

## SISTEMA DE CHAMADA ESCOLAR COM RECONHECIMENTO FACIAL UTILIZANDO OPENCV

Luiz Fellipe de Oliveira ALMEIDA

[1800935@escolas.anchieta.br](mailto:1800935@escolas.anchieta.br)

Ciência da Computação, Centro Universitário Anchieta, UniAnchieta

Paula Akemi da Silva TANAKA

[1902879@escolas.anchieta.br](mailto:1902879@escolas.anchieta.br)

Ciência da Computação, Centro Universitário Anchieta, UniAnchieta

Victor Hugo Martins OLIVEIRA

[2004526@escolas.anchieta.br](mailto:2004526@escolas.anchieta.br)

Ciência da Computação, Centro Universitário Anchieta, UniAnchieta

Clayton Augusto VALDO

[clayton.valdo@anchieta.br](mailto:clayton.valdo@anchieta.br)

Orientador, Centro Universitário Anchieta, UniAnchieta

### Resumo

Pessoas reconhecem facilmente outras conhecidas por meio de suas características físicas, independente de iluminação, acessórios, envelhecimento, dentre outras características, é uma habilidade natural impregnada no cérebro humano. Com a evolução dos estudos de Inteligência Artificial (IA), foi possível fazer com que máquinas também pudessem reconhecer humanos por características físicas. Reconhecimento Vocal, Digital, Retina e de Íris são alguns exemplos de tecnologias de IA voltadas ao reconhecimento humano. O presente Artigo visa a explicar o que é e como funciona a tecnologia de Reconhecimento Facial, ou seja, reconhecimento de pessoas mediante sua fisionomia, desenvolvendo um sistema-modelo de chamada escolar utilizando a tecnologia.

### Palavras-Chave:

Reconhecimento Facial; Inteligência Artificial; OpenCV; Machine Learning.

### Abstract

People easily recognize other acquaintances through their physical characteristics, regardless of lighting, accessories, aging among other characteristics, it is a natural ability impregnated in the human brain. With the evolution of Artificial Intelligence (AI) studies, it was possible to make machines also able to recognize humans by physical characteristics. Voice, Digital, Retina and Iris Recognition are some examples of AI technologies aimed at human recognition. This article aims to explain what is and how Facial Recognition technology works, that is, recognition of people through their physiognomy, developing a model school attendance system using technology.

### Keywords:

Facial recognition; Artificial intelligence; OpenCV; Machine Learning.

## INTRODUÇÃO

A “*World Wide Web*” (rede mundial de computadores) conecta os inúmeros dados que estão em constante crescimento. Assim, o termo *Web 3.0* foi cunhado para definir o período tecnológico atual em que todos consomem constantemente dessa teia integrada de informações (MARKOFF, 2006).

Trata-se de uma “nova era”, em que a busca por inteligência artificial e o cruzamento de dados ganham notoriedade para diferentes finalidades (SILVA; SILVA, 2019).

Um sistema computacional de reconhecimento de imagens é capaz de identificar padrões complexos, como texturas, objetos, textos e padrões biométricos, que utilizam características humanas únicas, como impressões digitais, íris, voz e face, o que permite a diferenciação entre seres humanos (ZHAO et al., 2003). Tais sistemas utilizam-se de taxas de acerto percentual para representar o grau de acuidade do processo de identificação esperado (OMAIA et al., 2009).

O reconhecimento de faces abrange uma larga escala de funções, algumas são atividades simples, que buscam agilidade em ações a aplicações voltadas à segurança. Assim, métodos tradicionais para garantir a identidade de um usuário (como o uso de senhas e cartões) podem ser perdidos, roubados, ou até manipulados, com o objetivo de burlar sistemas de proteção, tornando, portanto, o sistema de reconhecimento facial uma alternativa mais eficaz para a segurança digital (ANDREZZA, 2015).

A tecnologia de reconhecimento facial tem sido utilizada em sistemas públicos de segurança e monitoramento mundo afora, o que permite identificar criminosos ou pessoas desaparecidas e impactando os índices de criminalidade drasticamente, como nos Estados Unidos da América e na China. No Brasil, o sistema piloto foi aplicado no Rio de Janeiro e, no ano de 2019, muitos criminosos foram identificados e presos graças a essa tecnologia (NOGUEIRA; DOS SANTOS, 2019).

Uma questão social tem sido levantada devido às bases de dados mal treinadas, em termos de raça e gênero. O problema é refletido em amplas taxas de erros entre os diferentes grupos sociais: com homens de pele mais clara, a taxa de erro é de cerca de 0,8%, enquanto ela dispara para até 34,7% entre mulheres com tonalidades mais escuras de pele (BUOLAMWINI; GEBRU, 2018).

Este trabalho apresenta o desenvolvimento de um sistema em *Python* de chamada escolar, com reconhecimento facial utilizando, principalmente, a biblioteca *OpenCV*.

## DESENVOLVIMENTO

Reconhecimento Facial (RF) é uma ferramenta biométrica semelhante a outras ferramentas, como Reconhecimento de Digital e Reconhecimento de Íris. O funcionamento de um sistema com RF pode ser definido em três principais etapas: captura e tratamento de imagem (1), detecção (2) e reconhecimento (3) dos rostos humanos.

A detecção de faces pode ser feita de diversas maneiras, mas de acordo com o professor Shree Nayar (2021), o algoritmo de Viola e Jones (2003) é um dos melhores para a solução desse problema, por ser muito eficiente computacionalmente e utilizar Imagem Integral (ou *Integral Image* em inglês), e esse foi o método utilizado no desenvolvimento do sistema apresentado neste Artigo.

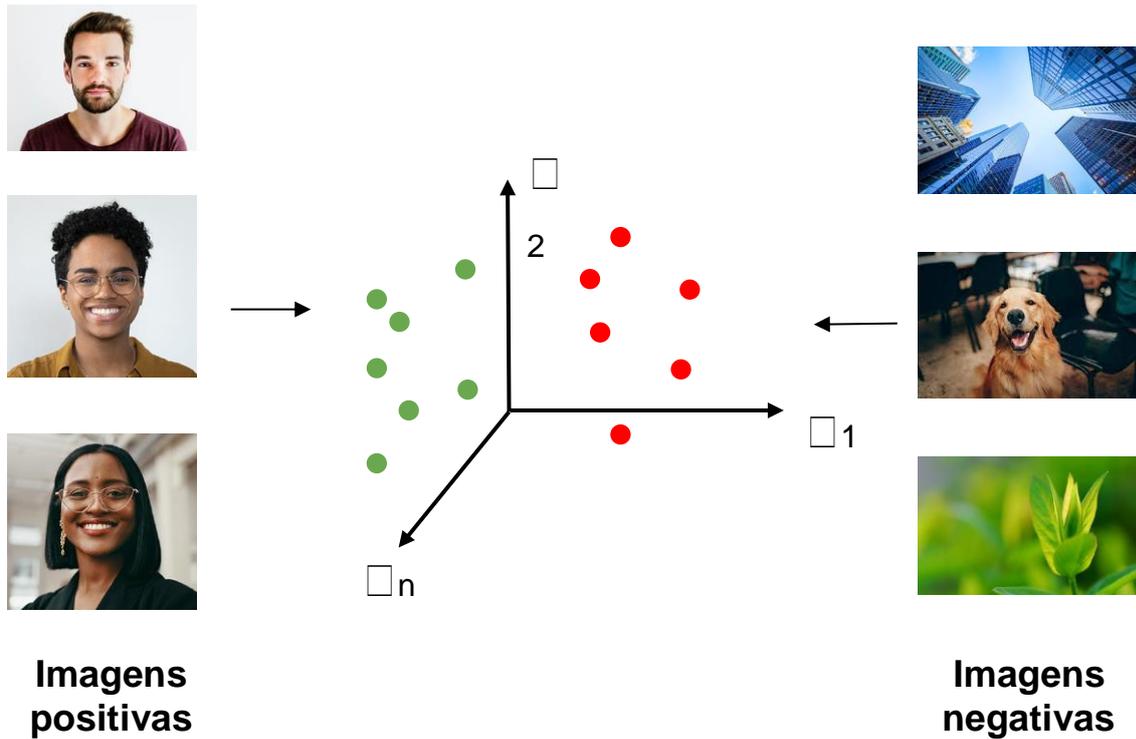
De acordo com Viola e Jones (2003), um classificador *Haar* é treinado utilizando imagens positivas e negativas. No contexto de RF, imagens positivas são aquelas que contêm rostos e negativas são aquelas que não contêm rostos. Considere a figura 1 abaixo como exemplos de imagens positivas e negativas, da esquerda para a direita:

**Figura 1.** Exemplo de imagens positivas (esquerda) e negativas (direita).



Durante a fase de treinamento, cada resultado de busca por Características de *Haar* encontradas em uma imagem é inserido em um espaço, que pode ser representado pelo gráfico na figura 2, abaixo:

**Figura 2.** Gráfico que representa o resultado de um treinamento de um Classificador de Haar.



Para facilitar o entendimento, a teoria apresentada neste Artigo é apoiada pelo desenvolvimento de uma aplicação que parte do pressuposto de que, em todas as imagens que serão testadas, o rosto estará em direção frontal em relação à câmera e a uma distância fixa.

### Capturando e Tratando a Imagem

A primeira etapa do sistema é a captura da imagem que será utilizada posteriormente para reconhecimento de rostos. Considere como exemplo a figura 3, mostrada abaixo:

**Figura 3.** *Frame* capturado da câmera.



Considerando que a imagem acima respeita a premissa definida anteriormente sobre direção e posição do rosto, o sistema irá extrair o rosto da posição pré-determinada para prosseguir com o tratamento:

**Figura 4.** *Frame* cortado nas dimensões pré-determinadas.



De acordo com as definições de Viola e Jones (2003) do programa *Haar Cascade*, a imagem a ser testada deve estar em escala de cinza. Considere como exemplo a figura 5, abaixo:

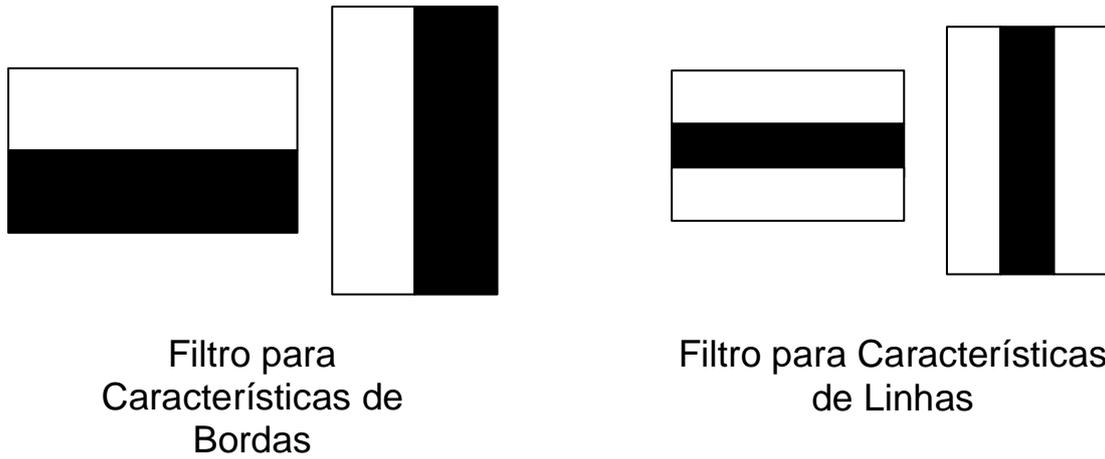
**Figura 5.** *Frame* convertido em escalas de cinza.



### **Detectando um Rosto**

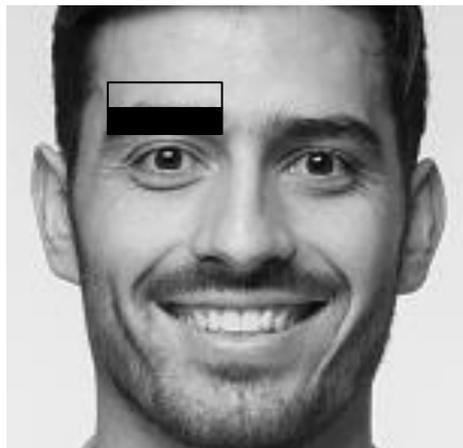
De acordo com o professor Shree Nayar (2021), as Características de *Haar* utilizam Filtros de *Haar*, que são *Haar-wavelets* (ou *Ondaletas de Haar*). São uma sequência de funções quadradas e redimensionadas, parecidas com Análise Harmônica Clássica (ou *Fourier-analysis*). Esses filtros indicam regiões que, em um rosto humano, são mais claras e mais escuras. Existem dois principais tipos de características: de bordas e de linhas, que podem ser representadas pelo gráfico na figura 6, abaixo:

**Figura 6.** Exemplos de Características de *Haar*



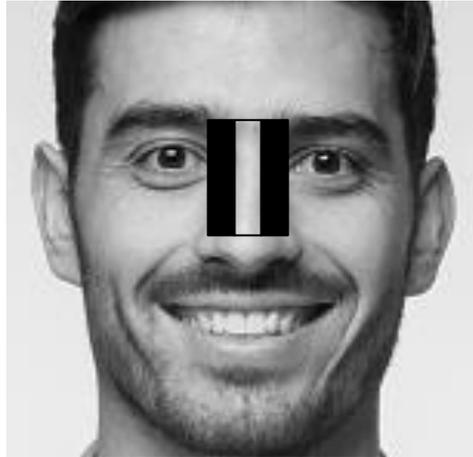
Um exemplo de uma Característica de Bordas em um rosto humano seria a região das sobrancelhas. De acordo com Gupta (2019), a própria sobrancelha se destaca como uma área escura, enquanto o início da testa, logo acima da sobrancelha, é uma área mais clara:

**Figura 7.** Exemplo de aplicação do Filtro de Bordas.



Um exemplo de uma Característica de Linhas em um rosto humano seria a ponte do nariz. De acordo com Gupta (2019), a ponte se destaca como uma região mais clara em relação às suas laterais:

**Figura 8.** Exemplo de aplicação do Filtro de Bordas.



Em cada região delimitada pelo filtro, se calcula a luminosidade dos pixels ali agrupados. A luminosidade de um pixel se dá pela fórmula (NAYAR, 2021):  $L = (R + G + B) / 3$

Por se tratar de uma imagem em escalas de cinza, os valores de R, G e B são iguais, então é possível simplificar a fórmula para:  $L = R$

O exemplo a seguir demonstra um pedaço pequeno recortado da sobrancelha do rosto-modelo com os valores de luminosidade calculados:

**Figura 9.** Exemplo de valores de luminosidade de pixels calculados.

34	41	38	58	56	Região escura do filtro
56	55	67	63	57	
96	92	91	93	72	Região clara do filtro
110	102	119	130	116	

O algoritmo de Viola e Jones (2003) se dá pela fórmula da Figura 10:

**Figura 10.** Algoritmo de Viola-Jones.

$$\Delta = \sum_{\text{claros}}^n I(x) - \sum_{\text{escuros}}^n I(x)$$

O algoritmo de Viola e Jones (2003) irá utilizar a diferença entre os valores das regiões claras e escuras do filtro para definir se a região onde o filtro foi aplicado é uma característica de *Haar* (delta alto) ou não (delta baixo). Seguindo o modelo identificado na imagem 7, a fórmula seria:  $\Delta = 1021 - 525 = 496$

Como indicado pelo professor Shree Nayar (2021), quanto maior o valor de  $\Delta$ , mais a região se aproxima do caso-perfeito do Filtro de *Haar* e maior a chance de ser uma Característica de *Haar*.

A partir desse cálculo já é possível verificar uma qualidade-chave das Características de *Haar*: é calculado utilizando somente somas, o que é, computacionalmente, significativamente mais barato do que multiplicações e divisões. O custo computacional da aplicação de *Haar* nas dimensões  $N \times M$  em uma imagem será de:  $C = N \times M - 1$  somas por filtro.

Viola e Jones (2003), torna, ainda, esse cálculo mais eficiente com o uso de Imagens Integrais (*Integral Images*, em inglês), um método que foi trazido para a área de computação gráfica em 1984 por Franklin C. Crow.

De acordo com Crow (1984), uma imagem integral é uma tabela com a soma de todos os pixels à esquerda e acima de um pixel, incluindo ele mesmo.

Utilizando os valores de luminosidade na Figura 9, a imagem integral seria:

**Figura 11.** Resultado do cálculo de Imagem Integral

34	41	38	58	56	34	75	113	171	227
56	55	67	63	57	90	186	291	412	525
96	92	91	93	72	186	374	570	784	969
110	102	119	130	116	296	586	901	1245	1546
<b>I. Imagem original</b>					<b>II. Imagem Integral</b>				

Com a imagem integral, agora é possível calcular o valor da luminosidade em retângulos arbitrários na imagem com uma soma simples definida por Crow (1984). Por exemplo, considere que, na imagem abaixo, precisa-se calcular o valor de luminosidade na região quadrada de C4 a D5:

**Figura 12.** Imagem 9 (anterior) com a delimitação da região quadrada de C4 a D5.

	1	2	3	4	5
A	34	75	113	171	227
B	90	186	291	412	525
C	186	374	570	784	969
D	296	586	901	1245	1546

O valor de luminosidade da região destacada se dará pela fórmula:  $L = D5 - D3 - B5 + B3$

Agora, se o Filtro de *Haar* for aplicado sobre uma Imagem Integral, o cálculo será mais simples comparado ao original, pois como visto no exemplo acima, a luminosidade de uma região inteira de pixel pode ser calculada com apenas 3 adições. Considere o exemplo abaixo como sendo a imagem integral da imagem 9, mencionada anteriormente:

**Figura 13.** Resultado da Imagem Integral da imagem 7 com as regiões de filtro destacadas.

	1	2	3	4	5	
A	34	75	113	171	227	Região escura do filtro
B	90	186	291	412	525	
C	186	374	570	784	969	Região clara do filtro
D	296	586	901	1245	1546	

O valor da luminosidade na região escura do filtro será o valor de B5 (525), pois este já contém a soma de luminosidade de todos os pixels acima e à esquerda desta posição. O valor da luminosidade na região clara do filtro se dará pelo cálculo:  $L_{clara} = D5 - B5 \rightarrow L_c = 1546 - 525 = 1021$

Portanto o delta das luminosidades se dará pelo cálculo:  $\Delta = 1021 - 525 = 496$

Aplicam-se os filtros relevantes para *Haar features* (como identificados na figura 6), o que irá resultar em um vetor de deltas de luminosidade. Este vetor é aplicado no classificador *Haar* previamente treinado que, por meio do Algoritmo KNN (*K-Nearest Neighbors* ou K-Vizinhos mais próximos), irá definir se essas características formam um rosto ou não (NAYAR, 2021).

### Reconhecendo um Rosto

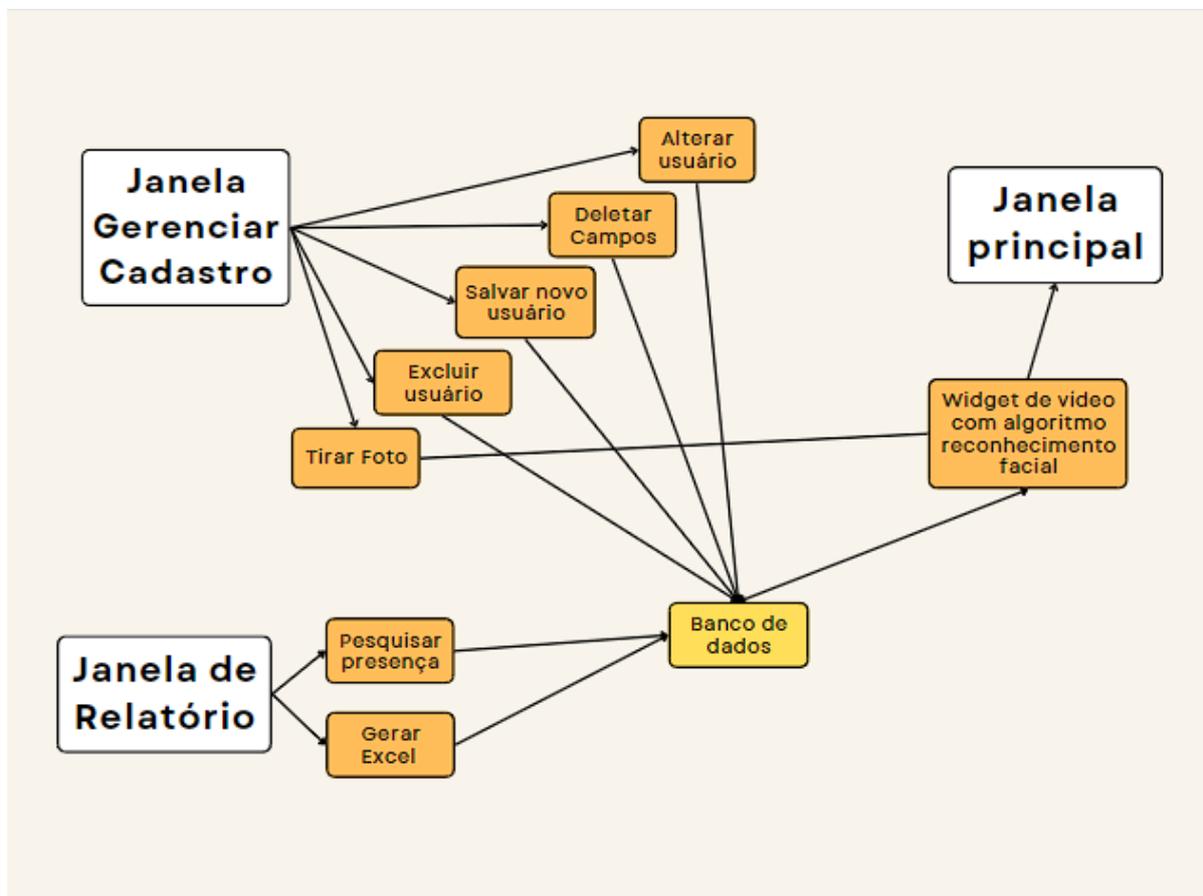
De acordo com o professor Nayar (2021), com as características do rosto calculadas, é possível verificar a distância delas para cada um dos rostos conhecidos pelo sistema, ou seja, rostos previamente cadastrados. Levando em consideração uma margem de erro aceitável, o rosto que tiver a menor diferença dentro da margem será considerado como o rosto da pessoa dona da imagem.

## RESULTADOS E DISCUSSÃO

Este trabalho apresenta o desenvolvimento de um sistema em *Python* de chamada escolar, com reconhecimento facial, utilizando, principalmente, a biblioteca *face\_recognition*, desenvolvida por Adam Geitgey.

O funcionamento do sistema proposto pelo Artigo pode ser descrito pelo fluxograma da Figura 14:

**Figura 14.** Fluxograma - identificando o funcionamento do sistema desenvolvido.



A janela principal do sistema contém o componente que abre a câmera principal da máquina e utiliza seus *frames* para detectar rostos de possíveis alunos para computar a presença na aula. A figura 15 demonstra essa tela:

**Figura 15.** Tela Principal



Sistema reconhecendo a  
aluna Paula



Sistema reconhecendo a  
aluna Paula, mesmo com  
óculos

Na figura 16 é demonstrada a tela de Gestão de Estudantes, na qual é possível cadastrar, editar e excluir estudantes do sistema. As informações utilizadas no cadastro são: RA, nome e foto (para reconhecimento facial):

**Figura 16.** Tela de Gestão de Estudantes

Gerenciar Estudantes

GERENCIAR CADASTRO

RA: 1902879

Nome: Paula Akemi Tanaka

Tirar Foto

ESTUDANTES CADASTRADOS

RA	Name
5433575	Tom holland? nao
2312355	Tom Hanks
2301231	Jorge Aragao
1901246	Paula Akemi Tanaka
1800935	Luiz
1201231	Cynci Lauper
1200146	Victor Oliveira

Na figura 17 é demonstrada a tela de Relatório de Presença, que lista todas as presenças computadas pelo sistema com base nos rostos detectados. Nesta janela é possível visualizar quais alunos estiveram presentes em datas específicas e, também, gerar um arquivo em formato Excel para análises extras:

**Figura 17.** Tela de Relatório de Presença.

	01/11/2022	02/11/2022	03/11/2022	07/11/2022	10/11/2022	11/11/2022	12/11/2022	13/11/2022	14/11/2022	15/11/2022	16/11/2022	17/11/2022	18/11/2022
1200146	presente	presente	presente	faltou	faltou	presente	prese						
1201231	faltou												
1800935	faltou	faltou	faltou	presente	presente	faltou							
1901246	faltou												
1902879	faltou												
2301231	faltou												
5433575	faltou												

A classe principal do código desenvolvido é chamada *VideoWidget* e possui o método *myLoop()*, que é dividido em três etapas.

A primeira etapa consiste na captura e tratamento de imagem, ações possibilitadas pelo uso da biblioteca *OpenCV*. A captura é feita com a junção de dois métodos:

- 1: `camera = cv2.VideoCapture(0)`
- 2: `_, frame = camera.read()`
- 3: `frameCinza = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)`

O código acima acessa a câmera do dispositivo (linha 1), captura o *frame* atual (linha 2) e converte a imagem para escala de cinza (linha 3).

A próxima etapa consiste em detectar um rosto na imagem resultante. Isso é possível com o uso de dois métodos:

- 4: `cascade = cv2.CascadeClassifier('./db/haarcascade_frontalface_alt2.xml')`
- 5: `faces = cascade.detectMultiScale(frameCinza, 1.1, 4)`

O código acima carrega um classificador de *Haar* já treinado (linha 4) e lista todos os rostos detectados no *frame* tratado no código anterior (linha 5). Isso é utilizado para que o sistema possa informar ao usuário se o rosto foi detectado ou não.

A terceira etapa consiste em comparar o rosto encontrado com os rostos cadastrados no banco de dados. Para isso, são utilizados dois métodos:

```
6: rostoCodificado = fr.face_encodings(frameCinza)[0]
```

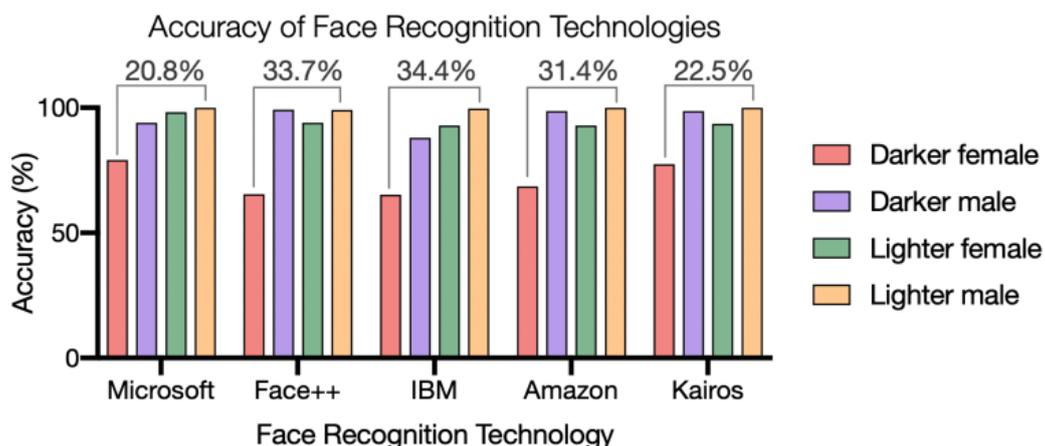
```
7: correspondencias = fr.compare_faces(rostosCadastrados, rostoCodificado)
```

O código acima codifica o rosto no *frame* em escalas de cinza (linha 6) para que possa ser utilizado no método de comparação de rostos (linha 7), no qual o primeiro parâmetro é a lista de rostos cadastrados no banco de dados e o segundo parâmetro é o próprio rosto codificado. A variável correspondente é uma lista de *booleanos*, o qual será verdadeiro no índice do rosto cadastrado que corresponder ao rosto codificado. Com esse resultado, é possível registrar a presença do(a) aluno(a) detectado(a).

Como indicado por Andrezza (2015), RF abrange uma larga escala de funções, principalmente voltadas à segurança, e essa implementação já é vista em diversas áreas: desde condomínios residenciais até instituições financeiras.

De acordo com uma pesquisa realizada por Joy Buolamwini e Timmit Gebru, em 2018, os algoritmos tendem a falhar drasticamente, principalmente com mulheres negras entre 18 e 30 anos, como visto na figura 18 abaixo:

**Figura 18.** Auditoria de 5 tecnologias de reconhecimento facial. Fonte: (Harvard University, 2020)



## CONSIDERAÇÕES FINAIS

Apesar da sua alta relevância, atualmente os algoritmos de reconhecimento facial não estão preparados para lidar com a pluralidade de raças e etnias pelo mundo. Como visto na figura 18, os algoritmos tendem a falhar drasticamente, principalmente com mulheres negras entre 18 e 30 anos. Em empresas como Microsoft e IBM já existem projetos que visam a melhorar estes classificadores, com o intuito de diminuir a disparidade apontada nos testes. Espera-se que projetos como estes sejam contínuos e progressivos, pois com o aumento e a migração de serviços cada vez maiores para o âmbito digital, é preciso garantir maior igualdade perante todos, para que os sistemas os reconheçam de forma eficaz.

## REFERÊNCIAS BIBLIOGRÁFICAS

- ANDREZZA, Igor Lucena Peixoto. *Análise de técnicas de normalização aplicadas ao reconhecimento facial*, 2015. 75 f. Dissertação (Programa de Pós-Graduação em Informática) - Universidade Federal da Paraíba, João Pessoa, 2015.
- BUOLAMWINI, Joy; GEBRU, Timnit. Gender shades: Intersectional accuracy disparities in commercial gender classification. In: *Conference on fairness, accountability and transparency*. PMLR, 2018. p. 77-91.
- CROW, Franklin. *Summed-area tables for texture mapping*. SIGGRAPH '84: Proceedings of the 11th annual conference on Computer graphics and interactive techniques. p. 207–212, 1984.
- FRANÇA, Tiago Cruz et al. Big Social Data: Princípios sobre coleta, tratamento e análise de dados sociais. *XXIX Simpósio Brasileiro de Banco de Dados–SBBDD*, v. 14, 2014.
- GUPTA, Rohan. *Breaking Down Facial Recognition: The Viola-Jones Algorithm*. [S. l.], 6 ago. 2019. Disponível em: <https://towardsdatascience.com/the-intuition-behind-facial-detection-the-viola-jones-algorithm-29d9106b6999>. Acesso em: 24 nov. 2022.
- HAFED, Ziad M.; LEVINE, Martin D. Face recognition using the discrete cosine transform. *International Journal of Computer Vision*, v. 43, n. 3, p. 167-188, 2001.
- MARINHO, Adriano da Silva. *Uma nova versão de um sistema de detecção e reconhecimento de face utilizando a Transformada Cosseno Discreta*. 2012. 80 f. Dissertação (Mestrado em Informática) - Universidade Federal da Paraíba, João Pessoa, 2012.
- MARKOFF, John. Entrepreneurs see a web guided by common sense. *The New York Times*, nov. 2006. Tradução por Fabiano Caruso. Disponível em: [https://www.mailarchive.com/bib\\_virtual@ibict.br/msg01199.html](https://www.mailarchive.com/bib_virtual@ibict.br/msg01199.html). Acesso em: 10 nov. 2022.
- NAYAR, Shree. *First Principles of Computer Vision*, 2021. Disponível em <https://fpcv.cs.columbia.edu/>. Acesso em: 12 nov. 2022.
- NOGUEIRA, Gustavo Rodrigues Guerra; DOS SANTOS, Felipe Gonçalves. Desenvolvimento de protótipo de fechadura eletrônica com Reconhecimento Facial. In: *Anais da VII Escola Regional de Informática de Goiás*. SBC, 2019. p. 333-339.
- NAJIBI, Alex. *Racial Discrimination in Face Recognition Technology*. [S. l.], 24 out. 2020. Disponível em: <https://sitn.hms.harvard.edu/flash/2020/racial-discrimination-in-face-recognition-technology/>. Acesso em: 2 dez. 2022.

SILVA, Rosane Leal da; SILVA, Fernanda dos Santos Rodrigues da. Reconhecimento facial e segurança pública: os perigos do uso da tecnologia no sistema penal seletivo brasileiro. In: *Congresso Internacional de Direito e Contemporaneidade*, Santa Maria, RS, Brasil. 2019.

SHERMINA, J. Illumination invariant face recognition using discrete cosine transform and principal component analysis. In: *2011 International Conference on Emerging Trends in Electrical and Computer Technology*. IEEE, 2011. p. 826-830.

IOLA, Paul; MICHAEL J., Jones. *Robust Real-Time Face Detection*. International Journal of Computer Vision, [S. l.], p. 1-19, 11 jul. 2003.

ZHAO, Wenyi et al. Face recognition: A literature survey. *ACM computing surveys (CSUR)*, v. 35, n. 4, p. 399-458, 2003.

## **AGRADECIMENTOS**

Agradecemos aos professores por se dedicarem a passar seus conhecimentos a nós, aos amigos e familiares pela paciência e apoio, ao nosso comprometimento com o curso, e todos aqueles que, de alguma forma, contribuíram para este momento e não estão mais entre nós.