Universidade Federal do ABC
Centro de Engenharia, Modelagem e Ciências Sociais Aplicadas
Trabalho de Graduação em Engenharia de Informação

Validação do Algoritmo de Viola-Jones para Detecção Facial

Gabriel Pitali de Carvalho

Santo André

Maio de 2020

Sumário

Introdução

Reconhecimento facial é uma tarefa trivial para humanos e há décadas tem sido um desafio para visão computacional e aprendizado de máquina, segundo a referência 1, desde os anos 90 o tema emerge em diferentes conferencias e com o aumento do poder computacional dos dias atuais, sua capacidade se expande muito, fazendo com que tal assunto receba enorme atenção, principalmente devido ao seu grande valor comercial e as mais diversas aplicações possíveis, como verificação de identidade, controle de acesso, segurança, investigação de imagens em bancos de dados, vigilância, entretenimento ou realidade virtual. [2] [1]

O processo de reconhecimento facial de forma automatizada é separado em 4 principais etapas, conforme detalhado no livro 3, primeiramente deve ser feita a *detecção facial*, que consiste em validar e localizar a existência de alguma face na imagem ou video, a segunda etapa consiste no *alinhamento facial*, para que todas faces da base de dados sigam o mesmo padrão, a terceira etapa é a *extração de características* que permite a obtenção de informação efetiva que será útil na distinção das diferentes faces, a quarta e última etapa consiste na *correspondência de características*, onde as características extraídas anteriormente são comparadas com outras já conhecidas para que sejam identificadas.

Aprofundando o estudo da primeira etapa, de *detecção facial*, a referência 4 indica duas diferentes metodologias, a primeira baseada em características e a segunda baseada em imagens, ambas posteriormente podem ser separadas em diversas técnicas mais específicas, como por exemplo a análise de características por constelação ou a análise de imagens com redes neurais, onde cada técnica espécifica possui seus prós e contras em relação as demais.

0.1 Objetivos e Motivação

Este trabalho tem como objetivo encontrar a melhor forma de atuar sobre a primeira etapa (*detecção facial*) e a segunda etapa (*alinhamento facial*) do processo de reconhecimento facial, avaliando o desempenho qualitativo e quantitativo de diferentes metodologias e ferramentas disponíveis e permitindo a rápida identificação de imagens que não possuem uma face, para satisfazer a necessidade descrita a seguir.

Atualmente empresas e órgãos públicos possuem a necessidade de manter cadastros pessoais mas existe grande demanda para que estes cadastros sejam feitos de forma totalmente virtual pela população, pois isso evita o deslocamento de pessoas até os pontos de cadastro e torna todo o processo muito mais ágil. Certos cadastros incluem fotos de

identificação e isto traz a necessidade de uma verificação feita por humanos para validar se a mesma consiste em uma foto de face frontal, conforme é necessário para o cadastro.

A validação citada já ocorre e é feita de forma totalmente manual, onde funcionários tem que verificar cada uma das imagens recebidas e muitas vezes se deparam com fotos sem nenhuma face frontal ou sem condições de serem identificadas (desfocadas, por exemplo), que são rejeitadas para que uma nova imagem seja solicitada. Estas imagens claramente inválidas por estarem em desacordo com o padrão esperado (foto de face frontal), poderiam facilmente ser eliminadas por uma filtragem anterior, reduzindo grande parte do trabalho que é feito hoje manualmente.

1 Materiais e Métodos

Com o objetivo de obter melhor entendimento sobre possíveis ferramenta de visão computacional e detecção facial, este trabalho descreve a execução de testes utilizando a linguagem de programação Python e a implementação da ferramenta OpenCV na mesma linguagem.

1.1 OpenCV

A ferramenta OpenCV, que pode ser encontrada no endereço 5, é uma biblioteca de código aberto focada na solução de problemas utilizando visão computacional em tempo real, desenvolvida pela Intel e posteriormente pela Itseez, com suporte a múltiplas plataformas e uso gratuito sobre a licença de código aberto BSD. A ferramenta apresenta suporte a frameworks de aprendizado profundo, como TensorFlow, Pytorch e Caffe e contempla tanto funções básicas, para aplicações como processamento de imagem, alteração de cor ou resolução, até aplicações avançadas, como detecção facial, identificação de características e biometria. [6]

Neste trabalho, será utilizada a função de detecção de faces da ferramenta OpenCV, que utiliza um classificador em cascata baseado características, este é um método eficiente para reconhecimento de faces em imagens proposto por Paul Viola and Michael Jones, amplamente conhecido como método Viola-Jones, onde uma função é treinada com muitos exemplos positivos (imagens que contém o objeto a ser detectado) e negativos (imagens que não contém o objeto a ser detectado) e então utilizada para detectar as mesmas características em outras imagens. [7]

A detecção de faces utilizando OpenCV consiste em duas etapas principais, o treinamento do modelo, onde são apresentadas diversas imagens já identificadas para que o modelo identifique padrões positivos e negativos. Após o treinamento, o modelo obtido pode ser utilizado para identificar, em novas imagens, características semelhantes as vistas nas imagens do treinamento. Neste projeto, será utilizado um modelo fornecido em conjunto com a ferramenta OpenCV, já treinado com diversos exemplos de faces frontais.

O procedimento utilizado permite ainda ajuste de parâmetros para a execução do algoritmo de Viola-Jones, neste teste, serão utilizados: o fator de escala, que é o fator pelo qual as dimensões da imagem serão multiplicadas na tentativa de encontrar faces de diferentes tamanhos (quanto menor, maior a chance de encontrar faces) e o número mínimo de vizinhos, que é número mínimo de detecções, após varias iterações, para uma parte da imagem ser considerada uma face (quanto menor, maior a chance de encontrar

faces).

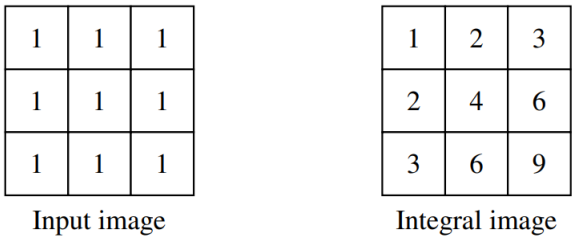
1.2 Algoritmo Viola-Jones

O algoritmo Viola-Jones foi publicado em 2001, no paper "Rapid object detection using a boosted cascade of simple features" [8] e é famoso por sua capacidade de detecção de faces com muita velocidade, isso ocorre devido a 3 principais técnicas utilizadas: o cálculo da imagem integral, o algoritmo de impulsão *AdaBoost* e o classificador em cascata.

1.2.1 Imagem Integral

A primeira etapa do algoritmo Viola-Jones consiste em transformar a imagem original em uma imagem integral, isto é feito calculando o valor de cada pixel como a soma de todos os pixels que estão acima ou a esquerda do mesmo, como ilustrado na figura 1.

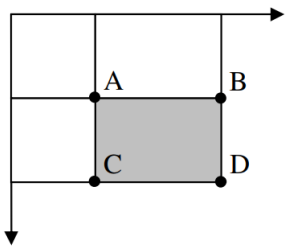
Figura 1 – Imagem original (esquerda) e imagem integral (direita).



Fonte: Jensen (2008)

A utilização desta técnica permite calcular facilmente o tamanho de qualquer retângulo formado entre quatro pixels da imagem, conhecendo apenas o valor dos seus cantos, possibilitando assim a análise rápida de diversas partes da imagem. Tal calculo é feito definindo o retângulo a ser análisado e então aplicando a equação 1.1.

Figura 2 – Representação da área da imagem a ser analisada.

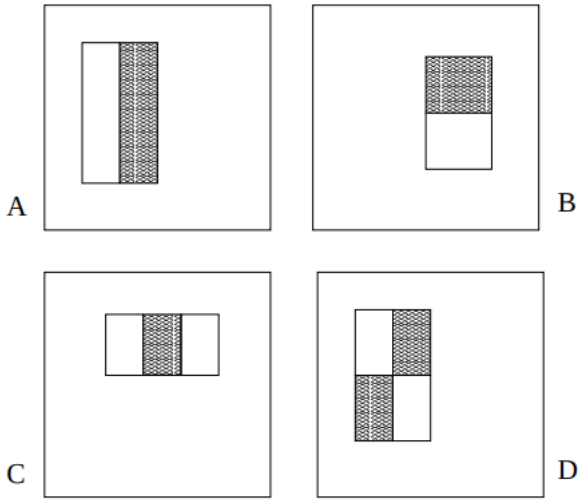


Fonte: Jensen (2008)

$$\text{Soma do retângulo cinza} = D - (B + C) + A \tag{1.1}$$

Com a possibilidade de calcular facilmente a soma dos pixels de um retângulo arbitrário de forma rápida, o algoritmo para detecção pode analisar diversos trechos da imagem, chamados aqui de características, fazendo a comparação de duas ou mais áreas retângulares predefinidas, como os exemplos ilustrado na figura 3.

Figura 3 – Alguns exemplos de características retângulares analisadas.



Fonte: Viola e Jones (2001)

O valor final de cada característica é definido pela soma do valor dos pixels sob o retângulo cinza menos a soma do valor dos pixels sob o retângulo branco.

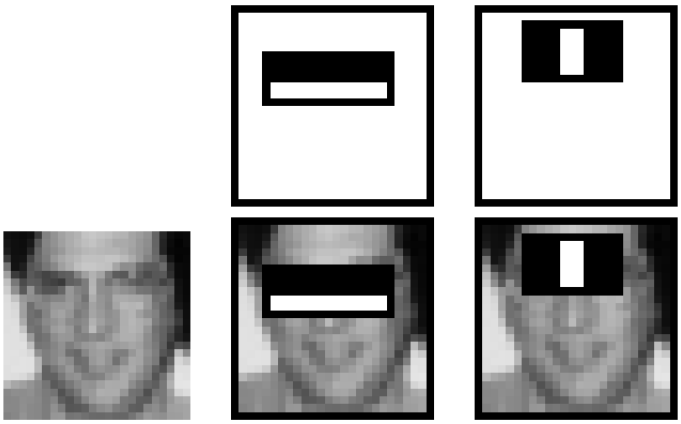
1.2.2 Algoritmo de Impulsão AdaBoost

As características demonstradas anteriormente, são definidas basicamente como duas ou mais áreas retângulares de qualquer tamanho, tal simplicidade implica na possibilidade da criação de uma enorme variação das mesmas que precisariam ser calculadas diversas vezes, para cada parte de imagem e com diferentes tamanhos, isso implica em um alto custo de processamento, para evitar tal problema, durante a etapa de treinamento do modelo, é utilizado o algoritmo de impulsão *AdaBoost*, que identifica quais são as características com maior probabilidade de acerto.

O *AdaBoost*, que tem seu nome derivado de *adaptative boosting* (traduzido como impulsão adaptativa), é um método de aprendizado de máquina que utiliza a combinação de vários classificadores fracos para obter uma classificação forte, no caso da detecção facial, o algoritmo é utilizado tanto para selecionar um conjunto de características mais eficientes como para treinar o classificador. [10]

A figura 4 retrata as melhores características registradas por Viola e Jones (2001), fica claro que as mesmas se destacam por evidenciar as regiões dos olhos e do nariz.

Figura 4 – Características retângulares mais eficientes para detecção facial.



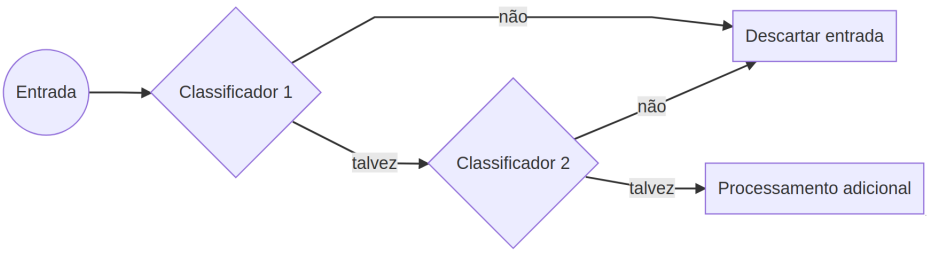
Fonte: Viola e Jones (2001)

1.2.3 Classificador em Cascata

Pensando que, na maioria dos casos, uma face não ocupa a maior parte de uma imagem a ser identificada, é necessário encontrar uma forma rápida de descartar os elementos do fundo da mesma e concentrar o poder de processamento nos elementos que tem maior probabilidade de serem reconhecidos como uma face, isso leva a uma formulação para o problema onde ao contrário de encontrar faces, é necessário um algoritmo que descarte as "não faces".

Para tal problema, o classificador em cascata apresenta uma ótima solução, esta consiste na utilização de uma série de classificadores que são aplicados de forma sequencial, conforme ilustrado na imagem 5, permitindo que imagens que certamente não possuem faces sejam rapidamente descartadas logo nas primeiras iterações, enquanto imagens com possíveis faces são classificadas por toda cascata, trazendo um elevado nível de confiança ao resultado.

Figura 5 – Diagrama do funcionamento do classificador em cascata.



Fonte: Viola e Jones (2001)

Um classificador comum, com um único estágio normalmente aceitaria muitos casos de falso negativo, para reduzir a taxa de falsos positivos e de descarte de imagens

relevantes, mas no classificador em cascata, falsos positivos nos primeiros estágios não são um problema, pois serão analisados em outros diversos estágios e provavelmente eliminados.

A utilização desse modelo combinada com o algoritmo *AdaBoost*, possibilita a análise das característica mais eficientes logo no início e consequentemente o descarte muito mais rápido dos casos negativos nos primeiros estágios.

1.3 Conjuntos de imagens para teste

Para analisar a eficiência da implementação do algoritmo *Viola-Jones* na biblioteca *OpenCV*, assim como o seu modelo previamente treinado, foram utilizados dois conjuntos de imagens. Tendo em mente o objetivo deste projeto, de identificar rapidamente imagens que não possuem uma face, são tratadas como imagens **positivas** as imagens que **não possuem faces** e como **negativas** as imagens que **possuem ao menos uma face**.

O primeiro dataset, nomeado *Hotels-50k* [11], será utilizado para representar imagens **positivas**, portanto consiste em um conjunto de mais de 50 mil imagens de diversos quartos de hotel vazios, pois imagens com características semelhantes a estas são muitas vezes submetidas erroneamente em cadastros pessoais, ao invés de uma imagem da face a ser cadastrada.

O segundo dataset, nomeado *UTKFace*, disponível em 12, será utilizado para representar imagens **negativas**, portanto consiste em um conjunto de mais de 20 mil imagens, com uma única face em cada, de diversas pessoas entre 0 e 116 anos de idade, catalogadas de acordo com idade, raça e sexo, alguns exemplos das imagens contidas no dataset podem ser vistos na figura 6.

Figura 6 – Exemplos de imagens do dataset *UTKFace*.

Fonte: *UTKFace dataset* [12])

Para os testes, foram selecionadas aleatoriamente 17130 do dataset *Hotels-50k* e selecionadas outras 17130 imagens do dataset *UTKFace*, mas sendo essas apenas imagens de pessoas entre 18 e 60 anos de idade. Assim, foi totalizado um conjunto de 34260, onde

metade delas eram positivas (não possuíam faces) e a outra metade negativas (possuíam ao menos uma face).

1.4 Metodologia de Análise

Para melhor analisar os resultados dos testes, foi necessário especificar com clareza como poderiam ser agrupadas as imagens, dada a sua origem e o resultado observado no teste, para isso foram utilizadas as definições da tabela 1.

Tabela 1 – Grupos observados

Grupo	**Descrição**	**Quantidade**
A	Imagens que não contêm nenhuma face	17130
$\overline{A}$	Imagens que contêm uma face	17130
B	Imagens onde o algoritmo não identificou nenhuma face	Variável
$\overline{B}$	Imagens onde o algoritmo identificou uma ou mais faces	Variável

Definidos os grupos, pode-se utilizar a tabela de contingência 2 para facilitar a análise da relação entre os grupos definidos anteriormente. Na tabela 2 são observados os grupos a (verdadeiro positivo) onde o algoritmo identifica corretamente uma face em cada uma das imagens que realmente contêm uma face, b (falso negativo) onde o algoritmo erroneamente não reconheceu nenhuma face, apesar das imageens conterem uma face cada, c (falso positivo) onde o algoritmo erroneamente identificou ao menos uma face, mesmo as imagens não contendo nenhuma e d (verdadeiro negativo) onde o algoritmo identificou corretamente que não existia nenhuma face nas imagens. É importante destacar que os quatro grupos destacados na tabela de contingência são mutuamente excludentes. [13]

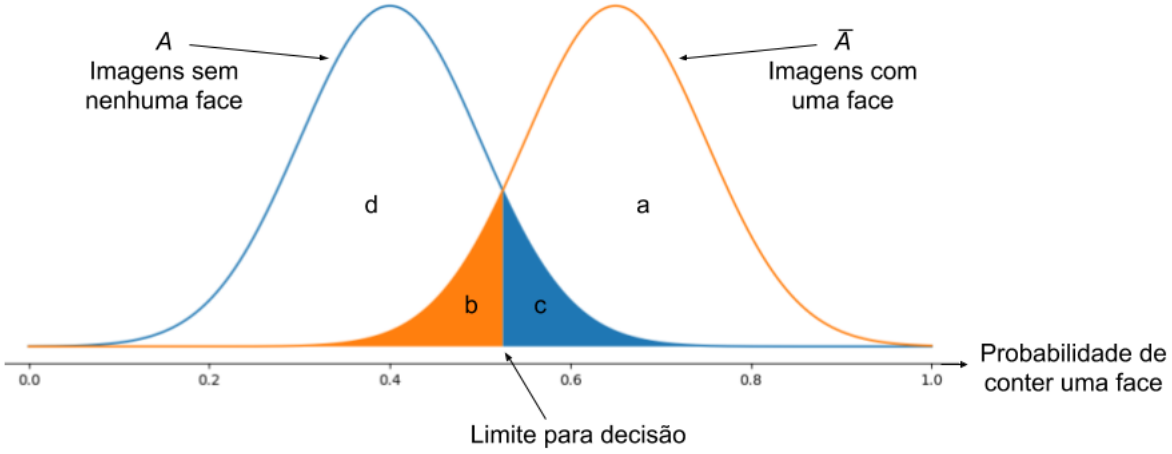
Tabela 2 – Tabela de contingência

	B	$\overline{B}$
A	a (verdadeiro positivo)	b (falso negativo)
$\overline{A}$	c (falso positivo)	d (verdadeiro negativo)

Para melhor entender os agrupamentos da tabela de contingência, pode-se observar na imagem 7 as distribuições que representam a quantidade de imagens dos grupos A e $\overline{A}$ dada a sua probabilidade de conter uma face.

Nas distribuições são destacados os grupos b (falso negativo) e c (falso positivo) e fica evidente que, devido a sobreposição das distribuições A e $\overline{A}$, é necessário definir uma limite para decisão, que pode ser ajustado conforme a necessidade, mas que independente do seu ajuste, sempre existirá um grupo categorizado de forma incorreta.

Figura 7 – Possível distribuição normal dos grupos observados na tabela de contingência.

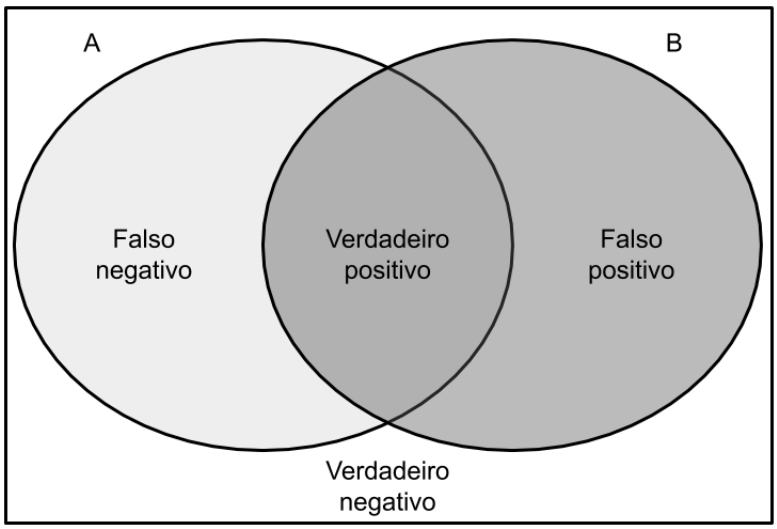


A tabela de contingência também pode ser escrita em termos probabilísticos (tabela 3), incluindo as probabilidades marginais ou pode ser visualizada no diagrama de Venn correspondente (figura 8).

Tabela 3 – Tabela de contingência com probabilidades marginais

	B	$\overline{B}$	Soma
A	$P(A \cap B)$	$P(A \cap \overline{B})$	$P(A)$
$\overline{A}$	$P(\overline{A} \cap B)$	$P(\overline{A} \cap \overline{B})$	$P(\overline{A})$
Soma	$P(B)$	$P(\overline{B})$	1

Figura 8 – Diagrama de Venn para as imagens analisadas.



Por fim, podem ser calculadas as medidas tradicionais de sensitividade (a probabilidade condicional do algoritmo identificar ao menos uma face dado que a imagem contém uma face) e especificidade (a probabilidade condicional do algoritmo não identificar nenhuma face em uma imagem que realmente não contém nenhuma face) conforme as equações 1.2 e 1.3.

$$\text{sensitividade, } P(B|A) = P(A \cap B)/(P(A \cap B) + P(A \cap \overline{B})) = a/(a+b) \tag{1.2}$$

$$\text{especificidade, } P(\overline{B}|\overline{A}) = P(\overline{A} \cap \overline{B})/(P(\overline{A} \cap \overline{B}) + P(\overline{A} \cap B)) = d/(d+c) \tag{1.3}$$

2 Resultados e Discussão

Após entender o funcionamento do algoritmo de Viola-Jones e definir a metodologia para análise, foram executadas duas iterações de detecção sobre o conjunto de imagens, com diferentes parâmetros, na primeira foram ajustados os parâmetros de fator de escala para 1.3 e número mínimo de vizinhos para 5, após analisar todas imagens foi possível preencher a tabela de contingência 4 e calcular a respectiva sensitividade $P(B|A) = 98.5872\%$ e especificidade $P(\overline{B}|\overline{A}) = 86.6024\%$.

Tabela 4 – Tabela de contingência com os resultados do primeiro teste

	B	$\overline{B}$	Soma
A	49.2936%	00.7064%	50%
$\overline{A}$	06.6988%	43.3012%	50%
Soma	55.9924%	44.0076%	100%

Na segunda iteração, ajustando o fator de escala para 1.05 e o número mínimo de vizinhos para 3, foram obtidos os resultados apresentados na tabela 5 e a respectiva sensitividade $P(B|A) = 66.3923\%$ e especificidade $P(\overline{B}|\overline{A}) = 98.3479\%$.

Tabela 5 – Tabela de contingência com os resultados do segundo teste

	B	$\overline{B}$	Soma
A	33.1962%	16.8038%	50%
$\overline{A}$	00.8260%	49.1740%	50%
Soma	34.0222%	65.9778%	100%

Observando as duas iterações, o primeiro ponto de destaque é o impacto dos parâmetros nos resultados, comparando as medidas de B na primeira e segunda iterações, percebe-se que a redução dos valores ajustados tanto nos parâmetros de fator de escala e número de vizinhos, fez com que mais faces forem detectadas, conforme o esperado, consequentemente, ouve um grande aumento no número de falsos negativos, onde o algoritmo encontra uma face apesar da mesma não existir.

Analisando a sensitividade, percebe-se que na segunda iteração essa foi bastante reduzida, ou seja, dado o grupo de imagens que não possuíam nenhuma face, na primeira iteração 98.5872% delas foram classificadas corretamente como imagens sem nenhuma face, já na segunda iteração somente 66.3923% foram classificadas corretamente.

Já analisando a especificidade, o valor teve uma redução na segunda iteração, mas não foi uma variação tão significativa quanto a variação da sensitividade, este dado pode ser entendido da seguinte forma, dado o grupo de imagens que possuíam uma face, na

primeira iteração 86.6024% delas foram classificadas corretamente como imagens com ao menos uma face e na segunda iteração 98.3479% foram classificadas corretamente.

Vale enfatizar que ambas análises sensitividade e especificidade estão de acordo com os resultados esperados devido aos parâmetros utilizados que fazem com que a segunda iteração detecte uma quantidade maior de faces.

Outro ponto a ser destacado é a porcentagem de imagens classificadas corretamente, ou seja, a soma das porcentagens de verdadeiros positivos e verdadeiros negativos, que na primeira iteração foi igual a 92.5948% e na segunda iteração foi igual a 82.3702%, esses valores demonstram que a ajuste de parâmetros não influencia apenas no limite de decisão observado na figura 7, mas também nas distribuições.

Relembrando o objetivo do projeto, que consiste em negar rapidamente imagens que certamente não possuem nenhuma face, e observando os resultados obtidos e análisados, pode-se concluir que o algoritmo de Viola-Jones implementado na biblioteca OpenCV poderia ser utilizado de forma satisfatória para solucionar tal problema. Além disso, a utilização dos parâmetros ajustados na primeira iteração se mostra mais adequada, devido a maior quantidade de imagens classificadas corretamente, que torna possível negar de forma correta e automática 98.5872% das imagens que não possuem faces, mas é importante lembrar que 13.3976% das imagens que possuem uma face seriam também automáticamente negadas erroneamente. Para os casos onde negar uma imagem correta poderia causar maiores problemas, é recomendável utilizar os mesmos parâmetros da segunda iteração, que tornaria possível negar de forma correta e automática 66.3923% das imagens que não possuem faces, mantendo em apenas 1.6521% o volume de imagens que possuem uma face seriam automáticamente negadas erroneamente.

Referências

1 ZHAO, W. et al. Face recognition: A literature survey. *ACM Comput. Surv.*, ACM, New York, NY, USA, v. 35, n. 4, p. 399–458, dez. 2003. ISSN 0360-0300. Disponível em: <http://doi.acm.org/10.1145/954339.954342>. Citado na página 3.

2 PARMAR, D.; MEHTA, B. Face recognition methods & applications. *International Journal of Computer Technology and Applications*, v. 4, p. 84–86, 01 2014. Citado na página 3.

3 LI, S. Z.; JAIN, A. K. *Handbook of Face Recognition*. 2nd. ed. [S.l.]: Springer Publishing Company, Incorporated, 2011. ISBN 085729931X, 9780857299314. Citado na página 3.

4 HJELMåS, E.; LOW, B. K. Face detection: A survey. *Computer Vision and Image Understanding*, v. 83, n. 3, p. 236 – 274, 2001. ISSN 1077-3142. Disponível em: <http://www.sciencedirect.com/science/article/pii/S107731420190921X>. Citado na página 3.

5 ITSEEZ. *Open Source Computer Vision Library*. 2015. <https://github.com/itseez/opencv>. Citado na página 5.

6 WIKIPEDIA. *OpenCV — Wikipedia, The Free Encyclopedia*. 2019. <http://en.wikipedia.org/w/index.php?title=OpenCV&oldid=906212825>. [Online; accessed 31-July-2019]. Citado na página 5.

7 ITSEEZ. *The OpenCV Reference Manual*. 2.4.9.0. ed. [S.l.], 2014. Citado na página 5.

8 VIOLA, P.; JONES, M. Rapid object detection using a boosted cascade of simple features. In: . [S.l.: s.n.], 2001. v. 1, p. I–511. ISBN 0-7695-1272-0. Citado 3 vezes nas páginas 6, 7 e 8.

9 JENSEN, O. H. *Implementing the Viola-Jones face detection algorithm*. Dissertação (Mestrado) — Technical University of Denmark, DTU, DK-2800 Kgs. Lyngby, Denmark, 2008. Citado na página 6.

10 SANTANA, L. M. Queiroz de; ROCHA, F. Processo de detecção facial utilizando viola;jones. *Interfaces Científicas - Exatas e Tecnológicas*, v. 1, 02 2015. Citado na página 7.

11 STYLIANOU, A. et al. Hotels-50k: A global hotel recognition dataset. *CoRR*, abs/1901.11397, 2019. Disponível em: <http://arxiv.org/abs/1901.11397>. Citado na página 9.

12 IMAGING, A.; PROCESSING, C. I. *UTK Face Dataset*. [Online; accessed 04-August-2019]. Disponível em: <https://susanqq.github.io/UTKFace/>. Citado na página 9.

13 DOUGHERTY, G. *Pattern Recognition and Classification: An Introduction*. [S.l.]: Springer Publishing Company, Incorporated, 2012. ISBN 1461453224, 9781461453222. Citado na página 10.