

# Aplicação Delphi com OpenCV para Reconhecimento de Rostos em *Streaming*

Paulo Victor de Aguiar<sup>1</sup>, Carlos Augusto Silveira de Oliveira<sup>2</sup>, Jorge Manuel Lages Fernandes<sup>3</sup>

<sup>1</sup>Departamento de Ciência da Computação – Universidade do Estado de Santa Catarina (UDESC)  
89.219-710 -- Joinville -- SC -- Brazil

<sup>2</sup>Bacharel Design Digital – Universidade Anhembi Morumbi  
04.705-000 -- São Paulo -- SP -- Brazil

pavaguiar@gmail.com, cardevisi@gmail.com, jorge@youngarts.com.br

**Abstract.** *The objective of this work is to propose the use of face recognition technologies in television and broadcast media, automating operational tasks during a live video streaming. Building a low cost prototype but with high performance using open source technologies. For this is use the OpenCV library and the Delphi programming language, so that in future versions, the prototype will be update to an application that can focus on the speaker, switch the camera plane, apply subtitles and other content automatically.*

**Resumo.** *O objetivo deste trabalho é propor o uso de tecnologias de reconhecimento facial no meio televisivo e broadcast, automatizando tarefas operacionais durante uma transmissão ao vivo em streaming de vídeo. Construindo um protótipo de baixo custo, mas de alta performance, fazendo uso de tecnologias open-source. Para isto é utilizada a biblioteca OpenCV e a linguagem de programação Delphi, para que em versões futuras, o protótipo evolua para uma aplicação capaz de dar foco no falante, alterar o plano de câmera, aplicar legendas e outros conteúdos de forma automatizada.*

**Palavras-Chave:** *OpenCV, Reconhecimento Facial, Delphi, Broadcast*

## 1. Introdução

Nos últimos anos uma grande transformação digital de empresas e negócios vem contribuindo para a automatização das mais variadas atividades em todos os setores da economia, dentre eles os meios de comunicação. Com o aumento dos negócios digitais e por consequência o aumento da concorrência, torna-se necessário otimizar processos e rotinas, com fins de redução de custos e menor tempo de resposta para o mercado consumidor. Nesse sentido, uma área onde a automação pode trazer grandes benefícios é a mídia televisiva. Os avanços do *Deep Learning* (DL) nos últimos anos têm contribuído significativamente para a mudança de vários processos nesta área, principalmente em tarefas relacionadas a processamento de fala, audição e visão computacional.

Na contramão dessa transformação, muitos serviços multimídia usados no Brasil ainda estão focados apenas no uso tradicional das mídias (captura, armazenamento, transmissão e apresentação). Muitas emissoras de televisão realizam transmissões ao vivo, como debates, reportagens, festivais e outros tipos de programação, mantendo diversos processos manuais, como dar foco no rosto do falante ou colocar uma legenda. Esses processos acabam sendo repetitivos e ocupam considerável tempo dos

transmissores além do custo operacional. Já existem tecnologias para reconhecimento facial capazes de automatizar todo esse processo. Este trabalho propõe o desenvolvimento de um protótipo de reconhecimento facial para *Streaming* de vídeos com a finalidade de identificação dos falantes, para que em versões futuras, seja possível automatizar o foco no falante a troca de plano de câmera, aplicação de legendas e outros conteúdos de forma automática. Neste artigo é apresentado o **FaceDetect**<sup>1</sup>, uma solução de código aberto, desenvolvida em Delphi 10 e OpenCV (Open Source Computer Vision Library, biblioteca de visão computacional e machine learning open-source). O restante do artigo está organizado como segue. A seção 3 apresenta o algoritmo de reconhecimento facial por camadas Haar Cascade, algoritmo utilizado no sistema. A seção 4 apresenta o protótipo e as tecnologias utilizadas no desenvolvimento da aplicação. A seção 5 apresenta os testes e resultados obtidos. Por fim, na Seção 6 as considerações finais.

## 2. Trabalhos Relacionados

Durante o levantamento bibliográfico, foram levantados diversos trabalhos com diferentes exemplos de aplicação com OpenCV e técnicas de reconhecimento facial. Existem uma grande variedade de aplicativos que utilizam as tecnologias mencionadas, cada uma com suas particularidades que contribuíram para o desenvolvimento do trabalho.

Em [Viola et al. 2001] é apresentada uma abordagem baseada em aprendizado de máquina em que uma função cascata é treinada a partir de muitas imagens positivas e negativas (onde imagens positivas são aquelas em que o objeto a ser detectado está presente, negativas são aquelas em que não estão). Em outras palavras as imagens positivas seriam imagens de rostos de pessoas e as imagens negativas seriam imagens de *background* ou paisagens, imagens sem rostos de pessoas. O OpenCV oferece algoritmos de cascata Haar pré-treinados, organizados em categorias (rostos, olhos e assim por diante), dependendo das imagens em que foram treinados.

Em [Maia 2016] é proposto um modelo de reconhecimento facial utilizando redes neurais artificiais, uma das muitas técnicas de aprendizado de máquina. Inspirado nos neurônios, uma rede neural recebe várias entradas como parâmetro e gera uma única saída. No trabalho de Maia, são passados diversos rostos nos quais passam por um processo de percepção, onde é analisado se são da mesma pessoa ou não.

Em [Silva et al. 2012] é utilizada a biblioteca OpenCV para a implementação de uma ferramenta educativa para processamento de imagens. A biblioteca é usada para ter uma visão computacional sobre uma imagem ou vídeo. Com o uso de uma interface interativa, são feitas as ações de processamento, como alteração de brilho ou contraste.

## 3. Reconhecimento facial com OpenCV e Haar Cascade

Nesta seção, é apresentado o método *Haar Cascade*, um algoritmo de detecção de objetos de aprendizado de máquina proposto em [Viola et al. 2001]. A abordagem se baseia em aprendizado de máquina em que uma função cascata é treinada a partir de muitas imagens positivas e negativas (onde imagens positivas são aquelas em que o objeto a ser detectado está presente, negativas são aquelas em que não estão). Em outras palavras as imagens positivas seriam imagens de rostos de pessoas e as imagens negativas seriam imagens de *background* ou paisagens, imagens sem rostos de pessoas. O OpenCV oferece algoritmos

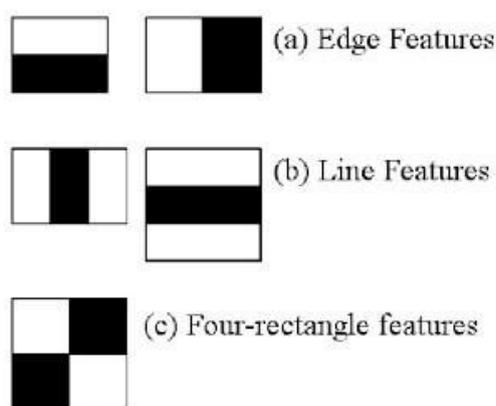
---

<sup>1</sup> <https://github.com/jmfjorge/FaceDetect>

de *Haas Cascade* pré-treinados, organizados em categorias (rostos, olhos e assim por diante), dependendo das imagens em que foram treinados.

No trabalho sobre detecção rápida de objetos citado anteriormente, os autores citam três de suas principais contribuições para com o meio científico. A primeira é a introdução de uma representação de imagem chamada *Integral Image*. A segunda é um classificador simples e eficiente, construído usando o algoritmo de aprendizado *AdaBoost*. E a terceira, um método para combinar classificadores em uma cascata. Nas próximas seções será mais aprofundado esses conceitos.

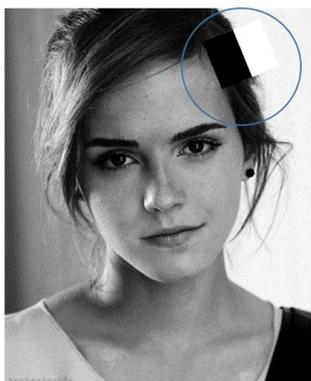
A ideia por trás do *Haas Cascade* é extrair *features* a partir de imagens usando um tipo de filtro, similar ao conceito de convolucional kernel. Estes filtros são chamados *Haar features* e são ilustrados conforme a figura 1.



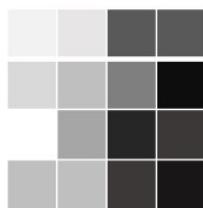
**Figura 1. Filtros Haar features.**

A ideia é passar esses filtros sobre a imagem, em uma espécie de janela imaginária, uma janela por vez ou uma pequena parte por vez da imagem. Então, para cada janela, todas as intensidades de pixels, respectivamente, as porções de pixels brancos e pretos são somadas. Finalmente, o valor obtido da subtração destes dois somatórios é o valor da *feature* extraída. Um valor grande de *feature* significa, que essa *feature* é relevante. Para exemplificar o processo de extração de *features* consideremos o exemplo da figura 2.

Se considerar o Edge feature (a) e aplicá-lo na imagem preto e branco anterior, obteremos um valor significativo, portanto o algoritmo retornará um edge feature com alta probabilidade. É claro que as intensidades reais dos pixels de uma imagem nunca são iguais a branco ou preto, e muitas vezes irão parecer com algo como na figura 3. Entretanto a ideia se mantém, ou seja, quanto maior a diferença da somatória dos pretos e dos brancos, maior a probabilidade dessa janela ser uma característica relevante.



**Figura 2. Filtros Haar features.**



**Figura 3. Exemplo mais realista dos pixels de uma imagem Preto&Branco.**

Com base no que foi visto anteriormente dá para imaginar a quantidade de *features* retornadas por este cálculo. Segundo [Alto] mesmo uma janela 24x24 resulta em torno de 160000 *features*, e dependendo do tamanho da imagem esse número varia bastante. Para tornar esse processo mais eficiente [Viola et al. 2004] introduziram a representação de imagem *Integral Image*, que permite calcular as *features* usadas pelo detector muito rapidamente. Essa representação é conceitualmente conhecida como *Summed-area table* ou tabela de área somada. É uma estrutura de dados e um algoritmo para calcular a soma de valores em um subconjunto retangular de um grid. Tem por objetivo reduzir o número de cálculos computacionais necessários para se obter a somatória de intensidade dos pixels dentro de uma janela.

O próximo passo do algoritmo Haar Cascade, também envolve eficiência e otimização. Além de numerosas, as *features* encontradas também podem ser irrelevantes. Entre as características obtidas no caso anterior, mais de 160000, como decidir quais são realmente boas. A resposta para isso se baseia no conceito de *Ensemblig method*, combinando muitos algoritmos fracos por definição, podemos criar um algoritmo forte. Aqui entra a segunda contribuição de [Viola et al. 2004] que foi o algoritmo de aprendizado baseado no *AdaBoost*, que seleciona um pequeno número de recursos visuais críticos a partir de um conjunto maior e gera classificadores extremamente eficientes. Esse algoritmo constrói um classificador “forte” como uma combinação linear de classificadores “fracos” simples ponderados.

O terceiro passo, último importante conceito introduzido por [Viola et al. 2004], também relacionado a otimização mas em termos de tempo de treinamento é um método para combinar classificadores em uma cascata, que permite que as regiões de fundo da imagem sejam rapidamente descartadas, assim focando mais na aparência facial. A aplicação de todos os recursos em todas as janelas leva muito tempo. Por esta razão foi proposto o conceito de cascata de classificadores: em vez de aplicar todas as *features* em uma janela, ele agrupa as *features* em diferentes estágios de classificadores e aplica uma

a uma. Se uma janela falha, a diferença entre os brancos e pretos é baixa no primeiro estágio, que normalmente inclui poucas *features*, o algoritmo a descarta e não considera as *features* restantes. Já se a *feature* é aprovada, valor do somatório alto, o algoritmo aplica o segundo estágio de *features* e continua o processo.

#### 4. Protótipo

O protótipo foi construído usando a linguagem de programação Delphi 10, tomando por base um projeto *open-source*<sup>2</sup> que utiliza a biblioteca OpenCV. Nos primeiros testes realizados foi utilizada uma *webcam*, mas em testes futuros, em um cenário mais real, a ideia é utilizar câmeras profissionais de transmissão ao vivo para capturar e reconhecer o rosto conforme ilustrado na figura 4.

O primeiro passo de uso do protótipo é fazer o reconhecimento facial a partir da webcam, utilizando o pré-treinamento de rostos disponíveis na aplicação. Feito o reconhecimento, a aplicação gera automaticamente uma foto da pessoa e a posiciona no canto superior direito para visualização. Nesse momento é possível colocar uma legenda na foto, identificando a pessoa recém capturada. Essas informações são salvas no banco de dados do protótipo como aprendizado do sistema, para que posteriormente ele possa reconhecer automaticamente o rosto do indivíduo.

Uma vez treinado o protótipo, o rosto da pessoa será destacado e legendado conforme as informações gravadas no sistema. Esse processo deve ser repetido para cada novo perfil de pessoa que se deseja reconhecer automaticamente através da aplicação.

A escolha das tecnologias utilizadas no projeto foram baseadas em critérios como, baixo custo, familiaridade e performance. O Delphi foi escolhido por permitir o desenvolvimento de aplicações multiplataforma, de alto desempenho com possibilidade de reuso de componentes de terceiros e por sua facilidade no desenvolvimento de interfaces. Atualmente o Delphi se encontra na versão 10.3, cuja a Embarcadero Technologies realiza correções e inovações todos os anos. A biblioteca OpenCV foi escolhida por ser de código aberto conforme mencionado na seção 1. Essas tecnologias são facilmente integráveis, com curva de aprendizado médio.

Na próxima seção é discutido os resultados obtidos, utilizando como critérios de avaliação e indicador de desempenho, a assertividade do reconhecimento. Dado o conjunto de rostos salvos no banco de dados, após o pré-treino, medindo quantas vezes o protótipo foi capaz de reconhecer o rosto de um falante, gerando um percentual em relação a quantidade total de reconhecimentos.

---

<sup>2</sup> <https://sourceforge.net/projects/delphi-opencv-face-detection/>

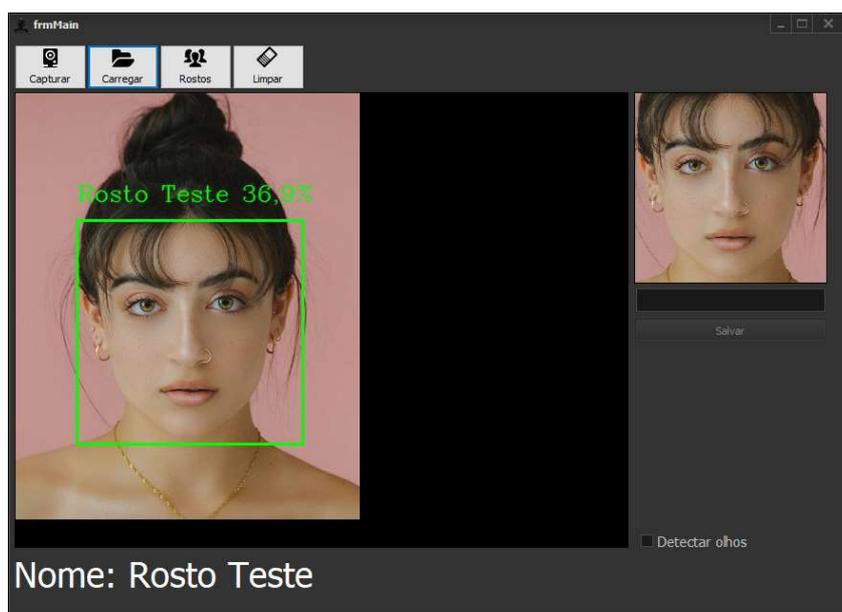


Figura 4. Protótipo do FaceDetect em Execução.

## 5. Resultados Obtidos

O protótipo foi testado com 4 voluntários da faixa etária entre 20 a 40 anos. Em cada 10 testes realizados, foram reconhecidos entre 7 a 9 rostos. Entretanto, se o rosto da pessoa estiver muito inclinado o algoritmo de reconhecimento perde acurácia, e o rosto acaba não sendo reconhecido. Em uma nova sequência de 10 testes, mas agora considerando inclinações variadas, o desempenho do algoritmo caiu consideravelmente, reconhecendo no máximo 5 rostos.

Os resultados são apresentados graficamente na tabela 1. A primeira coluna, refere-se aos voluntários que participaram dos testes. A segunda coluna, mostra os resultados obtidos na primeira sequência de 10 testes sem inclinação. A terceira coluna, mostra os resultados da sequência de 10 testes considerando a inclinação.

Voluntários	Acertos sem inclinação	Acertos com inclinação
1)	7	3
2)	8	2
3)	9	5
4)	8	3

Tabela 1. Resultados dos testes.

Como resultado final, entende-se que a aplicação ainda é bem limitada e que muitos testes e refinamentos ainda serão necessários para que o reconhecimento facial seja mais constante e preciso. Somente após esse refinamento é que será possível avançar para outras funcionalidades, como o foco automático na inclusão de legendas e outros recursos que foram comentados ao longo do trabalho.

## 6. Conclusão

Este trabalho apresenta uma aplicação de reconhecimento facial de rostos em streaming de vídeo, ainda em fase de protótipo, para ser utilizada em transmissões de vídeo ao vivo.

A intenção é contribuir com áreas multimídia, televisiva e broadcast, possibilitando a automatização de processos ainda feitos manualmente, como a aplicação de legendas, troca de plano de câmera e outras tarefas, com fins de manter essas áreas competitivas mesmo diante das fortes transformações que esse mercado vem sofrendo.

Apresentando o FaceDetect uma aplicação de reconhecimento facial desenvolvida em Delphi em conjunto com a biblioteca open-source de visão computacional OpenCV. Ainda apresentado os conceitos por trás do Haar Cascade, algoritmo para detecção de objetos de alta performance, que trouxe grande contribuição para o avanço e uso do reconhecimento facial em larga escala.

Ainda há um grande trabalho para transformar em realidade a aplicação de reconhecimento facial em *streaming* de vídeo durante uma transmissão ao vivo. O algoritmo se comporta bem em ambientes mais controlados, mas deixa muito a desejar em situações inesperadas, como a movimentação involuntária do objeto ou pessoa a ser detectado. Muitos testes, refinamentos e treinamentos ainda serão necessários para se obter uma acurácia maior.

## Referências

- Alto, V. Face recognition with opencv: Haar cascade. <https://medium.com/dataseries/face-recognition-with-opencv-haar-cascade-a289b6ff042a>.
- Maia, H. L. F. (2016). Detecção e reconhecimento facial por meio de aprendizado de máquina.
- Silva, M. A. M., Chaves, A. P. A., & Aquino, F. J. A. (2012). Projeto e desenvolvimento de uma ferramenta educativa para ensino de processamento de imagens baseado na biblioteca OpenCV. In *XL Congresso Brasileiro de Educação em Engenharia, Belém*.
- Viola, P., & Jones, M. (2001, December). Rapid object detection using a boosted cascade of simple features. In *Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001* (Vol. 1, pp. I-I). IEEE.
- Viola, P., & Jones, M. J. (2004). Robust real-time face detection. *International journal of computer vision*, 57(2), 137-154.