Tecnologias de Reconhecimento Facial e Biometria



Prof. Msc. Francisco de Assis de Souza Rodrigues

UniFAP
CENTRO UNIVERSITÁRIO PARAÍSO

🎯 Objetivo

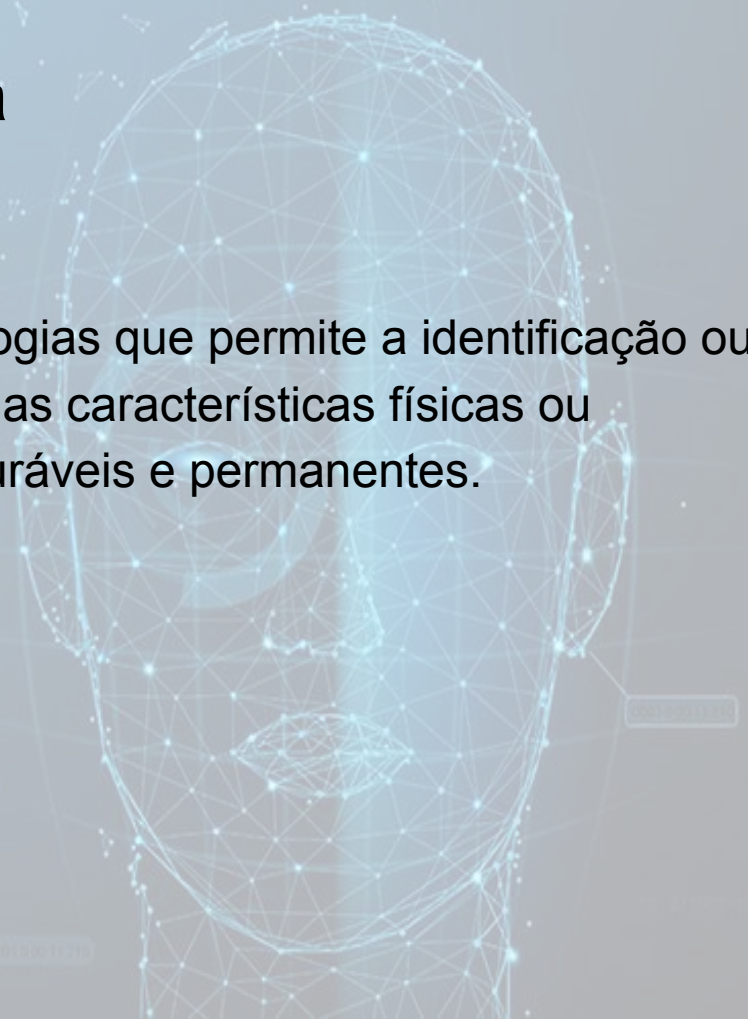
- Compreender os fundamentos das tecnologias de reconhecimento facial e biometria.
- Explorar os principais algoritmos utilizados em reconhecimento facial.
- Implementar uma solução básica de reconhecimento facial usando Python e OpenCV.



🧠 1. Introdução à Biometria

O que é?

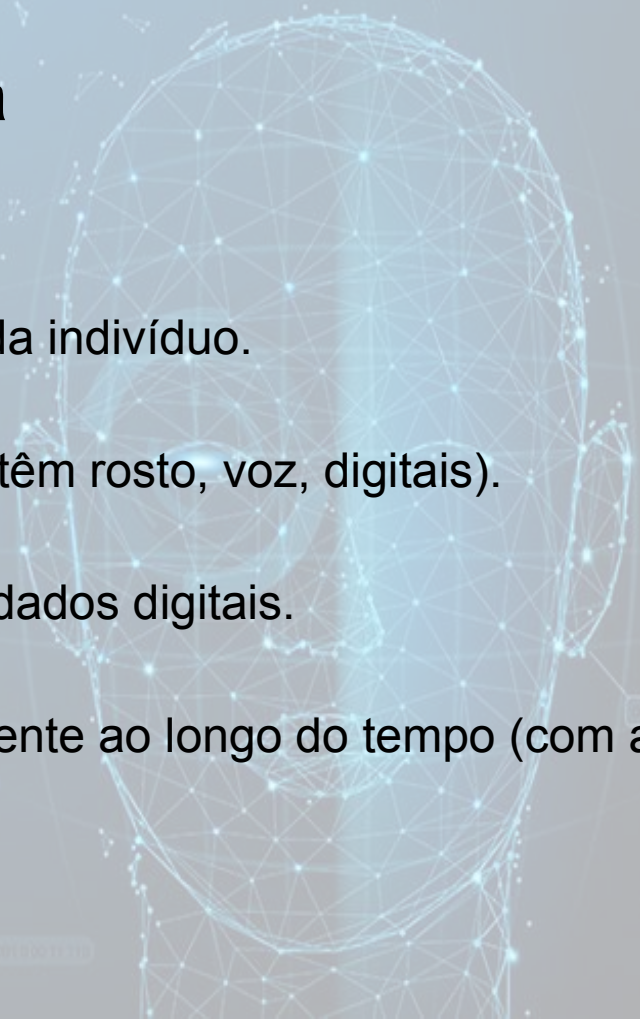
Biometria é um conjunto de tecnologias que permite a identificação ou autenticação de indivíduos com base em suas características físicas ou comportamentais, que sejam únicas, mensuráveis e permanentes.



🧠 1. Introdução à Biometria

🚀 Características da biometria:

- **Única:** Característica específica de cada indivíduo.
- **Universal:** Todos possuem (ex.: todos têm rosto, voz, digitais).
- **Mensurável:** Deve ser convertível em dados digitais.
- **Permanente:** Não muda significativamente ao longo do tempo (com algumas exceções).



🧠 1. Introdução à Biometria

🔍 Tipos de Biometria

🔹 Biometria Fisiológica

Relacionada a características físicas do corpo:

- **Impressão digital:** Uma das mais usadas, baseada nos sulcos da pele.
- **Reconhecimento facial:** Análise de geometria do rosto.
- **Íris:** Padrões únicos na íris do olho.
- **Retina:** Padrões de vasos sanguíneos no fundo do olho.
- **Reconhecimento de palma da mão:** Geometria e linhas da mão.
- **Geometria da mão ou dos dedos.**

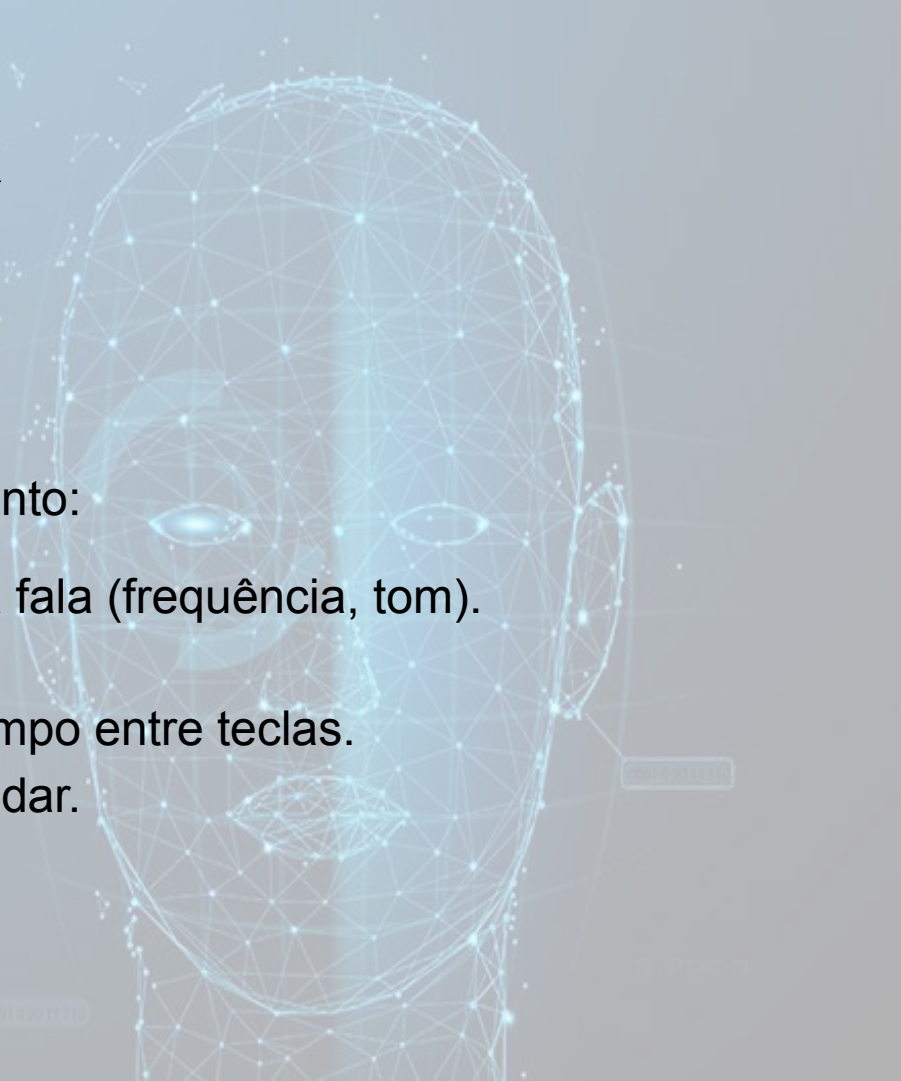
🧠 1. Introdução à Biometria

🔍 Tipos de Biometria

🔹 Biometria Comportamental

Relacionada a padrões de comportamento:

- **Voz:** Análise das características da fala (frequência, tom).
- **Assinatura:** Padrões na escrita.
- **Padrões de digitação:** Ritmo e tempo entre teclas.
- **Padrões de marcha:** Forma de andar.



🧠 1. Introdução à Biometria

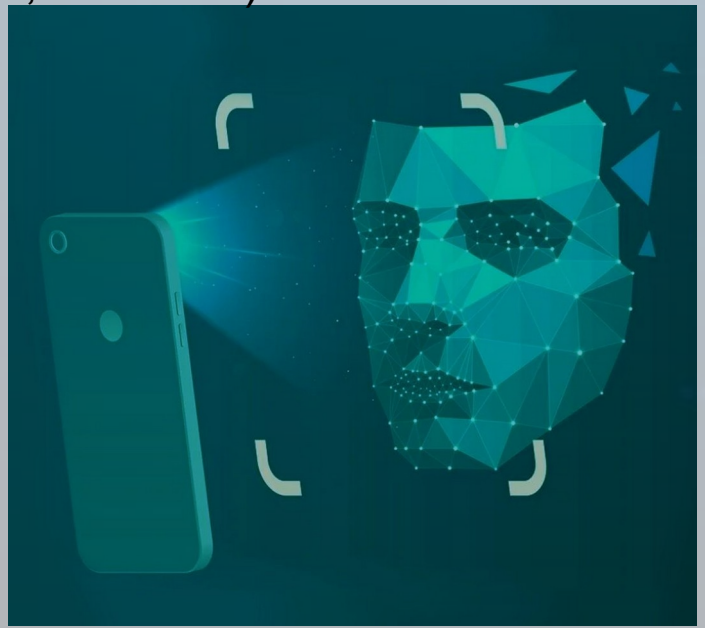
🏆 Vantagens da Biometria

- Conveniente (não depende de senhas).
- Difícil de falsificar (mas não impossível).
- Alta segurança quando bem implementada.



🤖 2. Reconhecimento Facial

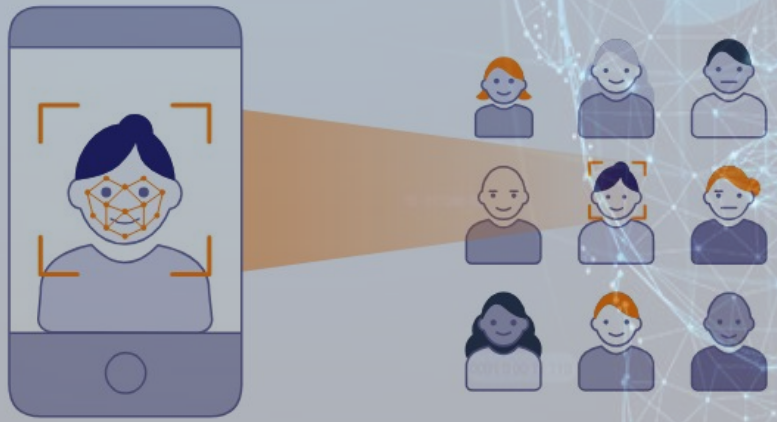
O reconhecimento facial é uma das formas mais comuns de biometria, pois não requer interação física e pode ser realizado remotamente (ex.: câmeras de vigilância, celulares).



🤖 2. Reconhecimento Facial

🏗️ Etapas do Funcionamento:

1. Detecção Facial
2. Alinhamento Facial
3. Extração de Características (Feature Extraction)
4. Comparação



🤖 2. Reconhecimento Facial

🏗️ Etapas do Funcionamento:

🔸 1. Detecção Facial

- Localizar rostos em uma imagem ou vídeo.
- Saída: Coordenadas (x, y, largura, altura) dos rostos detectados.

Ferramentas:

- Haar Cascades (OpenCV)
- HOG (Dlib)
- MTCNN (Deep Learning)



🤖 2. Reconhecimento Facial

- **2. Alinhamento Facial**
  - Corrigir rotação, inclinação ou escala do rosto.
  - Alinha olhos, nariz e boca para padronizar a imagem antes da análise.
  - Importante para melhorar a precisão do reconhecimento.



🤖 2. Reconhecimento Facial

🔸 **3. Extração de Características (Feature Extraction)**

- O rosto é transformado em um **vetor numérico (embedding)** que representa suas características únicas.
- Esse vetor é normalmente de alta dimensão (ex.: 128 ou 512 elementos).
- Esse vetor não é uma imagem, mas uma representação matemática.

🤖 2. Reconhecimento Facial

🔸 **4. Comparação**

- Compara-se os vetores para determinar se dois rostos são da mesma pessoa.

- Técnicas comuns:
  - **Distância Euclidiana:** Mede a "distância" entre dois pontos no espaço vetorial.
  - **Cosine Similarity:** Mede o ângulo entre os vetores (muito usada em machine learning).

- Se a distância for menor que um limite (threshold), considera-se uma correspondência.

🔍 3. Algoritmos e Modelos Populares

⚙️ Modelos Tradicionais (Machine Learning Clássico)

🔸 Haar Cascades (OpenCV)

- Algoritmo baseado em padrões simples de pixels (retângulos claros e escuros).
- Rápido, leve e eficiente para aplicações básicas.
- Baixa precisão em condições adversas (luz, ângulos, expressões).

🔍 3. Algoritmos e Modelos Populares

⚙️ Modelos Tradicionais (Machine Learning Clássico)

🔸 HOG + SVM (Dlib)

- HOG (Histogram of Oriented Gradients) extrai características baseadas em gradientes.
- SVM (Support Vector Machine) faz a classificação.
- Melhor que Haar em robustez, mas ainda limitado comparado a redes neurais.

🔍 3. Algoritmos e Modelos Populares

🤖 Modelos Baseados em Deep Learning (CNN)

- Muito mais precisos, robustos a variações de luz, ângulo e expressão.

🔹 FaceNet (Google)

- Transforma imagens faciais em vetores de 128 dimensões.
- Usa uma rede neural treinada para otimizar que rostos semelhantes fiquem próximos no espaço vetorial.

🔍 3. Algoritmos e Modelos Populares

🤖 Modelos Baseados em Deep Learning (CNN)

🔹 DeepFace (Meta/Facebook)

- Um dos primeiros modelos que ultrapassou a precisão humana em certos benchmarks.

- CNN profunda com múltiplas camadas de convolução.

🔍 3. Algoritmos e Modelos Populares

🤖 Modelos Baseados em Deep Learning (CNN)

🔹 Dlib ResNet

- Implementa uma rede ResNet leve que gera embeddings faciais de 128 dimensões.
- Usado por muitas bibliotecas de face recognition no Python.

🔍 3. Algoritmos e Modelos Populares

🤖 Modelos Baseados em Deep Learning (CNN)

🔹 MTCNN (Multi-task Cascaded Convolutional Networks)

- Focada na **detecção facial**, não no reconhecimento.

- Detecta e alinha rostos de forma extremamente precisa, mesmo com rostos em ângulos, expressões variadas ou obstruções.

🔍 3. Algoritmos e Modelos Populares

🔑 Face Embeddings

- Cada rosto é representado por um vetor.

- O modelo aprende um mapeamento de imagens faciais para um espaço vetorial onde:
  - Rostos da **mesma pessoa** estão **próximos**.
  - Rostos de **pessoas diferentes** estão **distantes**.

⚖️ 4. Aplicações e Desafios Éticos

🚀 Aplicações Práticas

- **Segurança e autenticação**
  - Desbloqueio facial em smartphones.
  - Controle de acesso em prédios e sistemas.

- **Vigilância pública**
  - Reconhecimento em espaços públicos (polícia, segurança).

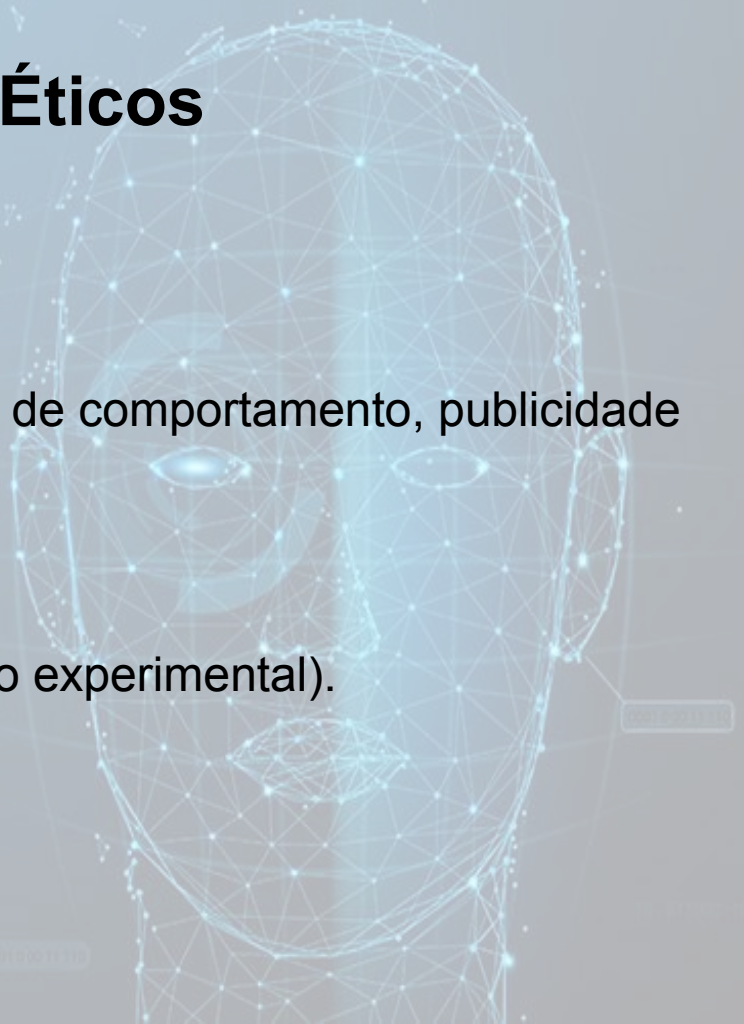


⚖️ 4. Aplicações e Desafios Éticos

🚀 Aplicações Práticas

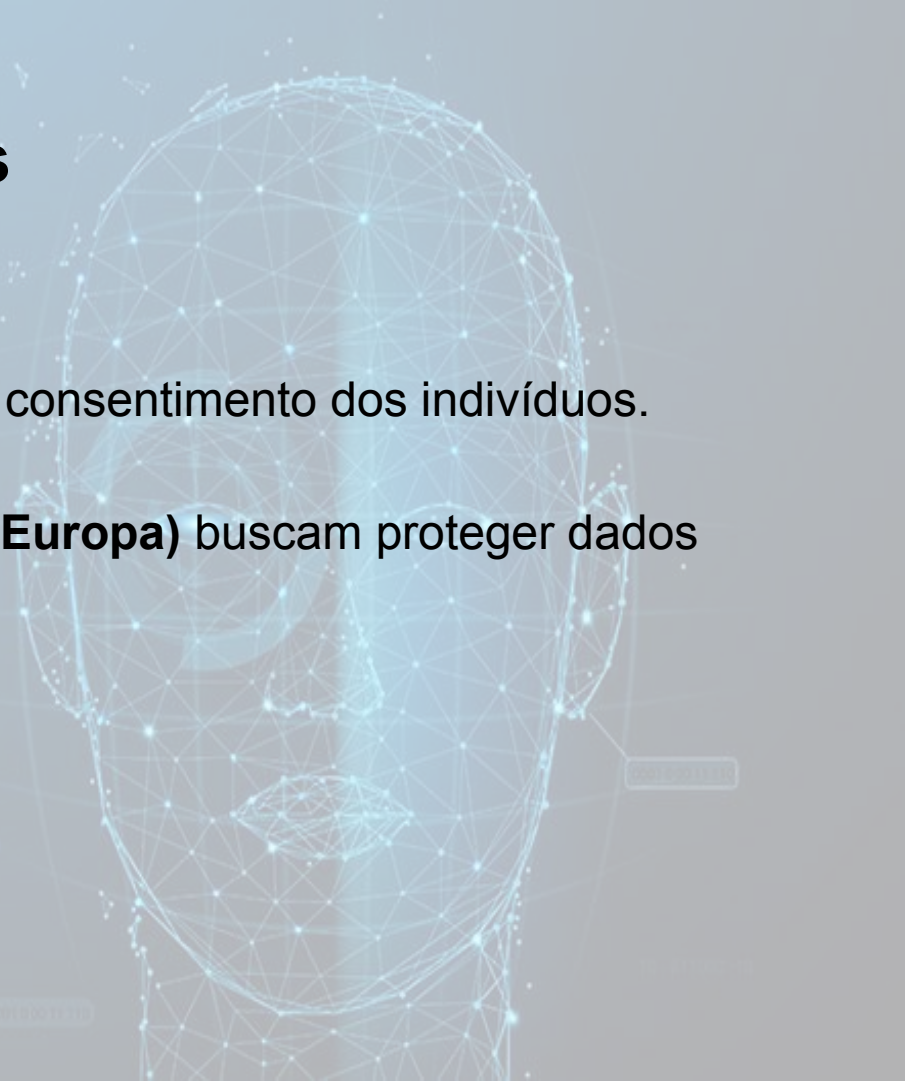
- **Comércio e marketing**
  - Monitoramento de clientes, análise de comportamento, publicidade personalizada.

- **Saúde**
  - Detecção de síndromes faciais (uso experimental).



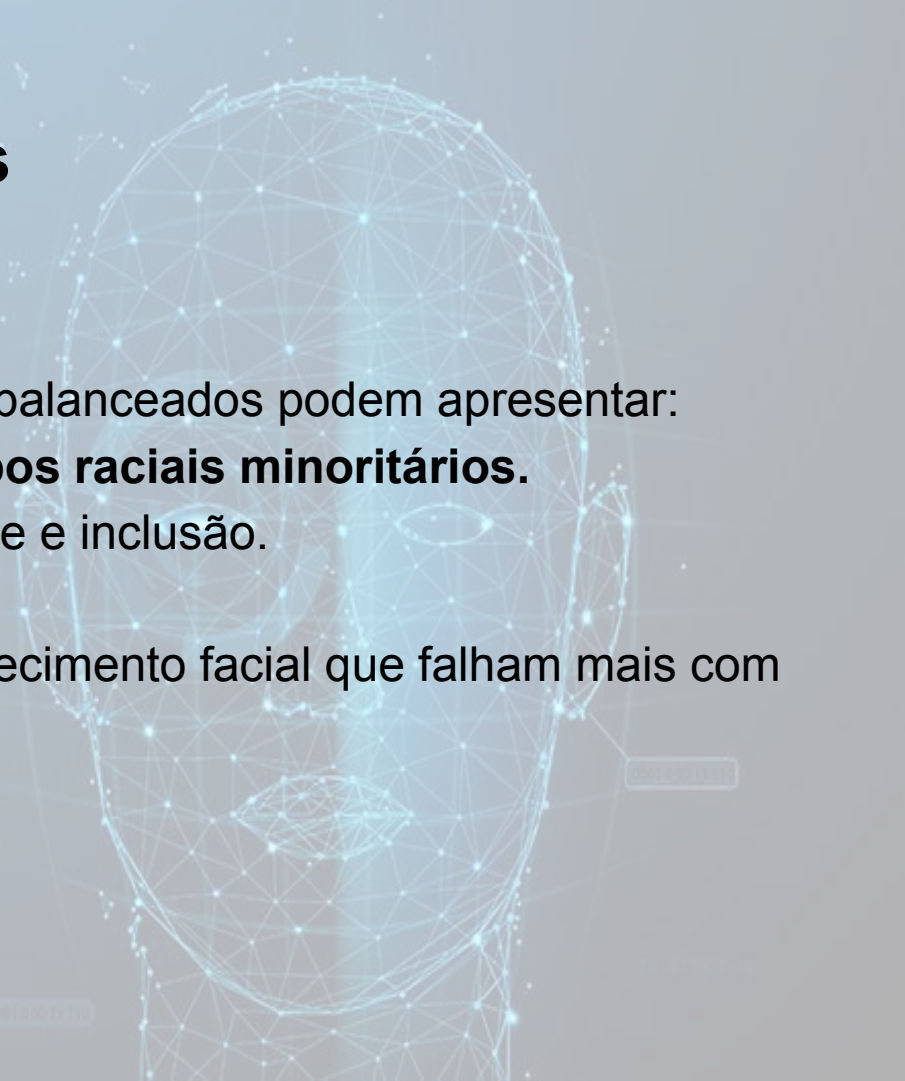
⚠️ Desafios Éticos e Sociais

- **Privacidade e Consentimento**
  - Muitas implementações ignoram o consentimento dos indivíduos.
  - Leis como **LGPD (Brasil)**, **GDPR (Europa)** buscam proteger dados biométricos.



⚠️ Desafios Éticos e Sociais

- **Viés Algorítmico**
  - Modelos treinados com dados desbalanceados podem apresentar:
    - **Maior taxa de erro para grupos raciais minoritários.**
    - Problemas de justiça, equidade e inclusão.
  - Exemplo real: Sistemas de reconhecimento facial que falham mais com pessoas negras e asiáticas.



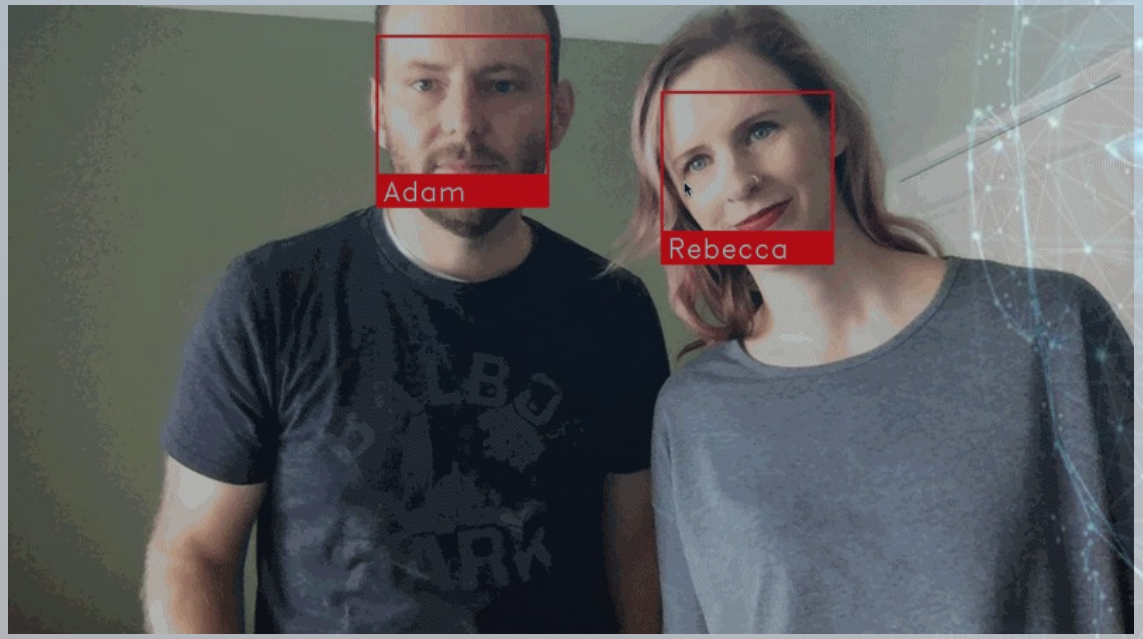
⚠️ Desafios Éticos e Sociais

- **Deepfakes e Spoofing**
  - Criação de vídeos falsos extremamente realistas (deepfakes).
  - Riscos para segurança, política, fake news e fraudes.



Biblioteca: Face Recognition

https://github.com/ageitgey/face_recognition



Leitura Complementar

SCHROFF, Florian; KALENICHENKO, Dmitry; PHILBIN, James. Facenet: A unified embedding for face recognition and clustering. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2015. p. 815-823.

https://arxiv.org/pdf/1503.03832

arXiv:1503.03832v3 [cs.CV] 17 Jun 2015

FaceNet: A Unified Embedding for Face Recognition and Clustering

Florian Schroff fschroff@google.com Google Inc.
Dmitry Kalenichenko dkalenichenko@google.com Google Inc.
James Philbin jphilbin@google.com Google Inc.

Abstract

Despite significant recent advances in the field of face recognition [10, 14, 15, 17], implementing face verification and recognition efficiently at scale presents serious challenges to current approaches. In this paper we present a system, called FaceNet, that directly learns a mapping from face images to a compact Euclidean space where distances directly correspond to a measure of face similarity. Once this space has been produced, tasks such as face recognition, verification and clustering can be easily implemented using standard techniques with FaceNet embeddings as feature vectors.

Our method uses a deep convolutional network trained to directly optimize the embedding itself, rather than an intermediate bottleneck layer as in previous deep learning approaches. To train, we use triplets of roughly aligned matching / non-matching face patches generated using a novel online triplet mining method. The benefit of our approach is much greater representational efficiency: we achieve state-of-the-art face recognition performance using only 128-bytes per face.

*On the widely used Labeled Faces in the Wild (LFW) dataset, our system achieves a new record accuracy of **99.63%**. On YouTube Faces DB it achieves **95.12%**. Our system cuts the error rate in comparison to the best published result [15] by 30% on both datasets.*

We also introduce the concept of harmonic *embeddings, and a* harmonic *triplet loss, which describe different versions of face embeddings (produced by different networks) that are compatible to each other and allow for direct comparison between each other.*

1. Introduction

In this paper we present a unified system for face verification (is this the same person), recognition (who is this person) and clustering (find common people among these faces). Our method is based on learning a Euclidean embedding per image using a deep convolutional network. The network is trained such that the squared L2 distances in the embedding space directly correspond to face similarity:

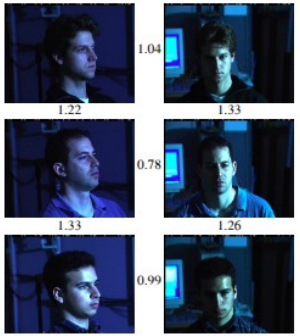


Figure 1. **Illumination and Pose invariance.** Pose and illumination have been a long standing problem in face recognition. This figure shows the output distances of FaceNet between pairs of faces of the same and a different person in different pose and illumination combinations. A distance of 0.0 means the faces are identical, 4.0 corresponds to the opposite spectrum, two different identities. You can see that a threshold of 1.1 would classify every pair correctly.

faces of the same person have small distances and faces of distinct people have large distances.

Once this embedding has been produced, then the aforementioned tasks become straight-forward: face verification simply involves thresholding the distance between the two embeddings; recognition becomes a k-NN classification problem; and clustering can be achieved using off-the-shelf techniques such as k-means or agglomerative clustering.

Previous face recognition approaches based on deep networks use a classification layer [15, 17] trained over a set of known face identities and then take an intermediate bottle-

