

SOFTWARE DE RASTREIO DE OLHAR COM MÚLTIPLAS INTERFACES

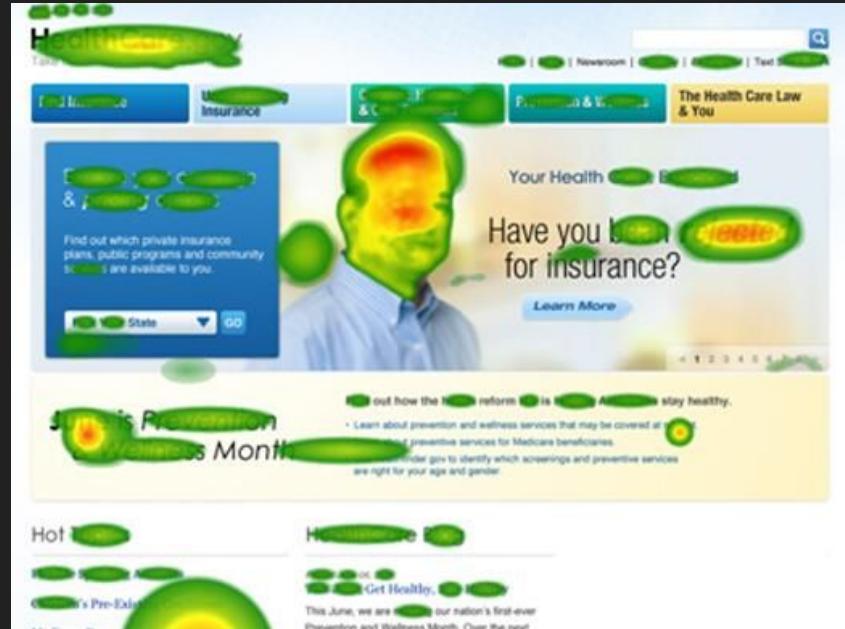
Marcelo Garcia Nuñez

Orientadora: Dra. Simone das Graças Domingues Prado

Introdução

Rastreamento de Olhar

- Área Interdisciplinar
 - Neurologia, psicologia, e computação
- Muito aplicado para publicidade



Fonte:

<https://www.usability.gov/how-to-and-tools/methods/eye-tracking.html>

Introdução

Eye Controllers



Fonte:

<https://www.pcworld.com/article/3014523/tobii-eyex-review-the-eye-mouse-is-magical-but-just-not-for-everyone.html>



Fonte:

https://www.researchgate.net/publication/334988473_Haptic_Feedback_to_Gaze_Events

Introdução

Aplicações de um *Eye Controller*

- *Jogos;*
- *Realidade virtual;*
- *Tecnologias assistivas;*

Introdução

Aplicações de um *Eye Controller*



Fonte: <https://www.mdpi.com/1424-8220/19/16/3630/xml>



Fonte: <https://www.youtube.com/watch?v=rp4zHlhm0L0>

Introdução

Problema

- Dispositivos caros;
- Podem ser desconfortáveis;
- Uma alternativa é o uso de câmeras comuns, como uma *webcam*.



Fonte:

https://www.researchgate.net/publication/334988473_Haptic_Feedback_to_Gaze_Events

Introdução

Objetivo

- Rastreamento de Olhar por meio de uma *webcam*;
- Simular uma entrada de dados;
- Interfaces adaptáveis;

Ferramentas Utilizadas

Ferramentas Utilizadas

Python e Bibliotecas

- *Python*;
- *OpenCV*;
- *Dlib*;
- *PyQt e QTDesigner*.

Desenvolvimento

Desenvolvimento

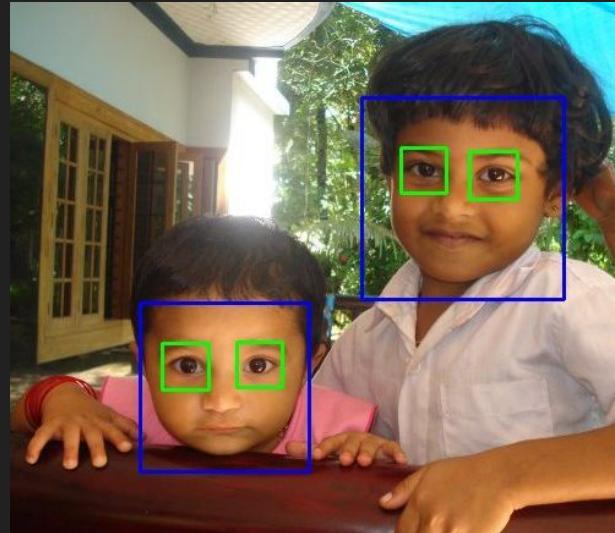
Processos do Rastreamento de Olhar

1. Reconhecimento facial;
2. Identificar localização dos olhos;
3. Identificar localização da pupila;
4. Estimar a direção do olhar.

Desenvolvimento

1. Reconhecimento Facial

- Limitar o local de busca pelos olhos;
- Baixo custo computacional;
- Algoritmo *Viola-Jones*;



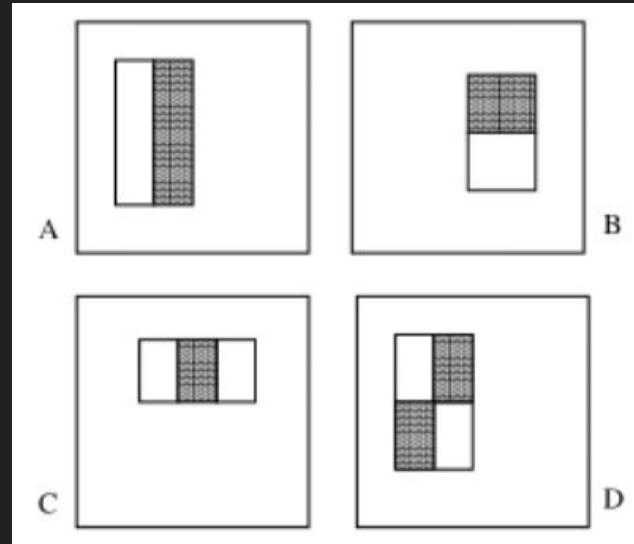
Fonte:

https://docs.opencv.org/3.4/db/d28/tutorial_cascade_classifier.html

Desenvolvimento

Algoritmo Viola-Jones

- Classificador implementado na *OpenCV*;
- Comparação de regiões retangulares dentro da própria imagem buscando padrões.



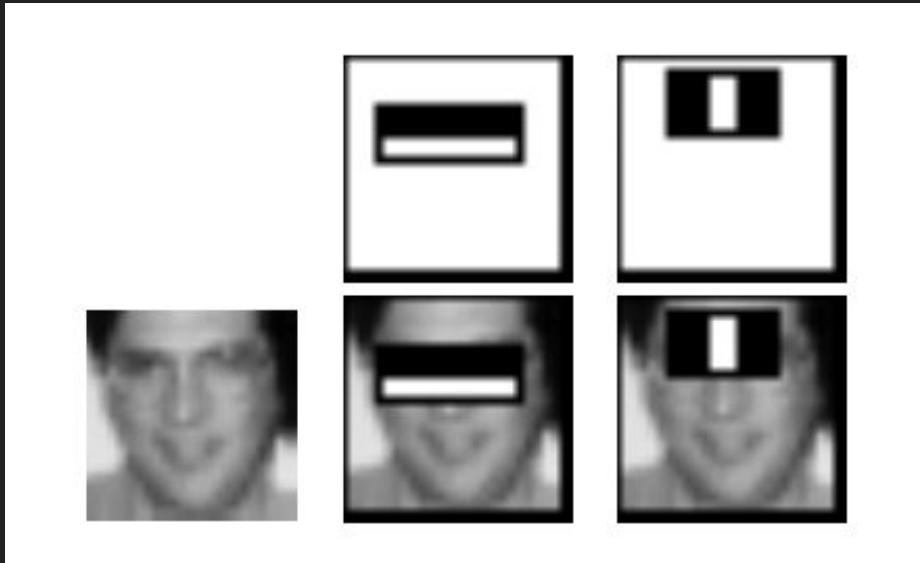
Fonte:

https://www.researchgate.net/publication/3940582_Rapid_Object_Detection_using_a_Boosted_Cascade_of_Simple_Features

Desenvolvimento

Feições *Haar-Like*

- Similaridades e padrões no rosto humano.



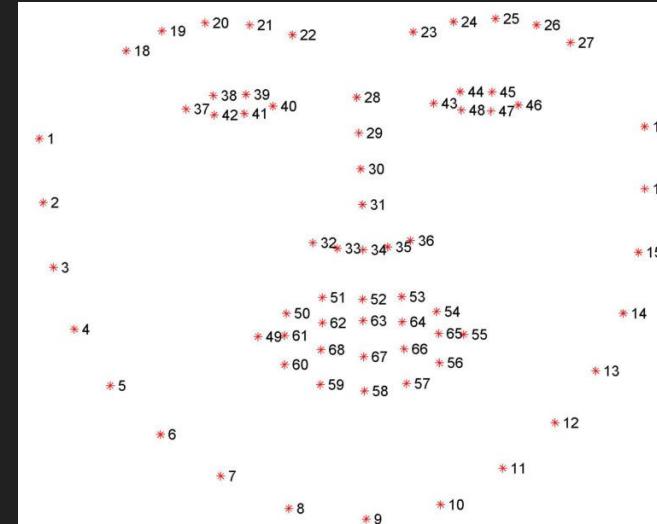
Fonte:

https://www.researchgate.net/publication/3940582_Rapid_Object_Detection_using_a_Boosted_Cascade_of_Simple_Features

Desenvolvimento

2. Identificar localização dos olhos

- Predictor;
- Modelo de 68 pontos - *Dlib*;
- Restringir área de busca.
- Corte da região da imagem utilizando as *landmarks*.

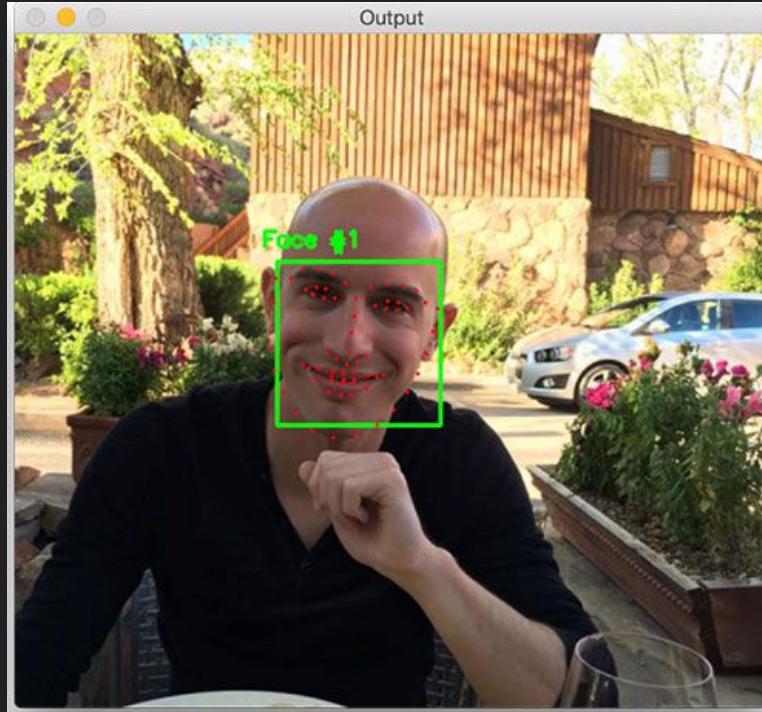


Fonte:

<https://www.pyimagesearch.com/2017/04/03/facial-landmarks-dlib-opencv-python/>

Desenvolvimento

2. Identificar localização dos olhos

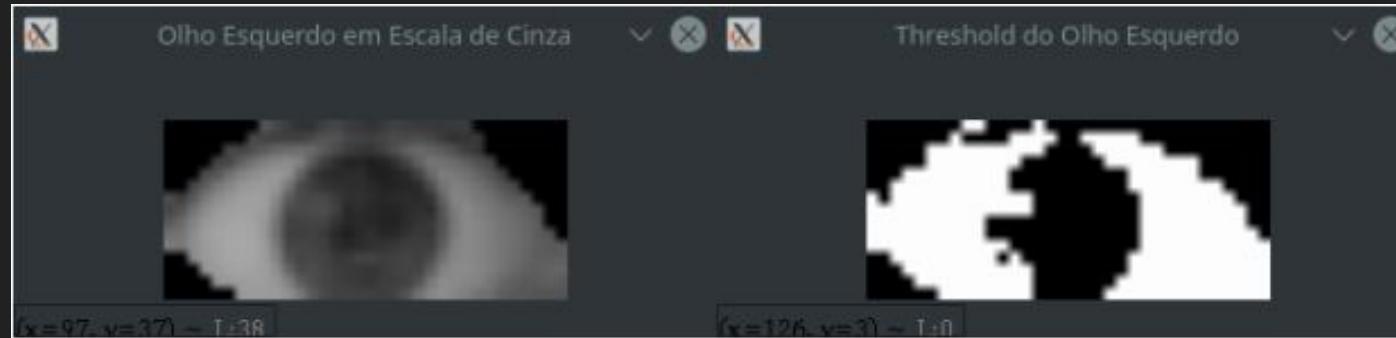


Fonte:

<https://www.pyimagesearch.com/2017/04/03/facial-landmarks-dlib-opencv-python/>

3. Identificar Localização da Pupila

- 3.1 Binarização
 - Feita através de limiarização
 - Limite binário para os valores da imagem.



Fonte: Elaborada pelo Autor.

4. Estimar a Direção do Olhar

- Dividir o olho ao meio verticalmente;
- “*Pixels*” diferentes de zero;
- Razão entre as regiões;
- Comparação com um valores definidos empiricamente.

Desenvolvimento

Avaliação da Direção do Olhar



Fonte: Elaborada pelo Autor.

Desenvolvimento

Piscadas

- Possível forma de entrada de dados.



Fonte: Elaborada pelo Autor.



Fonte: Elaborada pelo Autor.

Desenvolvimento

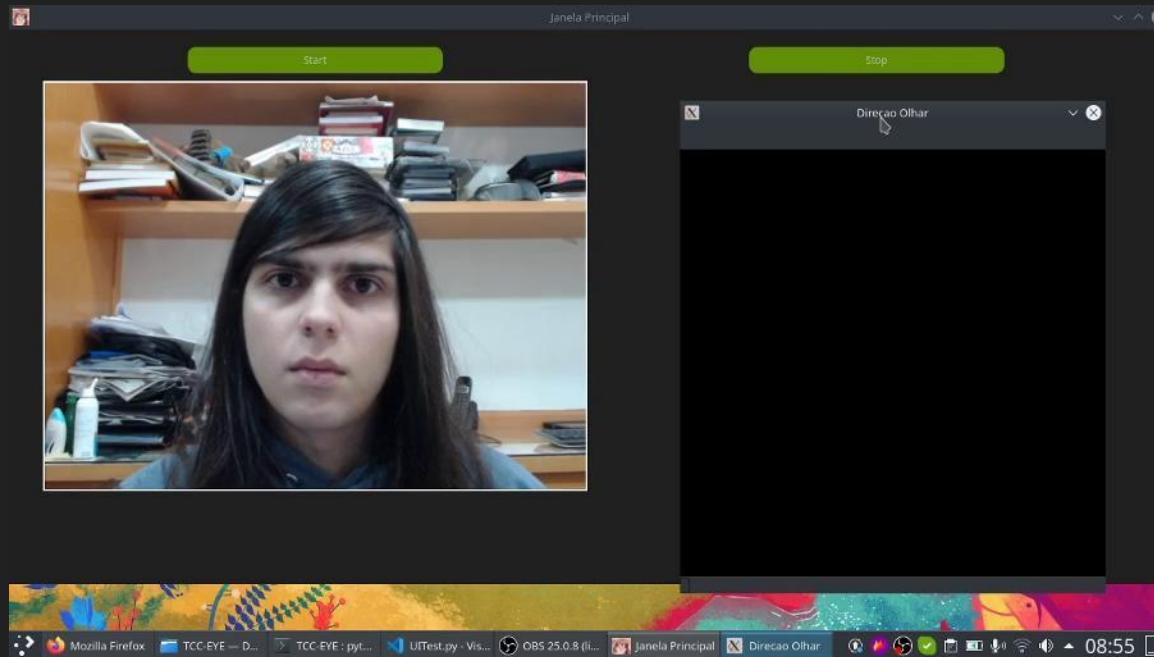
Piscadas - Vídeo



Fonte: Elaborada pelo Autor.

Desenvolvimento

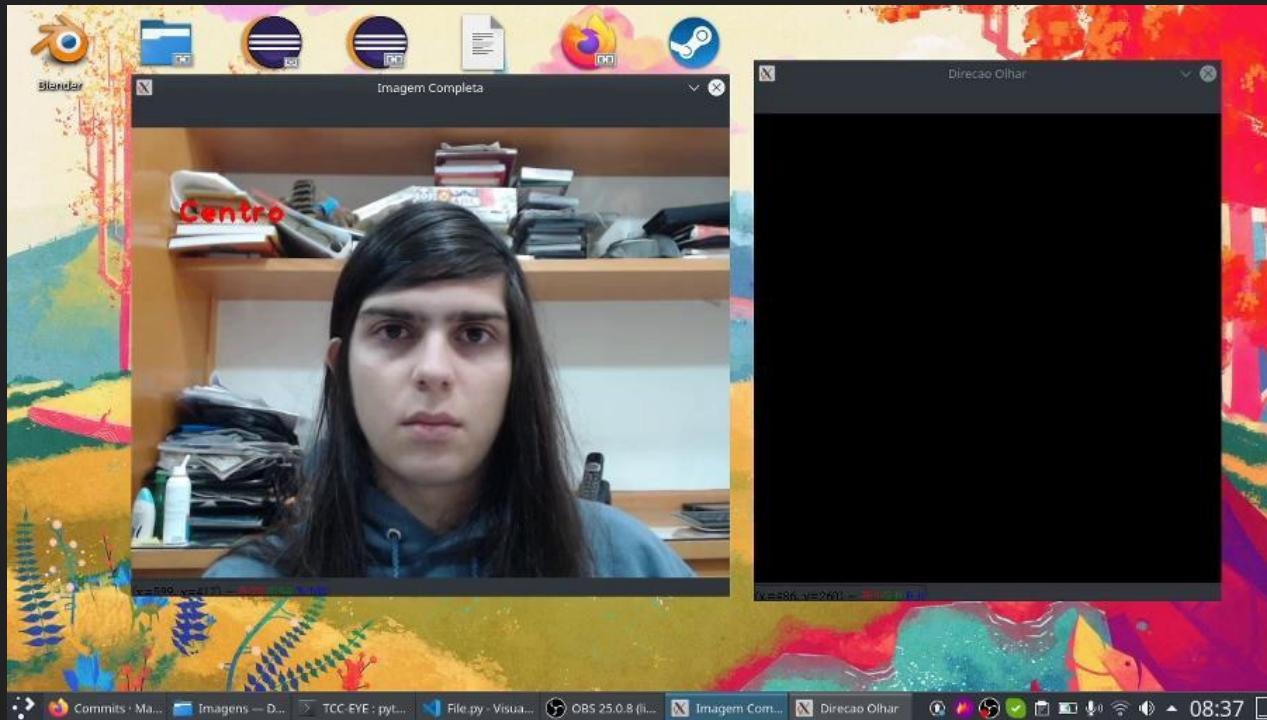
Interface Base



Fonte: Elaborada pelo Autor.

Desenvolvimento

Interface Simples



Fonte: Elaborada pelo Autor.

Conclusão

Conclusão

- Sistema funciona satisfatoriamente para o rastreamento da direção do olhar;
- Prova o conceito de que é possível rastrear a pupila sem equipamentos intrusivos;
- Os conceitos estudados já dão uma boa base para poder expandir a detecção para outras direções.

Conclusão

- Número reduzido de possíveis entrada de dados
- Movimentação pode não ser tão confortável para o usuário
- Face precisa estar sem oclusão e de frente para a *webcam*;

Conclusão

- Mesmo a detecção apenas das direções esquerda e direita já é útil, como em jogos mais simples;
- O rastreamento de esquerda e direita pode ser associado a outras funções, não apenas as direcionais:
 - Ex.: Jogo do dinossauro do Chrome
 - Esquerda = abaixar, Direita = saltar
- Sistema é portável para celulares, facilitando o acesso a ele.

Conclusão

Trabalhos Futuros

- Utilização de outras abordagens;
- Reconhecer mais eixos do movimento dos olhos além do horizontal;
- Permitir um processo de calibração integrado ao invés de usar valores empíricos;
- Integração com interfaces customizáveis.