INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE MINAS GERAIS - *CAMPUS* BETIM
BACHARELADO EM ENGENHARIA DE CONTROLE E AUTOMAÇÃO

Matheus Henrique Pereira Aganete dos Santos

# USO DE INTELIGÊNCIA ARTIFICIAL PARA AUXILIO AOS ÓRGÃOS DE SEGURANÇA PÚBLICA NA DETECÇÃO DE PESSOAS DESAPARECIDAS

Betim
2025

MATHEUS HENRIQUE PEREIRA AGANETE DOS SANTOS

# USO DE INTELIGÊNCIA ARTIFICIAL PARA AUXILIO AOS ÓRGÃOS DE SEGURANÇA PÚBLICA NA DETECÇÃO DE PESSOAS DESAPARECIDAS

Trabalho de Conclusão de Curso apresentado à banca examinadora do curso de Engenharia de Controle e Automação do Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais *Campus* Betim, como parte dos requisitos para obtenção do título de Bacharel em Engenharia de Controle e Automação.

**Orientador:** Prof. Me. Maurício Monteiro da Silva

**Coorientador:** Profª. Ma. Michelle Mendes Santos

Betim
2025

FICHA CATALOGRÁFICA

S327u Santos, Matheus Henrique Pereira Aganete dos

Uso de inteligência artificial para auxílio aos órgãos de segurança pública na detecção de pessoas desaparecidas / Matheus Henrique Pereira Aganete dos Santos. – 2025.

51 f. : il.

Trabalho de conclusão de curso (Bacharelado em Engenharia de Controle e Automação) - Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais, Câmpus Betim, 2025.

Orientação: Prof. Me. Maurício Monteiro da Silva
Coorientação: Profa. Ma. Michelle Mendes Santos

1. Inteligência artificial. 2. Reconhecimento facial. 3. Pessoas desaparecidas. 4. Visão computacional. 5. Engenharia de Controle e Automação. I Santos, Matheus Henrique Pereira Aganete dos. II. Título.

CDU: 004.8

Ficha catalográfica elaborada pelo Bibliotecário Denísio Pereira Marcos CRB-6/3142

Matheus Henrique Pereira Aganete dos Santos

# USO DE INTELIGÊNCIA ARTIFICIAL PARA AUXILIO AOS ÓRGÃOS DE SEGURANÇA PÚBLICA NA DETECÇÃO DE PESSOAS DESAPARECIDAS

Trabalho de Conclusão de Curso apresentado à banca examinadora do curso de Engenharia de Controle e Automação do Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais *Campus* Betim, como parte dos requisitos para obtenção do título de Bacharel em Engenharia de Controle e Automação.

Aprovado em: 01 / 08 / 2025 pela banca examinadora:

Prof. Me. Maurício Monteiro da Silva (Orientador) - IFMG

Profª. Me. Michelle Mendes Santos (Coorientador) - IFMG

Prof. Me. Virgil Del Duca Almeida - IFMG

Prof. Me. Felipe Augusto Rocha da Silva - IFMG

Prof. Dr. Leandro Freitas de Abreu - Universidade Federal de Minas Gerais - UFMG

# RESUMO

Este trabalho propõe o desenvolvimento de um sistema de reconhecimento facial com o objetivo de auxiliar na identificação de pessoas desaparecidas. A metodologia adotada seguiu um fluxo em quatro etapas: (i) coleta de dados utilizando a Yale Face Database, selecionada como base de dados, devido à inviabilidade de uso de dados reais; (ii) ampliação da base de dados mediante seis técnicas de *Data Augmentation*; (iii) estratificação da base de dados em subconjuntos (DB1-DB5) com variação progressiva de 1 a 9 fotos originais por pessoa e (iv) treinamento e avaliação do Eigenfaces com varredura de hiperparâmetros (threshold de similaridade e componentes principais), validando desempenho via acurácia e matriz de confusão.

O algoritmo foi treinado com diferentes combinações de parâmetros (número de componentes principais e limiar de rejeição) e testado em cinco bases distintas, variando a quantidade e variedade de imagens por indivíduo. Os resultados indicam que a acurácia do sistema está diretamente relacionada à diversidade de imagens por pessoa. A base mais completa, contendo imagens originais e artificialmente aumentadas, atingiu uma média de acurácia de 65,39%, com valor máximo de até 73,3% nas melhores configurações.

**Palavras-chave**: Reconhecimento facial. Pessoas desaparecidas. Eigenfaces. Data Augmentation. Visão computacional.

# ABSTRACT

This work proposes the development of a facial recognition system aimed at assisting in the identification of missing persons. The adopted methodology followed a four-stage workflow: (i) data collection using the Yale Face Database, selected as the benchmark dataset due to the unfeasibility of using real-world sensitive data; (ii) dataset expansion through six Data Augmentation techniques; (iii) dataset stratification into subsets (DB1-DB5) with progressive variation of 1 to 9 original photos per person; and (iv) Eigenfaces training and evaluation with hyperparameter sweep (similarity threshold and principal components), validating performance via accuracy and confusion matrix.

The algorithm was trained using different parameter combinations (number of principal components and rejection threshold) and tested on five different datasets, varying in the number and diversity of images per individual. The results indicate that the system's accuracy is directly related to the diversity of images per person. The most complete dataset, containing both original and artificially augmented images, reached an average accuracy of 65.39%, with a maximum of 73.3% under the best configurations.



**Keywords**: Facial recognition. Missing persons. Eigenfaces. Data Augmentation. Computer vision.

# LISTA DE ILUSTRAÇÕES

# LISTA DE TABELAS

# LISTA DE ABREVIATURAS E SIGLAS

| | |
|---|---|
| ABNT | Associação Brasileira de Normas Técnicas |
| IFMG | Instituto Federal de Minas Gerais |
| LGPD | Lei Geral de Proteção de Dados |
| PCMG | Polícia Civil de Minas Gerais |
| API | Interface de Programação de Aplicações (Application Programming Interface) |
| IA | Inteligência Artificial |
| SVM | Support Vector Machine (Máquina de Vetores de Suporte) |
| PCA | Principal Component Analysis (Análise de Componentes Principais) |
| 3DMM | 3D Morphable Models (Modelos Morfáveis 3D) |
| GANs | Generative Adversarial Networks (Redes Adversariais Generativas) |
| VAEs | Variational Autoencoders (Autoencoders Variacionais) |
| Eigenfaces | Técnica de reconhecimento facial baseada em componentes principais |
| LBPH | Local Binary Patterns Histograms (Técnica de reconhecimento facial) |
| FaceNet | Algoritmo de reconhecimento facial por rede neural profunda |
| Requests | Biblioteca Python para requisições HTTP |
| Beautiful Soup | Biblioteca Python para parsing de HTML/XML |
| Scrapy | Framework de Web Scraping em Python |
| imgaug | Biblioteca Python para Data Augmentation |
| YAML | Formato de serialização de dados legível por humanos |

# LISTA DE SÍMBOLOS

| | |
|---|---|
| $\Gamma_i$ | Vetor coluna da $i$-ésima imagem facial (dimensão $N^2 \times 1$) |
| $N$ | Largura (e altura) da imagem facial em pixels |
| $N^2$ | Total de pixels da imagem (vetorizada) |
| $Z$ | Número total de imagens na base de dados |
| $M$ | Matriz de dados com $Z$ linhas (imagens) e $N^2$ colunas (pixels) |
| $\Psi$ | Face média (vetor médio de todas as imagens) |
| $\Phi_i$ | Vetor da imagem $i$ centralizado: $\Phi_i = \Gamma_i - \Psi$ |
| $A$ | Matriz cujas colunas são os vetores $\Phi_i$ |
| $C$ | Matriz de covariância (calculada como $C = A^T A$) |
| $\lambda_i$ | $i$-ésimo autovalor da matriz de covariância |
| $v_i$ | $i$-ésimo autovetor da matriz de covariância (Eigenface) |
| $\alpha$ | Limiar de variância acumulada para seleção de componentes |
| $\omega_i^{(j)}$ | Peso (coeficiente) da imagem $j$ projetada na $i$-ésima eigenface |
| $\Omega_j$ | Vetor de pesos da imagem $j$ em todas as eigenfaces |
| $\|\Gamma - \Gamma_{\text{reconst}}\|$ | Norma da diferença entre a imagem original e sua reconstrução (erro de reconstrução) |
| DBL_MAX | Valor máximo para threshold (usado como sem rejeição) |

# SUMÁRIO

# 1 INTRODUÇÃO

Segundo o artigo 2º, inciso I, da Lei nº 13.812/2019, considera-se pessoa desaparecida "todo ser humano cujo paradeiro é desconhecido, não importando a causa de seu desaparecimento, até que sua recuperação e identificação tenham sido confirmadas por vias físicas ou científicas". Este fenômeno, além de afetar milhares de famílias todos os anos, representa um importante desafio social, jurídico e tecnológico no Brasil e no mundo.

O desaparecimento de pessoas apresenta motivações heterogêneas, incluindo crimes (como sequestros e homicídios), fugas voluntárias motivadas por conflitos familiares, violência doméstica, transtornos psiquiátricos, uso abusivo de substâncias psicoativas, situações de extrema pobreza, desastres naturais, e até fenômenos migratórios. Uma vez fora do convívio familiar, a pessoa desaparecida tende a se ver envolvida em situações de maior vulnerabilidade, enfrentando riscos de violência, exploração, escassez de recursos e dificuldades de acesso a direitos básicos ((DECDACRIM/SIIP/PCMG); (DRPD/DHPP/SIPJ/PCMG), 2023). A multiplicidade de causas e a falta de um padrão dificultam a atuação dos órgãos competentes, tornando o processo de localização e identificação lento e, muitas vezes, ineficaz.

No Estado de Minas Gerais, dados recentes apontam que aproximadamente 40% das pessoas desaparecidas são localizadas posteriormente[1]. Entretanto, tal percentual revela que uma parcela significativa permanece sem solução, evidenciando a necessidade de aprimoramento dos métodos de busca e identificação.

Nesse contexto, o emprego de tecnologias baseadas em Inteligência Artificial (IA) emerge como alternativa promissora para otimizar os processos de localização e identificação de pessoas desaparecidas. Em especial, a utilização de algoritmos de reconhecimento facial, capazes de analisar e comparar características biométricas extraídas de imagens, demonstra grande potencial para acelerar a busca e aumentar a assertividade na identificação de indivíduos. Cada ser humano possui padrões faciais únicos, determinados pelo formato do rosto, proporção e distância entre olhos, boca, nariz, e outros traços específicos, que podem ser capturados, processados e utilizados como "impressões digitais faciais" por sistemas automatizados (SILVA; CINTRA, 2015).

A implantação de bancos de dados integrados, alimentados com imagens de pessoas desaparecidas, associada à aplicação de técnicas avançadas de Inteligência Artificial e visão computacional, pode contribuir significativamente para a redução do tempo de resposta, o aumento da precisão na localização de pessoas e o amparo às famílias afetadas. Nesse contexto, a visão computacional atua como um dos principais pilares, permitindo que sistemas automáticos interpretem, analisem e extraiam informações relevantes de imagens e vídeos, potencializando todo o processo de busca e reconhecimento. Conforme destacado por Cabral *et al.* (CABRAL *et al.*, 2024), o desenvolvimento de ferramentas de reconhecimento facial representa um avanço importante no auxílio à busca de pessoas desaparecidas, integrando tecnologias de análise de

[1] Diagnóstico de pessoas desaparecidas e localizadas nas Regiões Integradas de Segurança Pública de Minas Gerais entre 2020 e 2022

imagens a sistemas de identificação automatizada.

Diante desse cenário, este trabalho propõe o desenvolvimento e análise de técnicas de reconhecimento facial, fundamentadas em visão computacional, aplicadas à identificação de pessoas desaparecidas. Além disso, são destacados os desafios técnicos, sociais e éticos envolvidos, bem como os potenciais benefícios dessas abordagens para a segurança pública e para a sociedade em geral.

## 1.1 Justificativa

A escolha deste tema se fundamenta em sua dupla relevância: social e tecnológica. Sob a perspectiva social, o desaparecimento de pessoas é uma questão sensível e persistente, com média de 7 mil casos anuais apenas em Minas Gerais e uma taxa de solução relativamente baixa (40%). As famílias afetadas enfrentam não apenas a dor da incerteza, mas também a ineficiência dos métodos tradicionais de busca, que dependem excessivamente da mobilização comunitária e da divulgação passiva de informações.((DECDACRIM/SIIP/PCMG); (DRPD/DHPP/SIPJ/PCMG), 2023)

No âmbito tecnológico, a pesquisa aborda um dos campos mais crescente da inteligência artificial aplicada: a visão computacional para reconhecimento facial em contextos de segurança pública. Enquanto países como Estados Unidos e China já utilizam essas tecnologias em larga escala, o Brasil ainda carece de soluções adaptadas às suas particularidades socioeconômicas e infraestrutura.

## 1.2 Objetivos

### 1.2.1 *Objetivo geral*

O objetivo geral deste trabalho é verificar viabilidade de implementação e desenvolver um sistema baseado em inteligência artificial, utilizando técnicas de visão computacional de Eigenfaces, para auxiliar os órgãos de segurança pública na identificação e localização de pessoas desaparecidas.

### 1.2.2 *Objetivos específicos*

- Implementar técnicas de *web scraping* para extração automatizada de registros de desaparecidos dos portais oficiais;
- Implementar um sistema de tratamento de imagens;
- Implementar um sistema baseado em inteligência artificial, utilizando técnicas de visão computacional de Eigenfaces, para verificação da viabilidade de identificação e localização de pessoas desaparecidas;

- Avaliar este algorítmo de reconhecimento facial quanto à taxa de acerto.

## 1.3 Organização do Texto

Este trabalho está organizado da seguinte forma:

**CAPÍTULO 1** - Introdução: Apresenta o contexto dos desaparecimentos em Minas Gerais, onde apenas 40% dos casos são solucionados. Discute o impacto social e jurídico do problema, justificando a pesquisa pela dupla relevância: social (vulnerabilidade das vítimas e sofrimento familiar) e tecnológica (lacuna em soluções adaptadas à realidade brasileira). Define como objetivo geral desenvolver um sistema de reconhecimento facial baseado em Eigenfaces, com objetivos específicos como implementação de *web scraping*, tratamento de imagens e avaliação comparativa de algoritmos.

**CAPÍTULO 2** - Fundamentação Teórica: Explora cinco eixos fundamentais: (1) Técnicas de *Web Scraping* e sua implementação em Python; (2) Estratégias de *Data Augmentation* para ampliação de bases de treino; (3) Métodos de tratamento de imagens e seu impacto crítico na acurácia; (4) Fundamentos de reconhecimento de padrões e biométrica facial; e (5) Algoritmo Eigenfaces/PCA, detalhando cálculo matemático, vantagens e limitações. Aborda ainda aspectos éticos e legais da LGPD, classificando dados biométricos como sensíveis e discutindo bases legais para seu uso.

**CAPÍTULO 3** - Metodologia: Descreve o fluxo experimental em quatro etapas: (1) Coleta de dados utilizando a *Yale Face Database* (165 imagens de 15 indivíduos) devido à não possibilidade de aplicação real para testes; (2) Aplicação de *Data Augmentation* com seis técnicas (inversão horizontal, transformação afim, modulação de brilho, etc.) gerando 1.080 imagens; (3) Divisão em cinco subconjuntos (DB1-DB5) com variação de 1 a 9 fotos originais por pessoa; e (4) Treinamento do Eigenfaces com variação de hiperparâmetros (*threshold* e componentes principais), utilizando acurácia e matriz de confusão para avaliação.

**CAPÍTULO 4** - Resultados e Discussões: Analisa 56 testes demonstrando correlação direta entre tamanho da base de dados e acurácia: DB1 (42,50%) até DB5 (65,39%). Evidencia que *thresholds* baixos prejudicam desempenho (20-66,67%) por aumentar falsos negativos, enquanto configurações ótimas (0 componentes e *threshold* máximo) atingem 73,3% em DB5.

**CAPÍTULO 5** - Conclusão: Sintetiza que diversidade de imagens reais por indivíduo e calibração de hiperparâmetros são determinantes para eficácia do Eigenfaces. Destaca contribuições para segurança pública e propõe trabalhos futuros: (1) Integração do classificador com bancos policiais e testes em câmeras urbanas; e (2) Análise comparativa de algoritmos (LBPH para hardware limitado e FaceNet para alta precisão), avaliando métricas como tempo de inferência e robustez.

# 2 REVISÃO BIBLIOGRÁFICA

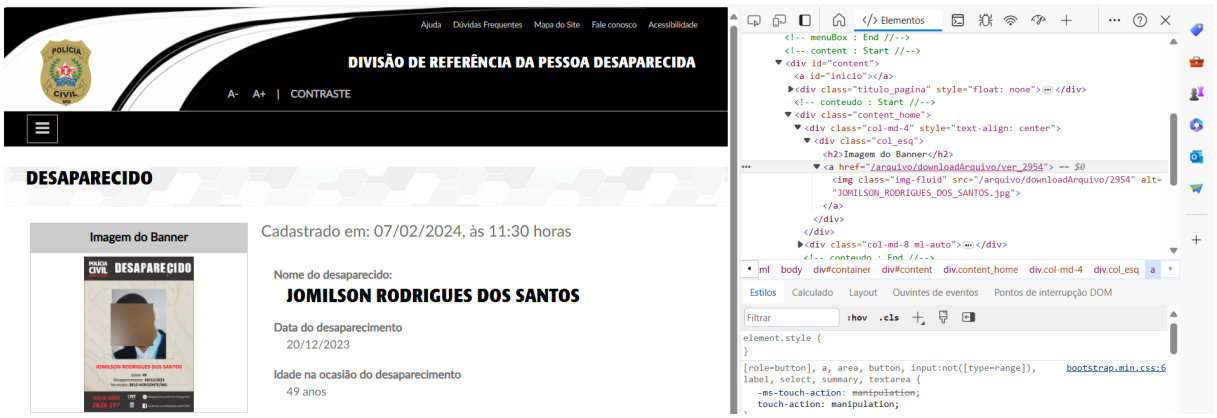
## 2.1 *Web Scraping*

O *Web Scraping* consiste em uma técnica para extração automatizada de dados não estruturados presentes na *internet*, frequentemente disponíveis em formato HTML, transformando-os em dados estruturados que podem ser armazenados e analisados posteriormente (SIRISURIYA, 2015). Tradicionalmente, a coleta de dados via técnicas manuais de copiar e colar se mostra ineficiente e trabalhosa, tornando o processo de obtenção de informações um procedimento demorado e suscetível a erros humanos. Com o avanço das técnicas de *Web Scraping*, tornou-se possível automatizar e organizar a coleta de informações de forma mais eficiente. No entanto, mesmo com a automação, permanece o desafio de lidar com dados presentes em formatos inadequados para análise direta, exigindo técnicas robustas para sua estruturação e o processamento em tempo hábil (DOGRA; NIRWAN; CHAUHAN, 2023).

Para a análise da estrutura dos sites, empregou-se a ferramenta DevTools do navegador, a qual consiste em um conjunto de utilitários integrados, utilizados por desenvolvedores, que permitem a inspeção, depuração e otimização de páginas web. Tal ferramenta possibilita examinar de forma detalhada a estrutura do HTML (DOM), os estilos CSS, os scripts JavaScript, bem como o tráfego de rede da aplicação, configurando-se, assim, como um recurso relevante no desenvolvimento e na resolução de inconsistências técnicas.

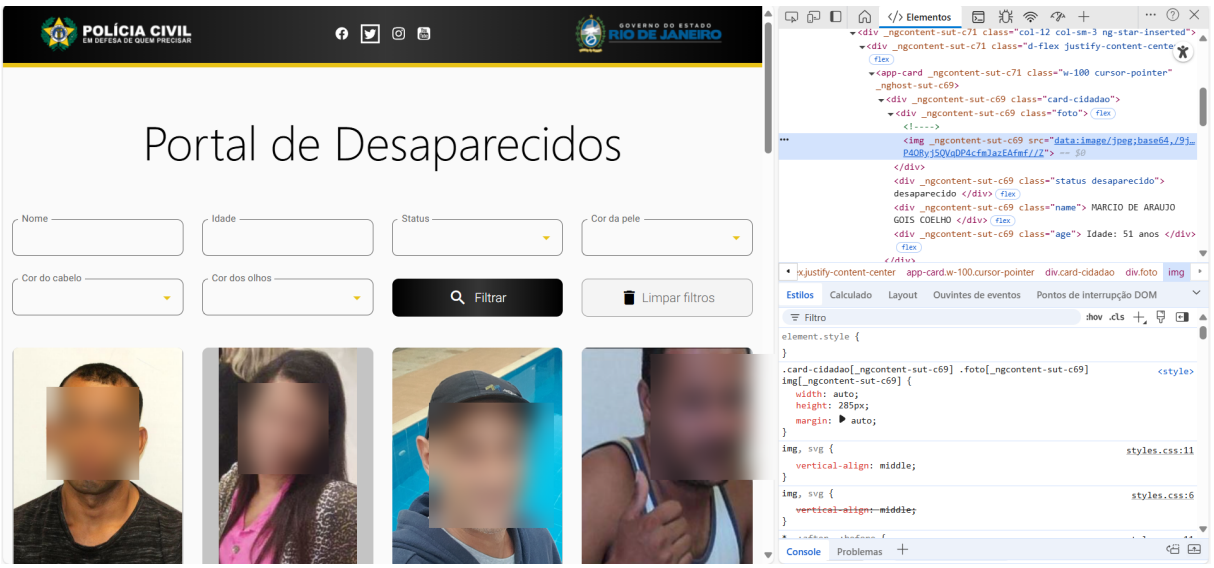
As Figuras 1 e 2 apresentam, de forma parcial, a estrutura HTML de sites pertencentes às Polícias de dois estados distintos: a Figura 1 refere-se ao estado de Minas Gerais e a Figura 2 ao estado do Rio de Janeiro. Conforme pode ser observado, as estruturas possuem diferenças, o que torna evidente a dificuldade em se estabelecer um padrão único para a elaboração de um código que vise à criação de uma base de dados unificada.

Figura 1 – Demonstração de navegação no site da Polícia Civil de Minas Gerais.




Fonte: https://www.desaparecidos.policiacivil.mg.gov.br/desaparecido/exibir/2599, 2024.

Figura 2 – Demonstração de navegação no site da Polícia Civil do Rio de Janeiro.




Fonte: https://www.desaparecidos.pcivil.rj.gov.br/pesquisar, 2024.

A técnica de web scraping será fundamental para a construção da base de dados necessária ao desenvolvimento deste trabalho, uma vez que a disponibilidade de dados estruturados é um requisito para o treinamento de modelos de inteligência artificial. Por meio dessa abordagem, será possível coletar imagens da base de desaparecidos da Polícia Civil de Minas Gerais (PCMG) de forma automatizada.

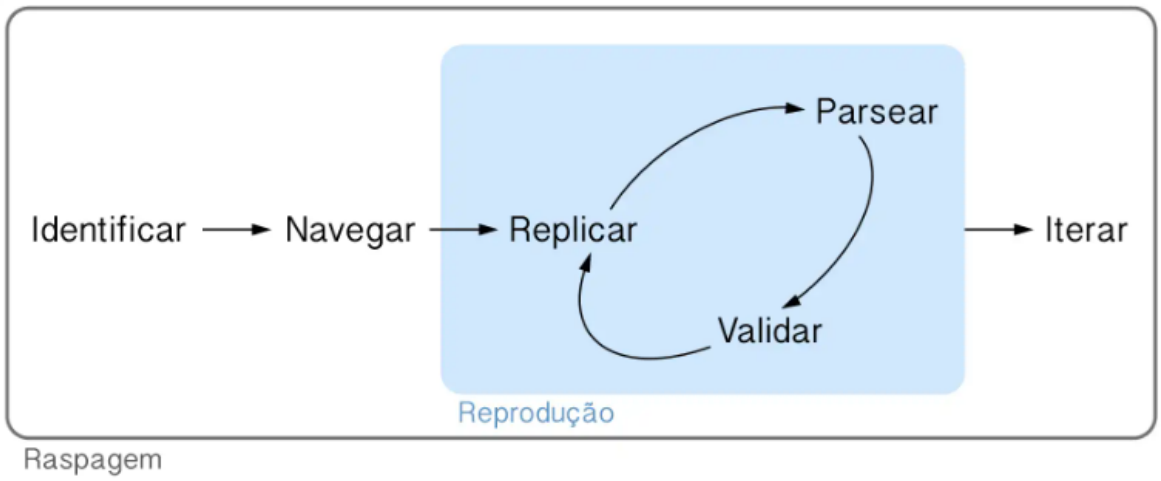
### 2.1.1 Aquisição de Dados por Meio de *Web Scraping*

Com o grande número de informações disponibilizadas na *internet* e com a predominância de dados não estruturados disponíveis, demonstra uma grande necessidade do uso de técnicas de web scraping. Embora a web seja caracterizada por uma quantidade excessiva de informações compartilhadas na estrutura do conteúdo, a maioria das páginas é projetada para usuários finais humanos e não para uso automatizado, o que gera uma lacuna relevante: há uma grande quantidade de dados que não são prontamente legíveis por máquinas ou facilmente exportáveis.

Mesmo com as tecnologias e modernização do processos de informação das polícias, as Interfaces de Programação de Aplicações (APIs) disponibilizadas pelos departamentos de segurança atuais, apresentam limitações tanto em relação à quantidade quanto ao tipo de dados disponibilizados, além de, por vezes, não serem gratuitas para uso. Nesse contexto, várias limitações citadas acima, demonstra que o uso web scraping se torna uma abordagem mais rápida e eficaz para obter as informações necessárias para a aquisição automatizada de dados.

Segundo Hadley (WICKHAM; RUNDEL; GROLEMUND, 2023), o processo de *Web Scraping* pode ser sistematizado em seis etapas principais: identificar, navegar, replicar persear, validar e iterar.

Figura 3 – Fases do *Web Scraping*.




Fonte: WICKHAM; RUNDEL; GROLEMUND.

A Figura 3 ilustra o processo de aquisição de dados, estruturado em seis etapas distintas: Identificar, Navegar, Replicar, Parsear, Validar e Iterar. Esse fluxo evidencia a complexidade inerente às operações de *web scraping*, as quais demandam atenção especial à dinamicidade das páginas web, bem como à robustez e à eficiência dos algoritmos implementados para garantir a extração precisa e consistente das informações.

Na etapa de **identificação**, busca-se delimitar as informações de interesse, bem como compreender padrões presentes no ambiente web, tais como títulos, estruturas de imagens e blocos de texto. Esse reconhecimento inicial tem como objetivo mapear a estrutura visual e lógica da página, facilitando as etapas subsequentes.

A fase de **navegação** envolve a análise da origem dos dados a serem extraídos. Utiliza-se, nesse contexto, ferramentas de desenvolvedor do navegador (geralmente acessadas pelo atalho F12) para inspecionar o código-fonte HTML da página e localizar os elementos que contêm os dados de interesse.

Em seguida, ocorre a etapa de **replicação**, na qual são realizadas requisições HTTP para o site desejado, permitindo o acesso ao conteúdo necessário. Esse estágio depende do correto entendimento adquirido na etapa anterior, já que é fundamental saber, com precisão, onde e como a informação é disponibilizada pela página web. Para exemplificar, é comum realizar requisições do tipo GET para obter o conteúdo completo de páginas específicas.

O processo de **parsing** refere-se à extração e análise detalhada do conteúdo retornado pelo servidor web. Nessa etapa, são empregados métodos automáticos para localizar e extrair, de forma eficiente, os dados específicos desejados, como, por exemplo, aqueles presentes em marcações HTML do tipo <a href></a>.

A etapa de **validação** consiste em verificar se as fases anteriores foram executadas

corretamente, conferindo se todos os dados desejados foram, de fato, extraídos. Recomenda-se aplicar o processo em diferentes páginas, de modo a garantir a robustez do método desenvolvido. Em caso de inconsistências, é necessário retornar às etapas de replicação e parsing para ajustar o procedimento.

Por fim, a fase de **iteração** visa consolidar a solução desenvolvida, encapsulando o procedimento em funções ou rotinas que permitam a repetição automática do processo para múltiplas páginas ou conjuntos de links. Dessa forma, garante-se a escalabilidade e a eficiência do algoritmo implementado, permitindo sua aplicação em diferentes cenários e volumes de dados.

A sistematização do *Web Scraping* apresentada envolve desafios inerentes, como a variabilidade na estrutura dos sites e as restrições impostas por políticas de acesso. Contudo, sua aplicação destaca-se como uma estratégia eficaz para a obtenção de grandes volumes de dados em pesquisas e projetos de engenharia.

Portanto, é vantajoso o uso de *Web Scraping* para raspagem de dados, porque além de ágil a fonte de dados é muito grande, pois, o maior repositório de informações disponíveis na atualidade é a *internet*. Além disso, é preciso cuidado na mineração dos dados e usar o *Web Scraping* de forma correta, pois alguns sites podem proibir o download de seu conteúdo ou considerar ilegal a prática de raspagem de dados, (DOGRA; NIRWAN; CHAUHAN, 2023).

### 2.1.2 Construindo o *Web Scraping* na linguagem Python

A linguagem Python consolidou-se como uma das principais escolhas para a implementação de técnicas de *web scraping* devido à sua sintaxe clara, vasta comunidade e amplo ecossistema de bibliotecas especializadas. De acordo com o ranking RedMonk (STEPHENS, 2021), Python é atualmente a segunda linguagem de programação mais utilizada, o que evidencia sua relevância e ampla adoção no desenvolvimento de soluções tecnológicas.

O processo de *web scraping* pode ser estruturado em três etapas principais: a coleta dos links de interesse, a extração dos dados contidos nesses links e, por fim, o armazenamento das informações, que permitem uma análise posterior de forma organizada e eficiente. Entre as bibliotecas mais utilizadas nesse contexto, destacam-se:

- **Requests**: essencial para a realização de requisições HTTP, permitindo o envio e recebimento de dados através de métodos como GET, POST, PUT e DELETE. Esta biblioteca também fornece suporte para autenticação, manipulação de cookies e gerenciamento de sessões, aspectos cruciais para a coleta automatizada de dados.

- **Beautiful Soup**: amplamente utilizada para a análise de documentos HTML e XML, possibilita uma navegação e extração de informações mais intuitiva e eficiente, especialmente quando comparada ao uso de expressões regulares. Esta biblioteca também simplifica o

tratamento de diferentes codificações de caracteres, reduzindo a complexidade inerente ao processamento de diversos tipos de documentos (KHDER, 2021).

A presença de ferramentas como o framework *Scrapy*, desenvolvido em Python, amplia ainda mais as possibilidades no âmbito do *web scraping*, proporcionando funcionalidades avançadas para a raspagem em larga escala, como o *crawling* assíncrono, o gerenciamento de requisições e a integração com bancos de dados (THOMAS; MATHUR, 2019).

As técnicas de extração de dados, tradicionalmente utilizadas para coleta de informações disponíveis na web, evoluíram em paralelo aos avanços nas tecnologias de bancos de dados e mineração de dados. Assim, o uso do Python configura-se como uma escolha estratégica para pesquisadores e profissionais que necessitam de soluções eficientes e escaláveis para a extração e análise de dados não estruturados.

"Devido à enorme comunidade e recursos de bibliotecas para Python e à clareza do estilo de codificação da linguagem, ela é a mais apropriada para raspar dados desejados de sites de interesse" (THOMAS; MATHUR, 2019).

Em síntese, a adoção do Python no contexto do *web scraping* justifica-se por sua facilidade de uso, ampla comunidade e um ecossistema de ferramentas que em diferentes projetos, consolidando-se como um ambiente colaborativo e eficiente para o desenvolvimento de soluções voltadas à extração de dados na web.

## 2.2 Data Augmentation

O *Data Augmentation*, consiste no aumento artificial da base de dados a partir da aplicação de transformações sobre as imagens originais. O objetivo dessas técnicas é expandir o conjunto de treinamento supervisionado, permitindo que o algoritmo desenvolva maior acurácia, o que tende a resultar em um desempenho mais preciso (OLIVEIRA, 2019)

No contexto da ampliação de dados faciais, os métodos podem ser categorizados de diversas formas, refletindo a evolução e a complexidade das técnicas empregadas. Uma classificação comum divide as abordagens em três categorias principais: (1) transformações básicas de imagem, que ampliam a diversidade dos dados de forma mais direta; (2) abordagens baseadas em modelos, como os Modelos Morfáveis 3D (3DMM), que permitem a síntese de rostos com variações controladas de pose, iluminação e expressões; e (3) técnicas avançadas baseadas em aprendizado profundo, como Redes Generativas Adversariais (GANs) e Autoencoders Variacionais (VAEs), capazes de gerar imagens realistas e preservar identidades durante transformações complexas. Adicionalmente, existem métodos híbridos, como CycleGAN e StyleGAN, que combinam diferentes estratégias para otimizar a qualidade e a diversidade das imagens sintéticas (WANG; WANG; LIAN, 2019). Este trabalho se aprofundará na primeira dessas categorias: as transformações básicas de imagem.

As transformações geométricas constituem uma categoria fundamental de operações em *data augmentation*, pois alteram a disposição espacial dos pixels de uma imagem, modificando sua geometria e, consequentemente, ampliando a variabilidade dos dados de entrada. Essas operações são realizadas por meio do remapeamento dos valores de pixel para novas posições coordenadas na imagem de saída, preservando o rótulo original da imagem e enriquecendo o conjunto de treinamento (SHORTEN; KHOSHGOFTAAR, 2019).

As transformações geométricas aplicáveis são diversas e abrangem diferentes manipulações da imagem. Entre elas, destacam-se a translação, que consiste no deslocamento da imagem ao longo dos eixos horizontal ou vertical, e a rotação, que gira a imagem em torno de um ponto central. Outra técnica comum é o espelhamento (ou reflexão), que inverte a imagem em relação a um eixo definido. Além disso, é possível ajustar o tamanho aparente da imagem por meio do zoom ou redimensioná-la usando escalas para diferentes resoluções.

O corte permite focar em regiões específicas da imagem, eliminando partes indesejadas ou simulando oclusões, enquanto o preenchimento (padding) adiciona pixels às bordas, geralmente com valores nulos ou reflexos da imagem original. Transformações mais complexas incluem a alteração de perspectiva, que simula diferentes ângulos de câmera, e distorções como a elástica, que deforma a imagem de forma não rígida, e a distorção de lente, que reproduz efeitos ópticos característicos de lentes reais.

Essas técnicas podem ser facilmente implementadas por meio de bibliotecas especializadas, como `imgaug`, que oferecem suporte a diversas operações de aumento, incluindo rotação, translação, inversão, corte, preenchimento, distorção elástica, escala, transformação por partes e perspectiva, entre outras. (REVISION, 2025)

Considerando-se a limitação da base de dados disponível sobre pessoas desaparecidas, torna-se necessário ampliar a quantidade de imagens disponíveis para o treinamento do modelo.

## 2.3 Tratamento de dados

Em sistemas desenvolvidos com base em inteligência artificial (IA) necessita da aplicação de uma etapa chamada comumente de tratamento de dados, essencial para o treinamento da IA. O processo de tratamento de dados pode ser entendido com uma etapa entre a aquisição de dados brutos e o funcionamento correto da IA.

A etapa de tratamento dos dados são necessário para evitar ruídos, artefatos e inconsistências que comprometem irreversivelmente o desempenho dos modelos. No contexto de visão computacional, essa pré-processamento envolve operações como remoção de background, correção de iluminação, normalização de contraste e redução de artefatos de compressão (LITJENS *et al.*, 2017). A negligência nesta fase propaga vieses sistêmicos que invalidam generalizações, em suma, mais dados ruins geram resultados piores que menos dados bons.

Há relação causal entre tratamento de imagens e desempenho algorítmico. Estudos comparativos em classificação de melanoma demonstraram que modelos treinados com imagens pré-processadas (filtragem de hair noise, equalização de cor) obtiveram uma alta taxa de detecção 23% superior aos que utilizaram dados crus (ESTEVA *et al.*, 2017). Analogamente, em diagnósticos por ressonância magnética, a aplicação de técnicas de *bias field correction* elevou a acurácia de segmentação tumoral de 74% para 89% (TUSTISON *et al.*, 2010). Esses resultados corroboram que a qualidade do dado pré-processado é o fator isolado mais impactante na eficiência dos resultados das inteligências artificiais.

Em nosso trabalho a frequência de ruídos nos dados obtidos pela *internet* é algo predominante, necessitando um tratamento de dados para eliminar os ruídos e inconsistências que comprometem a eficácia do treinamento de modelos de inteligência artificial. No contexto de bases como a de desaparecidos da Polícia Civil de Minas Gerais (PCMG), onde imagens contêm elementos como cartazes e fundos heterogêneos, o pré-processamento atua na seleção e isolamento de regiões de interesse que são os rostos dos indivíduos.

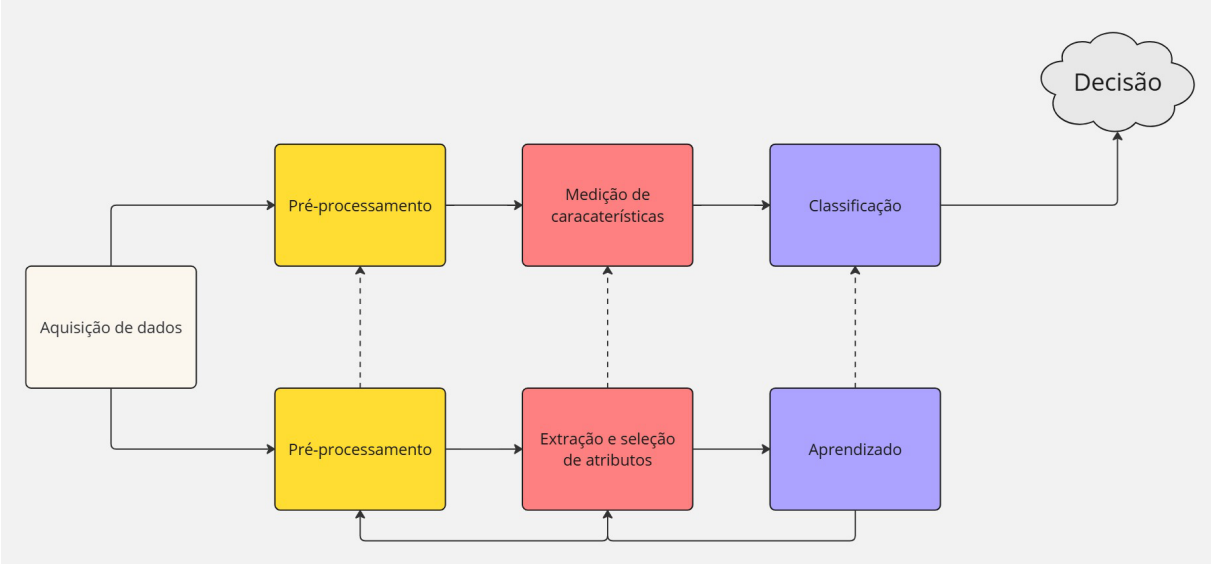
## 2.4 Reconhecimento de Padrões

Os sistemas de reconhecimento facial são fundamentados em técnicas de reconhecimento de padrões. Segundo (CHIEN, 1974), essas etapas consistem em aquisição de dados, pré-processamento, extração e seleção de características e classificação.

A etapa de aquisição de dados consiste na captura de imagens através de câmeras de vigilância, dispositivos móveis ou banco de dados públicos. Após a aquisição, as imagens passam por um pré-processamento, que envolve técnicas para melhorar a qualidade dos dados e remoção de ruídos. Com as imagens devidamente processadas, o sistema parte para a extração de características faciais. Esse processo identifica pontos específicos e relevantes do rosto, conhecidos como *landmarks*, que servem para distinguir um indivíduo dos demais. Nem todas as características extraídas são igualmente relevantes. Por isso, o sistema avalia e seleciona aquelas que mais contribuem para a diferenciação entre indivíduos, eliminando informações redundantes ou irrelevantes.

Na etapa final, o sistema utiliza os vetores de características para comparar a face analisada com as imagens armazenadas em um banco de dados. Algoritmos de classificação, baseados em técnicas estatísticas ou de aprendizado de máquina, determinam se há correspondência entre o rosto apresentado e algum registro existente. Quando uma correspondência é encontrada, o sistema identifica ou autentica o indivíduo; caso contrário, pode indicar que a pessoa não está cadastrada no sistema.

Figura 4 – Sistema genérico de reconhecimento de padrões.




Fonte: Elaborado pelo autor, 2025.

A Figura 4 apresenta as etapas do reconhecimento de padrões: aquisição de dados, pré-processamento, extração/seleção de características e classificação. Esses processos envolvem desde a captura de imagens até a análise por algoritmos de aprendizado de máquina, visando identificar com base em características faciais, conforme descrito por (CHIEN, 1974).

O reconhecimento facial é uma tarefa que, do ponto de vista humano, é natural e cotidiana. Desde cedo, utilizamos essa habilidade para identificar e diferenciar pessoas ao nosso redor, como um bebê que reconhece sua mãe ou um pai que identifica seu filho. Isso ocorre porque cada indivíduo possui características únicas, estudadas no campo da biometria. A biometria consiste em quantificar essas características individuais, demonstrando a singularidade de cada pessoa (JAIN; FLYNN; ROSS, 2007).

Para medir a eficiência biométrica precisa-se define-se atributos, os quais são:

1. Acessibilidade: Refere-se à facilidade de leitura das características biométricas, como as impressões digitais, que podem ser capturadas por dispositivos de leitura.
2. Desempenho: Mede a taxa de reconhecimento, ou seja, se a tecnologia biométrica identifica corretamente o indivíduo.
3. Evasão: Analisa a possibilidade de falsificação da biometria. Por exemplo, dois gêmeos idênticos podem ter características faciais muito semelhantes, o que dificulta a distinção.
4. Mensurabilidade: Verifica se a característica biométrica pode ser medida de forma eficaz.
5. Persistência: Avalia a consistência de uma característica biométrica ao longo do tempo, como a íris do olho, que mantém sua aparência ao longo da vida de um indivíduo.

6. Unicidade: Examina se a característica é única para cada indivíduo, como o DNA, garantindo uma representação exclusiva.

7. Universalidade: Refere-se à presença de características biológicas em todos os indivíduos, como rostos, dedos, olhos, etc.

A face possui alta acessibilidade, baixo desempenho, alta evasão, médio mensurabilidade, média permanência, média unicidade e alta universalidade (SILVA, 2016). Devido ao baixo desempenho da face, quando é necessário transferir a tarefa de reconhecimento para o computador a complexibilidade, devido à tradução de reconhecimento de padrões, aumenta (MORAES, 2010). Portanto, faz-se necessário o uso de técnicas de visão computacional.

O objetivo da visão computacional é tomar decisões úteis sobre objetos físicos reais (SHAPIRO; STOCKMAN, 2001) ou ter como objetivo a construção de descrições de cenas a partir de imagens. De maneira geral, a visão computacional tenta simular a visão humana. Atualmente, existem diferentes técnicas de reconhecimento facial, tais como: *Suport Vector Machine* (SVM), *Principal Component Analysis* (PCA), métodos utilizando redes neurais e extração de características baseados nas distâncias dos elementos locais da face. Neste trabalho usará a técnica de Eigenfaces que é baseada na técnica de PCA.

## 2.5 Eigenfaces

A superação dos desafios no reconhecimento facial é frequentemente abordada por meio de técnicas de visão computacional, que visam simular a capacidade humana de interpretar imagens e extrair informações úteis sobre objetos reais (KSHIRSAGAR; BAVISKAR; GAIKWAD, 2011). Entre essas técnicas, destaca-se o método *Eigenfaces*, que busca caracterizar um conjunto de características faciais independentes da geometria, como olhos, boca, nariz e orelhas, para representar o rosto de forma eficiente.

O *Eigenfaces* utiliza o algoritmo de redução de dimensionalidade conhecido como *Principal Component Analysis* (PCA), também denominado método *Karhunen-Loeve* no contexto de reconhecimento facial (EJAZ *et al.*, 2019). O PCA é uma análise estatística que explora a redundância e a variância nos dados, reduzindo sua dimensionalidade sem perder informações essenciais, mantendo assim a integridade dos resultados (YANG; KRIEGMAN; AHUJA, 2002).

Conforme destacado por (YANG; KRIEGMAN; AHUJA, 2002), o *Eigenfaces* é classificado como um método baseado em aparência, pois não requer conhecimento prévio sobre a face a ser reconhecida. Durante o processo de reconhecimento, o algoritmo identifica os componentes principais, que são os autovetores que descrevem as características faciais.

Uma das principais vantagens do método *Eigenfaces* é a redução de dimensionalidade, possibilitada pelo uso da *Principal Component Analysis* (PCA), que permite a compactação dos dados, otimizando tanto o armazenamento quanto o processamento de grandes conjuntos de

imagens (EJAZ *et al.*, 2019). As informações utilizadas para representar o rosto são utilizadas para esse fim. (YANG; KRIEGMAN; AHUJA, 2002). Porém, deve-se enfatizar que o algoritmo Eigenface é sensível às condições de iluminação e a certos tipos de ruído, como baixa qualidade e expressões faciais, o que pode afetar sua eficiência e reduzir a precisão do sistema (MULYONO *et al.*, 2019).

### 2.5.1 PCA - Análise de Componente Principal

A alta dimensionalidade contida em uma imagem aumenta a complexabilidade no reconhecimento facial. Isso posto, o uso de PCA se mostra apropriado. O objetivo principal do método PCA é identificar padrões de alta dimensionalidade, através de análise estatística, simplificar a complexidade dos dados mantendo o máximo de confiabilidade das informações. Essencialmente, o PCA transforma um conjunto de variáveis correlacionadas em um novo conjunto de variáveis não correlacionadas chamadas componentes principais (PC).

Cada componente principal é uma combinação linear das variáveis originais, onde as novas variáveis são relacionadas à projeção das variáveis originais derivadas da matriz de covariância do conjunto original (SANTOS, 2005). Conjuntos menores de variáveis correlacionadas facilitam a compreensão e a análise de suas interações do que conjuntos com um grande número de variáveis correlacionadas. A técnica foi originalmente criada por Pearson (1901) e desenvolvida de forma independente por Hotelling (1933).

Ao utilizar PCA para reconhecimento facial, pode-se encontrar padrões que representam características faciais de forma mais eficaz. Isso significa que o PCA auxilia a identificar as características mais importantes das imagens faciais. Isso é feito descobrindo padrões nos pixels de uma imagem e usando esses padrões para criar vetores que descrevem as diferenças entre os rostos.

Consequentemente, o método *Eigenfaces* não se concentra em identificar características faciais individuais, como boca, olhos e nariz. Em vez disso, ele realiza uma análise estatística das imagens, destacando como os diferentes tons de cinza dos pixels se correlacionam entre si, independentemente de onde esses pixels estão na imagem ou qual parte do rosto eles representam. Essa abordagem estatística permite capturar padrões gerais nas imagens faciais que podem ser utilizados para reconhecer e comparar rostos.

Simplificando, quando aplicamos a matemática às imagens faciais, procuramos encontrar os principais componentes que representam a distribuição dessas faces. Esses componentes principais são como padrões fundamentais que capturam mudanças nas imagens faciais em relação à média do grupo.

Cada um desses padrões, chamados de autovetor ou "autoface", pode ser pensado como uma face abstrata, refletindo um aspecto diferente das faces da coleção. Cada região da imagem contribui exclusivamente para cada vetor de características, o que significa que cada vetor de

características captura uma parte diferente da informação facial. Quando olhamos para um rosto em uma coleção, podemos descrevê-lo como uma combinação desses vetores de características.

O número de vetores de características importantes é limitado pelo número de faces no conjunto de treinamento. No entanto, só podemos usar o melhor autovetor, que está relacionado ao maior autovalor. Esses autovalores indicam as maiores variações ou diferenças entre as faces no conjunto, ajudando a representar as faces de forma mais eficiente e compacta (TURK; PENTLAND, 1991).

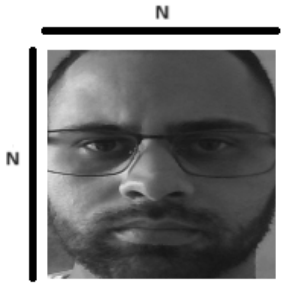
## 2.6 Cálculo do *Eigenfaces*

O método Eigenfaces, baseado na *Principal Component Analysis* (PCA), transforma o problema de reconhecimento facial em um problema de álgebra linear no espaço de características. Através de um conjunto de dados contendo $Z$ imagens faciais, cada uma com dimensões $N \times N$, o processo começa convertendo cada imagem em um vetor unidimensional de tamanho $N^2$. Esses vetores são então organizados em uma matriz $M$ de dimensões $Z \times N^2$, onde cada linha representa uma imagem facial vetorizada.

### 2.6.1 Pré-processamento dos Dados

O pré-processamento é crucial para garantir a eficácia do método Eigenfaces. Na conversão de imagem para vetor cada imagem facial de dimensão $N \times N$ é transformada em um vetor coluna $\Gamma_i$ de tamanho $N^2 \times 1$, onde os pixels são dispostos linearmente (e.g., linha por linha). Isso permite representar a imagem como um ponto em um espaço de alta dimensão.

Figura 5 – Representação $N \times N$ de uma imagem.




Fonte: Elaborado pelo autor, 2025.

A Figura 5 ilustra uma imagem facial de dimensão $N \times N$, que representa a forma original dos dados antes do processamento.

A construção da matriz de dados organiza-se $Z$ imagens vetorizadas na matriz $M$:

$$M = \begin{bmatrix} \Gamma_1^T \\ \Gamma_2^T \\ \vdots \\ \Gamma_Z^T \end{bmatrix} = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,N^2} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,N^2} \\ \vdots & \vdots & \ddots & \vdots \\ x_{Z,1} & x_{Z,2} & \cdots & x_{Z,N^2} \end{bmatrix}$$

### 2.6.2 Algoritmo Eigenfaces

1. **Cálculo da face média ($\Psi$):**

$$\Psi = \frac{1}{Z}\sum_{i=1}^{Z}\Gamma_i \tag{2.1}$$

A face média é obtida pela média aritmética de todos os vetores de imagem do conjunto de treinamento ($\Gamma_i$). Essa etapa, descrita no artigo (KSHIRSAGAR; BAVISKAR; GAIKWAD, 2011), centraliza os dados em torno da origem, removendo características comuns e destacando variações individuais.

2. **Centralização dos dados ($\Phi_i$):**

$$\Phi_i = \Gamma_i - \Psi \tag{2.2}$$

Cada imagem é ajustada subtraindo-se a face média. Isso gera um conjunto de dados com média zero, condição essencial para o cálculo da matriz de covariância. Os $\Phi_i$ representam desvios das características médias, capturando variações únicas das faces.

3. **Cálculo da matriz de covariância ($C$):**

$$C = \frac{1}{Z}AA^T \quad \text{onde} \quad A = [\Phi_1, \Phi_2, \ldots, \Phi_Z] \tag{2.3}$$

$$C_{\text{reduzida}} = \frac{1}{Z}A^TA \tag{2.4}$$

$$v_i = Au_i \tag{2.5}$$

A matriz de covariância quantifica as relações entre pixels. Para eficiência computacional, usa-se a versão reduzida de tamanho $Z \times Z$ (em vez de $N^2 \times N^2$). Os autovetores $v_i$ são obtidos via $v_i = Au_i$, sendo $u_i$ os autovetores de $C_{\text{reduzida}}$.

4. **Decomposição em autovalores:**

$$Cv_i = \lambda_i v_i \tag{2.6}$$

- $\lambda_i$: Autovalores (ordenados $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_Z$)
- $v_i$: Autovetores (Eigenfaces)

Os autovetores $v_i$ (Eigenfaces) e autovalores $\lambda_i$ de $C$ são calculados. Cada Eigenface ($v_i$) representa uma direção ortogonal de variação significativa nos dados, enquanto $\lambda_i$ indica sua importância (variância explicada).

5. **Seleção das componentes principais:** Seleciona-se $k$ eigenfaces que capturam $\alpha$ (threshold) da variância:

$$\frac{\sum_{i=1}^{k} \lambda_i}{\sum_{j=1}^{Z} \lambda_j} \geq \alpha \tag{2.7}$$

Segundo a documentação oficial da OpenCV, "não há uma regra fixa para o número de componentes principais $k$; recomenda-se experimentar diferentes valores, sendo que 80 componentes geralmente são suficientes para uma boa reconstrução"(OPENCV, 2024).

O artigo (KSHIRSAGAR; BAVISKAR; GAIKWAD, 2011) ressalta que $k < N^2$ reduz dimensionalidade sem perda crítica de informação. Na prática (OpenCV), $k = 80$ é comum para reconstruções eficientes.

6. **Projeção no espaço eigenface:** Para cada imagem $j$:

$$\omega_i^{(j)} = v_i^T \Phi_j \tag{2.8}$$

$$\Omega_j = \begin{bmatrix} \omega_1^{(j)} & \omega_2^{(j)} & \cdots & \omega_k^{(j)} \end{bmatrix}^T \tag{2.9}$$

Cada imagem centralizada $\Phi_j$ é projetada no subespaço das Eigenfaces selecionadas. O vetor $\Omega_j$ (coordenadas no espaço reduzido) serve como *feature vector* para reconhecimento, armazenando padrões únicos da face de forma compacta.

### 2.6.3 Reconhecimento Facial

1. **Projeção da imagem de teste:** A imagem de teste $\Gamma_{\text{teste}}$ é centralizada ($\Phi_{\text{teste}} = \Gamma_{\text{teste}} - \Psi$) e projetada no subespaço das eigenfaces selecionadas. O vetor de pesos $\Omega_{\text{teste}} = [\omega_1^{(\text{teste})}, \ldots, \omega_k^{(\text{teste})}]^T$ captura características únicas da face em um espaço reduzido, conforme descrito em (TURK; PENTLAND, 1991). Esta compactação permite comparação eficiente com o banco de dados.

2. **Cálculo de distâncias:** Utiliza-se distância euclidiana $\varepsilon_j = \|\Omega_{\text{teste}} - \Omega_j\|$ para comparar o vetor de teste com os vetores armazenados $\Omega_j$. Segundo Kshirsagar (KSHIRSAGAR; BAVISKAR; GAIKWAD, 2011) esta métrica quantifica similaridades faciais no espaço reduzido. Valores pequenos de $\varepsilon_j$ indicam alta similaridade, enquanto valores grandes sugerem faces distintas.

3. **Classificação:** A identidade é atribuída à classe $j$ que minimiza $\varepsilon_j$, desde que $\min_j \varepsilon_j \leq \theta$. O limiar $\theta$ (threshold) é determinado empiricamente para balancear False Acceptance Rate (FAR) e False Rejection Rate (FRR)(YANG; KRIEGMAN; AHUJA, 2002). Se $\min_j \varepsilon_j > \theta$, a face é classificada como "desconhecida".

### 2.6.4 Reconstrução de Faces

A face original pode ser reconstruída através da combinação linear:

$$\Gamma_{\text{reconst}} = \Psi + \sum_{i=1}^{k} \omega_i v_i \tag{2.10}$$

onde $\Psi$ é a face média, $v_i$ são as eigenfaces, e $\omega_i$ os pesos projetados. Conforme (TURK; PENTLAND, 1991), o erro de reconstrução diminui monotonicamente com o aumento de $k$. Como reconstruções com $k$ reduzido (e.g., 10-20 eigenfaces) retêm características essenciais, enquanto $k$ elevado (e.g., >80) reproduz detalhes finais. Esta propriedade é explorada em compressão de imagens, onde tipicamente 90-95% da variância é preservada com $k < N^2$.

Variações de iluminação afetam a projeção das imagens no espaço *eigenface*, comprometendo a qualidade da representação. Além disso, oclusões parciais tendem a aumentar o erro de reconstrução, representado por $\|\Gamma - \Gamma_{\text{reconst}}\|$, dificultando o reconhecimento. Mudanças de pose também impactam negativamente o desempenho do método, sendo necessário um realinhamento prévio das imagens.

## 2.7 LGPD - Lei Geral de Proteção de Dados

Em 14 de agosto de 2018 foi criada a Lei nº 13.709 a Lei Geral de Proteção dos Dados Pessoais (LGPD) (BRASIL, 2018) que visa criar medidas preventivas, proativas na manutenção e privacidade dos dados de terceiros.

Quando se trata de tratamento de dados de pessoas, a cautela com o manuseio e armazenamento dele deve ser redobrada por conta da Lei Geral de Proteção de Dados - LGPD. Essa lei em seu artigo 1° diz: “Esta Lei dispõe sobre o tratamento de dados pessoais, inclusive nos meios digitais, por pessoa natural ou por pessoa jurídica de direito público ou privado, com o objetivo de proteger os direitos fundamentais de liberdade e de privacidade e o livre desenvolvimento da personalidade da pessoa natural.” e no Artigo. 2º A disciplina da proteção de dados pessoais tem como fundamentos:

I. o respeito à privacidade;

II. a autodeterminação informativa;

III. a liberdade de expressão, de informação, de comunicação e de opinião;

IV. a inviolabilidade da intimidade, da honra e da imagem;

V. o desenvolvimento econômico e tecnológico e a inovação;

VI. a livre iniciativa, a livre concorrência e a defesa do consumidor; e

VII. os direitos humanos, o livre desenvolvimento da personalidade, a dignidade e o exercício da cidadania pelas pessoas naturais.

Os tipos de dados são:

**Dados pessoais:** refere-se a informações relacionadas a uma pessoa específica que pode ser identificada ou identificável, como nome, endereço, número de identificação, entre outros

**Dados sensíveis:** refere-se aos dados pessoais com potencial de impacto nos direitos e liberdades fundamentais das pessoas, como informações sobre origem racial ou étnica, convicções religiosas, opiniões políticas, saúde, vida sexual, dados genéticos ou biométricos.

**Dados de acesso público:** refere-se a informações disponíveis publicamente e que podem ser acessados por qualquer indivíduo, como propriedade de imóveis, participação em empresas, atividades de órgãos públicos, entre outros

**Dados Anonimizados:** refere-se a dados que passaram por um processo que elimina ou altera qualquer informação que possa identificar um indivíduo específico, garantindo assim que não esteja vinculada à identidade desse indivíduo. Isso significa que, uma vez anonimizados, os dados não estão sujeitos à Lei Geral de Proteção de Dados (LGPD). Deve-se enfatizar que os dados só são considerados anônimos se o caminho para a identificação do titular dos dados não puder ser reconstruído por meios técnicos ou outros. Caso sejam identificáveis, os dados não serão considerados anônimos, mas sim pseudonimizados, e estarão sujeitos à jurisdição da LGPD.

As principais bases legais para o tratamento de dados pessoais na LGPD são:

1. Consentimento do titular: O tratamento de dados pessoais é permitido com o consentimento do titular, que deve ser livre, informado e inequívoco.

2. Cumprimento de obrigação legal ou regulatória pelo controlador: O tratamento de dados pode ser realizado para o cumprimento de uma obrigação legal ou regulatória pelo controlador.

3. Execução de políticas públicas: O tratamento de dados pode ser feito para a execução de políticas públicas previstas em leis e regulamentos.

4. Estudos por órgão de pesquisa: O tratamento de dados pode ser realizado para estudos por órgão de pesquisa, garantindo, sempre que possível, a anonimização dos dados pessoais.

5. Proteção da vida ou da incolumidade física do titular ou de terceiros: O tratamento de dados pode ser feito para proteger a vida ou a integridade física do titular ou de terceiros

6. Tutela da saúde: O tratamento de dados pode ser realizado para tutela da saúde, exclusivamente, em procedimentos realizados por profissionais de saúde, serviços de saúde ou autoridade sanitária

7. Interesses legítimos do controlador ou de terceiros: O tratamento de dados pode ser realizado para atender aos interesses legítimos do controlador ou de terceiros, desde que não prevaleçam os direitos e liberdades fundamentais do titular que exijam a proteção dos dados pessoais

8. Proteção do crédito: O tratamento de dados para proteção do crédito visa ampliar e facilitar a concessão de crédito, melhorar análises de risco e impulsionar o mercado de consumo, devendo estar em conformidade com normas como o Código de Defesa do Consumidor e a lei do cadastro positivo

Essas bases legais estabelecem os fundamentos para o tratamento de dados pessoais na LGPD, garantindo que as informações sejam processadas de forma legal, legítima e transparente, respeitando os direitos e liberdades dos titulares.

A possibilidade de utilizar dados disponíveis em sites da Polícia Civil ou de órgãos públicos depende do contexto em que essas informações foram disponibilizadas e da finalidade para a qual serão utilizadas. É importante considerar que, mesmo que os dados sejam de acesso público, é necessário verificar se o uso dessas informações está em conformidade com a legislação de proteção de dados pessoais, como a LGPD.

Deve-se considerar o contexto em que a informação foi disponibilizada e a compatibilidade entre o uso dos dados e as circunstâncias em que foram tornados públicos. Além disso, destaca que, mesmo que os dados sejam considerados públicos, não deixam de ser pessoais, sendo necessário sempre considerar a finalidade da circulação e o que justifica sua disponibilização.

Portanto, ao utilizar dados disponíveis em sites da Polícia Civil ou de órgãos públicos, é recomendável analisar cuidadosamente o contexto, a finalidade e a legalidade do tratamento dessas informações, garantindo que esteja em conformidade com os princípios e diretrizes estabelecidos na LGPD e em outras normas aplicáveis de proteção de dados pessoais (TEFFÉ; VIOLA**, 2020).

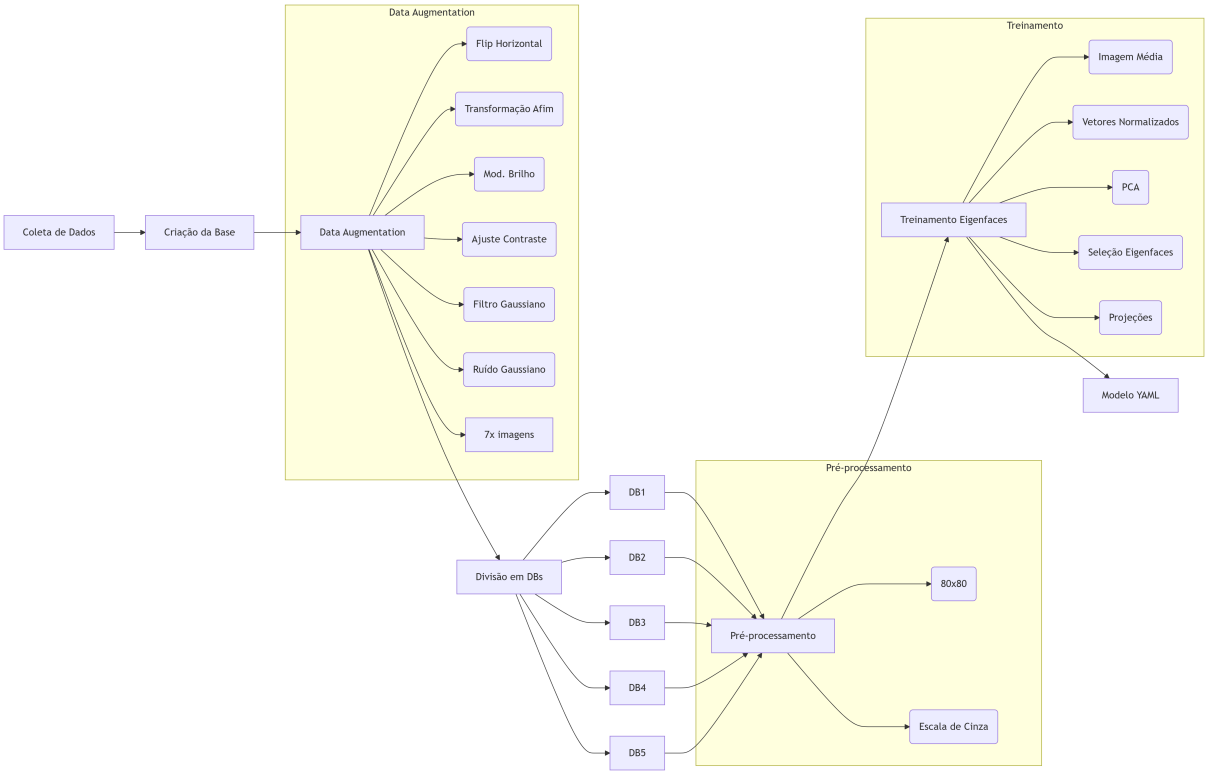
# 3 METODOLOGIA

A metodologia adotada para este trabalho foi estruturada em quatro etapas principais: coleta de dados, criação da base de dados, tratamento das imagens e e aplicação do algoritmo de reconhecimento facial. A etapa de coleta consistiu na obtenção de imagens faciais por meio de *Web Scraping*, observando critérios legais da LGPD descrita na seção 2.7 e a disponibilidade pública dos dados. Em seguida, as imagens foram submetidas ao tratamento, com o objetivo de remover ruídos Esse processo incluiu o recorte das imagens, de modo a preservar apenas a região facial de interesse.

Posteriormente, foi realizada a criação da base de dados, por meio da aplicação de técnicas de *Data Augmentation*, como rotação, espelhamento e variação de brilho. Essa etapa teve como finalidade ampliar a variabilidade das amostras, o que contribui para a melhoria do desempenho e da capacidade de generalização do modelo. Por fim, procedeu-se à etapa de reconhecimento facial, utilizando-se a técnica de *Eigenfaces*, baseada na Análise de Componentes Principais (PCA). Essa abordagem permite a extração de características discriminantes dos rostos, possibilitando a identificação ou verificação de indivíduos com base em projeções em um espaço de características reduzido.

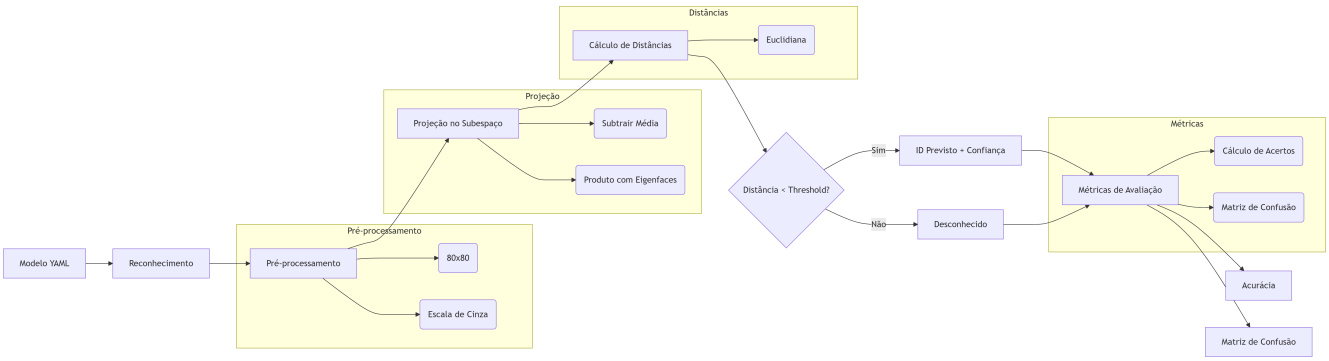
Nas Figuras 6 e 7, observa-se a representação do fluxo metodológico seguido no desenvolvimento do sistema de reconhecimento facial, contemplando as etapas de coleta, tratamento, aumento da base de dados e aplicação do algoritmo Eigenfaces.

Figura 6 – Metodologia: Pré-processamento e Treinamento.




Fonte: Elaborado pelo autor, 2025.

Figura 7 – Metodologia: Reconhecimento e Avaliação.




Fonte: Elaborado pelo autor, 2025.

## 3.1 Coleta de Dados

A primeira fase do trabalho consiste na coleta de dados. Inicialmente, foram consideradas as imagens de pessoas desaparecidas disponibilizadas pelo site da Polícia Militar. Contudo, devido à limitação de apenas uma foto por pessoa, a base de dados se mostrou insuficiente para a

aplicação eficaz do algoritmo Eigenfaces, que requer um conjunto mais robusto de imagens para treinamento e teste.

Para contornar a limitação mencionada, optou-se por utilizar a *Yale Face Database* (UNIVERSITY, 1997), um banco de dados de faces disponibilizado pela Universidade de Yale. Esse conjunto de dados consiste em 165 imagens em escala de cinza (formato GIF) contendo 15 indivíduos distintos, com 11 imagens por sujeito representando diferentes expressões faciais e condições de iluminação: luz central, com óculos, expressão feliz, luz lateral esquerda, sem óculos, expressão neutra, luz lateral direita, expressão triste, sonolência, surpresa e piscar de olhos. Dessas onze fotos, nove foram selecionadas para treino e duas fotos de cada individuo foram separadas para teste e nove para treinamento, assegurando representatividade em ambos os conjuntos.

## 3.2 Criação da base de dados através de *Data Augmentation*

Através do *Data Augmentation*, será possível gerar imagens artificiais para aumentar a base de treinamento do IA, sendo um componente crucial para o desempenho do algoritmo seja satisfatório e apresente maior acurácia é a base de dados utilizada no treinamento, pois ela precisa ser mais abrangente possível.

O aumento de dados foi realizado utilizando a biblioteca `imgaug`, que permite a aplicação sistemática de transformações geométricas e fotométricas às imagens originais. Foram selecionadas seis técnicas distintas, com combinações aleatórias das seis técnicas de *data augmentation* para produzir sete versões aumentadas de cada imagem original:

1. **Flip Horizontal** (`Fliplr`) - Inversão lateral da imagem com probabilidade de 50% ($p = 0.5$), simulando variações de perspectiva.

2. **Transformação Afim** (`Affine`) - Aplicação de rotação (faixa de $\pm 20^{\circ}$), escala fixa ($0.8\times$) e translação aleatória (até 10% nas direções $x$ e $y$).

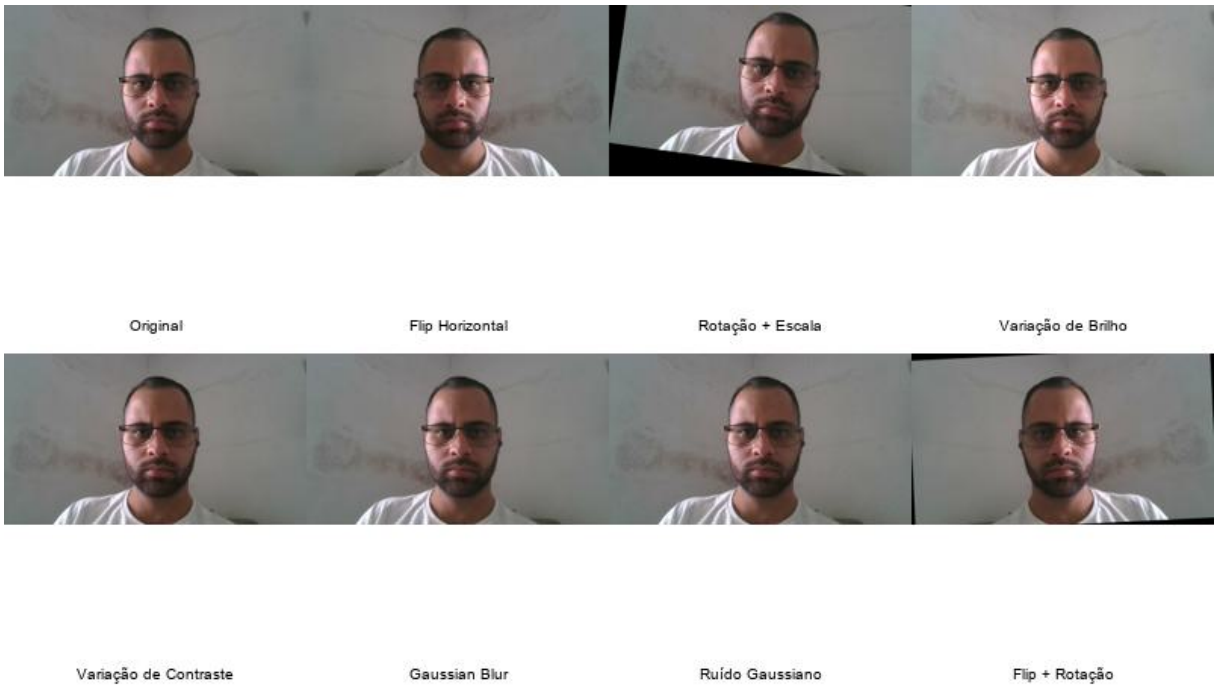
3. **Modulação de Brilho** (`Multiply`) - Multiplicação dos valores dos pixels por um fator aleatório entre 0.8 e 1.2, simulando variações de iluminação.

4. **Ajuste de Contraste** (`LinearContrast`) - Variação linear do contraste com fator entre 0.8 e 1.2, aumentando a robustez a mudanças na distribuição de intensidades.

5. **Filtro Gaussiano** (`GaussianBlur`) - Aplicação de *blur* com desvio padrão $\sigma$ variando entre 0.0 e 1.0, simulando desfoques naturais.

6. **Ruído Gaussiano Aditivo** (`AdditiveGaussianNoise`) - Inserção de ruído aleatório com amplitude de até 5% do valor máximo do pixel (scale $= 0.05 \times 255$), replicando artefatos de aquisição.

Figura 8 – Grid de técnicas de Data Augmentation aplicadas




Fonte: Elaborado pelo autor, 2025.

A Figura 8 ilustra um grid comparativo das técnicas de Data Augmentation utilizadas para aumento da base de dados.

### 3.2.1 Divisão da Base de Dados

Após aumento no números das imagens, a base de dados foi organizada e dividida em conjuntos de treino e teste, etapa essencial para o desenvolvimento do modelo de *machine learning*. O conjunto de treino deve conter a maioria dos dados, garantindo aprendizado adequado, enquanto o conjunto de teste é reservado para avaliação do desempenho do modelo.

A base de treino foi subdividida em cinco grupos para simular diferentes condições de disponibilidade de imagens:

1. **Primeira subdivisão (DB1):** Contém uma única foto por indivíduo, simulando a situação mais restrita, próxima ao que o site da PCMG disponibiliza para treinamento do modelo.
2. **Segunda subdivisão (DB2):** Totaliza 120 imagens, sendo 15 originais (uma por indivíduo) e 105 imagens aumentadas artificialmente, ampliando a variabilidade dos dados.
3. **Terceira subdivisão (DB3):** Composta por 360 imagens, incluindo 45 originais (três fotos diferentes por indivíduo) e 315 aumentadas, aumentando ainda mais a diversidade do conjunto.

4. **Quarta subdivisão (DB4):** Contém 720 imagens, sendo 90 originais (seis fotos diferentes por indivíduo) e 630 aumentadas, proporcionando maior riqueza de informações para o treinamento.

5. **Quinta subdivisão (DB5):** Totaliza 1080 imagens, com 135 originais (nove fotos diferentes por indivíduo) e 945 aumentadas, representando o conjunto mais completo para o aprendizado do modelo.

A divisão dos dados foi realizada de forma aleatória para evitar vieses relacionados a características como etnia ou iluminação, garantindo que o modelo fosse treinado e testado em amostras representativas e diversificadas, favorecendo a generalização e o desempenho em cenários reais.

## 3.3 Treinamento do Algoritmo Eigenfaces - Reconhecimento Facial

Para a implementação do reconhecimento facial baseado em Eigenfaces, utilizou a biblioteca OpenCV em python com a base de dados com variação do número de fotos conforme descrito 3.2.1 e da variação de hiperparâmetros, conforme descrito em 2.6.2 no item 5.

Conforme detalhado na Tabela 1, o classificador `EigenFaceRecognizer` foi configurado com diferentes combinações de parâmetros para avaliação de desempenho. O número de componentes principais (`num_components`) variou entre 20 e 100, permitindo analisar o *trade-off* entre precisão, generalização e eficiência do algoritmo. Se aumentar o número de componentes, é possível melhorar a precisão do modelo, pois mais detalhes dos dados são capturados. No entanto, isso também eleva o custo computacional e o risco de *overfitting*, capturando não apenas os padrões reais, mas também ruídos e variações aleatórias. Por outro lado, ao reduzir a quantidade de componentes, o modelo se torna mais rápido, mas pode acabar perdendo informações importantes, o que pode comprometer a acurácia.

Além disso, foram testados diferentes valores para o limiar de decisão (`threshold`), incluindo valores fixos (2000 e 4000) e um valor padrão (`DBL_MAX`[2]), este último desativando a rejeição automática de rostos desconhecidos.[3]

[2] DBL_MAX = $1.7976931348623157 \times 10^{308}$

[3] No método Eigenfaces, o número máximo de componentes principais (Eigenfaces) é determinado pela matriz de covariância, conforme 2.6.2, portanto, para DB1 será utilizado somente a variação do threshold.

Tabela 1 – Configurações de parâmetros testadas no classificador Eigenfaces

| num_components | threshold |
|---|---|
| 0 | DBL_MAX |
| 20 | DBL_MAX |
| 20 | 2000 |
| 20 | 4000 |
| 40 | DBL_MAX |
| 40 | 2000 |
| 40 | 4000 |
| 80 | DBL_MAX |
| 80 | 2000 |
| 80 | 4000 |
| 100 | DBL_MAX |
| 100 | 2000 |
| 100 | 4000 |

Fonte: Elaborado pelo autor, 2025.

Após a criação dos classificadores, treina-se o modelo relacionando as faces da base de dados e sua identificação através do método `eigen_classifier.train(faces, ids)`. O método `train` implementa o núcleo matemático, conforme 2.6.2, do algoritmo Eigenfaces.

Ele se inicia com o cálculo da imagem média, conforme 2.6.2 item 1 do conjunto de treinamento, onde todos os rostos são somados pixel a pixel e divididos pelo número total de imagens, gerando uma representação facial média que serve como ponto de centralização dos dados.

Em seguida, cada rosto é transformado em um vetor unidimensional e normalizado pela subtração dessa imagem média, eliminando características comuns e destacando variações individuais, conforme 2.6.2 item 2.

Esses vetores normalizados são organizados em uma matriz de covariância, conforme 2.6.2 item 3, cuja decomposição em autovalores e autovetores via PCA identifica as eigenfaces – os autovetores que correspondem às direções de maior variância nos dados, ordenados de forma decrescente pelos autovalores associados. A definição do parâmetro num_components determina quantas dessas eigenfaces (as mais significativas) são retidas para compor o subespaço facial reduzido.

Os rostos de treino são projetados nesse subespaço através de uma multiplicação matricial entre os vetores normalizados e a matriz de eigenfaces, conforme 2.6.2 item 6,convertendo cada imagem original de 6.400 pixels (80×80),conforme 2.6.1 em um vetor compacto de características com dimensão igual ao número de componentes (num_components), que encapsula essencialmente a "assinatura facial"única de cada indivíduo.

Através do método `eigen_classifier.write` salva-se um modelo treinado no formato YAML *(YAML Ain't Markup Language)*, que é uma linguagem de serialização de dados usada para escrever arquivos de configuração. Este arquivo armazena integralmente o estado interno do classificador Eigenfaces, incluindo: parâmetros de configuração (como num_components e threshold), as eigenfaces (componentes principais que definem o subespaço facial), a imagem média do conjunto de treinamento, os autovalores associados a cada componente, as projeções reduzidas de cada rosto de treino e seus IDs correspondentes.

Após o carregamento do modelo Eigenfaces treinado através de `eigen_classifier.read()`, o sistema realiza a predição nas imagens de teste utilizando o método `classificador.predict()`. Este método recebe como entrada cada face pré-processada (redimensionada para 80×80 pixels e convertida para escala de cinza) e retorna uma tupla com dois valores fundamentais: o ID numérico previsto (representando a identidade do indivíduo reconhecido, como "subject15"para ID 15) e o nível de confiança da predição, calculado como a distância Euclidiana entre a projeção da face teste e a projeção mais próxima no espaço de características reduzido. Quanto menor o valor de confiança, maior a similaridade com o indivíduo previsto, sendo que o limiar configurado (threshold) determina o valor máximo aceitável para considerar uma predição válida - valores acima deste limiar são tratados como "desconhecidos".

Após a obtenção das previsões do modelo para todas as imagens de teste, procede-se ao cálculo das métricas de avaliação de desempenho, acurácia e matriz de confusão, que serão apresentados na seção 4.

A acurácia é uma métrica usada para medir o quanto um modelo de classificação acerta suas previsões em relação ao total de tentativas realizadas. Ela representa a proporção (ou percentual) de previsões corretas — tanto de exemplos positivos quanto negativos — em relação ao total de exemplos avaliados. A fórmula geral da acurácia é:

$$\text{Acurácia} = \frac{\text{Número de Previsões Corretas}}{\text{Número Total de Amostras}} \tag{3.1}$$

A matriz de confusão é uma tabela n×n que resume as predições de um classificador comparando-as com os rótulos verdadeiros. Cada linha corresponde à classe real e cada coluna à classe prevista, permitindo verificar onde o modelo acertou ou errou. A estrutura básica é:

Tabela 2 – Matriz de Confusão - Resultados de Classificação

| | **Previsto Classe A** | **Previsto Classe B** |
|---|---|---|
| **Real Classe A** | Verdadeiros Positivos (VP) | Falsos Negativos (FN) |
| **Real Classe B** | Falsos Positivos (FP) | Verdadeiros Negativos (VN) |

Fonte: Elaborado pelo autor, 2025.

A escolha da acurácia e da matriz de confusão como métricas de avaliação, em detrimento de outras, está relacionado com a capacidade de oferecer clareza interpretativa e diagnóstico direcionado na avaliação do modelo. A acurácia foi escolhida por fornecer uma medida intuitiva de desempenho — resumindo a taxa de acertos totais apresentada de forma numérica que é acessível a indivíduos não técnicos —, enquanto a matriz de confusão desagrega esse resultado em padrões de erro específicos (como falsos positivos ou falsos negativos), revelando onde e como o modelo falha, o que é crítico para eventuais ajustes.

# 4 RESULTADOS

No total, foram realizados 56 testes, organizados em cinco bases de dados distintas (DB1 a DB5), conforme apresentado nas tabelas abaixo. Cada tabela contém informações sobre os testes, incluindo o número do teste, a quantidade de imagens originais (Origin), o total de imagens aumentadas por técnicas de data augmentation (Adic.), o número de componentes analisados (N°Comp), o valor do threshold aplicado e a acurácia obtida em cada teste.

A tabela 3 apresenta os testes de 1 a 4, que contém base mínima, sem imagens aumentadas, apresentou acurácia predominante de 50,00% em três dos quatro testes (1, 2 e 4), onde foram utilizados thresholds elevados (DBL_MAX ou 4000). O número de componentes (0 ou 15) não influenciou o resultado nessas configurações. Entretanto, quando o threshold foi reduzido para 2000 (Teste 3), a acurácia despencou para 20,00%, evidenciando a extrema sensibilidade do modelo a limiares baixos mesmo em bases pequenas.

Tabela 3 – Resultados do Algoritmo Eigenfaces (Testes 1-4)

| **Teste** | **Orig.** | **Adic.** | **Total** | **N° Comp.** | **Threshold** | **Acurácia** |
|---|---|---|---|---|---|---|
| 1 | 15 | 0 | 15 | 0 | DBL_MAX | 50,00% |
| 2 | 15 | 0 | 15 | 15 | DBL_MAX | 50,00% |
| 3 | 15 | 0 | 15 | 15 | 2000 | 20,00% |
| 4 | 15 | 0 | 15 | 15 | 4000 | 50,00% |

Fonte: Elaborado pelo autor, 2025.

A tabela 4 apresenta os testes de 5 a 17, que possui a adição de 105 imagens sintéticas, observou-se um padrão dicotômico. Configurações com threshold 2000 (Testes 6, 9, 12 e 15) mantiveram acurácia mínima de 20,00% independentemente do número de componentes (20–100). Já sob thresholds altos (4000/DBL_MAX), a acurácia estabilizou em 50,00% em todos os demais testes, demonstrando que o aumento de dados sem ajuste adequado do limiar não trouxe ganhos.

Tabela 4 – Resultados do Algoritmo Eigenfaces (Testes 5-17)

| Teste | Orig. | Adic. | Total | N° Comp. | Threshold | Acurácia |
|---|---|---|---|---|---|---|
| 5 | 15 | 105 | 120 | 0 | DBL_MAX | 50,00% |
| 6 | 15 | 105 | 120 | 20 | 2000 | 20,00% |
| 7 | 15 | 105 | 120 | 20 | 4000 | 50,00% |
| 8 | 15 | 105 | 120 | 20 | DBL_MAX | 50,00% |
| 9 | 15 | 105 | 120 | 40 | 2000 | 20,00% |
| 10 | 15 | 105 | 120 | 40 | 4000 | 50,00% |
| 11 | 15 | 105 | 120 | 40 | DBL_MAX | 50,00% |
| 12 | 15 | 105 | 120 | 80 | 2000 | 20,00% |
| 13 | 15 | 105 | 120 | 80 | 4000 | 50,00% |
| 14 | 15 | 105 | 120 | 80 | DBL_MAX | 50,00% |
| 15 | 15 | 105 | 120 | 100 | 2000 | 20,00% |
| 16 | 15 | 105 | 120 | 100 | 4000 | 50,00% |
| 17 | 15 | 105 | 120 | 100 | DBL_MAX | 50,00% |

Fonte: Elaborado pelo autor, 2025.

A tabela 5 apresenta os testes de 18 a 30, que possui a expansão para 315 imagens revelou maior resiliência. Em 11 dos 13 testes com threshold $\geq 4000$, a acurácia atingiu 66,67% mesmo com componentes variando (0–100). Contudo, combinações com threshold 2000 e componentes elevados ($\geq$80) causaram quedas bruscas (Testes 25 e 28: 40,00%). Destaca-se o Teste 19 (20 componentes + threshold 2000) que alcançou 60,00%, sugerindo que bases médias podem tolerar melhor limiares baixos com poucos componentes.

Tabela 5 – Resultados do Algoritmo Eigenfaces (Testes 18-30)

| Teste | Orig. | Adic. | Total | N° Comp. | Threshold | Acurácia |
|---|---|---|---|---|---|---|
| 18 | 45 | 315 | 360 | 0 | DBL_MAX | 66,67% |
| 19 | 45 | 315 | 360 | 20 | 2000 | 60,00% |
| 20 | 45 | 315 | 360 | 20 | 4000 | 66,67% |
| 21 | 45 | 315 | 360 | 20 | DBL_MAX | 66,67% |
| 22 | 45 | 315 | 360 | 40 | 2000 | 46,67% |
| 23 | 45 | 315 | 360 | 40 | 4000 | 66,67% |
| 24 | 45 | 315 | 360 | 40 | DBL_MAX | 66,67% |
| 25 | 45 | 315 | 360 | 80 | 2000 | 40,00% |
| 26 | 45 | 315 | 360 | 80 | 4000 | 66,67% |
| 27 | 45 | 315 | 360 | 80 | DBL_MAX | 66,67% |
| 28 | 45 | 315 | 360 | 100 | 2000 | 40,00% |
| 29 | 45 | 315 | 360 | 100 | 4000 | 66,67% |
| 30 | 45 | 315 | 360 | 100 | DBL_MAX | 66,67% |

Fonte: Elaborado pelo autor, 2025.

A tabela 6 apresenta os testes de 31 a 43. Esta base registrou o primeiro salto significativo: os Testes 42–43 (100 componentes + threshold $\geq$ 4000) atingiram 70,00% de acurácia. Notou-se variação atípica com 40 componentes (Testes 36–37: 63,33% vs. 66,67% em outras configurações). Thresholds baixos continuaram limitando o desempenho ($\leq$60,00%), com pior resultado no Teste 41 (43,33%), reforçando que dados abundantes não compensam limiares inadequados.

Tabela 6 – Resultados do Algoritmo Eigenfaces (Testes 31-43)

| **Teste** | **Orig.** | **Adic.** | **Total** | **N° Comp.** | **Threshold** | **Acurácia** |
|---|---|---|---|---|---|---|
| 31 | 90 | 630 | 720 | 0 | DBL_MAX | 66,67% |
| 32 | 90 | 630 | 720 | 20 | 2000 | 60,00% |
| 33 | 90 | 630 | 720 | 20 | 4000 | 66,67% |
| 34 | 90 | 630 | 720 | 20 | DBL_MAX | 66,67% |
| 35 | 90 | 630 | 720 | 40 | 2000 | 56,67% |
| 36 | 90 | 630 | 720 | 40 | 4000 | 63,33% |
| 37 | 90 | 630 | 720 | 40 | DBL_MAX | 63,33% |
| 38 | 90 | 630 | 720 | 80 | 2000 | 43,33% |
| 39 | 90 | 630 | 720 | 80 | 4000 | 66,67% |
| 40 | 90 | 630 | 720 | 80 | DBL_MAX | 66,67% |
| 41 | 90 | 630 | 720 | 100 | 2000 | 43,33% |
| 42 | 90 | 630 | 720 | 100 | 4000 | 70,00% |
| 43 | 90 | 630 | 720 | 100 | DBL_MAX | 70,00% |

Fonte: Elaborado pelo autor, 2025.

A tabela 7 apresenta os testes de 44 a 56. O ápice do estudo ocorreu aqui: o Teste 44 (0 componentes + threshold DBL_MAX) alcançou 73,33%. Sob thresholds altos ($\geq$4000), a acurácia manteve-se estável (66,67–70,00%), mesmo com componentes entre 0–100. Quando o threshold foi fixado em 2000, a acurácia variou de 46,67% (Teste 54) a 66,67% (Teste 45), comprovando que bases maximizadas atenuam, mas não eliminam, os efeitos de limiares baixos.

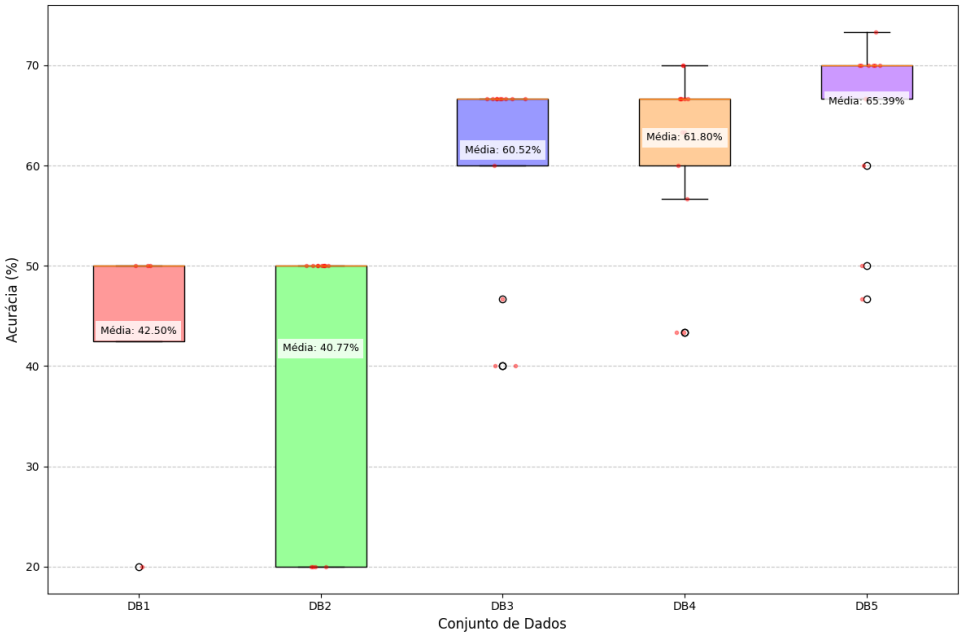
Tabela 7 – Resultados do Algoritmo Eigenfaces (Testes 44-56)

| Teste | Orig. | Adic. | Total | N° Comp. | Threshold | Acurácia |
|---|---|---|---|---|---|---|
| 44 | 135 | 945 | 1080 | 0 | DBL_MAX | 73,33% |
| 45 | 135 | 945 | 1080 | 20 | 2000 | 66,67% |
| 46 | 135 | 945 | 1080 | 20 | 4000 | 66,67% |
| 47 | 135 | 945 | 1080 | 20 | DBL_MAX | 66,67% |
| 48 | 135 | 945 | 1080 | 40 | 2000 | 60,00% |
| 49 | 135 | 945 | 1080 | 40 | 4000 | 70,00% |
| 50 | 135 | 945 | 1080 | 40 | DBL_MAX | 70,00% |
| 51 | 135 | 945 | 1080 | 80 | 2000 | 50,00% |
| 52 | 135 | 945 | 1080 | 80 | 4000 | 70,00% |
| 53 | 135 | 945 | 1080 | 80 | DBL_MAX | 70,00% |
| 54 | 135 | 945 | 1080 | 100 | 2000 | 46,67% |
| 55 | 135 | 945 | 1080 | 100 | 4000 | 70,00% |
| 56 | 135 | 945 | 1080 | 100 | DBL_MAX | 70,00% |

Fonte: Elaborado pelo autor, 2025.

Após a análise das tabelas, foi realizada uma avaliação estatísticas dos dados, e o gráfico exibe as informações como média, mediana, desvio padrão, mínimo, máximo e intervalo de acurácia por divisão da base de dados, conforme a seção 3.2.1.

Figura 9 – Distribuição de Acurácia por Conjunto de Dados




Fonte: Elaborado pelo autor, 2025.

A figura 9 apresenta comparações estatísticas pra as cinco bancos de dados (DB1 a DB5).

Percebe-se diferenças no desempenho do algoritmo que resulta em diferentes acurácias, ou seja, o quanto cada conjunto foi preciso nas detecções de faces.

No caso do DB1, que possui apenas quatro amostras, observa-se a pior média, cerca de 42,5%. Também é possível notar uma grande variação entre os resultados: houveram valores mínimos de 20%, indicando muita inconsistência. Essa diferença fica nítida ao analisar a distância entre a média e a mediana, sugerindo que valores baixos puxaram a média para baixo. O desvio padrão de 12,99 indica grande dispersão nos dados.

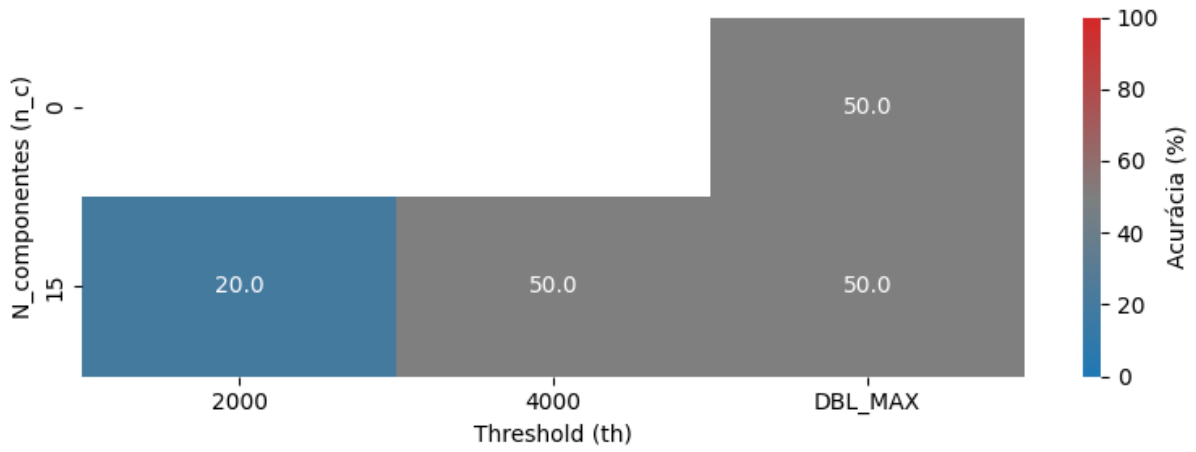
O DB2, com uma amostragem maior (13 amostras), não apresenta melhorias significativas: a média é de 40,77%, ainda baixa, e a variação permanece elevada, com desvio padrão de 13,85. A diferença entre média e mediana continua indicando a influência de alguns resultados muito baixos.

Já no DB3 também com 13 amostras, observa-se uma melhora: a média aumenta para 60,52% e os resultados apresentam menor dispersão (desvio padrão de 10,28). A mediana de 66,67% sinaliza que a maioria dos resultados está acima de 60%. O DB4, com 13 amostras, apresenta pequena melhoria, com média de 61,80% e menor variação (desvio padrão de 8,64), evidenciando maior estabilidade nos resultados.

Por fim, o DB5, com 13 amostras, destaca-se pelo melhor desempenho: média de 65,39%, desvio padrão reduzido para 7,90 e mediana de 70%. Isso indica que metade das observações alcançou pelo menos 70%, e mesmo o resultado mínimo dessa base foi superior aos mínimos dos outros conjuntos, sugerindo que o aumento da base dados melhora a acurácia do algoritmo Eigenfaces.

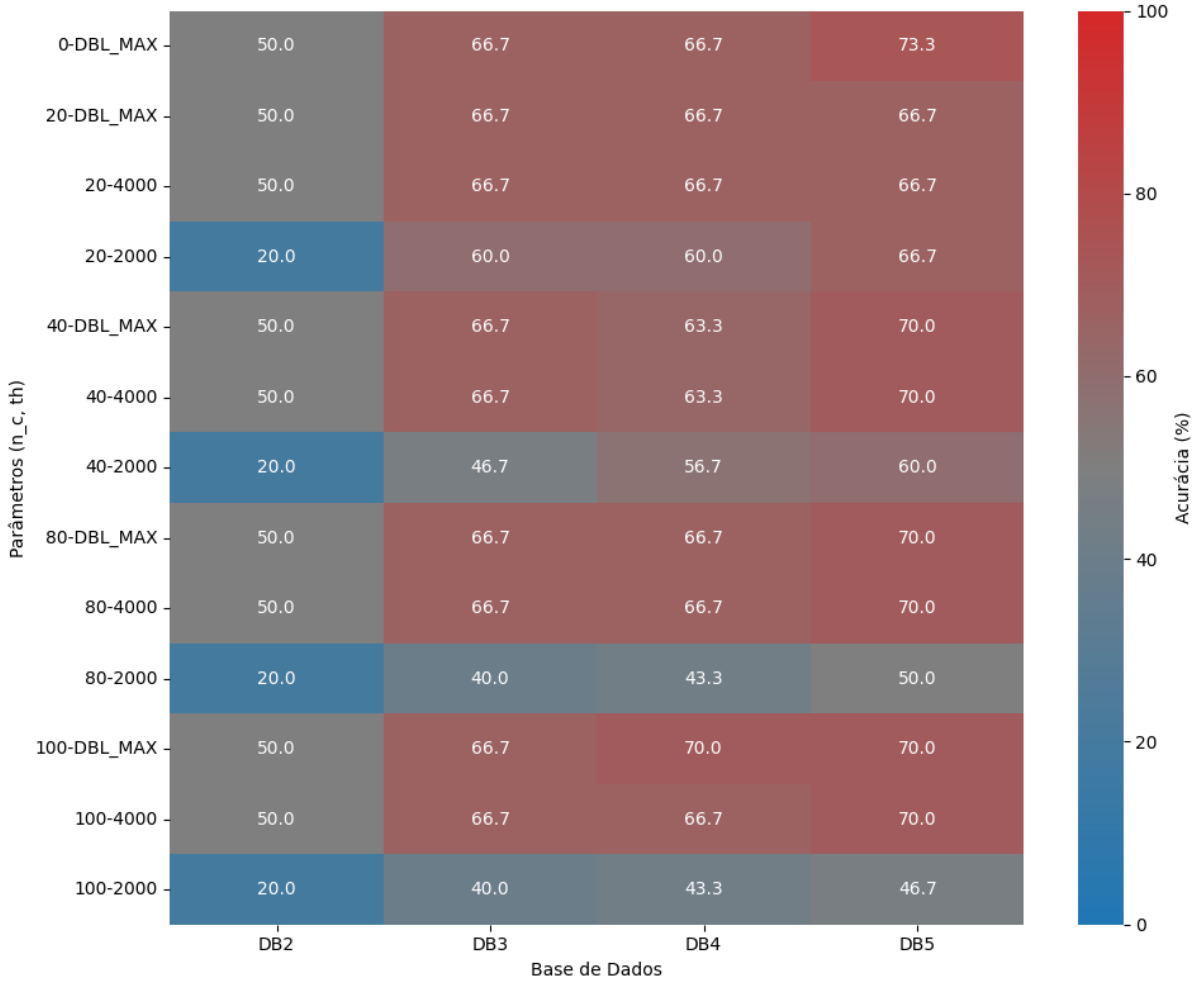
Ao analisar separadamente cada conjunto de dados de acordo com diferentes combinações de número de componentes e threshold, conforme tabela 1, obtem o resultado apresentado nas figuras 10 e 11.

Figura 10 – Acurácia por Combinação de Parâmetros e Base de Dados BD1




Fonte: Elaborado pelo autor, 2025.

Figura 11 – Acurácia por Combinação de Parâmetros e Base de Dados BD2 a BD5




Fonte: Elaborado pelo autor, 2025.

As Figuras 10 e 11 apresentam dois mapas de calor codificado por cores, cuja escala varia do azul ao vermelho. Nesse gradiente, os tons de azul indicam as menores acurácias, os tons de cinza representam valores intermediários e os tons de vermelho correspondem às maiores acurácias observadas.

Além disso, essas figuras evidencia que a acurácia tende a aumentar proporcionalmente ao número de imagens utilizadas na base de dados. Observa-se acurácia média e baixa em bases menores, enquanto bases maiores apresentam regiões predominantemente avermelhadas, indicando desempenho superior.

Adicionalmente, verifica-se que o valor do *threshold* constitui outro fator crítico para o desempenho do modelo: limiares baixos — como 2000 — comprometem significativamente a acurácia (20–66,67%), ao passo que *thresholds* mais elevados (como 4000 ou `DBL_MAX`) mantêm ou até ampliam o desempenho (46,67–73,33%).

Esse comportamento se deve ao fato de que *thresholds* baixos aumentam a incidência de falsos negativos, rejeitando amostras que, na realidade, pertencem à classe correta, conforme discutido na Seção 2.6.3, item 3. Quando associados a um número elevado de componentes principais no modelo — por exemplo, 100 —, *thresholds* baixos intensificam os efeitos da alta dimensionalidade: as maiores distâncias no espaço de características amplificam os erros de classificação, resultando em quedas acentuadas na acurácia, especialmente em conjuntos maiores, como o DB5.

Para cada combinação de parâmetros e bases de dados, foram geradas matrizes de confusão correspondentes aos valores de acurácia obtidos. Especificamente:

- Para a base de dados DB1, foram produzidas 4 matrizes de confusão, correspondentes às 4 combinações de parâmetros testadas
- Para cada uma das demais bases de dados (DB2 a DB5), foram geradas 13 matrizes de confusão cada, representando todas as combinações paramétricas avaliadas

Considerando que o foco desta análise recai sobre os resultados com maior desempenho, examinaremos em detalhes a matriz de confusão associada à base de dados DB5, especificamente para a configuração de parâmetros especificada na Tabela 1 com número de componentes principais igual a 0 e threshold igual a DBL_MAX ($1.7976931348623157 \times 10^{308}$).

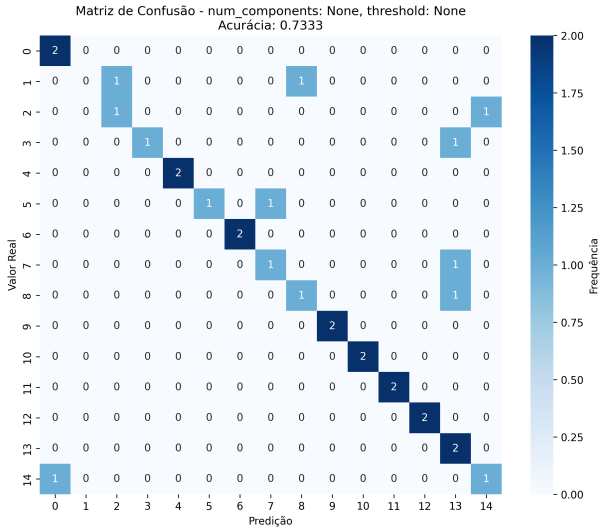
No contexto deste estudo, onde trabalhamos com 15 classes distintas (cada uma representando uma pessoa diferente), a matriz assume dimensões de 15×15 (índices de 0 a 14), oferecendo uma visão detalhada do comportamento do classificador.

Os elementos ao longo da diagonal principal da matriz representam os casos em que o modelo acertou a classificação, ou seja, quando uma imagem foi corretamente identificada como

pertencente à pessoa correspondente. Por outro lado, os valores fora da diagonal indicam erros de classificação, nos quais o algoritmo confundiu uma pessoa com outra

No DB5, com nove fotografias por indivíduo (135 originais) e suas derivações, obteve-se um total de 1080 imagens (135 originais + 945 aumentadas). A acurácia atingiu 73.3% – o ápice da série experimental, dados parâmetros de número de componentes e threshold . Esses dados reforçam que a quantidade e diversidade de imagens originais, potencializadas por técnicas de aumento, são determinantes para a eficácia do modelo.

Figura 12 – Matriz de confusão DB5 ($n_c = 0, th = DBL\_MAX$)




Fonte: Elaborado pelo autor, 2025.

Conforme evidenciado na Figura 12, que representa a matriz de confusão correspondente a este teste, observa-se uma concentração de acertos mais consistente em determinadas classes, especialmente em seis indivíduos: 4, 6, 9, 10, 11 e 12. Esses foram reconhecidos corretamente duas vezes cada, sem apresentar nenhuma confusão com outras classes, o que demonstra um ótimo desempenho na identificação desses rostos. Esse resultado sugere que suas imagens — tanto originais quanto aumentadas — mantêm boa qualidade, variabilidade adequada e características faciais bem distintas.

Os indivíduos 0 e 13 também foram reconhecidos corretamente duas vezes, porém com confusões adicionais, indicando que embora o modelo seja capaz de identificá-los, ainda encontra semelhanças com outras classes, o que pode comprometer a confiabilidade em cenários reais.

O indivíduo 2 é um caso de atenção, pois não foi corretamente reconhecido nenhuma vez, o que representa uma falha completa na classificação para essa classe. Isso pode estar relacionado à baixa qualidade ou diversidade das imagens associadas a esse indivíduo, ou ainda à similaridade facial com outros indivíduos da base, o que torna sua identificação particularmente difícil. Indivídios como 5 e 14 foram reconhecidos uma vez apenas.

# 5 CONCLUSÃO E TRABALHOS FUTUROS

## 5.1 Conclusão

Diante dos resultados apresentados[1], conclui-se que a diversificação e ampliação da base de imagens por indivíduo são fundamentais para a eficácia do reconhecimento facial. Os testes demonstraram correlação direta entre volume de dados (originais e aumentados) e acurácia, com a base DB5 (1.080 imagens) alcançando 73,33% de precisão, superando significativamente bases menores como DB1 (42,50%). Essa superioridade deve-se à maior capacidade de capturar características faciais robustas e generalizáveis quando múltiplas imagens por indivíduo estão disponíveis.

Paralelamente, a calibração do *threshold* mostrou-se crítica para o desempenho. Limiares baixos (ex: 2000) resultaram em quedas bruscas na acurácia (20,00% a 46,67%), mesmo em bases ampliadas, devido ao aumento de falsos negativos. Em contraste, *thresholds* elevados (4000 ou `DBL_MAX`) mantiveram ou melhoraram os resultados (até 73,33%), por reduzirem rejeições indevidas de amostras válidas.

Quanto ao número de componentes principais (PCA), seu impacto foi secundário frente ao *threshold*, mas revelou nuances importantes. Em bases menores, a quantidade de componentes (0-100) não alterou significativamente os resultados. Porém, em conjuntos maiores (DB5), configurações específicas - como 0 componentes com *threshold* `DBL_MAX` - atingiram desempenho máximo, sugerindo que a redução de dimensionalidade nem sempre é benéfica com dados suficientes e bem calibrados.

A sinergia entre parâmetros técnicos e diversidade amostral demonstrou ser determinante. Bases amplas atenuaram, mas não eliminaram, os efeitos negativos de *thresholds* inadequados (ex: DB5 com *threshold* 2000: 46,67%). Contudo, a otimização conjunta - como em DB5 com *threshold* alto e 0 componentes - maximizou a assertividade, comprovando que a interação entre volume/qualidade de dados e ajuste paramétrico é fundamental.

Recomenda-se prioritariamente:

1. **Expandir a base por indivíduo** com múltiplas imagens reais (expressões, ângulos) e técnicas de *data augmentation*
2. **Priorizar *thresholds* elevados** ($\geq$4000) para minimizar falsos negativos
3. **Avaliar configurações de componentes** conforme tamanho da base, optando por menor dimensionalidade em conjuntos limitados

[1] Os resultados obtidos foram gerados em ambiente controlado, utilizando imagens estáticas. Em situações reais, com ambientes dinâmicos e câmeras diferentes, o desempenho do sistema pode variar devido a fatores como variações na iluminação, ângulo de captura, movimento das pessoas e qualidade das câmeras, o que pode impactar na precisão e consistência dos resultados.

Portanto, este trabalho atingiu seus objetivos gerais e específicos ao desenvolver e validar um sistema de inteligência artificial baseado no método *Eigenfaces* para auxiliar na identificação de pessoas desaparecidas, demonstrando a viabilidade técnica através de resultados favorável na taxa de acurácia. A implementação completa do processo — desde a extração de dados em *sites* oficiais, realizada em conformidade com a Lei Geral de Proteção de Dados (LGPD), passando pelo tratamento das imagens até a aplicação do algoritmo de reconhecimento facial — confirma o potencial da solução como instrumento de apoio para órgãos de segurança pública.

## 5.2 Trabalhos Futuros

Para consolidar os avanços deste trabalho, propõem-se dois eixos de desenvolvimento futuro. Primeiramente, sugere-se a implementação de um classificador padronizado em YML (YAML) integrado ao banco de dados da Polícia Militar, com testes em câmeras públicas urbanas. Essa validação prática deve focar em: Desempenho em cenários reais (variação lumínica, movimento e multidões) e redução de falsos positivos/negativos cruzados com registros históricos.

Em segundo plano, sugere-se uma análise comparativa de algoritmos de reconhecimento facial, contemplando diferentes complexidades computacionais. Como referência de baixa complexidade, o algoritmo LBPH (Local Binary Patterns Histograms) seria avaliado por sua eficiência em hardware limitado (ex.: câmeras IoT) e adaptabilidade a condições lumínicas adversas. Para alta complexidade, seria escolhida opção por soluções baseadas em redes neurais profundas como FaceNet, que oferecem precisão superior mediante *embeddings* e robustez a ângulos críticos. A comparação utilizaria métricas como acurácia, tempo de inferência, consumo de memória e escalabilidade.

# REFERÊNCIAS

BRASIL. Lei nº 13.709 de 14 de agosto 2018. dispõe sobre a proteção de dados pessoais e altera a lei nº 12.965, de 23 de abril de 2014 (marco civil da internet). **Diário Oficial [da] República Federativa do Brasil**, Brasília, DF, 2018. Citado na página 30.

CABRAL, G. A. *et al.* Ferramenta de reconhecimento facial para auxiliar na busca de pessoas consideradas desaparecidas. **Revista Científica da UNIFENAS-ISSN: 2596-3481**, v. 6, n. 5, 2024. Citado na página 14.

CHIEN, Y. Pattern classification and scene analysis. **IEEE Transactions on Automatic Control**, v. 19, n. 4, p. 462–463, 1974. Citado 2 vezes nas páginas 23 e 24.

(DECDACRIM/SIIP/PCMG), D. de Estatística e Análise Criminal da Superintendência de Informações e Inteligência Policial da Polícia Civil do Estado de M. G.; (DRPD/DHPP/-SIPJ/PCMG), D. E. de Referência a Pessoa Desaparecida Departamento Estadual de Investigação de Homicídios e Proteção à Pessoa da Superintendência de Investigação e P. J. Relatório estatístico: Diagnóstico de pessoas desaparecidas e localizadas nas regiões integradas de segurança pública de minas gerais de 2020 a 2022. p. 44, 2023. Citado 2 vezes nas páginas 14 e 15.

DOGRA, K. S.; NIRWAN, N.; CHAUHAN, R. Unlocking the market insight potential of data extraction using python-based web scraping on flipkart. In: **2023 International Conference on Sustainable Emerging Innovations in Engineering and Technology (ICSEIET)**. [S.l.: s.n.], 2023. p. 453–457. Citado 2 vezes nas páginas 17 e 20.

EJAZ, M.; ISLAM, M.; SIFATULLAH, M.; SARKER, A. Implementation of principal component analysis on masked and non-masked face recognition. In: . [S.l.: s.n.], 2019. p. 1–5. Citado 2 vezes nas páginas 25 e 26.

ESTEVA, A. *et al.* Dermatologist-level classification of skin cancer with deep neural networks. **nature**, Nature Publishing Group UK London, v. 542, n. 7639, p. 115–118, 2017. Citado na página 23.

JAIN, A. K.; FLYNN, P.; ROSS, A. A. **Handbook of Biometrics**. California: Springer; 2008 th edition (October 29, 2007), 2007. ISBN 978-0387710402. Citado na página 24.

KHDER, M. A. Web scraping or web crawling : State of art, techniques, approaches and application. Int. J. Advance Soft Compu. Appl, v. 13, n. 3, 2021. Citado na página 21.

KSHIRSAGAR, V.; BAVISKAR, M.; GAIKWAD, M. Face recognition using eigenfaces. In: **2011 3rd International Conference on Computer Research and Development**. [S.l.: s.n.], 2011. v. 2, p. 302–306. Citado 3 vezes nas páginas 25, 28 e 29.

LITJENS, G. *et al.* A survey on deep learning in medical image analysis. **Medical image analysis**, Elsevier, v. 42, p. 60–88, 2017. Citado na página 22.

MORAES, J. L. de. **Controle de Acesso Baseado em Biometria Facial**. 102 p. Dissertação (Mestrado) — Universidade Federal do Espírito Santo, Vitória, 2010. Citado na página 25.

MULYONO, I. *et al.* Performance analysis of face recognition using eigenface approach. In: . [S.l.: s.n.], 2019. p. 1–5. Citado na página 26.

OLIVEIRA, W. D. G. de. **Data Augmentation via Generative Adversarial Networks aplicado em classificação de imagens**. Tese (Doutorado) — Universidade Federal de São Paulo–UNIFESP, 2019. Citado na página 21.

OPENCV. **cv::face::EigenFaceRecognizer Class Reference**. 2024. Disponível em: <https://docs.opencv.org/4.x/dd/d7c/classcv_1_1face_1_1EigenFaceRecognizer.html#a22c8392f27a20b24d04351b675e7b6db>. Acesso em: 02 jul. 2024. Citado na página 29.

REVISION, A. J. **Docs imgaug**. 2025. Disponível em: <https://imgaug.readthedocs.io/en/latest/#>.Acesso em: 05 de janeiro de 2025, 08:30:15. Disponível em: <https://imgaug.readthedocs.io/en/latest/#>. Citado na página 22.

SANTOS, A. R. dos. **Identificação de Faces Humanas Através de PCA-LCA e Redes Neurais SOM**. 154 p. Dissertação (Mestrado) — Escola de Engenharia de São Carlos da Universidade de São Paulo, São Carlos, 2005. Citado na página 26.

SHAPIRO, L.; STOCKMAN, G. **Computer Vision**. [S.l.]: Prentice Hall, 2001. ISBN 9780130307965. Citado na página 25.

SHORTEN, C.; KHOSHGOFTAAR, T. M. A survey on image data augmentation for deep learning. **Journal of Big Data**, Springer, v. 6, n. 1, p. 60, 2019. Citado na página 22.

SILVA, A. L. **Redução de Características para Classificação de Imagens de Faces**. 106 p. Dissertação (Mestrado) — Universidade do Estado do Rio Grande do Norte and a Universidade Federal Rural do Semi-Árido, Mossoró, 2016. Citado na página 25.

SILVA, A. L. da; CINTRA, M. E. Reconhecimento de padrões faciais : Um estudo. In: . [S.l.: s.n.], 2015. Citado na página 14.

SIRISURIYA, S. de S. A comparative study on web scraping. **International Research Conference, General Sir John Kotelawala Defence University, (KDU IRC 2015)**, 2015. Citado na página 17.

STEPHENS, R. **RedMonk Top 20 Languages Over Time: June 2021**. 2021. Disponível em: <https://redmonk.com/rstephens/2021/08/05/top-20-june-2021/>.Acesso em: 20 de março de 2024, 09:45:45. Disponível em: <https://redmonk.com/rstephens/2021/08/05/top-20-june-2021/>. Citado na página 20.

TEFFÉ, C. S. de; VIOLA**, M. Tratamento dedados pessoais na lgpd: estudo sobre as bases legais. **Civilistica.com**, 2020. Citado na página 32.

THOMAS, D. M.; MATHUR, S. Data analysis by web scraping using python. In: **2019 3rd International conference on Electronics, Communication and Aerospace Technology (ICECA)**. [S.l.: s.n.], 2019. p. 450–454. Citado na página 21.

TURK, M.; PENTLAND, A. Eigenfaces for recognition. **Journal of Cognitive Neuroscience**, v. 3, n. 1, p. 71–86, 1991. Citado 3 vezes nas páginas 27, 29 e 30.

TUSTISON, N. J. *et al.* N4itk: improved n3 bias correction. **IEEE transactions on medical imaging**, IEEE, v. 29, n. 6, p. 1310–1320, 2010. Citado na página 23.

UNIVERSITY, Y. **Yale Face Database**. 1997. Disponível em: <http://cvc.cs.yale.edu/cvc/projects/yalefaces/yalefaces.html>. Acessado em: 15 de setembro de 2024, 20:42:09. Disponível em: <http://cvc.cs.yale.edu/cvc/projects/yalefaces/yalefaces.html>. Citado na página 35.

WANG, X.; WANG, K.; LIAN, S. A survey on face data augmentation. **CoRR**, abs/1904.11685, 2019. Citado na página 21.

WICKHAM, H.; RUNDEL, M. Çetinkaya; GROLEMUND, G. **R for Data Science**. California: O'Reilly Media, Inc., 2023. ISBN 9781491910344. Citado 2 vezes nas páginas 18 e 19.

YANG, M.-H.; KRIEGMAN, D.; AHUJA, N. Detecting faces in images: A survey. **Pattern Analysis and Machine Intelligence, IEEE Transactions on**, v. 24, p. 34 – 58, 02 2002. Citado 3 vezes nas páginas 25, 26 e 29.