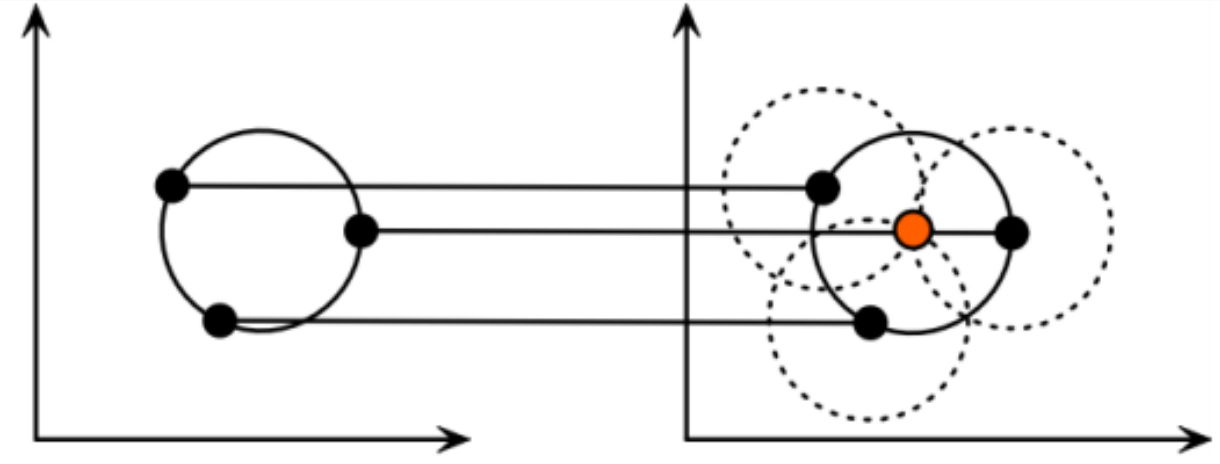
- Desenvolvida por Paul Hough em 1962
- Detecta características analiticamente representáveis em imagens, assim como linhas, círculos e elipses
- Atualmente possui estudos para análise de objetos mais complexos

Fundamentação Teórica
Transformada de Hough

- Pré-processamento com detector de bordas Canny
- Para detecção de círculos qualquer ponto de borda em uma imagem binária pode ser considerado parte de um círculo

Fundamentação Teórica
Transformada de Hough

- Desenhar o perímetro de um círculo para cada ponto de borda encontrado:



Fundamentação Teórica
Transformada de Hough

- A equação paramétrica, usada pela transformada de Hough para detecção de círculos:

$$x = a + r\cos\theta$$
$$y = b + r\sin\theta$$

Fonte: Rhody (2005, p. 1).

Requisitos

a) disponibilizar uma interface para que o usuário possa tirar a foto dos olhos (Requisito Funcional - RF);

b) validar se as imagens capturadas atendem aos requisitos mínimos de luminosidade e enquadramento para que possa se fazer a detecção (RF);

c) detectar indícios de estrabismo na imagem capturada, caso ocorra a patologia (RF);

d) disponibilizar uma interface com o resultado da detecção (RF);

e) disponibilizar uma interface com o histórico das detecções efetuadas pelo dispositivo (RF);

Requisitos

f) utilizar a linguagem de programação Java (Requisito Não-Funcional RNF);

g) extrair a localização da pupila utilizando a transformada de Hough (RNF);

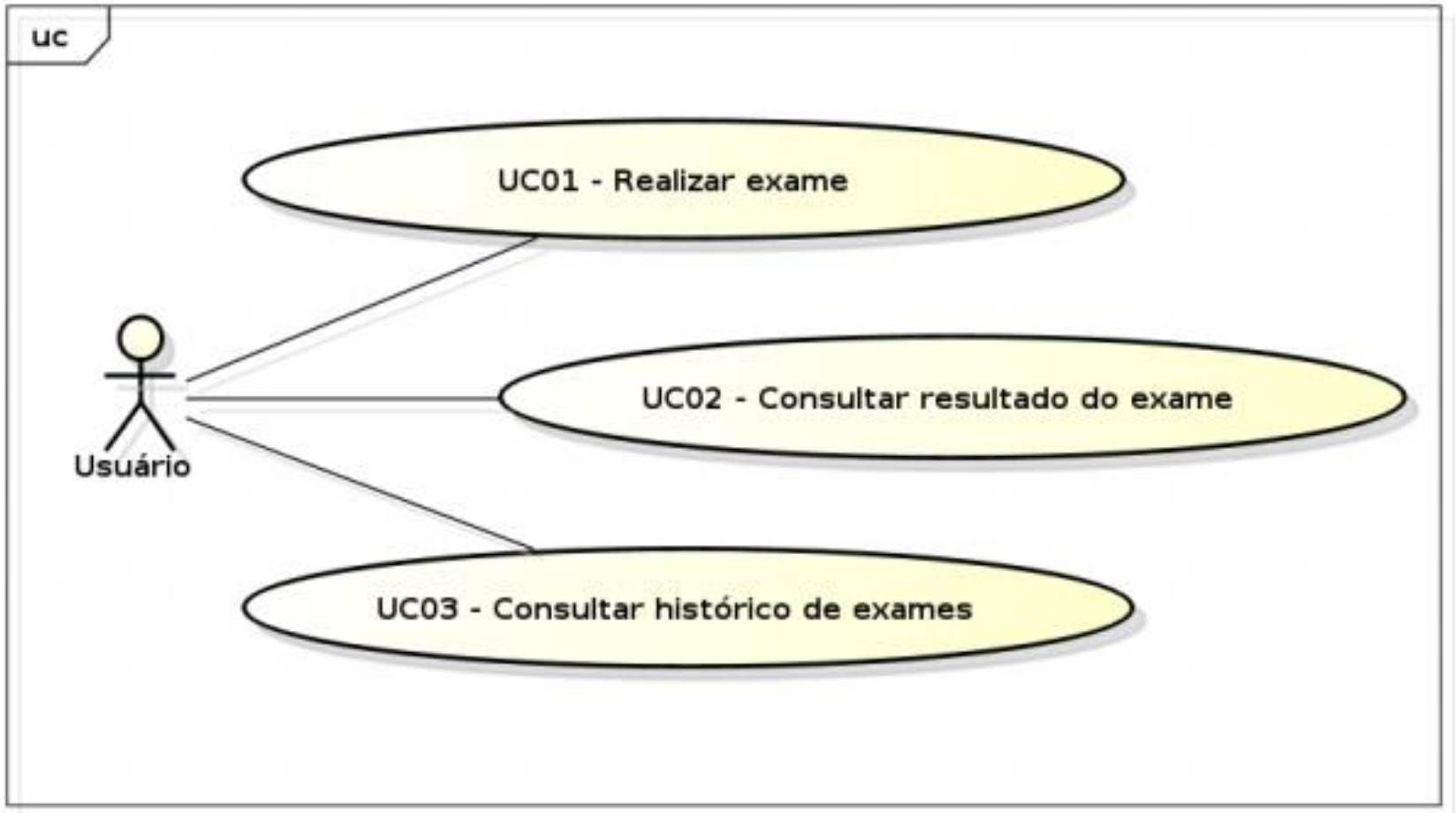
h) construir uma API para todo o processo de detecção do estrabismo que seja independente de plataforma (RNF);

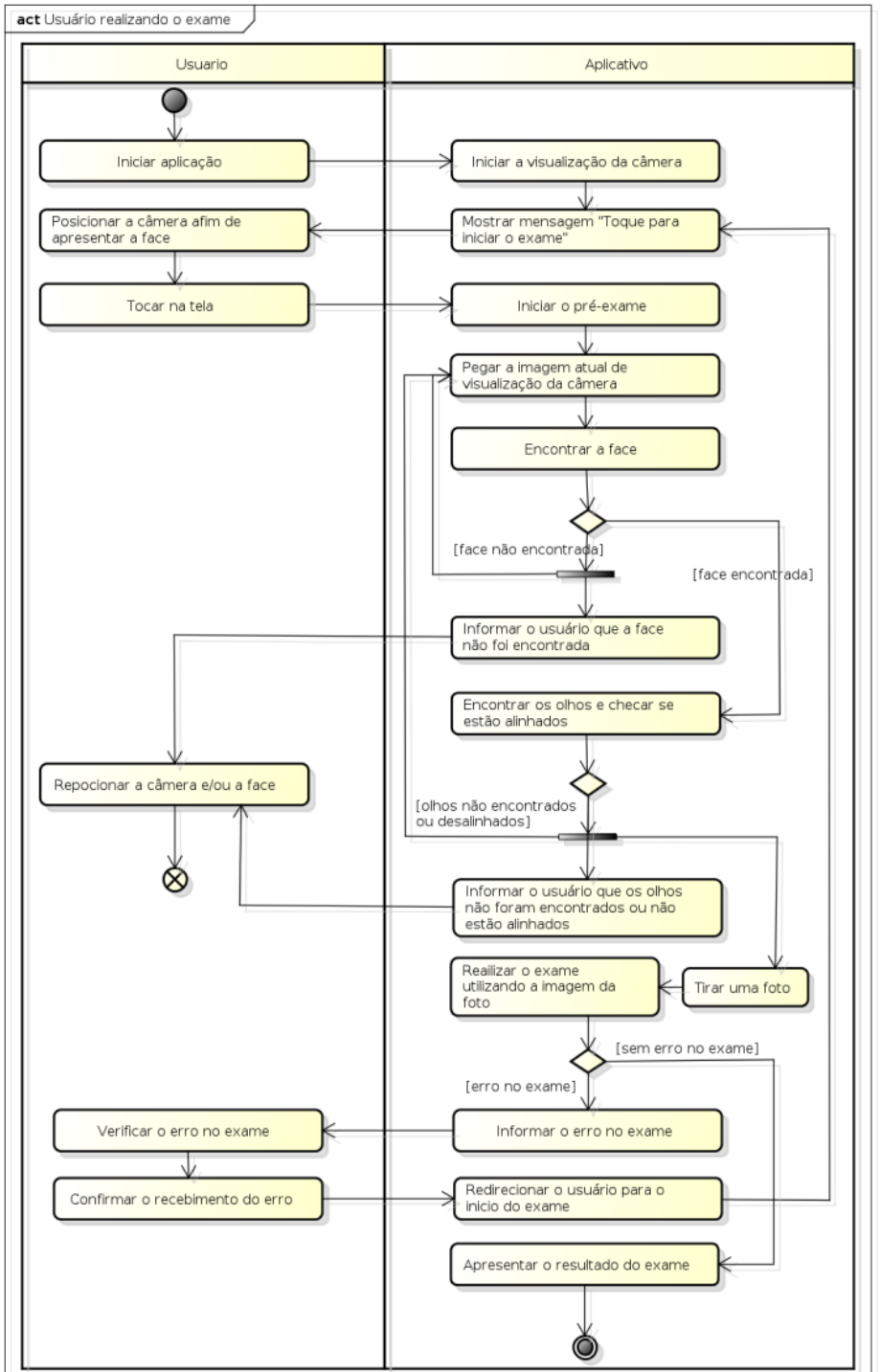
i) ser compatível com a plataforma Android (RNF);

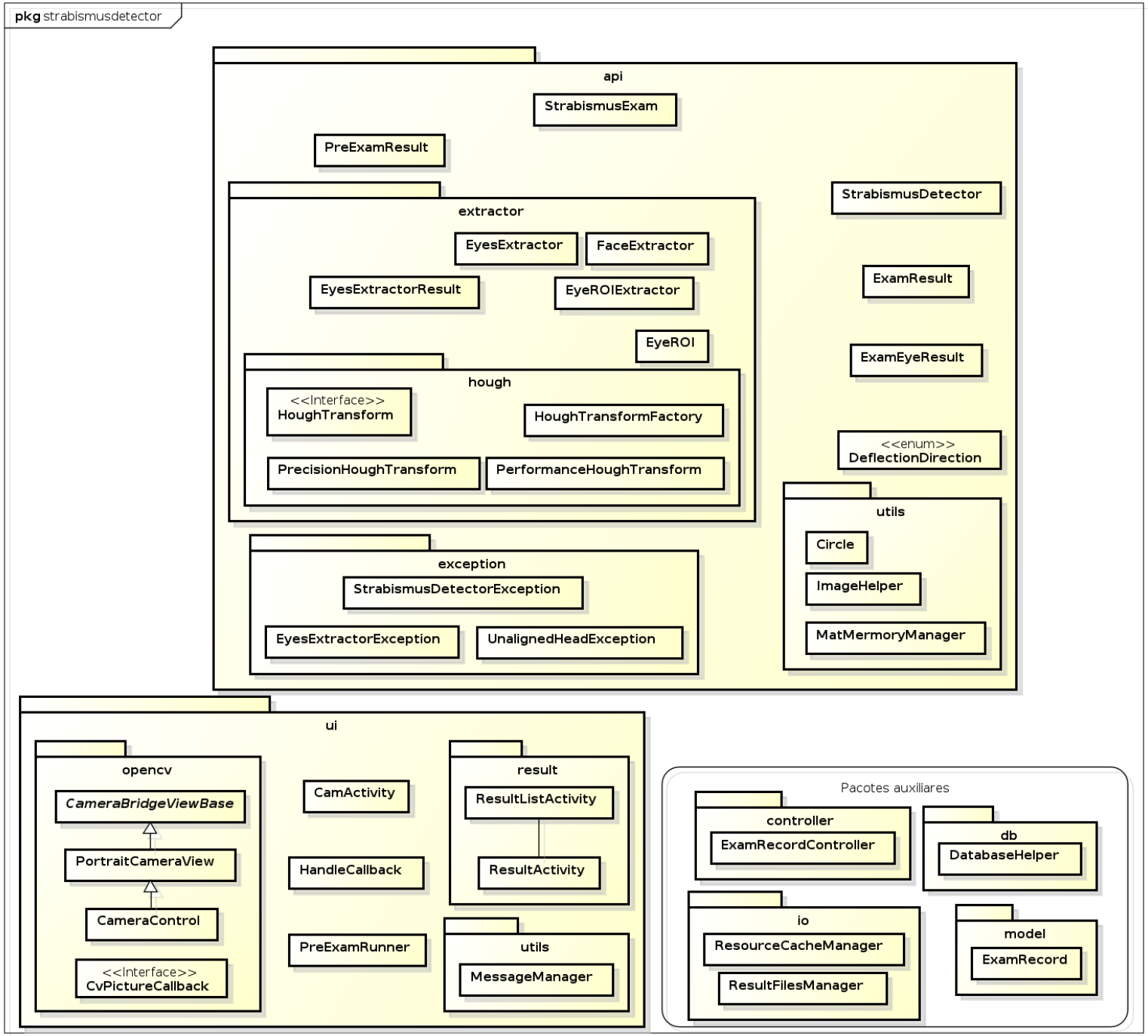
j) utilizar a biblioteca OpenCV para desenvolver a parte de processamento de imagem e visão computacional (RNF);

k) utilizar o ambiente de desenvolvimento Eclipse com o plugin de desenvolvimento para a plataforma Android (RNF).

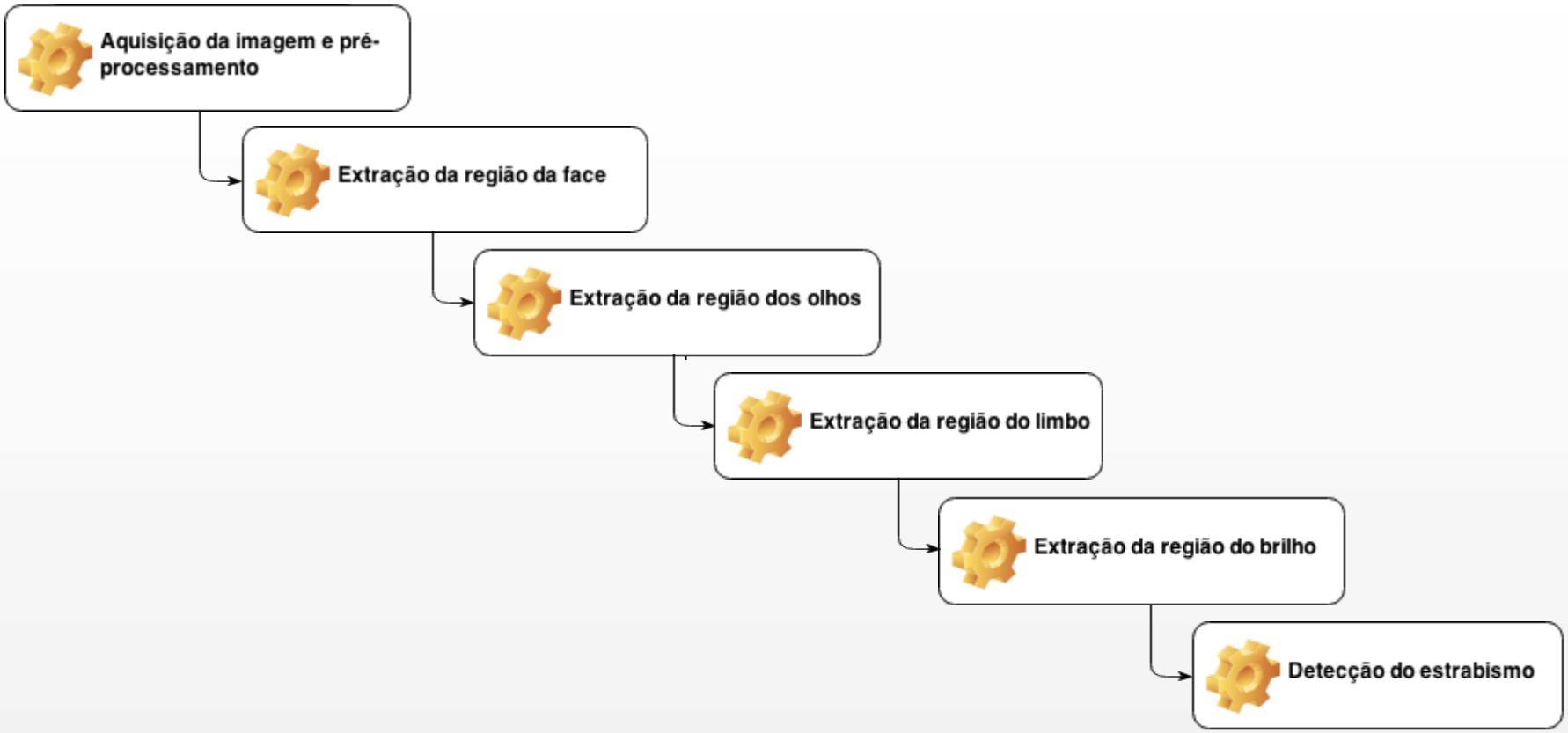
Especificação



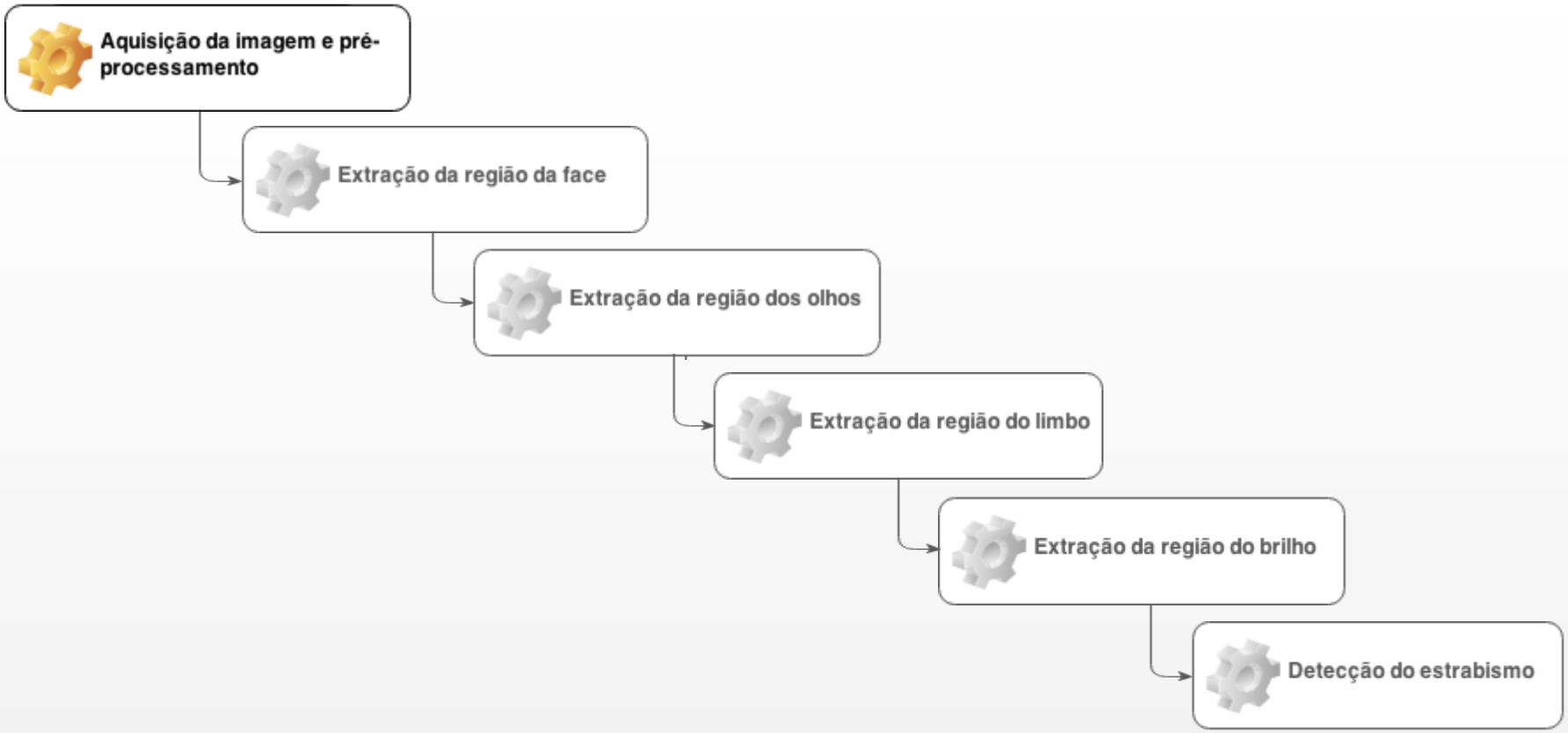



pkg strabismusdetector
api
StrabismusExam
PreExamResult
StrabismusDetector
extractor
EyesExtractor
FaceExtractor
EyesExtractorResult
EyeROIExtractor
ExamResult
EyeROI
ExamEyeResult
hough
<<Interface>>
HoughTransform
HoughTransformFactory
<<enum>>
DeflectionDirection
PrecisionHoughTransform
PerformanceHoughTransform
utils
Circle
ImageHelper
MatMermoryManager
exception
StrabismusDetectorException
EyesExtractorException
UnalignedHeadException
ui
opencv
CameraBridgeViewBase
PortraitCameraView
CameraControl
<<Interface>>
CvPictureCallback
CamActivity
HandleCallback
PreExamRunner
result
ResultListActivity
ResultActivity
utils
MessageManager
Pacotes auxiliares
controller
ExamRecordController
db
DatabaseHelper
io
ResourceCacheManager
ResultFilesManager
model
ExamRecord

Implementação - Fases



Implementação - Fases

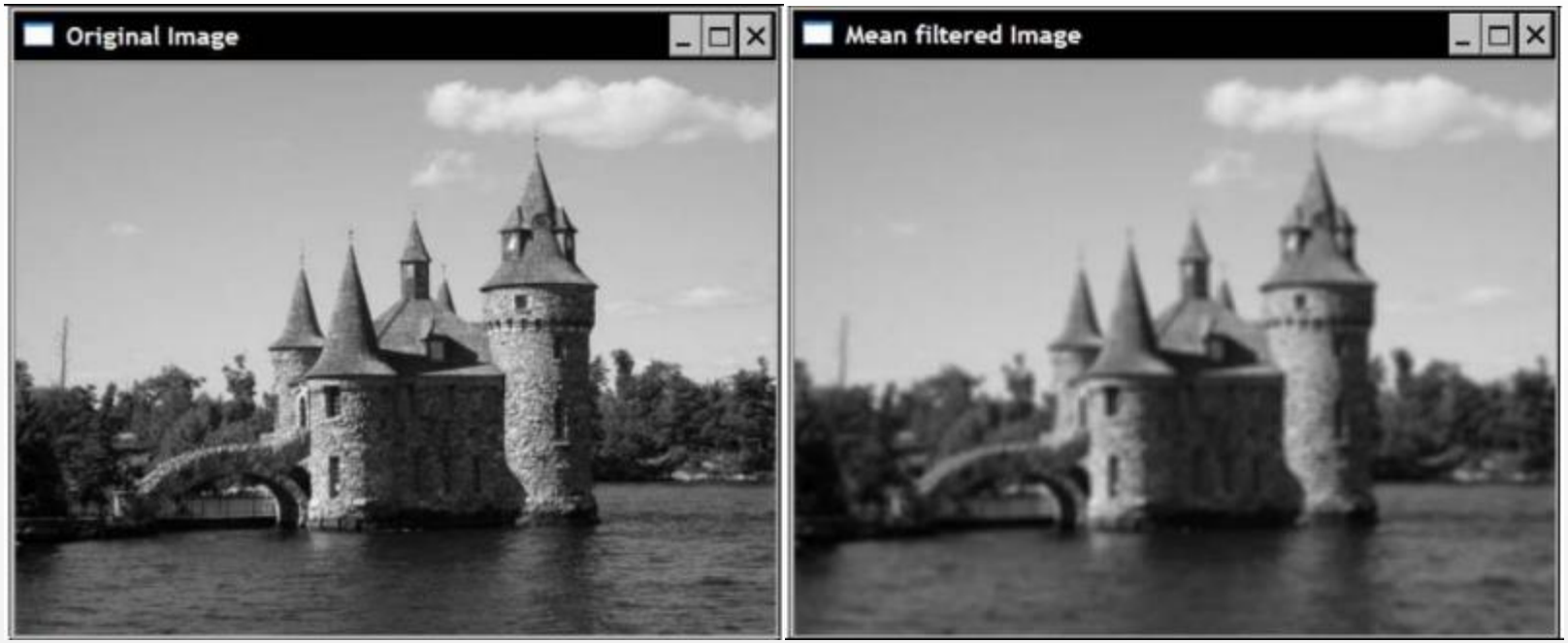


Aquisição da imagem e pré-processamento

- Uso da classe Camera disponibilizada pela biblioteca da plataforma
- Ativar o flash ao tirar a foto
- Realização do pré-exame

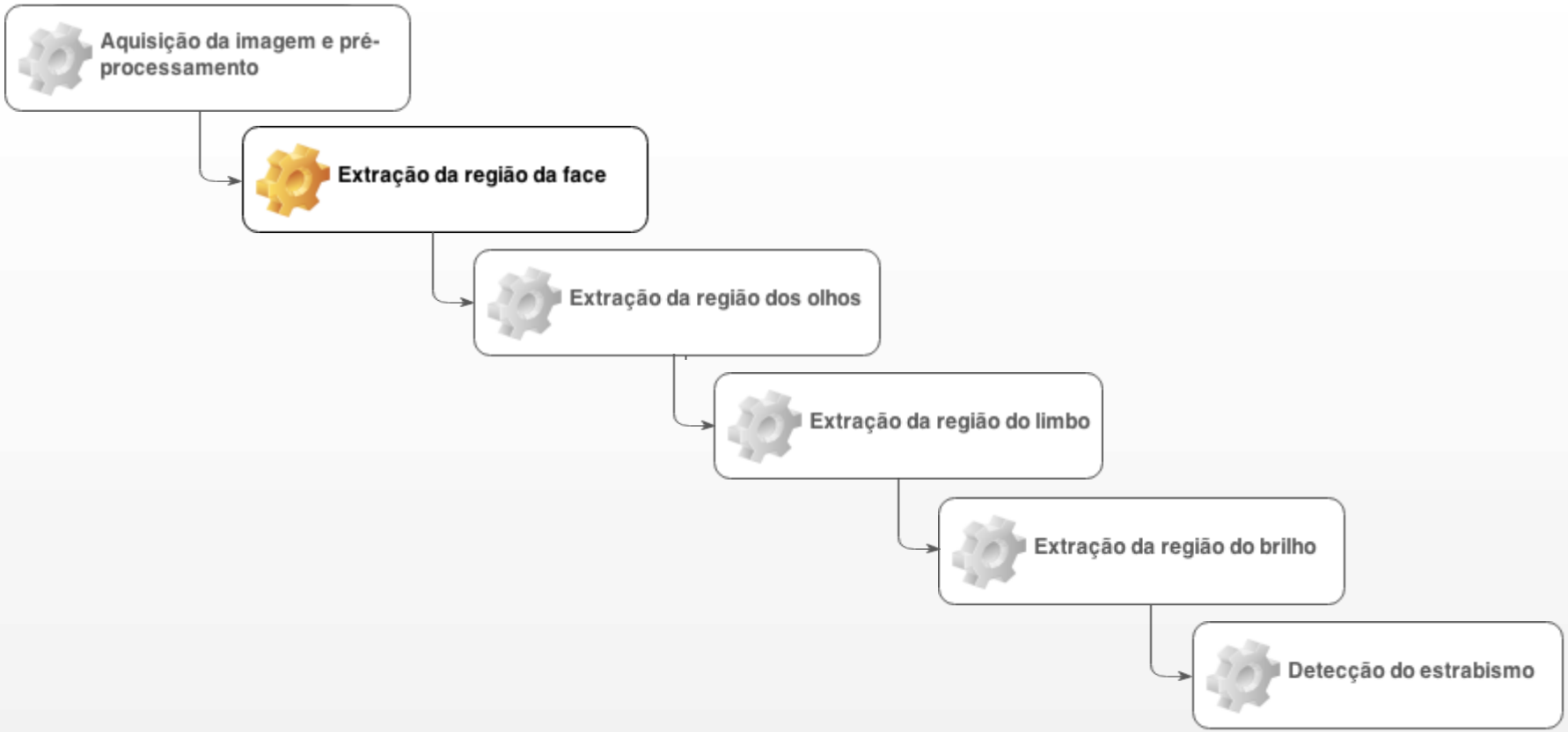
Aquisição da imagem e pré-processamento

- Suavização da imagem



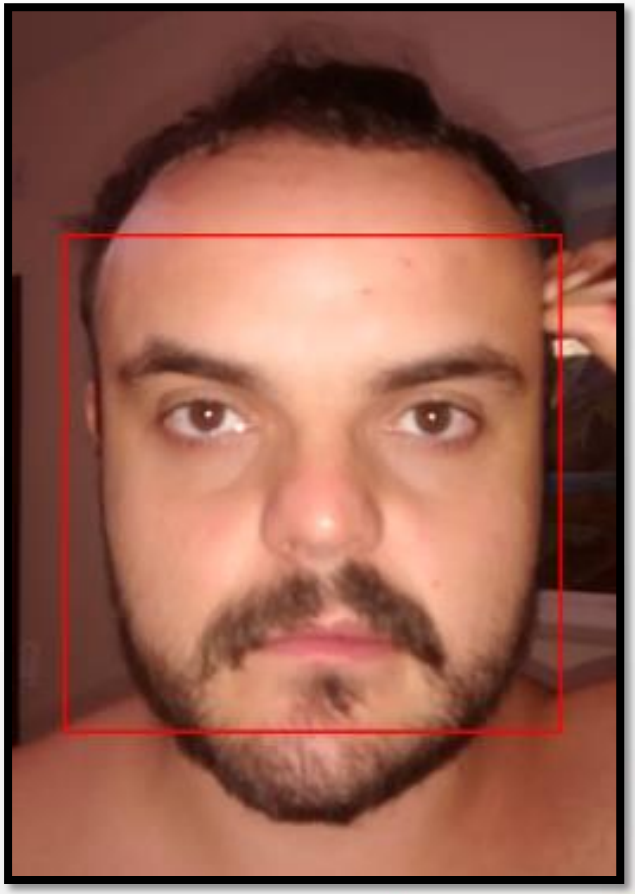
Fonte: adaptado de Laganière (2011, p. 142).

Implementação - Fases

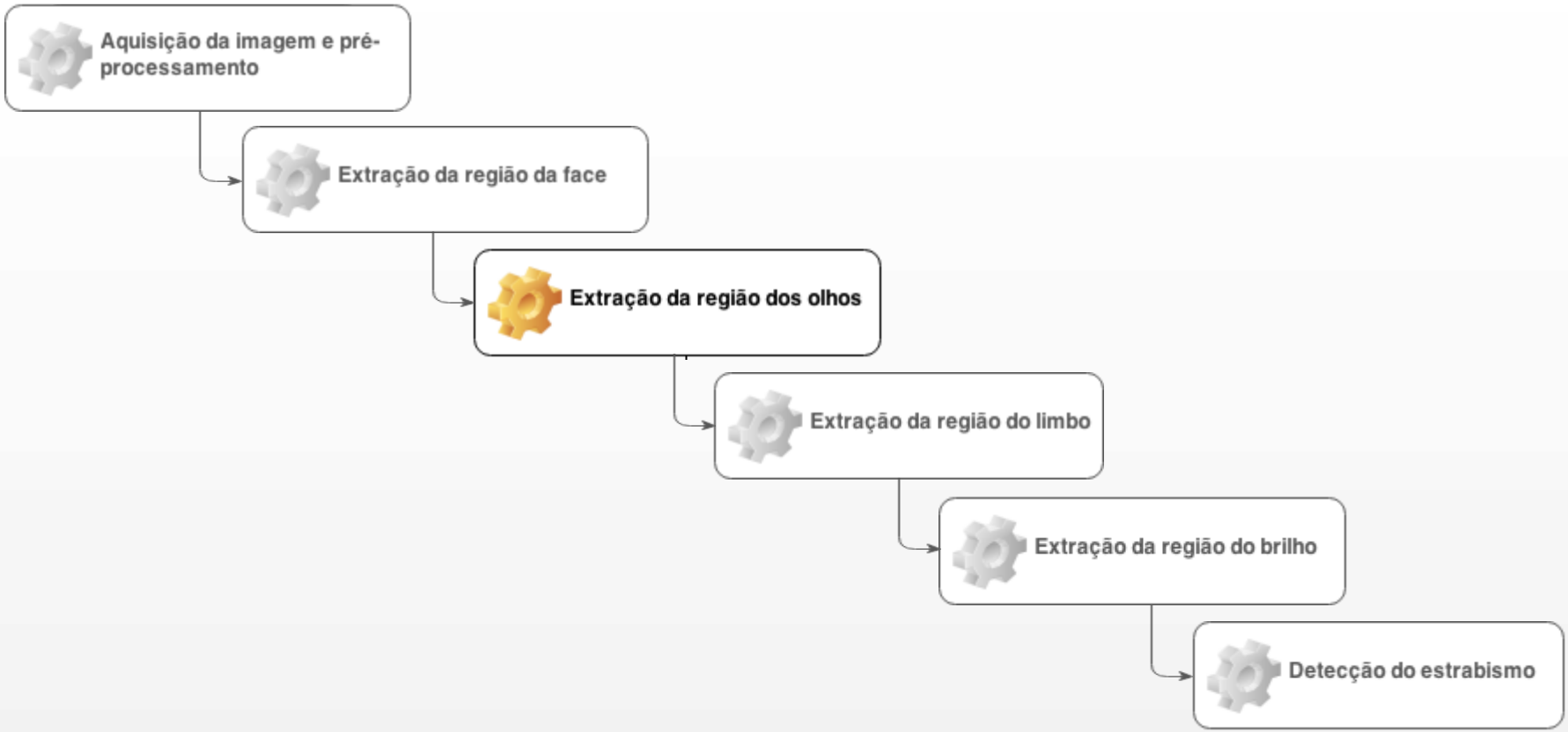


Implementação

- Realizada através da classe CascadeClassifier do OpenCV
- O arquivo com os pesos em números inteiros lbpcascade_frontalface.xml

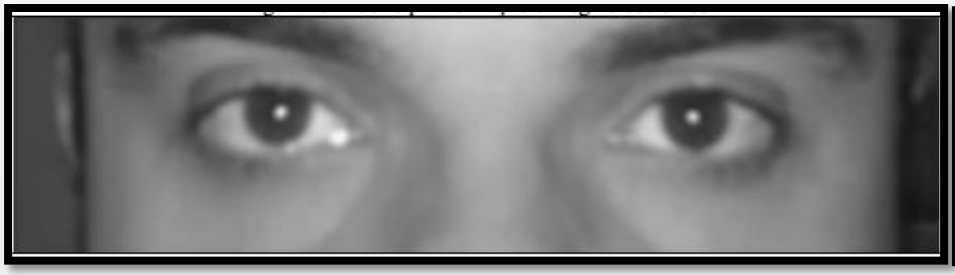


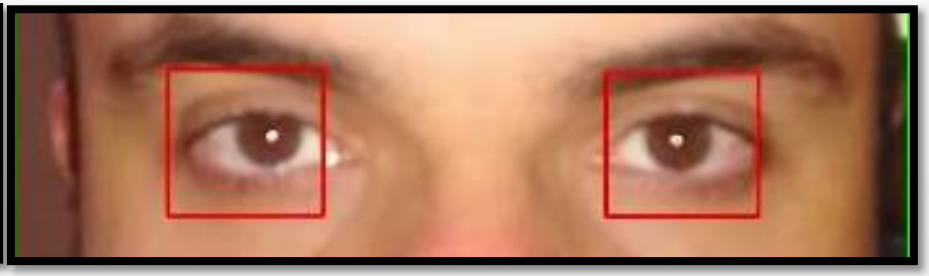
Implementação - Fases



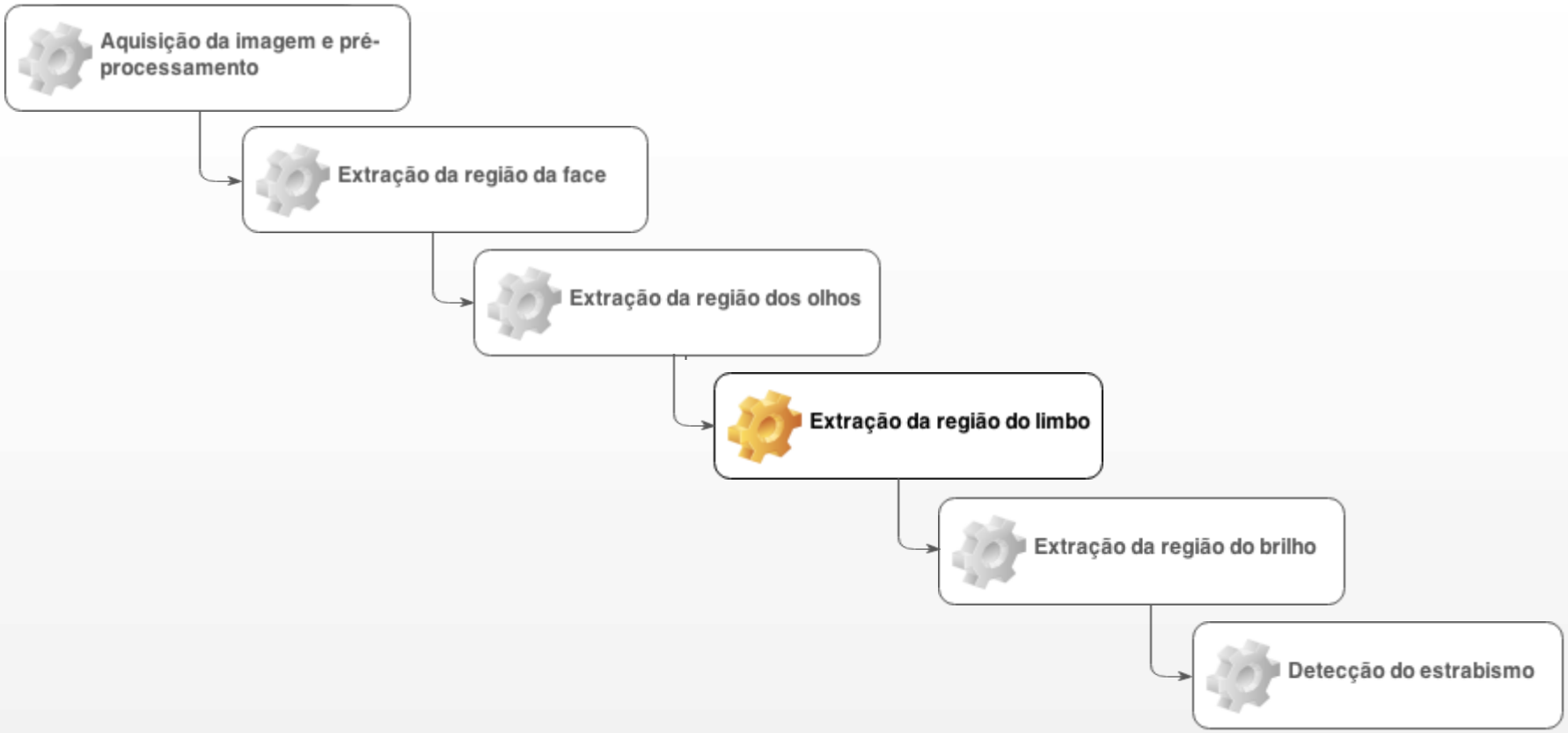
Implementação

- Realiza o corte na face encontrada
- Detecta através da classe CascadeClassifier do OpenCV
- O arquivo com os pesos haarcascade_eye.xml





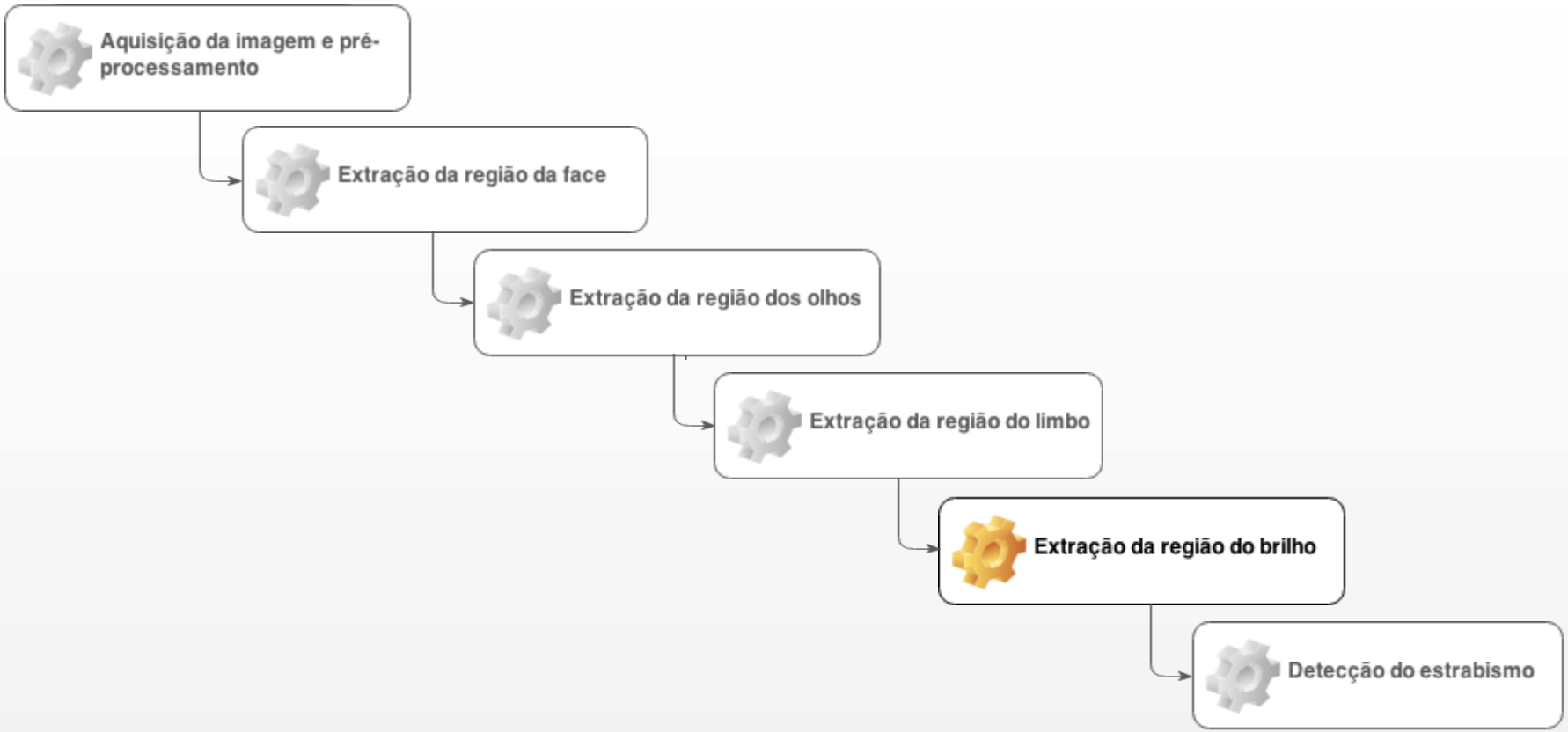
Implementação - Fases



Implementação

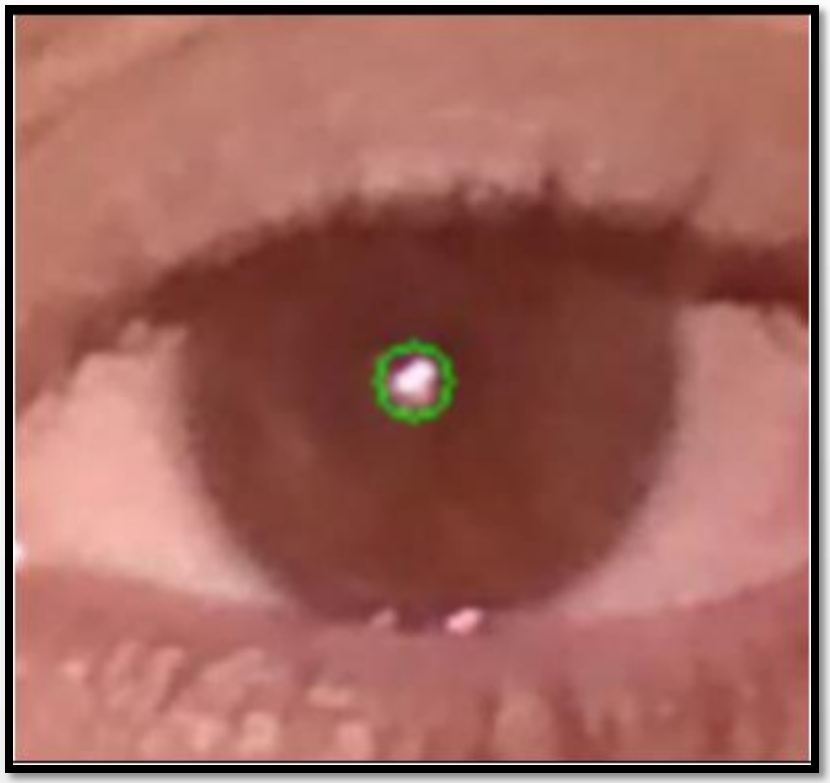
- Transformada de Hough aplicada à detecção de círculos
- Uso de ângulos de corte na chamada da transformada de Hough

Implementação - Fases

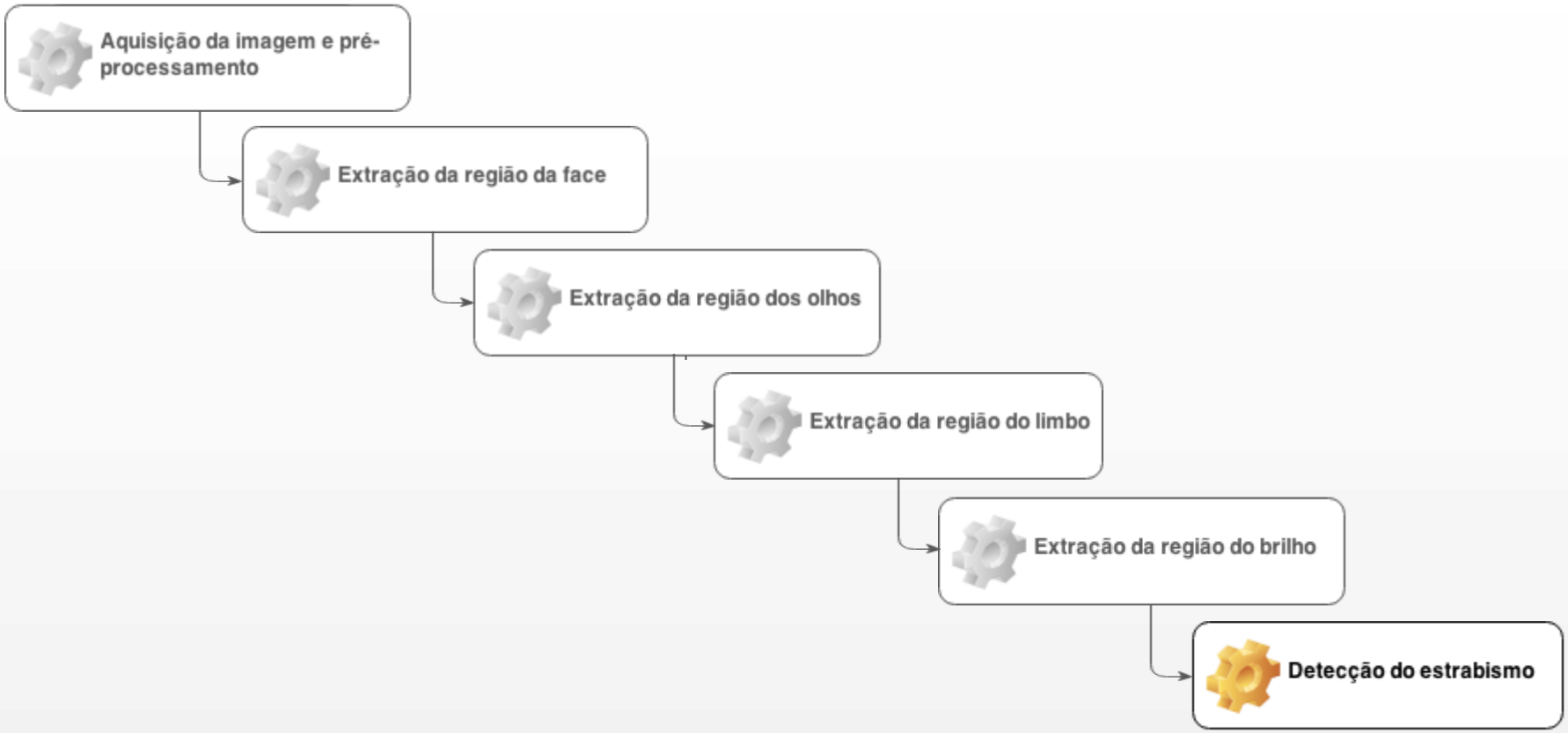


Implementação

- Transformada de Hough aplicada à detecção de círculos

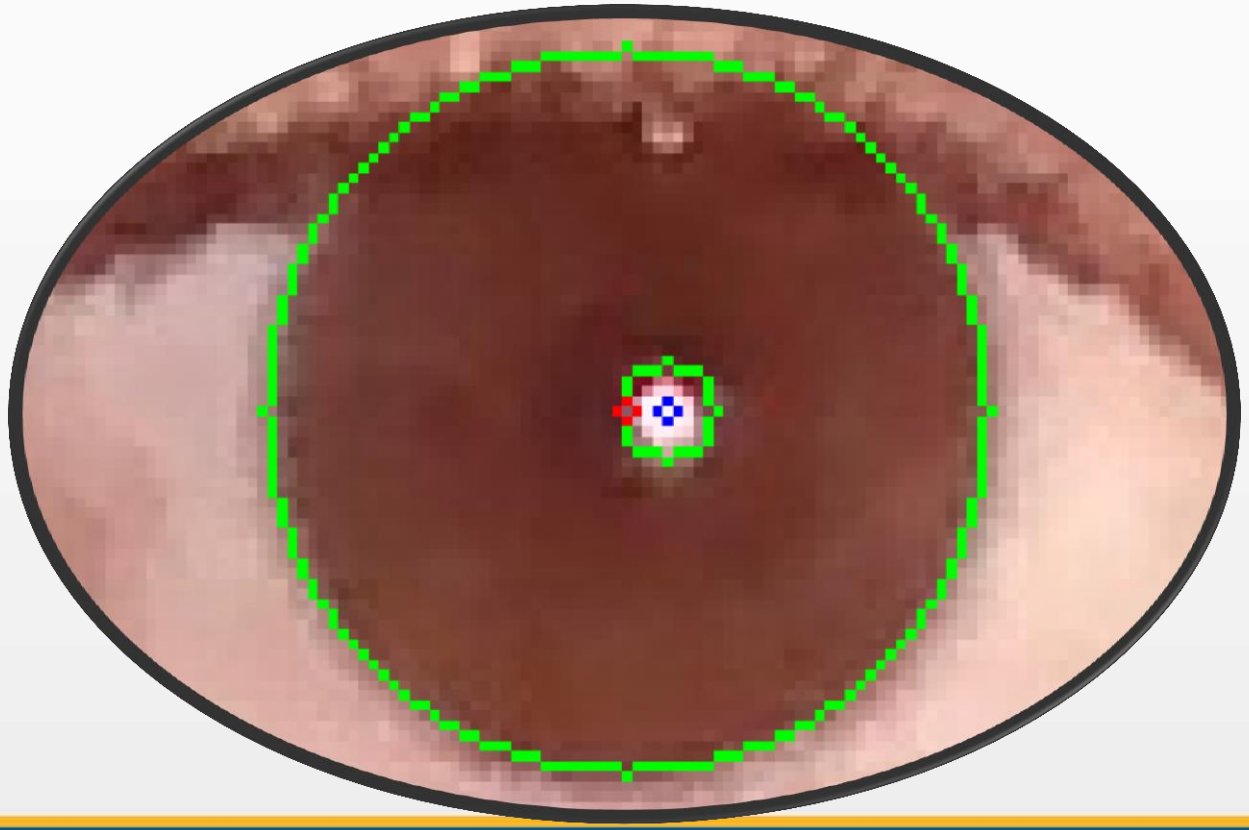


Implementação - Fases

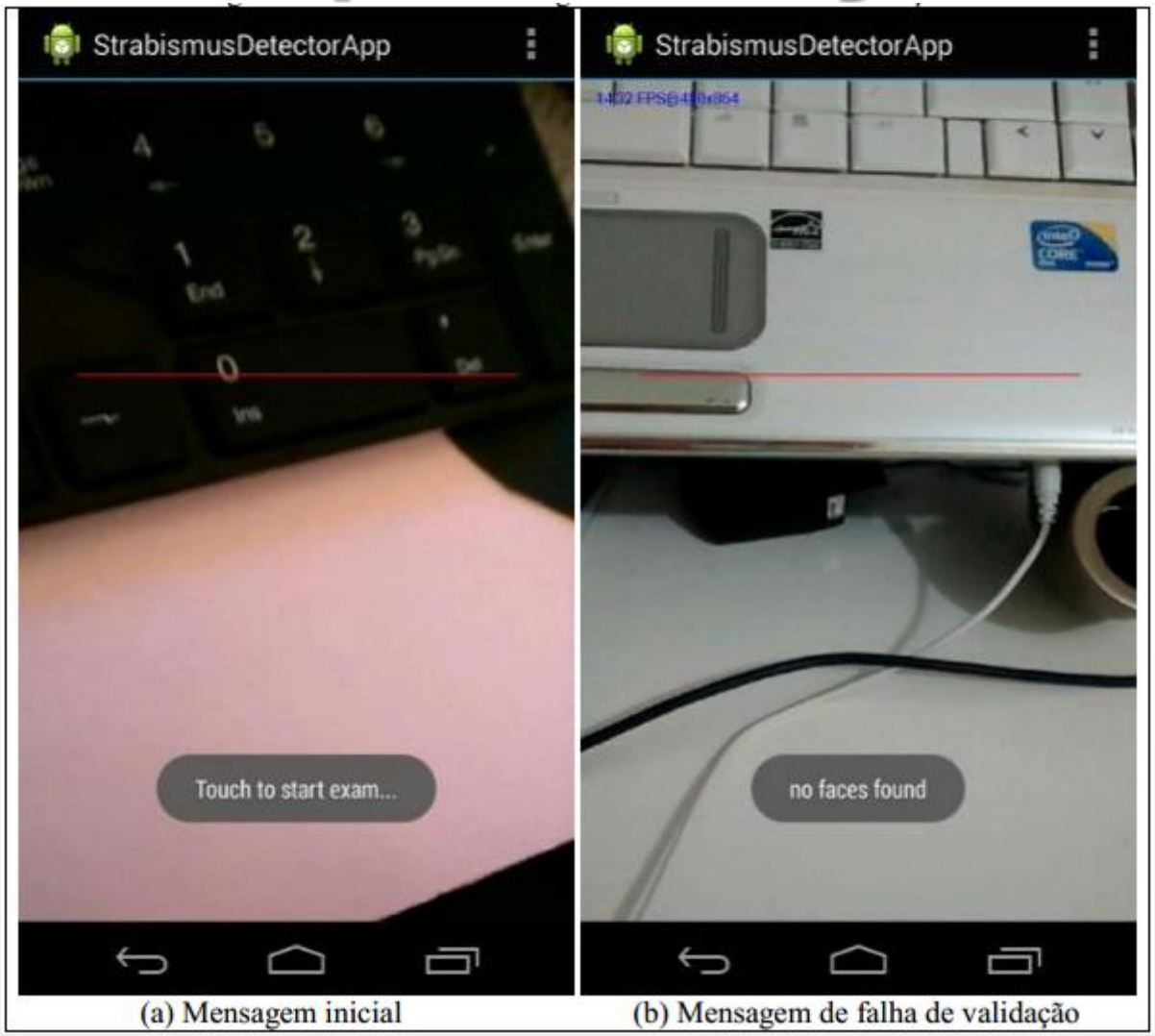


Implementação

- Cálculos em cima das regiões do limbo e do brilho
- Cálculo da proporcionalidade corneana

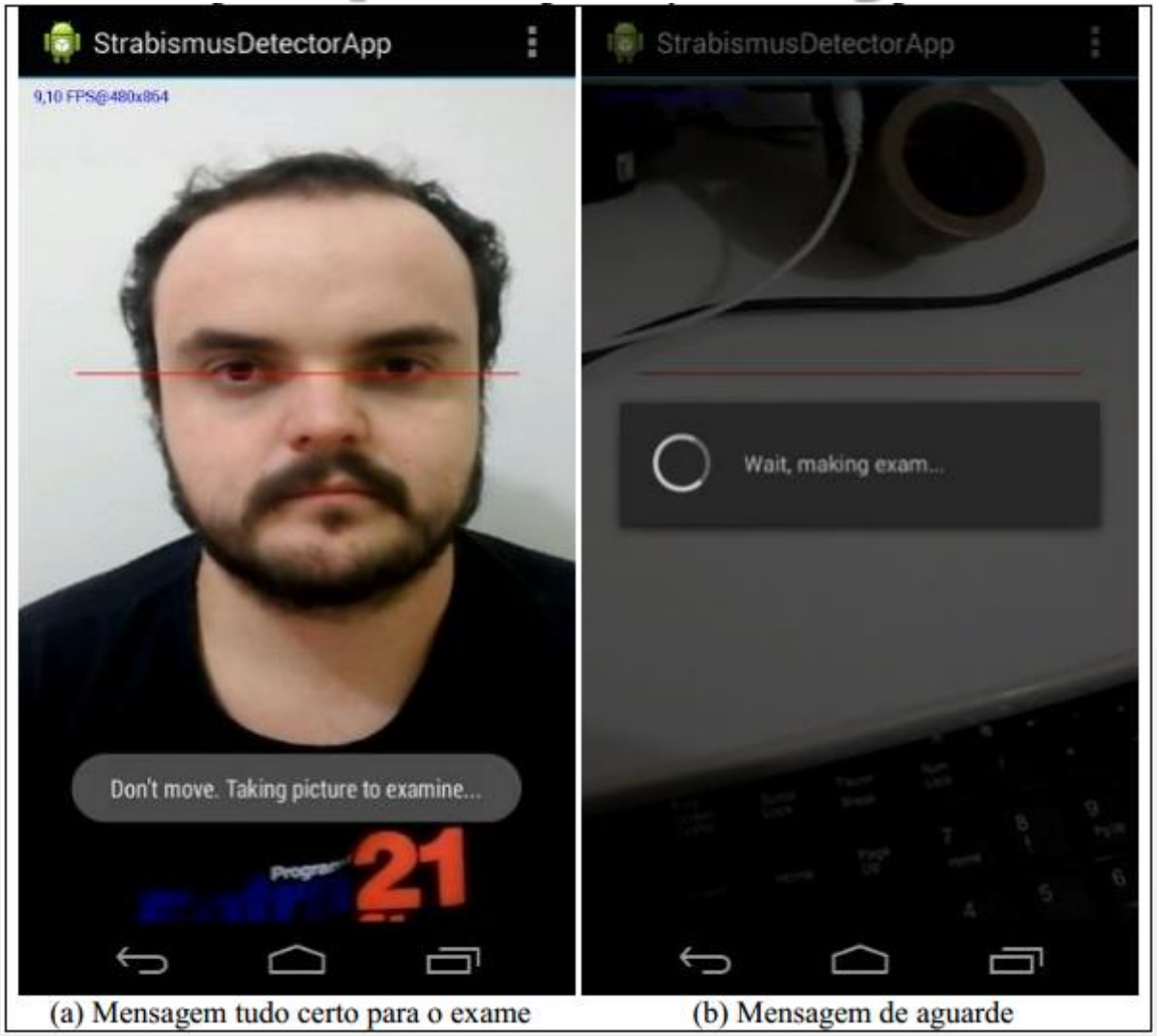


Operacionalidade da Implementação



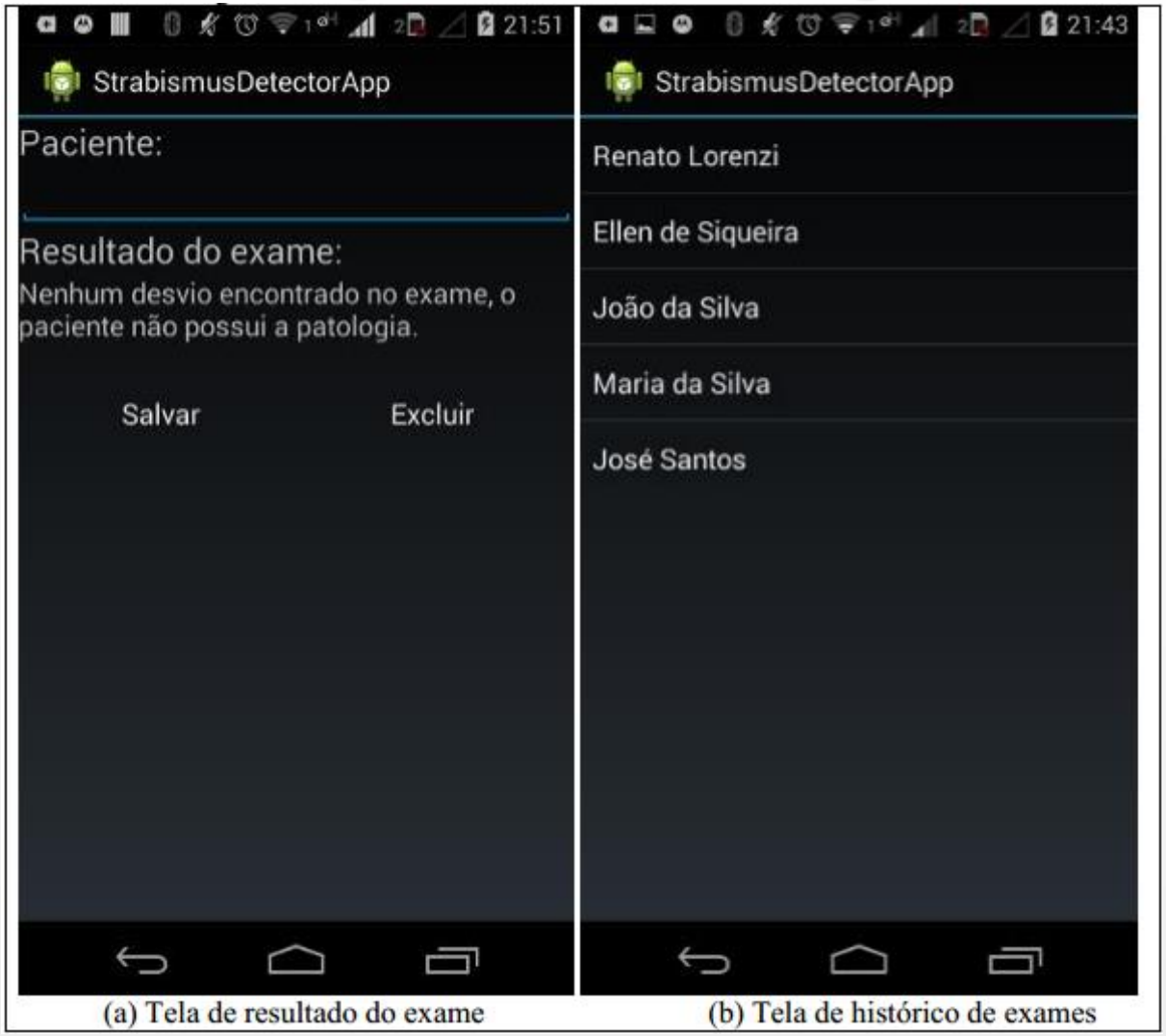
(a) Mensagem inicial (b) Mensagem de falha de validação

Operacionalidade da Implementação



(a) Mensagem tudo certo para o exame (b) Mensagem de aguarde

Operacionalidade da Implementação



(a) Tela de resultado do exame (b) Tela de histórico de exames

Resultados e Discussões

Precisão			
Fase	N° de pacientes	N° Amostras	% atingimento
Extração da região da face	23	110	100%
Extração da região dos olhos	23	110	100%
Extração da região do limbo	23	220	95.45%
Extração da região do brilho	23	220	96.36%
Detecção do estrabismo	21	52	85.71%

Resultados e Discussões

- A memória total alocada pela aplicação se manteve em torno de 16 a 18MB
- Pré-exame manteve uma taxa de 13 a 15 FPS

	Tempo (milissegundos)	
Fases	Smartphone	Notebook
Extração da região da face	25,6	15,2
Extração da região dos olhos	154	35,8
Extração da região do limbo	1516,2	159,8
Extração da região do brilho	143,6	16,2

Trabalhos Correlatos

Características	Medeiros (2008)	Almeida (2010)	Krueger (2012)	Lorenzi (2014)
dispositivos móveis			X	X
computadores convencionais	X	X		X
utilização do OpenCV	X		X	X
implementação em	Java	C++	Objective-C	Java
reconhecimento de olhos escuros	X	X		X
reconhecimento de olhos claros	X	X	X	X
Auxílio na captura das imagens				X
Quantidade de amostras mais relevantes		X		

Conclusões

- Técnica utilizada para a detecção do estrabismo foi o método de Hirshberg
- As técnicas de processamento de imagem utilizadas se mostraram eficiente
- Detecção de estrabismo, 85.71% deram o resultado esperado
- Criação da etapa de pré-exame

Sugestões

a) melhorar a etapa de pré-exame para que a mesma garanta que a amostra capturada seja válida;

b) estudar técnicas para definição de limiares de forma automática ao utilizar operador de Canny;

c) efetuar um trabalho de melhoria de desempenho em cima da transformada de Hough implementada neste trabalho;

d) aprimorar a quantidade e qualidade das amostras de teste.