

**FUNDAÇÃO CENTRO DE ANÁLISE, PESQUISA E INOVAÇÃO TECNOLÓGICA.
INSTITUTO DE ENSINO SUPERIOR FUCAPI
COORDENAÇÃO DE GRADUAÇÃO EM
CIÊNCIA DA COMPUTAÇÃO**

SISTEMA DE RECONHECIMENTO FACIAL UTILIZANDO INTELIGÊNCIA ARTIFICIAL

EDSON DE SOUZA JR

**MANAUS
2016**

EDSON DE SOUZA JR

SISTEMA DE RECONHECIMENTO FACIAL UTILIZANDO INTELIGÊNCIA ARTIFICIAL

Projeto de Inteligência Artificial no curso de
Ciência da Computação do Instituto de Ensino
Superior FUCAPI – CESF.

**MANAUS
2016**

RESUMO

O presente trabalho visa descrever a elaboração de um sistema capaz de reconhecer imagens, em especial a face de pessoas, identificando - a, culminando assim para o chamado reconhecimento facial. Para tanto será abordado o treinamento de um padrão de rosto, gerando um classificador que poderá ser posteriormente utilizado em qualquer outro sistema capaz de ler e utilizar tal classificador. O trabalho também abordará de forma simplificada a acurácia deste classificador.

SUMÁRIO

1.Introdução.....	5
2.Descrição do Sistema.....	6
2.1. Descrição básica do Sistema e Problemas.....	6
2.2. Algoritmos de Aprendizagem e Ferramentas Usadas.....	6
2.2.1. Opencv.....	6
2.2.2. Princípio do Algoritmo de Haar Cascade.....	7
2.2.3. Princípio Básico da Detecção de Face Usando Haar Cascade.....	8
2.2.4. Boosting/Imagens positivas e negativas.....	8
3. O protótipo.....	8
3.1. Descrição da Arquitetura do Sistema.....	9
3.2. Descrição dos Algoritmos que Compõem o Sistema.....	10
3.2.1. Algoritmo de Detecção Facial.....	10
3.2.2. Algoritmo de Checagem de Padrão - Reconhecimento Facial.....	11
3.2.3. Algoritmo de Replicação de Imagem.....	12
3.3. O sistema em Uso.....	12
3.3.1. Uso do Algoritmo de Replicação de Imagens.....	12
3.3.2. Opencv Traincascade.....	13
3.3.3. Algoritmo de Reconhecimento Facial.....	14
4. Testes.....	14
5. Conclusão.....	16
6. Referências.....	18

1.Introdução

Os processamento digital de imagens é uma área que vem evoluindo continuamente ao longo dos anos, com o aumento significativo de estudos envolvendo morfologia matemática, redes neurais, processamento de imagens, compressão de imagens, reconhecimento de imagens e sistemas de análise de imagens baseados em conhecimento. Não obstante, dentro do processamento digital de imagens é possível haver o reconhecimento de padrões, sendo esta uma área da ciência cujo o objetivo é a classificação de objetos através de categorias ou classes.

O rosto humano, apesar das variações existentes de pessoas para pessoa, possui uma composição básica, um padrão que não se altera. Esse padrão é interpretado por algoritmos a fim de verificar a existência de uma face ou várias na imagem que é passada como entrada, verificando ponto a ponto do rosto a fim de encontrar algo que se relaciona com o padrão de uma face, sendo que tal padrão “está embutido” no próprio algoritmo ou em algum classificador[15]. O reconhecimento de face vem ganhando cada vez mais espaço com passar do tempo, sendo utilizado hoje em dia como verificação de login, sistema de reconhecimento pela polícia e forças de inteligência, detecção de conteúdo explícito, averiguação de semelhanças entre indivíduos, etc [15]. É importante também enfatizar que existem diferenças entre detecção facial e reconhecimento facial, sendo a detecção o método de verificar a existência de rostos em uma foto, enquanto que o reconhecimento propõe - se em identificar rostos específicos [18].

Os primeiros trabalhos de reconhecimento facial começaram há muito tempo, tendo seu surgimento datado entre os anos 1960 e 1970, sendo que somente nos anos 70 é que se propuseram sistemas automatizados de reconhecimento, tendo desde então várias linhas de pesquisa a cerca do assunto, apresentando soluções cada vez mais robustas[16]. Porém, apesar do grande avanço tecnológico em relação aos sistemas de reconhecimento e detecção facial, ainda existem algumas barreiras com relação à detecção, principalmente em condições adversas, tais como por exemplo, iluminação inadequada, posição da face de perfil e não frontal, imagens tremidas ou borradas, entre outras. Tais condições podem inferir significativamente na detecção e reconhecimento do padrão, aumentando assim as taxas de erro[16][19]. De fato, é possível dizer que apesar do grande desenvolvimento na área, ainda sim não existem algoritmos que consigam detectar e reconhecer faces humanas com tanta precisão quanto uma pessoa em si é capaz de fazer, entretanto, dentro de ambientes controlados é possível se obter uma grande acurácia [19].

Diante das vastas possibilidades existentes que permeiam o reconhecimento de padrões, em especial o reconhecimento facial, o presente trabalho visa elaborar e descrever um sistema simplista de reconhecimento facial, utilizando como base o Opencv e o algoritmo de Haar Cascade, ambos amplamente utilizados na área de reconhecimento de padrões, com o intuito de realizar um teste de acurácia.

O documento está dividido em varias seções, a saber: seção 2: breve descrição do projeto e seu funcionamento incluindo uma breve descrição dos algoritmos utilizados; seção3: descrições de prototipação do sistema e descrições de funcionamento do mesmo; seção 4: descrição de testes realizados e tabulação dos resultados; seção 5: resultados obtidos.

2.Descrição do Sistema

Nesta seção serão abordados os conceitos básicos a respeito das ferramentas e algoritmos usados na implementação da solução.

2.1. Descrição básica do Sistema e Problema

De uma forma sucinta e direta, o sistema em si será responsável por ler uma imagem de entrada e identificar um padrão de rosto específico e, caso este padrão seja reconhecido, o sistema então irá apresentar a foto com o rosto marcado por um círculo, caso contrário, havendo falhas ou o sistema não sendo capaz de identificar o padrão de rosto, o mesmo simplesmente concluirá sua execução. A figura 1 demonstra o processo de execução.

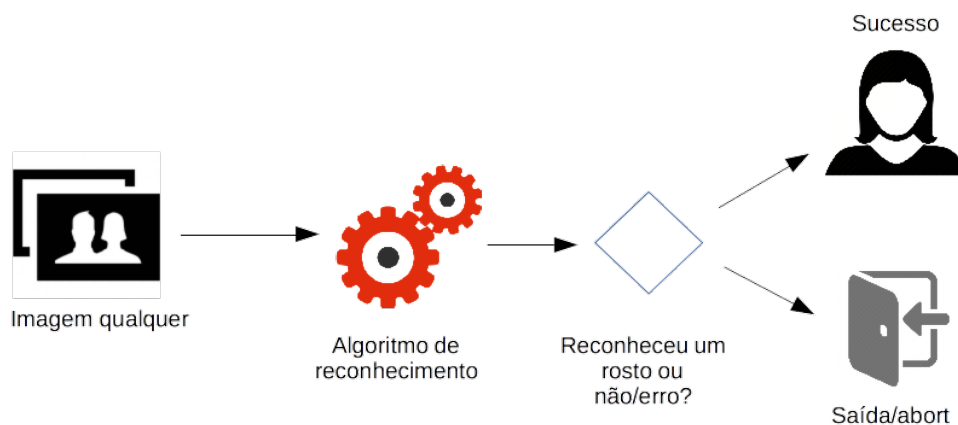


Figura 1. Esquema básico de funcionamento do algoritmo de detecção facial.

Como mencionado na seção de introdução, o reconhecimento de um rosto em uma imagem pode ser afetado por diversos fatores que podem acarretar em falhas de identificação, no entanto, em ambientes controlados, o reconhecimento facial tende a ter altas taxas de acerto. O presente trabalho visa treinar um classificador para reconhecer um tipo de rosto em específico, descrevendo os algoritmos usados e realizando testes a fim de verificar o poder de classificação do modelo treinado.

2.2. Algoritmos de Aprendizagem e Ferramentas Usadas

Nesta seção serão descritos de forma rápida e sucinta as ferramentas e algoritmos de aprendizagem utilizados no desenvolvimento deste trabalho.

2.2.1. Opencv

Em suma para o projeto será utilizado a biblioteca Opencv, tanto para a captura e tratamento de imagens quanto para o treinamento e detecção de padrões. O Opencv foi

inicialmente desenvolvido pela Intel como uma biblioteca de processamento digital de imagens em tempo real, em volta dos anos 2000, sob licença BSD [9]. Originalmente desenvolvido em linguagem C, teve sua mudança para a linguagem C++ a partir da versão 2.0 e hoje em dia apresenta versões e wrappers para as mais diversas linguagens, como Java, Python, MATLAB/Octave, C#, entre outras [9], além de suporte a várias plataformas, como Windows, Linux, MacOS, Android e mais recentemente iOS. As aplicações pelas quais o OpenCV pode ser utilizado englobam segmentação e reconhecimento de imagens, ferramentas 2D e 3D, reconhecimento e detecção facial, acompanhamento de movimentos, reconhecimento de gestos, HDR (high dynamic range) e realidade aumentada [9].

O OpenCV já vem com suporte ao reconhecimento de alguns padrões, como por exemplo faces, boca, nariz, olhos e corpos humanos. Não obstante, ele também vem com suporte para treinar o reconhecimento de algum padrão específico que se queira [9][10], utilizando em específico o algoritmo de Haar cascade.

2.2.2. Princípio do Algoritmo de Haar Cascade

Até os anos 2000, haviam várias técnicas distintas para procurar padrões em imagens, principalmente no que se tratava de rostos. No entanto, tais técnicas eram tidas como imprecisas ou demoradas demais. Em 2001 Paul Viola e Michael Jones desenvolveram uma técnica de classificador de objetos, conhecido classificador baseado em Haar Cascade, sendo posteriormente melhorado em 2002 por Lienhart e Maydt, constituindo assim um mecanismo de reconhecimento bastante preciso e rápido, com cerca de 95% de acurácia, podendo ser utilizado inclusive em reconhecimento de padrões em tempo real [10].

Um classificador Cascade nada mais é do que um conjunto de classificadores fracos que em conjunto formam um classificador forte [10]. O classificador basicamente consiste em um número de estágios, onde cada estágio nada mais é que um “reconhecedor” fraco, sendo que tal apresenta uma performance muito limitada e é incapaz de reconhecer algum padrão de forma satisfatória sozinho. Cada estágio é treinado usando uma técnica de inteligência artificial chamada de boosting, sendo que tal fornece a capacidade de treinar um classificador altamente preciso baseado nos dados fornecidos pelos “reconhecedores” fracos [10][11]. Cada estágio do classificador “rotula” um região da imagem, sendo que tal rotulação ocorre pela posição atual do classificador dentro da imagem, distribuindo rótulos tidos como positivos ou negativos, onde o positivo indica que um padrão foi detectado na imagem e o negativo indica o inverso. O classificador termina quando todos os estágios são alcançados [11].

2.2.3. Princípio Básico da Detecção de Face Usando Haar Cascade

O algoritmo de classificador Haar Cascade, ao ser usado para reconhecimento facial, procura por características que determinam uma face humana, utilizando para isso um arquivo de classificador que contenham dentro de si tais características. Quando o algoritmo encontra uma dessas características, então o ele "passa" para o próximo estágio de detecção, permitindo assim procurar por padrões que representam a face. Caso naquele atual estágio não seja encontrado um padrão de face, o algoritmo então procura por características em outros estágios, sendo que somente será validado se todos os estágios forem alcançados com sucesso [12].

2.2.4. Boosting / Imagens positivas e negativas

Sistemas de reconhecimento, sejam eles de face ou de qualquer outro objeto, foram concebidos com a idéia de serem capazes de detectar objetos pelos quais foram treinados em qualquer tipo de imagem, verificando a existência ou não deste objeto. Há muitas técnicas de aprendizagem de máquina que podem ser usadas para criar um classificador, no entanto, muitas destas precisam de uma grande quantidade de imagens para serem capazes de classifica - las [10]. Ao treinar um classificador, é necessário não somente imagens que tenham o objeto a ser reconhecido, mas também imagens que não tenham este objeto. As imagens que contem o objeto a ser classificado são tidas como “amostras positivas”, enquanto que as imagens que não contém são conhecidas como “amostras negativas” [10].

Como mencionado nas seção 2.2.2, Boosting é uma técnica de inteligência artificial supervisionada, usada em problemas que envolvam classificação de elementos. Seu princípio de funcionamento basicamente consiste em combinar classificadores “fracos” para gerar um classificador “forte”, obtendo uma excelente “precisão” na sua classificação. É importante verificar que o algoritmo de Boosting foi sofrendo alterações ao longo do tempo, apresentando variações, sendo a mais comum o AdaBoosting (Adaptive Boosting) [13][14].

3. O prototipo

Nas seções a seguir serão apresentados os elementos que compõem a arquitetura da solução, descrevendo seus módulos e seus os princípios de funcionamento.

3.1. Descrição da Arquitetura do Sistema

Em suma, o sistema é contemplado por três algoritmos simplistas distintos. Por se tratar de reconhecimento facial, existe um algoritmo responsável por fazer captura de rostos e tratamento dessas imagens, salvando - as para o treinamento posteriormente.

Existe também um algoritmo responsável por ler um arquivo XML contendo a definição do treinamento realizado com as imagens geradas pelo primeiro algoritmo e verificar as correspondências em uma imagem passada para ele como entrada. Por fim, há um pequeno algoritmo responsável por replicar imagens, gerando “ruídos” nas mesmas, cuja finalidade é simplesmente aumentar a base de imagens para o treinamento. Todos os algoritmos implementados funcionam de forma independente, onde a saída de um não é necessariamente a entrada de outro, no entanto também foram utilizadas ferramentas do OpenCV, responsáveis por realizar o treinamento do padrão em si. A figura 2 descreve a arquitetura do sistema.

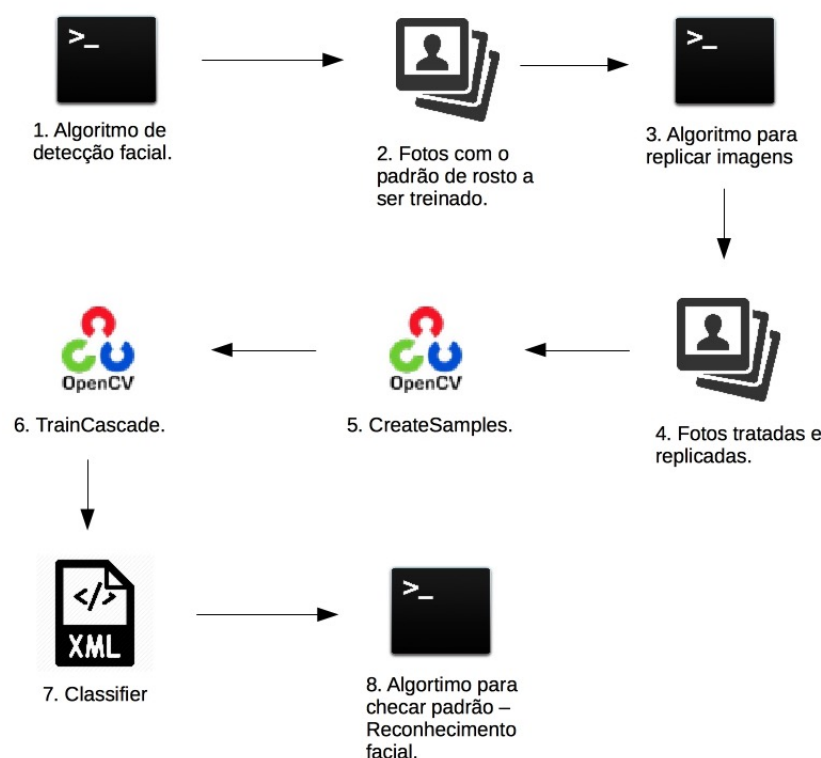


Figura 2. Esquema básico de funcionamento do algoritmo de detecção facial.

Em primeiro caso, é necessário detectar a face que se deseja reconhecer e para tanto utiliza - se o Algoritmo de Detecção Facial (figura 2 - 1), onde o rosto é identificado e recortado, gerando como saída fotos contendo somente a face desejada (figura 2 - 2). Para este trabalho, foi necessário haver a replicação de imagens, uma vez que a quantidade de imagens originais era tida como insuficiente, uma vez que para o reconhecimento facial é indicado a utilização de uma grande quantidade de imagens para treino [6], e para tanto usou - se o Algoritmo para Replicar Imagens (figura 2 - 3). Após a replicação das imagens, foi - se utilizada uma ferramenta do próprio OpenCV, o `opencv_createsamples`, cuja finalidade é criar um dataset que será utilizado posteriormente no treinamento[6] (figura 2 - 4 e 5). Com base no resultado do `opencv_createsamples` e das imagens tratadas, já é possível treinar o padrão de rosto a

ser reconhecido, utilizando a ferramenta `opencv_traincascade`, obtendo como saída um arquivo de classificador, no formato XML[6] (figura 2 - 6 e 7). Por fim, utiliza - se o Algoritmo de Reconhecimento Facial para testar o classificador gerado no treinamento (figura 2 - 8).

3.2. Descrição dos Algoritmos que Compõem o Sistema

A seguir será apresentado a arquitetura do sistema, contemplando o funcionamento de cada um dos algoritmos citados acima.

3.2.1. Algoritmo de Detecção Facial

O algoritmo de detecção facial é a primeira etapa da solução, pois é ele responsável por capturar o rosto que será usado no treinamento. É importante verificar que há uma diferença entre detecção facial e reconhecimento, sendo que a detecção refere - se somente a verificar se há rostos em uma imagem ou vídeo, enquanto o reconhecimento é responsável por detectar somente um tipo de rosto (ou tipos de rostos) dos quais foi treinado ou que estão em sua base de dados, sendo possível "dizer" quem está naquela imagem[1]. Neste primeiro momento não há ainda o reconhecimento facial, apenas detecção e, por ser a fase inicial para o treinamento, é importante que as imagens passadas a ele como entrada remetam o padrão de rosto a ser treinado. A figura 3 descreve os passos básicos deste algoritmo.

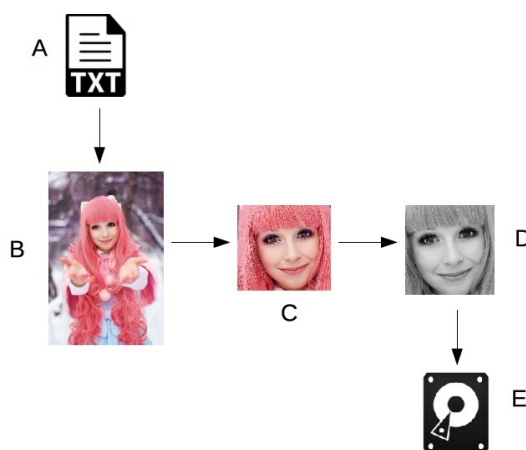


Figura 3. Esquema básico de funcionamento do algoritmo de detecção facial.

Primeiramente o algoritmo tem como entrada um arquivo texto contendo os caminhos para as imagens, cujo objetivo é tornar simples e conveniente o tratamento de uma grande quantidade de dados (figura 3 - A). Ao ler o caminho, o algoritmo procura por esta imagem no local indicado e, caso encontre, carrega - a (figura 3 - B). Após a carga da imagem, o algoritmo será responsável por detectar faces, utilizando para tal os próprios recursos de detecção facial do OpenCV, o que consiste em um classificador já pronto disponível [2], recortando o rosto detectado (figura 3 - C). Após o recorte do rosto a

imagem é colocada em escala de cinza, uma vez que a detecção facial funciona exclusivamente com imagens em escala de cinza [10] (figura 3 - D) e por fim salva em disco em um diretório passado como argumento para o algoritmo, além de permitir escolher uma resolução pré - definida para armazenar a imagem (figura 3 - E).

3.2.2. Algoritmo de Checagem de Padrão - Reconhecimento Facial

Este algoritmo é responsável por verificar se um determinado padrão de rosto é encontrado em uma imagem passada a ele como entrada. O algoritmo necessita de dois parâmetros de entrada, sendo o primeiro um arquivo XML que descreve o classificador e o segundo parâmetro uma imagem que pode ou não conter o rosto utilizado no treinamento.

O arquivo XML é gerado como saída do treinamento executado pelo `traincascade` do OpenCV [2], sendo responsável por fazer o reconhecimento do rosto de uma única pessoa. O arquivo de imagem passado pode conter ou não o rosto desta pessoa, sendo que se houver, o algoritmo abrirá uma janela mostrando o rosto da pessoa e o destacará com um círculo (figura 4 - A), caso contrário se não houver, simplesmente emitirá uma mensagem de erro e encerrará (figura 4- B).



Figura 4. Demonstração das imagens após aplicação de filtros.

3.2.3. Algoritmo de Replicação de Imagens

O algoritmo de replicação é responsável por basicamente ler uma imagem de entrada, aplicar um filtro de Blur[3] ou Bilateral na mesma e salvar uma determinada quantidade dessas imagens em disco. Como mencionado, o objetivo deste algoritmo é suprir a necessidade de haverem mais imagens para o treinamento, onde um pequeno grupo de imagens pode ser replicado alterando singelamente as características de cada uma das imagens através da aplicação de filtro. Em relação ao Blur, foram incorporados no algoritmo dois tipos diferentes, sendo: Median Blur, e o Homogeneous Blur, sendo também incorporado o filtro Bilateral[7]. A figura 5 demonstra como as imagens ficariam ao aplicar os filtros de Blur e o Bilateral.

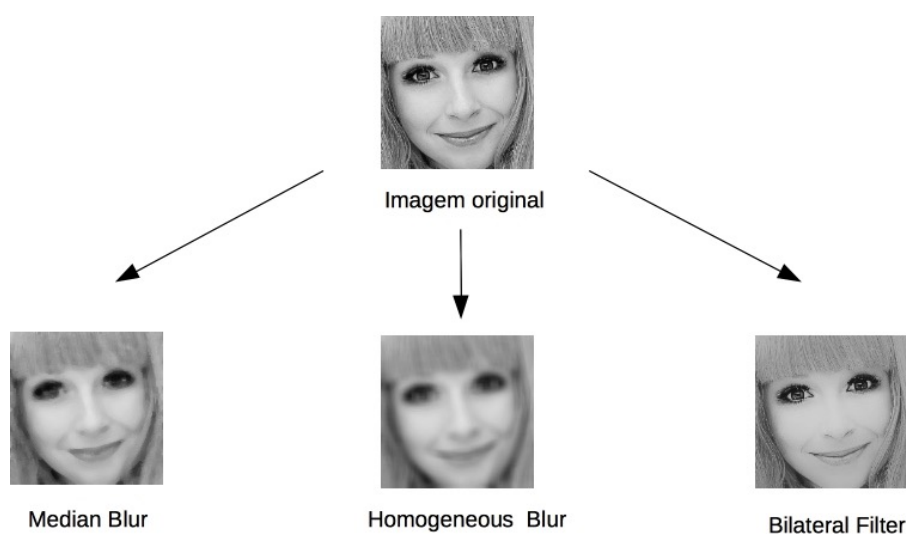


Figura 5. Demonstração das imagens após aplicação de filtros.

3.3. O sistema em Uso

A seguir será realizado uma descrição de como os componentes que compõem a solução, dando foco principal para a ferramenta de treinamento do Opencv, o Algoritmo de Replicação de Imagens e por fim o Algoritmo de Reconhecimento Facial.

3.3.1. Uso do Algoritmo de Replicação de Imagens.

O Algoritmo de Replicação de Imagens pode ser invocado via terminal utilizando a seguinte sintaxe:

```
./D_N_Images <path_imagem_original> <qt.Copias> <tipo_filtro> <nome_imagem_saida>
```

onde:

- `<path_imagem_original>` : representa o caminho da imagem de entrada, que será replicada pelo algoritmo.
- `<qt_Copias>` : representa a quantidade de cópias que serão feitas da imagem de entrada.
- `<tipo_filtro>` : Tipo de filtro a ser aplicado, que pode ser: 0 - apenas replica a imagem original sem aplicar filtros; 1 - Mediam Blur; 2 - Filtro Bilateral; 3 - Homogeneous Blur. Ao utilizar algum dos filtros, é solicitado do usuário um valor para MAX KERNEL.
- `<nome_imagem_saida>` : Corresponde a uma identificação que é dada para as imagens de saída.

A saída do algoritmo pode ser visualizada na figura 4 da seção 3.2.3.

3.3.2. Opencv Traincascade

Para o treinamento do padrão foi utilizado a ferramenta Traincascade, que vem inclusa no pacote de instalação do Opencv [2]. A ferramenta pode ser invocada através do seguinte comando:

```
opencv_traincascade -data <diretorio_cascade> -vec <Arquivo_vec> -bg  
<Arquivo_imagens_negativas> -numPos <qt.Imagens_positivas> -numNeg  
<qt.Imagens_Negativas> -numStages <numero_estágios> -w <largura_imagem> -h  
<altura_imagem>
```

Sendo:

- `-data`: caminho do diretório onde os arquivos de cascade serão salvos.
- `-vec`: caminho do arquivo de amostras. Um arquivo de amostras nada mais é do que um arquivo em formato binário contendo as imagens que serão utilizadas pelo Traincascade para o treinamento. Esse arquivo é gerado pelo `opencv_createsamples`, cuja extensão de arquivo é `.vec` [8].
- `-numPos` / `-numNeg`: quantidade de imagens positivas e negativas, respectivamente, que serão utilizadas no treinamento.
- `-numStages`: quantidade de estágios que aquele treinamento terá.
- `-w` / `-h`: Largura e altura, respectivamente, das imagens positivas.

- -bg: Arquivo de texto contendo o caminho para as imagens negativas que serão utilizadas no treinamento.

A saída do comando acima é um arquivo de classificador, no formato XML.

3.3.3. Algoritmo de Reconhecimento Facial

O Algoritmo de Reconhecimento Facial é o responsável por verificar se em uma determinada foto existe o padrão de rosto descrito no arquivo de classificador passado como entrada para ele. Sua sintaxe consiste em:

```
./TestCustonHaar <arquivo_classificador> <imagem>
```

onde:

- arquivo_classificador: caminho para o arquivo cascade, contendo o padrão de rosto treinado.
- imagem: imagem a ser analisada.

Caso o algoritmo consiga ler o arquivo de imagem e identificar a face, uma janela contendo a face será mostrada, sendo ela circulada. Caso o classificador não consiga encontrar nenhum padrão, o algoritmo termina informando um erro. Se a imagem ou o classificador estiverem corrompidos, o algoritmo aborta sua execução.

4. Testes

Os testes do sistema, principalmente no que diz respeito ao treinamento para o reconhecimento facial, foram realizados utilizando uma pequena quantidade de imagens positivas e negativas. Dentro do conjunto de imagens positivas haviam aquelas que continham somente o rosto da pessoa que foi utilizado no treinamento, além de outras imagens onde vários outros rostos foram postos lado a lado, na tentativa de “confundir” o classificador e verificar se ele era capaz de reconhecer somente um padrão de face. No conjunto de imagens negativas, foram usadas imagens com faces distintas, bem como imagens que continham somente paisagens. Foram estabelecidos os seguintes critérios para os testes:

- I. Positivas sem erro: Remete a imagens que contem o padrão de face treinado e que foram identificadas sem nenhum tipo de problema.
- II. Positivas com erro: Remete a imagens que contem o padrão de face treinado e foram identificadas pelo classificador, porem com alguma falha, como por exemplo ser destacada uma área que não continha o rosto, ou ser identificado mais de uma vez

o mesmo rosto, ou ainda identificar o rosto correto e um outro rosto que fez parte do treinamento.

III. Positivas perdidas: Remete a imagens que contem o padrão de face treinado, mas que por alguma razão o classificador não o identificou, fazendo com que o algoritmo de checagem terminasse emitindo um aviso de que nenhum objeto foi encontrado naquela foto.

IV. Negativas Identificadas: Remete ao conjunto de imagens negativas, que não contem o padrão de rosto treinado, mas que por alguma razão o classificador processou como sendo verdadeira.

Em relação a números, foram utilizadas 20 imagens tanto positivas quanto negativas para o teste, enquanto que para a classificação foram utilizados 4 arquivos de classificador distintos, onde todos eles passaram pelo mesmo treinamento, porém cada um com uma quantidade de estágios de treino diferentes. O primeiro arquivo de classificador obteve 5 estágios de treino, o segundo 7 estágios, o terceiro 10 estágios e o último 11 estágios. O treino para o reconhecimento facial foi realizado com 1010 fotos positivas e 2990 fotos negativas. A tabela 1 descreve os resultados obtidos nos testes.

Classificador de 5 estágios	
Positivas sem erro:	0
Positivas com erro:	16
Positivas perdidas:	4
Negativas Identificadas:	20
Classificador de 7 estágios	
Positivas sem erro:	0
Positivas com erro:	18
Positivas perdidas:	2
Negativas Identificadas:	20
Classificador de 10 estágios	
Positivas sem erro:	8
Positivas com erro:	6
Positivas perdidas:	6
Negativas Identificadas:	5
Classificador de 11 estágios	

Positivas sem erro:	10
Positivas com erro:	7
Positivas perdidas:	3
Negativas Identificadas:	4

Tabela 1. Resultado dos testes realizados

Apesar do teste proposto acima, é importante salientar que existe uma ferramenta inclusa no pacote do Opencv, o `opencv_performance`, cuja sua principal funcionalidade é realizar testes com os classificadores treinados, gerando como resultado uma estatística que mostra o quão bom foi o seu classificador diante de um conjunto de imagens usadas para teste[4][5] .

5. Conclusão

Em relação ao assunto em geral do trabalho, conclui - se que a área de reconhecimento de padrões, principalmente a de reconhecimento facial, é muito ampla e plausível de muitas pesquisas ainda. É importante salientar que, apesar deste trabalho abordar o método de Haar Cascade para o reconhecimento facial, existem muitos outros algoritmos que podem ser usados para identificar faces dentro de imagens, cada qual com sua peculiaridade.

Em relação ao treinamento utilizado, nota - se uma boa evolução da capacidade de classificação de acordo com o a quantidade crescente de estágios utilizados no treinamento. Vê - se que com 11 estágios, obteve - se pelo menos a metade das amostras classificadas sem nenhum tipo de erro, tendo ainda cerca de 35% das imagens positivas classificadas com algum tipo de falha, perdendo somente cerca de 15% do total. O número de imagens negativas classificadas como verdadeiras também é relativamente baixo, representando somente cerca de 20% do total, utilizando o classificador de 11 estágios. Diante dos testes realizados, é possível verificar que o classificador ainda necessita ser melhorado para atingir níveis mais satisfatórios, como por exemplo ter cerca de 70% das imagens positivas classificadas corretamente e nenhuma perda, além de ter menos de 20% de imagens negativas classificadas.

Diante do trabalho em si, no geral, nota - se uma limitação muito importante, correlacionada a capacidade de detecção de somente um tipo de rosto, onde somente um único arquivo de classificador pode ser carregado no sistema por vez, coisa que poderá ser melhorada em implementações futuras.

Com relação ao método utilizado para o treinamento, nota - se dois grandes pontos, sendo o primeiro a grande demanda de imagens e também o grande dispêndio de tempo para o treinamento, algo que pode consumir várias horas ou alguns dias. Em

contra partida, o resultado do treinamento pode ser muito bom, representando um classificador com alta acurácia, além de permitir que o mesmo possa ser reutilizado de maneira simples, uma vez que todo treinamento “está contido” dentro de um único arquivo.

Referências

- [1] - Dianin, Antonio Henrique. Diferenças entre Detecção de Rosto e Reconhecimento de Rosto, 2015. Disponível em: <https://www.portaleducacao.com.br/informatica/artigos/67656/as-diferencas-entre-deteccao-de-rosto-e-reconhecimento-de-rosto>
- [2] - OPENCV. Cascade Classifier, 2016. Disponível em: http://docs.opencv.org/2.4/doc/tutorials/objdetect/cascade_classifier/cascade_classifier.html
- [3] - http://docs.opencv.org/2.4/doc/tutorials/imgproc/gaussian_median_blur_bilateral_filter/gaussian_median_blur_bilateral_filter.html
- [4] - <http://vgl-ait.org/cvwiki/doku.php?id=opencv:tutorial:haartraining>
- [5] - ATHENANICHOL. Performance Testing your Clasifier, 2014. Disponível em: <http://athenanichol.com/blog/?p=185>
- [6] - OPENCV. Cascade Classifier Training, 2016. Disponível em: http://www.swarthmore.edu/NatSci/mzucker1/opencv-2.4.10-docs/doc/user_guide/ug_traincascade.html
- [7] - Yuan, Eric. Bilateral Filtering, 2013. Disponível em: <http://eric-yuan.me/bilateral-filtering/>
- [8] - OPENCV FORUM. What is a .vec file?, 2013. Disponível em: <http://answers.opencv.org/question/20872/what-is-a-vec-file/>
- [9] - Garcia, Gloria Bueno. et al. Learning Image Processing with Opencv.. Birmingham. Pack Publishing. Primera edição; 2015; p. 1 - 12
- [10] - Baggio Lélis Daniel. et al. Mastering Opencv with Practical Computer Vision Projects. Birmingham. Pack Publishing. Primeira edição; 2012; p. 266 - 292.
- [11] - MATLAB. Train a Cascade Object Detector, 2016. Disponível em: <https://www.mathworks.com/help/vision/ug/train-a-cascade-object-detector.html>
- [12] - Krishna M. Gopi. Face Detection System On AdaBoost Algorithm Using Haar Classifiers. International Journal of Modern Engineering Research; 2012, vol 2, p. 3556-3560.
- [13] - Brownlee, Jasson. Boosting and AdaBoost for Machine Learning, 2016. Disponível em: <http://machinelearningmastery.com/boosting-and-adaboost-for-machine-learning/>

- [14] - Ray, Sunil. Quick Introduction to Boosting Algorithms in Machine Learning, 2015. Disponível em: <https://www.analyticsvidhya.com/blog/2015/11/quick-introduction-boosting-algorithms-machine-learning/>
- [15] - Pista, Pedro. Como funciona o reconhecimento facial, 2012. Disponível em: <http://www.techtudo.com.br/artigos/noticia/2012/04/como-funciona-o-reconhecimento-facial.html>
- [16] - Faria, Alessandro. Biometria, Reconhecimento Facial Livre. Disponível em: <http://www.linhadecodigo.com.br/artigo/1813/biometria-reconhecimento-facial-livre.aspx>
- [17] - Processamento digital de Imagens. Disponível em: http://trabalhoic.awardspace.com/p_visaogeral.php [sem editor e data reconhecidos]
- [18] -Arubas, Eval. Face Detection and Recognition (Theory and Practice), 2013. Disponível em: <http://eyalarubas.com/face-detection-and-recognition.html>
- [19] - Braga, Luiz F. Zenicola. Sistema de Reconhecimento Facial. Universidade de São Paulo - USP, 2013.

