ÂNIMA EDUCAÇÃO

ARIEL SOARES FRANCO (RA: 12722210594)

GUSTAVO MAURI PINTO (RA: 12924112023)

MANUELLE REIS SANTANA (RA: 1272116405)

PATRICK RUBNER (RA: 1292212379)

TEORIA DA COMPUTAÇÃO E COMPILADORES (A3)

2025

ARIEL SOARES FRANCO (RA: 12722210594)
GUSTAVO MAURI PINTO (RA: 12924112023)
MANUELLE REIS SANTANA (RA: 1272116405)
PATRICK RUBNER (RA: 1292212379)

TEORIA DA COMPUTAÇÃO E COMPILADORES (A3)

Relatório do projeto realizado pelos alunos da turma da UC de Teoria da Computação e compiladores, apresentado ao professor responsável como requisito parcial de avaliação do semestre.

Orientador: Prof. Matusalém Carlos Rosa

2024

SUMÁRIO

1 INTRODUÇÃO

A segurança, tanto pública quanto patrimonial, tem se tornado um dos principais desafios da sociedade contemporânea. Em um cenário marcado pelo aumento da criminalidade e pela crescente sensação de insegurança, a identificação e rastreabilidade de indivíduos em ambientes públicos tornaram-se fundamentais para a prevenção e resolução de ocorrências. No entanto, a ausência de mecanismos eficientes para monitoramento e reconhecimento de pessoas com histórico judicial ou envolvimento em atividades ilícitas contribui significativamente para a vulnerabilidade desses espaços.

Nesse contexto, o uso de tecnologias baseadas em visão computacional, como o reconhecimento facial, surge como uma ferramenta promissora. Esses sistemas permitem automatizar a identificação de indivíduos a partir de imagens, promovendo maior controle, rastreamento e resposta rápida a situações de risco. Este trabalho propõe o desenvolvimento de um sistema simples de reconhecimento facial aplicado à análise de imagens, com foco na identificação de pessoas previamente cadastradas, possibilitando sua utilização em contextos diversos que demandam reforço na segurança e monitoramento.

2 PROPOSTA

- Carregamento automático de imagens contendo rostos conhecidos a partir de uma pasta chamada `faces`.
- Leitura de uma imagem de entrada a partir da pasta `frames`.
- Detecção dos rostos presentes na imagem de entrada.
- Reconhecimento dos rostos detectados comparando-os com os rostos conhecidos.
- Exibição da imagem com caixas ao redor dos rostos detectados, nomes dos indivíduos reconhecidos e percentual de confiança.
- Diferenciação visual usando cores para indicar o nível de confiança do reconhecimento:
    - Verde para rostos conhecidos com confiança ≥ 90%.
    - Roxo para rostos conhecidos com confiança < 90%.
    - Vermelho para rostos desconhecidos.
- Exibição da imagem final redimensionada para melhor visualização.

3 MÉTODOS APLICADOS

Categoria	Tecnologia	Versão	Função no Sistema
Linguagem de Programação	Python	3.13	Linguagem principal usada para desenvolvimento do sistema e integração das bibliotecas.
SGBD	SQLite	-	Sistema de banco de dados leve utilizado para armazenar informações locais, como registros de usuários, logs de detecção e metadados.
Biblioteca de Visão Computacional	OpenCV-Python	4.10.0.84	Usada para capturar, processar e exibir imagens, além de desenhar as caixas de identificação facial.
Biblioteca de Reconhecimento Facial	face-recognition	1.3.0	Responsável por detectar e codificar rostos, além de comparar rostos detectados com os conhecidos.
Modelos Pré-Treinados	face-recognition-models	≥ 0.3.0	Conjunto de modelos de machine learning usados internamente pela biblioteca `face-recognition` para realizar as detecções faciais com alta precisão.

4 REQUISITOS

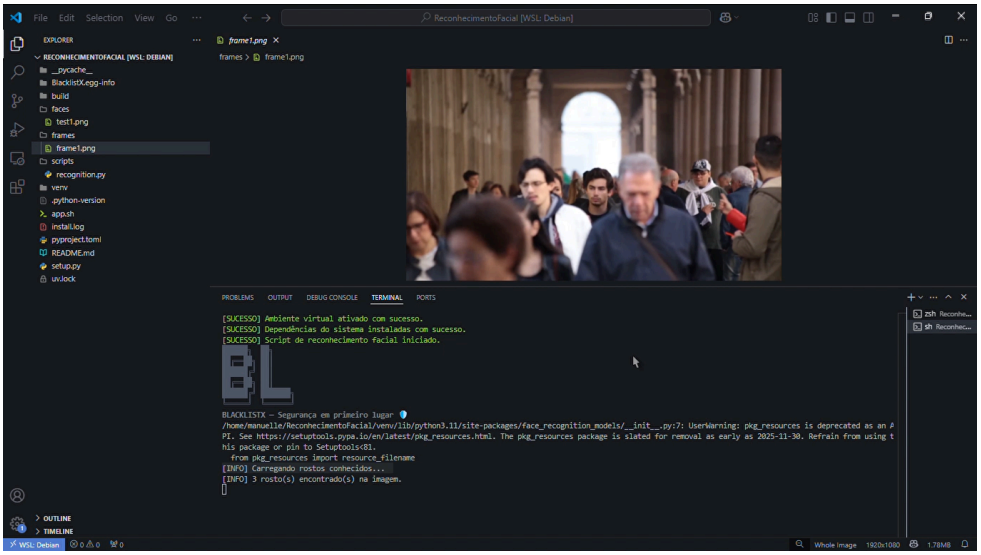
4.1 Requisitos Funcionais

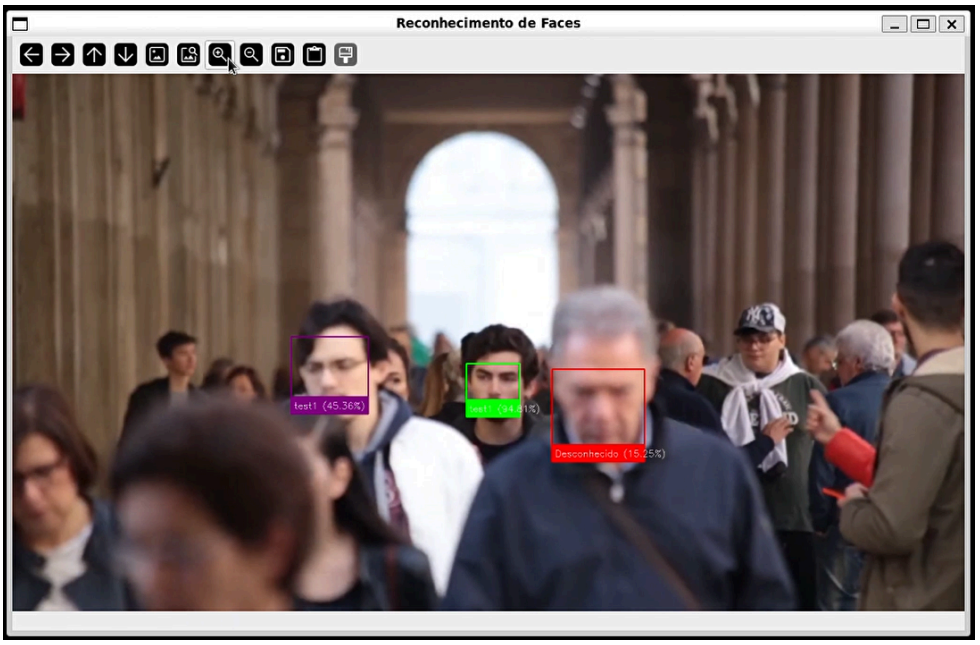
D	Requisito	Descrição
F01	Carregar imagens conhecidas	O sistema deve carregar todas as imagens da pasta `faces` e extrair suas codificações faciais.
F02	Carregar imagem de análise	O sistema deve carregar uma única imagem da pasta `frames` para reconhecimento.
F03	Detectar rostos na imagem	O sistema deve localizar todos os rostos presentes na imagem de entrada.
F04	Realizar reconhecimento facial	O sistema deve comparar as faces detectadas com as faces conhecidas para identificar indivíduos.
F05	Calcular percentual de similaridade	O sistema deve calcular um percentual que indica o nível de confiança no reconhecimento.
F06	Indicar visualmente o reconhecimento	O sistema deve desenhar caixas ao redor dos rostos, mostrando nomes e porcentagens.
F07	Diferenciar níveis de confiança por cor	Verde para confiança ≥ 90%, roxo para confiança < 90%, vermelho para desconhecidos.
F08	Redimensionar a imagem para exibição	A imagem final deve ser exibida em uma janela redimensionada, mantendo proporção e no máximo 1000px.

4.2 Requisitos Não Funcionais

D	Requisito	Descrição
NF1	Linguagem	O sistema deve ser implementado em Python.
NF2	Bibliotecas	Utilização das bibliotecas `face_recognition`, `opencv-python`, `numpy` e `os`.
NF3	Plataforma	O sistema deve rodar em ambientes Windows, Linux ou macOS que suportem as bibliotecas mencionadas.
NF4	Interface	Interface gráfica simples via janela OpenCV para exibição dos resultados.
NF5	Performance	O processamento deve ser feito em tempo real para imagens estáticas, com feedback imediato.

5 APLICAÇÃO E SEUS RESULTADOS





1. Inicialização do script sh.
2. Carregar as imagens conhecidas e extrair seus vetores de face.
3. Carregar a imagem de entrada.
4. Detectar rostos na imagem.
5. Para cada rosto detectado, calcular similaridade com rostos conhecidos.

6. Determinar se o rosto é conhecido ou desconhecido, e definir a cor da caixa conforme a confiança.
7. Desenhar as caixas, nomes e porcentagens na imagem.
8. Redimensionar e exibir a imagem com os resultados.
9. Aguardar interação do usuário para fechar a janela.

6 CONSIDERAÇÕES

- As imagens na pasta `faces` devem conter somente uma face clara para garantir precisão nas codificações.
- O sistema considera apenas uma imagem de entrada por execução, que deve estar na pasta `frames`.
- O percentual de confiança é baseado na distância entre as codificações faciais, conforme função `face_distance`.
- O sistema pode ser estendido para múltiplas imagens ou vídeos no futuro.