

UNIVERSIDADE ESTADUAL DE MARINGÁ
CENTRO DE TECNOLOGIA
DEPARTAMENTO DE INFORMÁTICA

YURI PIRES ALVES

Desenvolvimento de uma Ferramenta Baseada em Características
Grafométricas para Identificação de Manuscritos

Maringá
2024

YURI PIRES ALVES

Desenvolvimento de uma Ferramenta Baseada em Características
Grafométricas para Identificação de Manuscritos

Trabalho de Conclusão de Curso apresentado
à Universidade Estadual de Maringá, como
parte dos requisitos necessários à obtenção
do título de Bacharel em Informática.

Orientadora: Prof^a. Dr^a. Aline Maria
Malachini Miotto Amaral

Maringá
2024

FOLHA DE APROVAÇÃO

YURI PIRES ALVES

Desenvolvimento de uma Ferramenta Baseada em Características Grafométricas para Identificação de Manuscritos

Trabalho de Conclusão de Curso apresentado à Universidade Estadual de Maringá, como parte dos requisitos necessários à obtenção do título de Bacharel em Informática pela Comissão Julgadora composta pelos membros:

BANCA EXAMINADORA

Prof^ª. Dr^ª. Aline Maria Malachini Miotto Amaral
Universidade Estadual de Maringá — DIN/UEM

Prof. Me. Felipe Fernandes da Silva
Universidade Estadual de Maringá — DIN/UEM

Prof. Me. André Felipe Ribeiro Cordeiro
Universidade Estadual de Maringá — DIN/UEM

AGRADECIMENTOS

Em primeiro lugar, agradeço a Deus por toda proteção e saúde.

O início dessa jornada na UEM foi conturbado. Entrar em uma universidade no meio de uma pandemia, sem saber muito bem como tudo funcionava e sem o contato presencial com as pessoas foi estranho. Ainda bem que, logo no início, consegui, de forma repentina, uma grande amiga que me acompanhou em todos os momentos até o final. Além de agradecer à ela, agradeço a todos os meus amigos de turma que também sempre estiveram comigo.

Agradeço também ao PET-Informática UEM e a todos os amigos que fiz participando deste grupo, em especial a uma pessoa que me aproximei recentemente e construímos uma amizade incrível. O PET foi um alicerce muito importante pra mim em momentos difíceis e todo o aprendizado que tive neste grupo com certeza agregou muito em mim.

Agradeço à minha família por todo o apoio e suporte.

E, finalmente, agradeço à minha orientadora, professora Dra. Aline Maria Malachini Miotto Amaral pelo apoio e sugestões no desenvolvimento deste trabalho; e aos professores da banca, Me. Felipe Fernandes da Silva e Me. André Felipe Ribeiro Cordeiro, por aceitarem o convite.

Desenvolvimento de uma Ferramenta Baseada em Características Grafométricas para Identificação de Manuscritos

RESUMO

A identificação de autoria em manuscritos é um campo de estudo essencial na criminalística e na análise forense, dada a necessidade de determinar a autoria de documentos questionados. Além disso, devido à variabilidade da escrita humana e à similaridade entre diferentes autores, esta tarefa torna-se um problema desafiador. Este trabalho tem como objetivo propor uma ferramenta que busca padronizar e automatizar a identificação de manuscritos baseada em princípios jurídicos e nos preceitos da grafoscopia, além de técnicas de reconhecimento de padrões e aprendizado de máquina para extrair e analisar características grafométricas dos manuscritos, auxiliando o trabalho pericial. Espera-se que a pesquisa contribua para o desenvolvimento de um sistema automático de identificação de manuscritos preciso e confiável, que possa ser utilizado como ferramenta auxiliar em perícias grafotécnicas, reduzindo o tempo de análise e auxiliando na identificação de potenciais autores. A automação do processo não apenas acelera a análise, mas também aumenta a consistência dos resultados, minimizando o risco de erro humano. Dessa forma, a ferramenta poderá se tornar um recurso valioso tanto em investigações criminais quanto em processos judiciais, oferecendo uma base sólida e científica para a identificação de autoria de documentos manuscritos.

Palavras-chave: computação forense, manuscritos, características grafométricas, cartas forenses, inclinação axial, SVM, Random Forest.

Development of a Tool Based on Graphometric Features for Handwritten Document Identification

ABSTRACT

Authorship identification in handwritten documents is a crucial field of study in criminalistics and forensic analysis, given the necessity of determining the authorship of questioned documents. Furthermore, due to the variability of human handwriting and the similarity among different authors, this task becomes a challenging problem. This study aims to propose a tool that seeks to standardize and automate the identification of handwritten documents based on legal principles and the precepts of graphoscopy, in addition to employing pattern recognition and machine learning techniques to extract and analyze graphometric features of handwritten texts, supporting forensic work. The research is expected to contribute to the development of an accurate and reliable automatic manuscript identification system that can be used as an auxiliary tool in handwriting examinations, reducing analysis time and aiding in the identification of potential authors. Automating the process not only speeds up the analysis but also increases the consistency of results, minimizing the risk of human error. Thus, the tool could become a valuable resource for both criminal investigations and judicial processes, offering a solid and scientific basis for the authorship identification of handwritten documents.

Keywords: Forensic Computing, Handwritten Documents, Graphometric Features, Forensic Letters, Axial Slant, SVM, Random Forest.

LISTA DE FIGURAS

Figura - 3.1	Carta PUCPR CF00001_01.	19
Figura - 4.1	Carta PUCPR CF00001_01 Binarizada.	25
Figura - 4.2	Carta PUCPR CF00001_01 Dilatada.	26
Figura - 4.3	Carta PUCPR CF00001_01 Erodida.	26
Figura - 4.4	Carta PUCPR CF00001_01 Borda.	27
Figura - 4.5	Histogramas representando a inclinação axial nula, à esquerda e à direita, respectivamente.	29
Figura - 4.6	Início do arquivo CSV de treino.	30
Figura - 4.7	Início do arquivo CSV teste.	30
Figura - 4.8	Interface inicial da ferramenta.	34
Figura - 4.9	Visualização dos resultados pela interface.	35
Figura - 5.1	Protocolo dos experimentos.	38
Figura - 5.2	Comunicação entre <i>frontend</i> e <i>backend</i>	39
Figura - 5.3	Funcionamento do <i>backend</i>	40
Figura - 5.4	Matriz de confusão em uma análise de 60 autores.	41

LISTA DE TABELAS

Tabela - 5.1	Melhores resultados obtidos por cada modelo.	42
--------------	--	----

LISTA DE SIGLAS E ABREVIATURAS

API: *Application Programming Interface*

CSV: *Comma-Separated Values*

LOO: *Leave-One-Out*

PUCPR: Pontifícia Universidade Católica do Paraná

REST: *Representational State Transfer*

SVM: *Support Vector Machine*

SUMÁRIO

1	Introdução	11
1.1	Motivação	12
1.2	Objetivos	12
1.2.1	Objetivo Geral	12
1.2.2	Objetivos Específicos	13
1.3	Organização do trabalho	13
2	Fundamentação Teórica	14
2.1	A Escrita como Elemento Biométrico	14
2.1.1	A Fisiologia da Escrita e a Individualização do Gesto Gráfico	15
2.2	Características Grafométricas	16
2.2.1	Classificação das Características Grafométricas	16
2.2.2	A Importância das Características Grafométricas na Iden- tificação de Autoria	17
3	Materiais e métodos	18
3.1	Recursos computacionais	18
3.2	Base de Cartas Forenses Modelo PUCPR	18
3.3	Recursos metodológicos	20
4	Desenvolvimento	22
4.1	Preparação da base de dados	22
4.2	Refatoração de algoritmos	22
4.2.1	Criação da API	23
4.2.2	Processamento de imagens	24
4.2.3	Extração da <i>feature</i> Inclinação Axial	27
4.3	Criação e treinamento dos modelos	30
4.3.1	SVM (<i>Support Vector Machine</i>)	31
4.3.2	<i>Random Forest</i>	33
4.3.3	Avaliação dos modelos	33
4.4	Criação da interface	33
4.4.1	Integração com a API	36

5	Experimentos	37
5.1	Protocolo dos experimentos	37
5.2	Resultados	41
6	Análise e discussão dos resultados	43
7	Conclusão	45
	REFERÊNCIAS	46

Introdução

A identificação de autoria em documentos manuscritos é um desafio constante na área forense, sendo essencial para a elucidação de crimes, validação de assinaturas e resolução de disputas legais. A análise grafométrica, tradicionalmente realizada por peritos em documentoscopia, consiste na análise minuciosa de características da escrita, como a forma das letras, inclinação, pressão e espaçamento, buscando determinar a autoria de um documento questionado.

Entretanto, a demanda por perícias, a subjetividade inerente à análise manual e a busca por métodos mais ágeis e eficientes têm impulsionado a pesquisa e o desenvolvimento de soluções computacionais para automatizar este processo. Técnicas de reconhecimento de padrões, aprendizado de máquina e visão computacional têm se mostrado promissoras na análise de características grafométricas, abrangendo desde a extração automatizada de características até a classificação de autores.

Nesse cenário, este trabalho se propõe a desenvolver uma ferramenta computacional para auxiliar os peritos grafotécnicos na identificação da autoria de manuscritos. A ferramenta será desenvolvida utilizando a linguagem de programação *Python* e bibliotecas como *OpenCV* e *Scikit-learn*, que possibilitam a implementação de algoritmos para processamento de imagens, extração de características e classificação. Adicionalmente, uma interface *web* será desenvolvida para facilitar a utilização da ferramenta. Espera-se que a ferramenta desenvolvida neste trabalho contribua para a redução do tempo e esforço demandados na análise de manuscritos, a minimização da subjetividade inerente à análise humana e o aumento da precisão na identificação de autores.

1.1 Motivação

Como destacado anteriormente, a identificação de autoria em manuscritos, utilizando-se da análise de características grafométricas, é uma tarefa crucial em áreas como a criminalística e a ciência forense, desempenhando um papel fundamental na determinação da autoria de documentos questionados. Atualmente, essa análise é realizada por peritos em documentoscopia, com base em seus conhecimentos e experiência, o que, embora eficaz em diversos casos, apresenta desafios inerentes à própria natureza da análise manual.

A grande variabilidade da escrita humana, a similaridade entre diferentes autores e a subjetividade intrínseca ao processo de análise, especialmente quando lida com um grande volume de documentos, aumentam a complexidade da tarefa e podem impactar o tempo de resolução de casos, bem como a precisão dos resultados. Diante disso, a demanda por métodos automatizados, que auxiliem os peritos, tornando o processo mais ágil, preciso e menos suscetível a erros humanos, tem se tornado cada vez mais evidente.

Nesse contexto, este trabalho propõe o desenvolvimento de uma ferramenta computacional que utilize técnicas de aprendizado de máquina para automatizar a análise de características grafométricas, com o objetivo de auxiliar os peritos grafotécnicos na identificação de autoria de manuscritos. A ferramenta buscará extrair e analisar de forma objetiva e sistemática características como a inclinação axial, gerando resultados que possam ser utilizados como evidências em investigações criminais e processos judiciais.

Acredita-se que a implementação desta ferramenta, além de contribuir para a área de computação forense como um todo, trará benefícios diretos para o trabalho pericial, proporcionando uma maior agilidade na análise de documentos, redução do risco de erros humanos, aumento da precisão na identificação de autores e fortalecimento da credibilidade das provas documentais. A automatização deste processo, portanto, não apenas auxilia na resolução de casos de forma mais eficiente, mas também contribui para a construção de um sistema judicial mais justo e confiável.

1.2 Objetivos

1.2.1 Objetivo Geral

Este trabalho tem como objetivo geral desenvolver uma ferramenta para identificação de manuscritos com o uso de características grafométricas. Essa ferramenta deve servir de auxílio a peritos grafotécnicos durante o processo de identificação de autoria em manuscritos no âmbito jurídico.

1.2.2 Objetivos Específicos

- Refatorar algoritmos anteriores que implementam a extração de características grafométricas, com foco na inclinação axial, de forma modular, possibilitando assim a adição de novas características em estudos futuros.
- Elaborar a interface *web* para utilização da ferramenta.
- Realizar experimentos para avaliação da ferramenta desenvolvida.
- Analisar os resultados obtidos com os experimentos realizados.
- Contribuir com o trabalho dos peritos grafotécnicos, auxiliando na identificação de autoria de manuscritos de forma automatizada e sistemática.

1.3 Organização do trabalho

Este trabalho está estruturado da seguinte forma: O capítulo 1 define o problema da identificação de autoria em manuscritos, apresenta a motivação para este trabalho e descreve os objetivos. O capítulo 2 apresenta uma revisão de literatura sobre grafoscopia, escrita como elemento biométrico e características grafométricas. O capítulo 3 descreve os materiais e métodos utilizados no desenvolvimento da ferramenta, incluindo os recursos computacionais, a base de dados e a metodologia empregada. O capítulo 4 detalha o desenvolvimento da ferramenta, abrangendo o pré-processamento das imagens, a extração de características grafométricas, a criação, o treinamento e a avaliação dos modelos de aprendizado de máquina, e o desenvolvimento da interface *web*. O capítulo 5 apresenta os experimentos realizados para avaliar a performance da ferramenta e seus resultados. O capítulo 6 analisa e discute os resultados obtidos nos experimentos, comparando o desempenho dos diferentes modelos e avaliando a influência da quantidade de autores na acurácia. O capítulo 7 finaliza com as conclusões e propostas para trabalhos futuros.

Fundamentação Teórica

A análise da escrita manual tem sido um campo de estudo relevante em áreas como a psicologia, a pedagogia (Mendes, 2003), a linguística e, mais recentemente, na computação, especialmente no âmbito forense. Dentro deste contexto, a documentoscopia, ramo da criminalística dedicado à análise de documentos, utiliza características específicas da escrita para verificar a autenticidade e determinar a autoria de manuscritos. Essa prática tradicionalmente baseia-se na análise humana, realizada por peritos que examinam minuciosamente os documentos em busca de indícios que levem à identificação do autor. No entanto, a subjetividade inerente à análise humana e a demanda por métodos mais precisos e eficientes têm impulsionado a busca por soluções automatizadas.

Por isso, este trabalho se propõe a desenvolver uma ferramenta computacional que auxilie os peritos na identificação de autoria, utilizando como base a análise de características grafométricas. Para fundamentar esta proposta, exploraremos a escrita como um elemento biométrico e as particularidades das características grafométricas na individualização da escrita.

2.1 A Escrita como Elemento Biométrico

A escrita, um gesto complexo aprendido ao longo da vida, vai além da simples representação gráfica da linguagem. Cada indivíduo desenvolve um estilo próprio de escrita, influenciado por fatores fisiológicos, psicológicos, sociais e culturais, que se traduzem em nuances e padrões únicos, tornando a escrita uma verdadeira "impressão digital" gráfica. Essa singularidade da escrita humana a coloca como um elemento biométrico com alto potencial para a identificação de indivíduos.

2.1.1 A Fisiologia da Escrita e a Individualização do Gesto Gráfico

A escrita se inicia no cérebro, que comanda os movimentos coordenados dos dedos, mão, punho e braço, resultando em traços e formas sobre o papel. Compreender a fisiologia da escrita é fundamental para entender como esse processo complexo se traduz em características individualizadoras.

Saudek (1978) descreve sete princípios fisiológicos que regem a escrita e que contribuem para a individualidade do gesto gráfico:

- Princípio 01: a escrita, com a prática, se torna um gesto habitual, permitindo ao escritor experiente focar no conteúdo e não na mecânica da escrita. Cada indivíduo desenvolve seus próprios hábitos, que se refletem na forma, pressão e velocidade da escrita.
- Princípio 02: os músculos da mão trabalham de forma mais eficiente em ações rítmicas de contração e relaxamento. A fadiga muscular interfere nesse ritmo, impactando a fluidez e a consistência da escrita.
- Princípio 03: o relaxamento muscular é essencial para a escrita fluida e precisa. A fadiga dificulta o relaxamento, tornando os movimentos mais lentos e menos coordenados.
- Princípio 04: a perda da capacidade de relaxamento muscular pode gerar movimentos espasmódicos, afetando a pressão e a uniformidade do traçado.
- Princípio 05: a fadiga pode se estender além dos dedos, impactando a mão e o braço, comprometendo a escrita em diversos aspectos.
- Princípio 06: diante da fadiga ou desconforto, o escritor realiza ajustes na forma como segura a caneta, na postura ou na posição do papel, o que impacta a escrita.
- Princípio 07: a alternância entre tensão e relaxamento muscular durante a escrita é um processo natural. A interrupção desse ritmo, causada por fadiga ou outros fatores, pode ser um indicativo de escrita disfarçada ou forjada.

A combinação desses princípios, atuando de forma singular em cada indivíduo, contribui para a formação de um estilo de escrita único, que pode ser utilizado como base para a identificação de autoria.

2.2 Características Grafométricas

As características grafométricas são elementos mensuráveis da escrita que, em conjunto, revelam os padrões e as nuances que tornam a escrita de cada pessoa única.

2.2.1 Classificação das Características Grafométricas

As características grafométricas podem ser classificadas de acordo com o nível de detalhe que abarcam:

- **Globais:** são características extraídas do documento como um todo, de parágrafos ou de linhas. Abrangem aspectos como:
 - Inclinação axial: representa o ângulo médio de inclinação da escrita em relação a uma linha de base horizontal.
 - Espaçamento entre linhas: mede a distância média entre as linhas de texto.
 - Tamanho das margens: avalia a largura das margens superior, inferior, esquerda e direita do documento.
 - Alinhamento do texto: analisa se a escrita se mantém alinhada a uma linha de base horizontal, real ou imaginária.
 - Distribuição do texto: observa como o texto se distribui no espaço gráfico, se ocupa todo o espaço disponível ou se concentra em determinadas áreas.
- **Locais:** são características extraídas de palavras ou caracteres individualmente. Incluem aspectos como:
 - Forma das letras: analisa as formas específicas das letras, como a presença de laços, ornamentos, simplificações ou distorções.
 - Laços ascendentes e descendentes: observa a forma, o tamanho e a inclinação dos laços presentes em letras como "l", "h", "g" e "f".
 - Corte da letra "t": Analisa a forma, a posição e a direção do traço que corta a letra "t".
 - Pressão: avalia a força exercida sobre a caneta, observada na largura e intensidade do traço.
 - Relações de tamanho: compara o tamanho de letras ou segmentos de letras dentro da mesma palavra.

- Ligações entre letras: analisa a forma e a fluidez das ligações entre as letras dentro das palavras.
- Mínimos gráficos: examina detalhes como pontos, vírgulas, acentos e cedilhas, que podem revelar hábitos específicos do escritor.

A combinação de características globais e locais proporciona uma análise mais completa e precisa da escrita, aumentando a confiabilidade da identificação de autoria.

2.2.2 A Importância das Características Grafométricas na Identificação de Autoria

As características grafométricas são elementos chave na identificação de autoria em documentos manuscritos, tanto na análise manual realizada por peritos quanto em abordagens computacionais. A escolha por características grafométricas se justifica por sua sólida fundamentação científica, baseada em princípios e leis da escrita estabelecidos pela grafoscopia, uma área de estudo consolidada e reconhecida pela comunidade científica. Além disso, a interpretabilidade dessas características, por serem intuitivas e facilmente compreendidas por peritos e juízes, facilita a argumentação em processos judiciais. As características grafométricas também demonstram robustez, sendo muitas vezes resistentes a tentativas de disfarce, pois refletem hábitos profundamente enraizados na escrita do indivíduo. Sua aplicabilidade a documentos existentes, permitindo a análise de manuscritos já produzidos, torna essa abordagem útil em investigações criminais e processos judiciais.

No entanto, a extração e a análise automatizada de características grafométricas apresentam desafios, como a complexidade algorítmica, que exige o desenvolvimento de algoritmos robustos para extrair essas características de imagens de manuscritos, demandando conhecimento em processamento de imagens, visão computacional e inteligência artificial. Outro desafio reside na variação intrapessoal, pois a escrita de um mesmo indivíduo pode variar dependendo de fatores como o estado emocional, a fadiga, a velocidade de escrita e a intenção de disfarce.

Este trabalho busca superar alguns desses desafios, utilizando algoritmos eficientes e robustos para extrair e analisar um conjunto de características grafométricas selecionadas.

Materiais e métodos

Neste capítulo, estão descritos os recursos computacionais, a base de dados e os recursos metodológicos que serão utilizados no desenvolvimento deste trabalho.

3.1 Recursos computacionais

- **Hardware:** O *laptop* utilizado para este trabalho conta com um processador Intel Core i5 12450H, placa de vídeo Nvidia RTX 3050 4GB VRAM, 16GB de memória RAM e WSL com sistema operacional Ubuntu 22.04.
- **Software:** Para a implementação da ferramenta será utilizado o *Visual Studio Code* como editor de texto. Para o desenvolvimento do protótipo inicial será utilizada a ferramenta de prototipagem *Figma*. Para o desenvolvimento do *backend* será utilizada a linguagem de programação *Python* com as bibliotecas *Scikit-learn* para treinamento dos modelos utilizando SVM (*Support Vector Machine*) e *Random Forest* e a biblioteca *OpenCV* para utilização de visão computacional. E para o desenvolvimento do *frontend* será utilizada a linguagem de programação *TypeScript*, com a biblioteca *ReactJS* para construção da interface.

3.2 Base de Cartas Forenses Modelo PUCPR

A base de dados utilizada para a realização dos experimentos de avaliação da ferramenta desenvolvida foi a Base de Cartas Forenses Brasileira, também conhecida como Cartas Forenses PUCPR (Freitas et al., 2008). Esta base foi desenvolvida considerando as

particularidades da língua portuguesa, como a presença de acentos (á, à, ã, ê, ü) e do símbolo especial (ç), o que a torna ideal para este estudo, visto que o foco principal é a análise de manuscritos em português.

A base PUCPR contém 1800 exemplares de cartas, escritas por 600 autores distintos. Cada autor contribuiu com três cartas manuscritas, redigidas em folha A4 não pautada. Os documentos foram digitalizados em 300 dpi e estão disponíveis em escala de cinza (256 níveis de cinza). A Figura - 3.1 apresenta um exemplo de carta presente na base PUCPR.

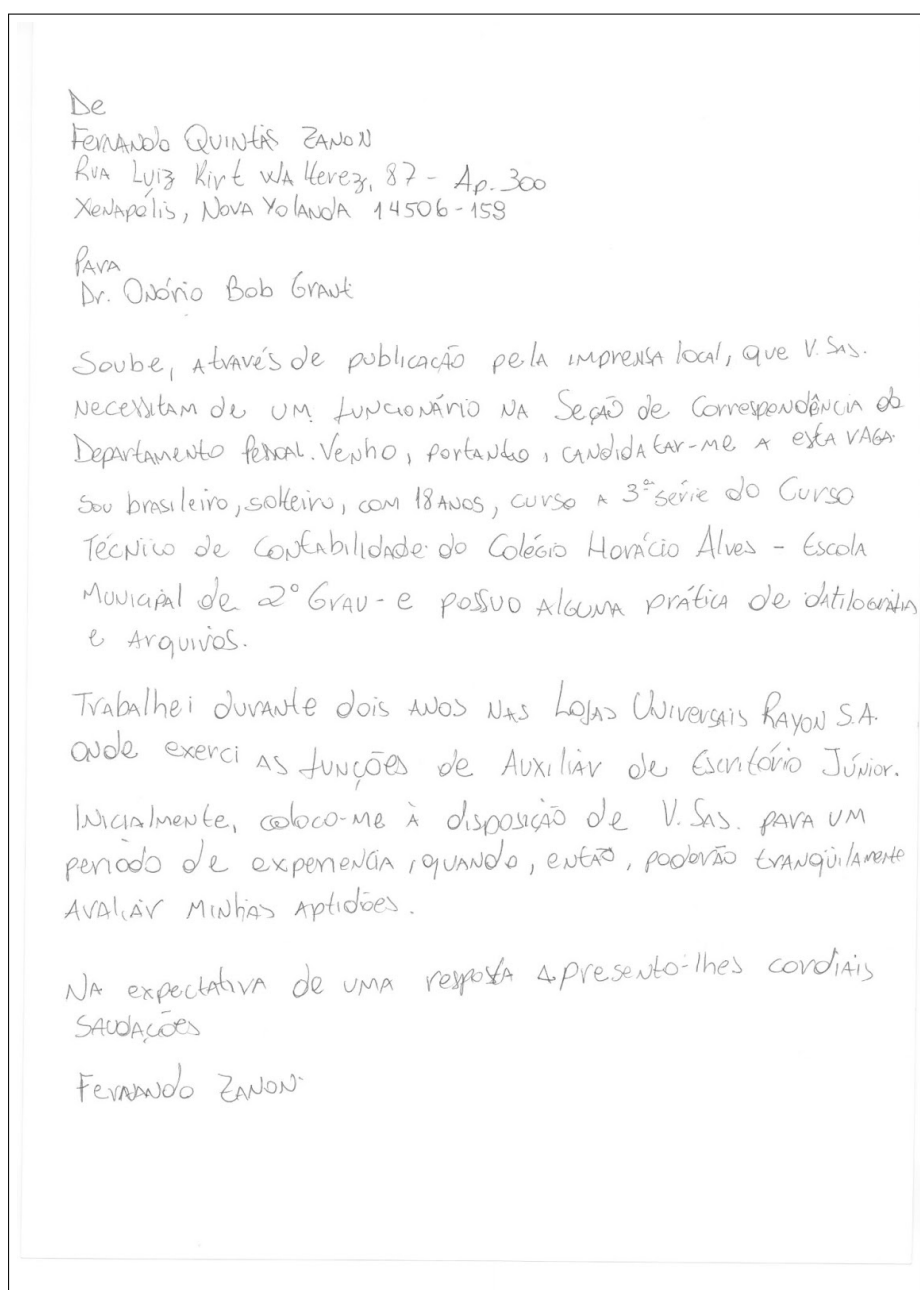


Figura 3.1: Carta PUCPR CF00001.01.

3.3 Recursos metodológicos

- **Revisão sistemática da literatura:** Uma busca abrangente foi realizada em bases de dados acadêmicas, como *IEEE Xplore*, para identificar estudos anteriores relevantes sobre reconhecimento de autoria de manuscritos, com foco em técnicas de aprendizado de máquina, grafoscopia, processamento de imagens e análise de escrita.
- **Prototipação da interface:** A interface da ferramenta foi inicialmente prototipada utilizando a ferramenta *Figma*. A prototipagem permitiu a criação de um modelo visual interativo da interface, auxiliando na avaliação de usabilidade e na comunicação com os usuários finais. Foram explorados diferentes *layouts*, elementos de interação e fluxos de uso da ferramenta.
- **Desenvolvimento da interface:** Após a validação da prototipagem, a interface da ferramenta foi desenvolvida utilizando a linguagem de programação *TypeScript* e a biblioteca *ReactJS*. A ferramenta é uma aplicação *web*, acessível através de navegadores e a interface foi desenvolvida com foco na usabilidade, buscando oferecer uma experiência intuitiva e eficiente aos usuários.
- **Refatoração de algoritmos:** Os algoritmos anteriores para a extração de características grafométricas, desenvolvidos por Amaral (2014) e Baranoski (2005), serão refatorados em *Python* utilizando a biblioteca *OpenCV*, visando uma implementação mais modular e eficiente. Esse processo envolveu a análise do código existente, melhorias estruturais e a implementação de testes para garantir a funcionalidade. Além disso, a refatoração substituiu o uso da ferramenta WEKA pela biblioteca *Scikit-learn* para a classificação SVM e *Random Forest*.
- **Preparação dos dados:** A base de dados de manuscritos PUCPR, contendo amostras de escrita de 600 autores, foi utilizada para treinar e avaliar a ferramenta. As imagens dos manuscritos são submetidas a um processo de pré-processamento, que inclui binarização, extração de contornos e segmentação, utilizando a biblioteca *OpenCV*. Este processo visa preparar as imagens para a extração das características grafométricas.
- **Extração de característica:** A característica grafométrica extraída das imagens pré-processadas foi implementada em *Python*. Além disso, a característica foi normalizada para garantir a compatibilidade entre os dados e o classificador.

- **Treinamento e avaliação dos modelos:** Os algoritmos *Support Vector Machine* (SVM) e *Random Forest* foram utilizados para treinar os modelos de classificação, utilizando as características grafométricas extraídas dos manuscritos. A biblioteca *Scikit-learn* foi utilizada para a implementação do SVM e *Random Forest*. Diferentes configurações de parâmetros do SVM foram avaliadas para determinar a melhor performance do modelo. A base de dados foi dividida em conjuntos de treinamento e teste, utilizando técnicas de validação cruzada, para avaliar a capacidade de generalização do modelo.

Desenvolvimento

Este capítulo descreve as etapas para o desenvolvimento da ferramenta para identificação de autoria em manuscritos, utilizando a característica grafométrica inclinação axial. A ferramenta é composta por dois módulos principais: *backend*, responsável pelo processamento e análise dos dados, e *frontend*, responsável pela interface com o usuário.

4.1 Preparação da base de dados

Para garantir a generalização do modelo de identificação de autoria e avaliar seu desempenho em dados não utilizados durante o treinamento, a base de dados foi dividida em dois conjuntos:

- **Conjunto de Treinamento:** Composto por duas cartas de cada autor, totalizando 1200 exemplares.
- **Conjunto de Teste:** Composto por uma carta de cada autor, totalizando 600 exemplares.

Essa divisão visa simular um cenário real, onde o modelo é treinado com dados de autores conhecidos e posteriormente avaliado em documentos de autoria desconhecida.

4.2 Refatoração de algoritmos

O desenvolvimento da ferramenta se baseou em pesquisas anteriores sobre identificação de autoria em manuscritos, como os trabalhos de Baranoski (2005), Amaral (2014),

Batista e Alves (2024), com foco na otimização da extração e análise da característica inclinação axial. A refatoração dos algoritmos existentes buscou garantir modernização, modularização e eficiência.

A ferramenta é composta por dois módulos separados:

- **Backend:** Responsável por todas as etapas de processamento e análise dos dados, incluindo:
 - Pré-processamento das imagens.
 - Extração da característica grafométrica.
 - Treinamento e teste dos modelos de identificação de autoria (SVM e *Random Forest*).
 - Envio dos resultados para o *frontend*.
- **Frontend:** Responsável pela interface com o usuário, com as seguintes funcionalidades:
 - Configuração dos parâmetros para o treinamento do modelo SVM e *Random Forest*.
 - Visualização dos resultados da análise de autoria.

4.2.1 Criação da API

A comunicação entre o *frontend* e o *backend* da ferramenta é realizada por meio de uma API REST, desenvolvida em *Python* utilizando a biblioteca FastAPI. Essa escolha se justifica pela performance e facilidade de desenvolvimento que a biblioteca oferece. A API atua como um intermediário, recebendo requisições do *frontend*, processando-as e retornando os resultados.

A estrutura da API é composta por uma rota principal, responsável por receber os parâmetros de configuração do modelo de identificação de autoria, como, por exemplo, a quantidade de classes (autores) que serão utilizadas no treinamento. Após receber a requisição com os parâmetros, a API aciona os módulos responsáveis pelas seguintes etapas:

1. **Processamento das imagens da base de dados:** O módulo de pré-processamento é acionado, preparando as imagens para a extração das características grafométricas.

2. **Criação e classificação dos modelos:** Com as imagens pré-processadas, o módulo de extração de características é executado, gerando os dados para o treinamento dos modelos SVM e *Random Forest*. Em seguida, cada modelo é treinado e testado com os dados fornecidos.

Finalmente, a API retorna os resultados do treinamento para o *frontend*. Esses resultados incluem métricas de desempenho do modelo, como a acurácia, permitindo ao usuário avaliar a qualidade do modelo gerado.

A utilização de uma API REST proporciona diversas vantagens para a ferramenta:

- **Modularidade:** Permite a separação clara entre *frontend* e *backend*, facilitando o desenvolvimento, a manutenção e a escalabilidade da ferramenta.
- **Flexibilidade:** A API pode ser acessada por diferentes tipos de clientes, como aplicações *web*, *mobile* e *desktop*, facilitando a implementação de aplicações futuras.
- **Padronização:** A utilização do padrão REST garante a interoperabilidade entre diferentes sistemas e facilita a integração com outras ferramentas.
- **Eficiência:** FastAPI é conhecida por sua alta performance, o que garante a rapidez na comunicação entre o cliente e o servidor.

A API, portanto, é um componente crucial para o funcionamento da ferramenta, viabilizando a comunicação eficiente e a troca de informações entre o *frontend* e o *backend*, garantindo a usabilidade e a robustez da ferramenta como um todo.

O código desenvolvido para a implementação da API, bem como outros módulos da ferramenta, está disponível publicamente em: <https://github.com/yuripiresalves/manuscriptus>.

4.2.2 Processamento de imagens

Pré-processamento

O pré-processamento visa uniformizar as imagens e remover informações irrelevantes, realçando elementos importantes para as etapas seguintes, como a segmentação e extração de características.

Primeiro, é realizada a binarização da imagem, Figura - 4.1, um processo crucial que simplifica a representação visual, convertendo uma imagem em escala de cinza de 256 níveis para uma imagem binária, composta apenas por *pixels* pretos e brancos. A principal vantagem da binarização é a redução da quantidade de dados a serem processados,

facilitando a extração de características relevantes da imagem e eliminando ruídos que podem prejudicar a análise. A técnica de binarização utilizada foi a de Otsu (Otsu, 1979) pois se trata de um método global e automatizado que calcula um limiar ideal para separar os *pixels* da escrita do fundo da imagem, minimizando a necessidade de ajustes manuais e garantindo a consistência na aplicação do método em todo o conjunto de dados.

A próxima etapa é a detecção de bordas por dilatação e erosão. A utilização de morfologia matemática na detecção de bordas por dilatação e erosão é um processo que faz extração de contornos bem definidos, usados posteriormente para a extração da inclinação axial (Baranoski, 2005). A dilatação modifica a imagem de um manuscrito deixando o traçado do autor mais espesso, Figura - 4.2. Já a erosão faz o processo inverso, deixa o traçado do autor mais fino, Figura - 4.3. Após a geração da imagem dilatada e imagem erodida, estas são sobrepostas e é feita a subtração dos *pixels* compatíveis nas duas imagens resultando na imagem de borda, Figura - 4.4.

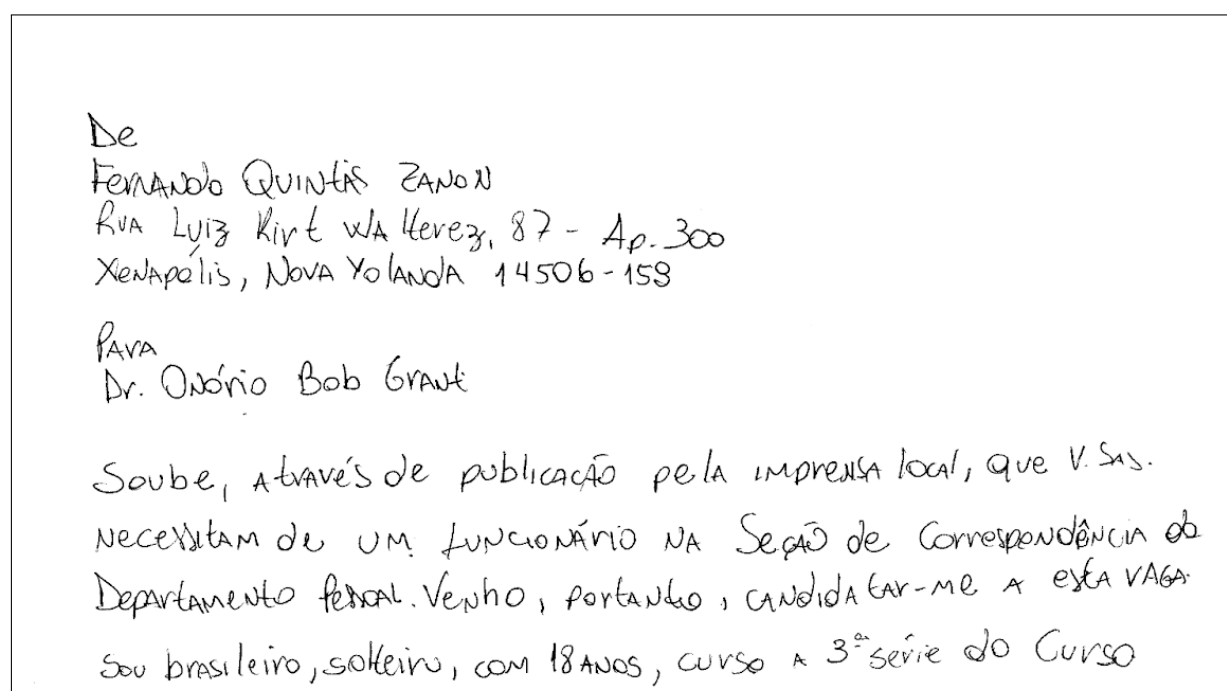


Figura 4.1: Carta PUCPR CF00001.01 Binarizada.

De
Fernando Quintis Zanon
RVA Lutz Kirt wla Kerez, 87 - Ap. 300
Xenapolis, Nova Yolanda 14506-158

Para
Dr. Onório Bob Grant

Soube, através de publicação pela imprensa local, que V. Ss. necessitam de um funcionário na Seção de Correspondência do Departamento Federal. Venho, portanto, candidatar-me a esta vaga sou brasileiro, solteiro, com 18 anos, curso a 3ª série do Curso

Figura 4.2: Carta PUCPR CF00001_01 Dilatada.

Figura 4.3: Carta PUCPR CF00001_01 Erodida.

De
 Fernando Quintis Zanon
 Rua Luiz Kyré via Herez, 87 - Ap. 300
 Xanxalís, Nova York 14506-159

Para
 Dr. Osório Bob Grant

Soube, através de publicação pela imprensa local, que V. Sas.
 necessitam de um funcionário na Seção de Correspondência do
 Departamento Fedat. Venho, portanto, candidatar-me a esta vaga.
 Sou brasileiro, solteiro, com 18 anos, curso a 3ª série do Curso

Figura 4.4: Carta PUCPR CF00001_01 Borda.

4.2.3 Extração da feature Inclinação Axial

Conforme apresentado no capítulo 2, a inclinação axial representa o ângulo médio de inclinação da escrita. Esta característica grafométrica é amplamente utilizada por peritos em suas análises e possui um importante poder discriminatório no processo de identificação de autoria (Amaral, 2014). Por essa razão, foi escolhida para ser implementada neste trabalho.

Para este trabalho, a abordagem escolhida para extração desta *feature* foi a técnica da distribuição de borda direcional. Segundo Baranoski (2005), nesta técnica, consideram-se as bordas, pois obtiveram melhores resultados na discriminação da inclinação axial do autor em relação à inclinação extraída diretamente do traçado e por serem mais finas, reduzem a influência da espessura do traçado sobre o cálculo.

Distribuição de borda direcional

A imagem pré-processada é representada por uma imagem de borda na qual apenas os *pixels* desta borda estão em preto. A imagem é então percorrida considerando-se o *pixel* da borda do traçado no centro do elemento estruturante retangular. Em seguida, são verificados os fragmentos de borda em todas as direções, partindo deste *pixel* central e conferindo os *pixels* posteriores com um operador lógico AND, finalizando nas extremi-

dades do elemento estruturante apenas se houver a presença de um fragmento de borda inteiro. Ou seja, se todos os *pixels* vizinhos forem pretos, considera-se o fragmento de borda e computa-se a posição do fragmento em um vetor de posições para a construção do histograma, Figura - 4.5.

Este vetor é normalizado para garantir que todas as características tenham a mesma escala e evitando que características com magnitudes maiores dominem o processo de aprendizado do modelo. Foi utilizada a técnica de normalização *min-max*, que escala os valores do vetor de características para um intervalo específico, geralmente $[0, 1]$. Essa técnica é aplicada individualmente a cada vetor de 17 posições, correspondente a um fragmento da escrita. A normalização *min-max* calcula, para cada elemento do vetor, a diferença entre o valor original e o valor mínimo do vetor, dividindo esse resultado pela diferença entre o valor máximo e o valor mínimo do vetor. Após a normalização, o vetor resultante representa a distribuição da inclinação axial do manuscrito, com valores próximos a 1 indicando uma maior frequência de *pixels* naquela direção (e, portanto, uma maior probabilidade de inclinação) e valores próximos a 0 indicando uma menor frequência.

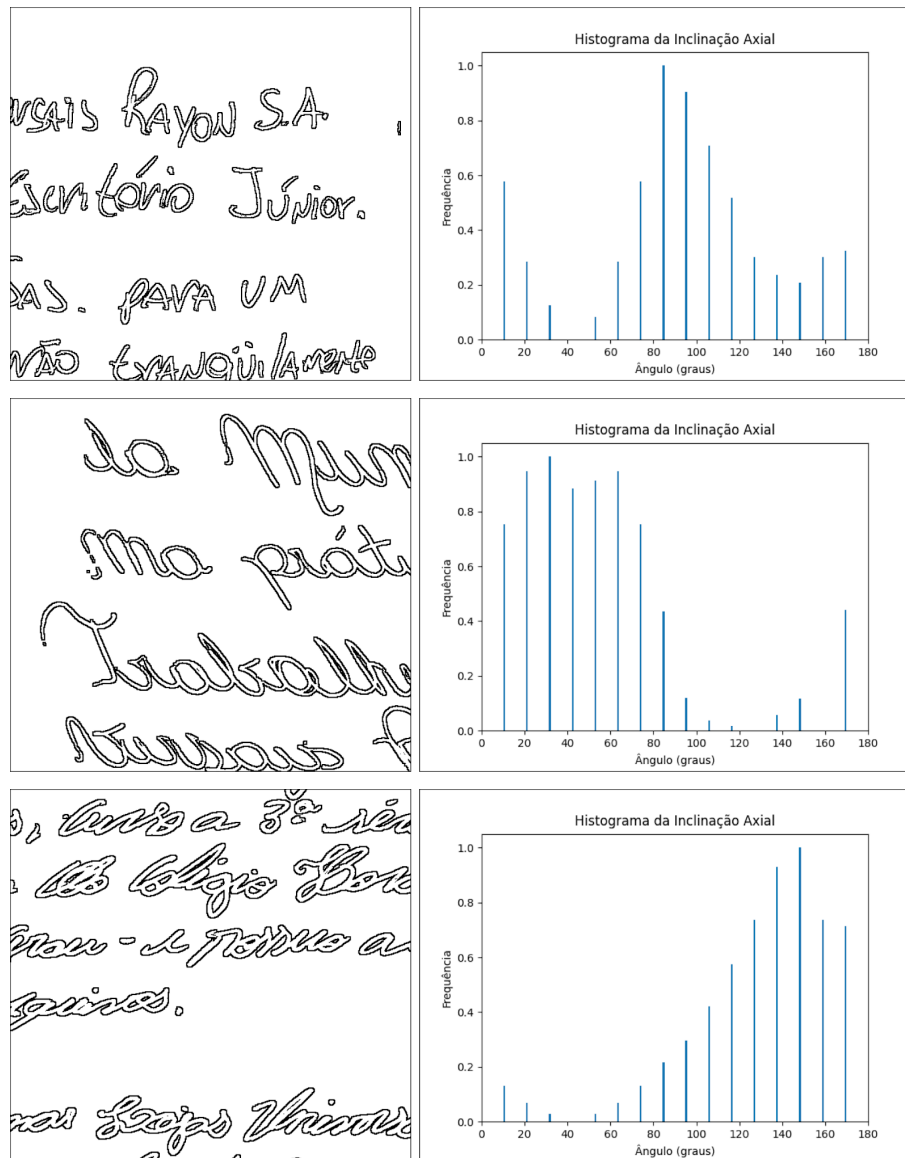


Figura 4.5: Histogramas representando a inclinação axial nula, à esquerda e à direita, respectivamente.

Preparação dos dados para os modelos de classificação

Após a extração da *feature* inclinação axial e a normalização dos vetores de distribuição de borda direcional, os dados estão prontos para serem utilizados pelos algoritmos de aprendizado de máquina. Para isso, são gerados dois arquivos CSV: um para o conjunto de treino, Figura - 4.6, e outro para o conjunto de teste, Figura - 4.7.

Cada linha dos arquivos CSV representa um manuscrito, e cada coluna, um valor da *feature* inclinação axial para aquele manuscrito. No caso, como o vetor de distribuição

possui 17 posições, cada linha do CSV terá 17 colunas representando esses valores. Uma coluna a mais é adicionada para identificar a classe (autor) à qual o manuscrito pertence.

A separação dos dados em conjuntos de treino e teste, como descrito na seção 4.1, é crucial para avaliar a capacidade de generalização dos modelos. O conjunto de treino será utilizado para ajustar os parâmetros dos algoritmos, enquanto o conjunto de teste servirá para avaliar sua performance em dados desconhecidos. Esses arquivos CSV serão, então, utilizados na próxima etapa para alimentar os modelos de aprendizado de máquina e dar início ao processo de treinamento e avaliação.

autor	inclinacao_0	inclinacao_1	inclinacao_2	inclinacao_3
a001	0.3658632150674197	0.5968195206312319	0.28402734617775016	0.12490880103764153
a001	0.32954545454545453	0.6061847988077497	0.28808896022010777	0.12669093201880088
a002	1.0	0.23980191265459783	0.09890470836438976	0.03759616963755574
a002	1.0	0.25756237032036455	0.11871457077902094	0.04636202086787321
a003	0.4480336903140347	0.5429564000349899	0.2454669282581258	0.1015583019881784
a003	0.4713869765163678	0.5331981822634629	0.23192467822253668	0.09666929712373126
a004	0.47907760324864396	0.4129069325828472	0.18490782849185405	0.07255889140998549
a004	0.4782795698924731	0.4435330261136713	0.19481822836661547	0.07524833589349718

Figura 4.6: Início do arquivo CSV de treino.

autor	inclinacao_0	inclinacao_1	inclinacao_2	inclinacao_3
a001	0.34759691057505765	0.5894188415876618	0.2771697353490245	0.1226740851910148
a002	1.0	0.26092990094976193	0.12023527614391567	0.048251979731622235
a003	0.43458008811017557	0.5346444666052336	0.23537282722579178	0.0970600487256845
a004	0.5229634854680318	0.4491959749148582	0.20063704797698692	0.07717318226405528

Figura 4.7: Início do arquivo CSV teste.

4.3 Criação e treinamento dos modelos

Após a extração e organização dos dados referentes à inclinação axial em arquivos CSV, a próxima etapa envolve a construção e o treinamento de modelos de aprendizado de máquina. O objetivo principal é capacitar a ferramenta a identificar padrões nos dados, permitindo a identificação automática da autoria. Para alcançar este objetivo, a ferramenta oferece a flexibilidade de utilizar dois algoritmos, o SVM e o *Random Forest*. Além disso, é possível escolher a quantidade de autores que será utilizada, sendo esses autores escolhidos de forma aleatória.

A escolha de qual(is) modelo(s) utilizar fica a cargo do usuário, que pode optar por utilizar apenas o SVM, apenas o *Random Forest* ou utilizar ambos os modelos, o

que permite comparar o desempenho de ambos os algoritmos no conjunto de dados, identificando qual deles apresenta melhor performance para a tarefa de identificação de autoria. Essa flexibilidade permite explorar diferentes abordagens e escolher a mais adequada para cada contexto e objetivo de análise. Adicionalmente, em trabalhos futuros, a ferramenta poderá ser expandida para incorporar outros modelos de aprendizado de máquina, enriquecendo ainda mais as opções de análise e adaptando-se às demandas específicas de cada pesquisa.

4.3.1 SVM (Support Vector Machine)

O algoritmo SVM é um método de aprendizado supervisionado bastante popular, utilizado para classificação e regressão. No contexto da identificação de autoria em manuscritos, o SVM atua classificando manuscritos em diferentes classes, onde cada classe representa um autor.

A ideia central do SVM é encontrar o hiperplano ótimo que melhor separa as classes em um espaço multidimensional. Cada dimensão deste espaço corresponde a um valor da *feature* extraída, e cada manuscrito é representado como um ponto neste espaço. O hiperplano ótimo é aquele que maximiza a margem, ou seja, a distância entre o hiperplano e os pontos de dados mais próximos de cada classe. Essa margem maximizada contribui para uma maior robustez do modelo, tornando-o menos suscetível a erros de classificação quando apresentado a dados não vistos durante o treinamento.

Para utilizar o SVM, a biblioteca *Scikit-learn* oferece uma implementação robusta e eficiente. A criação do modelo se inicia com a instanciação da classe `SVC`, definindo o tipo de *kernel* desejado como argumento.

Com o modelo instanciado, o treinamento é realizado através do método *fit*, utilizando como entrada a matriz de *features* de treinamento (X_{train}) e o vetor de rótulos correspondentes (y_{train}). A matriz X_{train} é estruturada de forma que cada linha represente um manuscrito e cada coluna um valor da *feature*. Já o vetor y_{train} indica o autor associado a cada manuscrito em X_{train} .

SVM com Grid Search

Visando não apenas construir um modelo funcional, mas também alcançar o desempenho ótimo, a técnica de *Grid Search* foi aplicada ao SVM. Este método consiste em uma busca exaustiva dentro de um espaço predefinido de hiperparâmetros, testando diferentes combinações em busca daquela que maximiza a performance do modelo. Para garantir uma avaliação robusta e utilizar ao máximo os dados de treinamento, especialmente em

cenários com um número limitado de exemplos, optou-se pela técnica de validação cruzada *Leave-One-Out* (LOO).

Os hiperparâmetros, por sua vez, atuam como reguladores do processo de aprendizado, influenciando diretamente a capacidade de generalização do modelo. Para o SVM, os hiperparâmetros otimizados via *Grid Search* foram:

- **kernel:** Define a função matemática responsável por projetar os dados em um espaço de *features* de maior dimensionalidade, com o objetivo de facilitar a separação das classes. Os *kernels* avaliados foram: '*linear*', '*poly*' e '*rbf*', cada um com suas características e aptidões para diferentes conjuntos de dados.
- **C:** Parâmetro de regularização que atua como um contrapeso entre maximizar a margem entre as classes e minimizar o erro de classificação. Valores altos de C dão mais ênfase à classificação correta dos exemplos de treinamento, enquanto valores baixos favorecem uma maior margem, buscando uma melhor capacidade de generalização.
- **gamma:** Este hiperparâmetro controla a influência de um único exemplo de treinamento na decisão do modelo. Valores baixos de *gamma* resultam em áreas de decisão mais amplas e suaves, enquanto valores altos criam áreas de decisão mais complexas e focadas em torno dos exemplos de treinamento.
- **degree:** Grau do polinômio para o *kernel* '*poly*'.

A implementação do Grid Search com o SVM segue um processo simples e estruturado. Primeiro, cria-se uma lista com os hiperparâmetros que serão testados e seus possíveis valores. Depois, um objeto *GridSearchCV* é gerado, onde informamos o modelo base, a lista de hiperparâmetros e o método de validação cruzada.

A busca pelos melhores parâmetros é realizada com os dados de treinamento. No final, o sistema identifica a combinação de parâmetros que oferece o melhor desempenho e ajusta o modelo com essas configurações.

Essa estratégia garante uma exploração completa do espaço de hiperparâmetros definido e a seleção do modelo com a melhor performance de generalização, mesmo com um conjunto de dados limitado. Além disso, a utilização do *Leave-One-Out* como método de validação cruzada contribui para uma estimativa mais precisa do desempenho do modelo em dados não vistos, tornando a escolha do modelo final mais robusta e confiável.

4.3.2 Random Forest

Assim como o SVM, o algoritmo *Random Forest* também foi utilizado na tarefa de identificação de autoria. Este algoritmo constrói uma floresta composta por várias árvores de decisão, onde cada árvore é treinada com uma amostra aleatória dos dados de treinamento.

Essa estratégia de aprendizado, é eficaz para evitar o problema de *overfitting*, quando o modelo se ajusta excessivamente aos dados de treinamento, comprometendo sua capacidade de generalização. Além disso, promove maior robustez ao lidar com dados não vistos anteriormente.

A implementação do *Random Forest* utilizando a biblioteca *Scikit-learn* se dá pela instanciamento da classe *RandomForestClassifier*. O número de árvores foi definido como 100, por meio do parâmetro *n_estimators*, e a reprodutibilidade foi garantida através do parâmetro *random_state*.

O treinamento do modelo *Random Forest* segue o mesmo padrão do SVM, utilizando o método *fit* com a matriz da *feature* de treinamento (*X_train*) e o vetor de rótulos (*y_train*) como entrada.

4.3.3 Avaliação dos modelos


Após o treinamento, os modelos SVM, SVM com *Grid Search* e *Random Forest* foram avaliados utilizando um conjunto de teste, separado previamente dos dados de treinamento. Essa etapa é essencial para verificar a capacidade de generalização de cada modelo, ou seja, sua habilidade em classificar corretamente dados que o modelo ainda não viu.

Neste trabalho, a métrica utilizada para avaliação de desempenho foi a acurácia, que representa a porcentagem de acertos do modelo. Os resultados de acurácia obtidos para os três modelos fornecem uma base sólida para a comparação de desempenho, e os detalhes dessa análise serão apresentados no próximo capítulo.

4.4 Criação da interface

Para a visualização e utilização da ferramenta de forma intuitiva e simples, foi desenvolvida uma interface web utilizando a biblioteca *ReactJS*, juntamente com o *framework Next.js*, a linguagem de programação *TypeScript* e os componentes visuais *shadcn*. Essa escolha se justifica pela versatilidade e amplo suporte da comunidade que essas tecnologias oferecem, permitindo a criação de uma interface moderna, responsiva e de fácil manutenção.

A interface tem como principal função enviar as configurações para o *backend*, que incluem o número de autores desejado e os modelos que serão utilizados para análise (SVM, SVM com *Grid Search* e/ou *Random Forest*). Após o processamento pelo *backend*, os resultados, como acurácia de cada modelo e os melhores parâmetros utilizados no *Grid Search* do SVM (caso o usuário tenha selecionado este modelo), são exibidos na interface de forma clara e concisa. A Figura - 4.8 ilustra a interface em seu estado inicial, apresentando os campos para configuração dos parâmetros de análise. Já a Figura - 4.9 demonstra a interface exibindo os resultados obtidos após o processamento de uma requisição, incluindo a acurácia dos modelos e os melhores parâmetros encontrados pelo *Grid Search*.



The image shows a web interface titled "Manuscritus" with the subtitle "Ferramenta baseada em características grafométricas para a identificação de manuscritos." Below the subtitle, there is a paragraph of instructions: "Ajuste o número de autores e selecione os modelos específicos para análise. Após definir as configurações, clique em **Obter resultados** para visualizar a acurácia de cada modelo." The interface includes a section for "Número de Autores" with a text input field containing "20" and a slider control. Below this, there is a section for "Modelos" with two checkboxes: "SVM" and "Random Forest". At the bottom of the form is a large black button labeled "Obter resultados".

Figura 4.8: Interface inicial da ferramenta.

Manuscritus
Ferramenta baseada em características grafométricas para a identificação de manuscritos.

Ajuste o número de autores e selecione os modelos específicos para análise. Após definir as configurações, clique em **Obter resultados** para visualizar a acurácia de cada modelo.

Número de Autores
Escolha um valor entre 20 e 200.

60

Modelos

- ☒ SVM
- ☒ Random Forest

Obter resultados

Acurácia:

Modelo	Acurácia
SVM	78.33%
SVM (Grid Search)	81.67%
Random Forest	78.33%

Melhores parâmetros (SVM):

Parâmetro	Valor
C	100
kernel	linear

Figura 4.9: Visualização dos resultados pela interface.

O código desenvolvido para a implementação da interface da ferramenta, está disponível publicamente em: <https://github.com/yuripiresalves/manuscritus>. Além disso, a ferramenta foi disponibilizada publicamente por meio de um *deploy online*, acessível pelo *link*: <https://manuscritus.vercel.app>.

4.4.1 Integração com a API

A comunicação entre o *frontend* (interface *web*) e o *backend* (lógica de processamento) é realizada por meio de uma API REST, desenvolvida utilizando a biblioteca FastAPI em *Python*. Essa arquitetura permite uma separação clara entre as camadas da aplicação, facilitando o desenvolvimento, manutenção e escalabilidade. A escolha do FastAPI se deve à sua alta performance, facilidade de uso e integração com o *framework web* utilizado no *backend*.

No *frontend*, a função *getResultsFromApi* é responsável por enviar as configurações selecionadas pelo usuário para o *endpoint* `/results` da API. Essa função utiliza o método *fetch* para realizar uma requisição POST, enviando os dados no corpo da requisição em formato JSON. A API, por sua vez, recebe esses dados, processa a requisição e retorna os resultados da análise também em formato JSON. A resposta da API é então tratada no *frontend*, onde os resultados são exibidos na interface. Essa integração entre *frontend* e *backend*, mediada pela API REST, garante uma comunicação eficiente e segura, permitindo que a ferramenta funcione de forma robusta e escalável. Além disso, a estrutura modular facilita a adição de novas funcionalidades e modelos de aprendizado de máquina no futuro.

Experimentos

Para avaliar os resultados alcançados com a ferramenta desenvolvidas neste trabalho foram conduzidos experimentos. Esses experimentos seguiram um protocolo específico, assegurando sua replicabilidade em diferentes cenários. Vários fatores são conhecidos por impactarem os resultados de pesquisas desse tipo, como o número de escritores, a abordagem de classificação escolhida e a base de dados empregada para o treinamento e teste. A seguir, é descrito o protocolo utilizado nos experimentos realizados.

5.1 Protocolo dos experimentos

Inicialmente, para cada autor presente na base de dados PUCPR, duas cartas foram alocadas ao conjunto de treinamento e uma carta ao conjunto de teste. O pré-processamento e a extração da inclinação axial foram realizados em cada imagem, e os vetores de características resultantes foram salvos nos arquivos CSV de treino e teste, como demonstrado na Figura - 5.1.

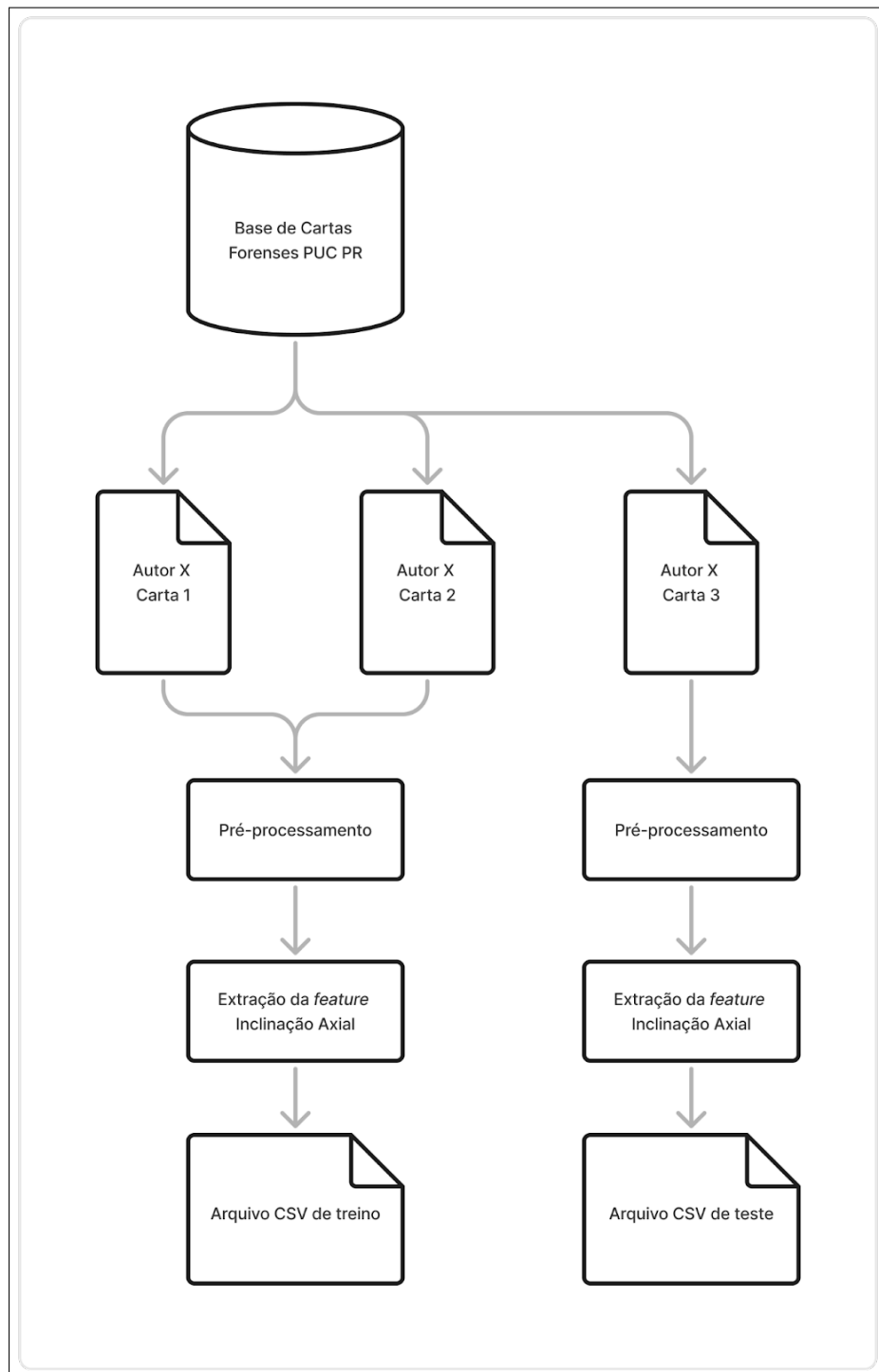


Figura 5.1: Protocolo dos experimentos.

A comunicação entre o *frontend* e o *backend* da ferramenta é realizada por meio de uma API REST (Figura - 5.2), que recebe as requisições do *frontend*, processa-as e retorna

os resultados. A API recebe como parâmetros o número de autores e os modelos a serem utilizados no teste (SVM e/ou *Random Forest*).

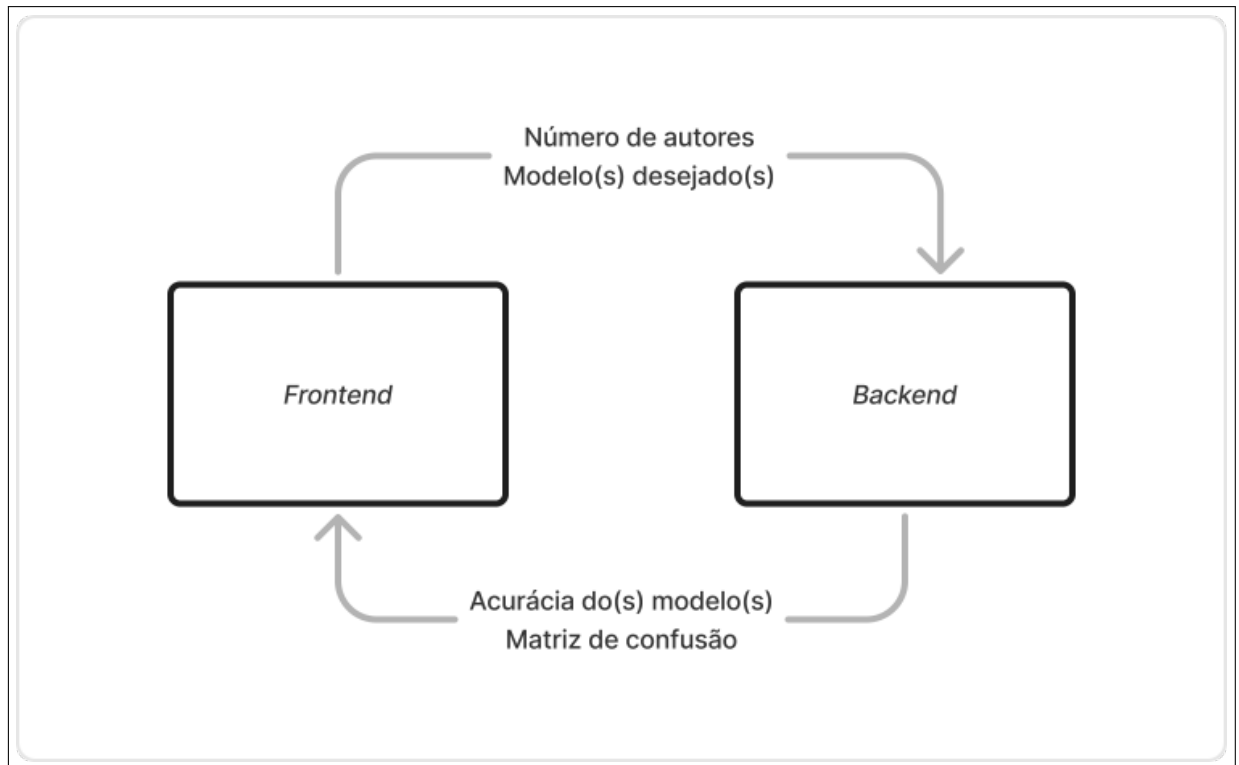


Figura 5.2: Comunicação entre *frontend* e *backend*.

Com os parâmetros recebidos, o *backend* (Figura - 5.3) carrega os dados de treino e teste dos arquivos CSV, seleciona aleatoriamente os autores de acordo com a quantidade especificada pelo usuário para compor os conjuntos de treino e teste e, finalmente, treina e testa os modelos selecionados pelo usuário, retornando as acurácias obtidas.

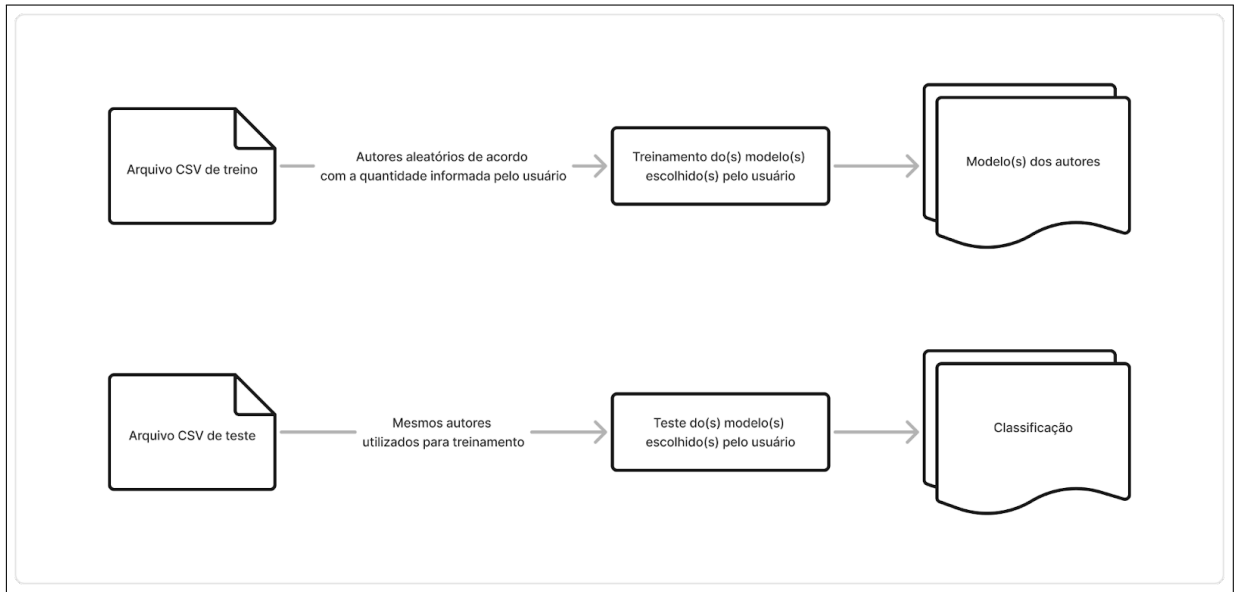


Figura 5.3: Funcionamento do *backend*.

Os resultados dos experimentos, incluindo a acurácia de cada modelo, são retornados para o *frontend* e exibidos ao usuário. Além da acurácia, o *backend* gera uma matriz de confusão (Figura - 5.4) para o modelo SVM, visualizando o desempenho do modelo em cada classe (autor) e permitindo a análise dos erros de classificação.

A próxima seção descreve os resultados obtidos nos experimentos com os modelos SVM e *Random Forest*, utilizando a característica inclinação axial para a identificação de autoria em manuscritos.

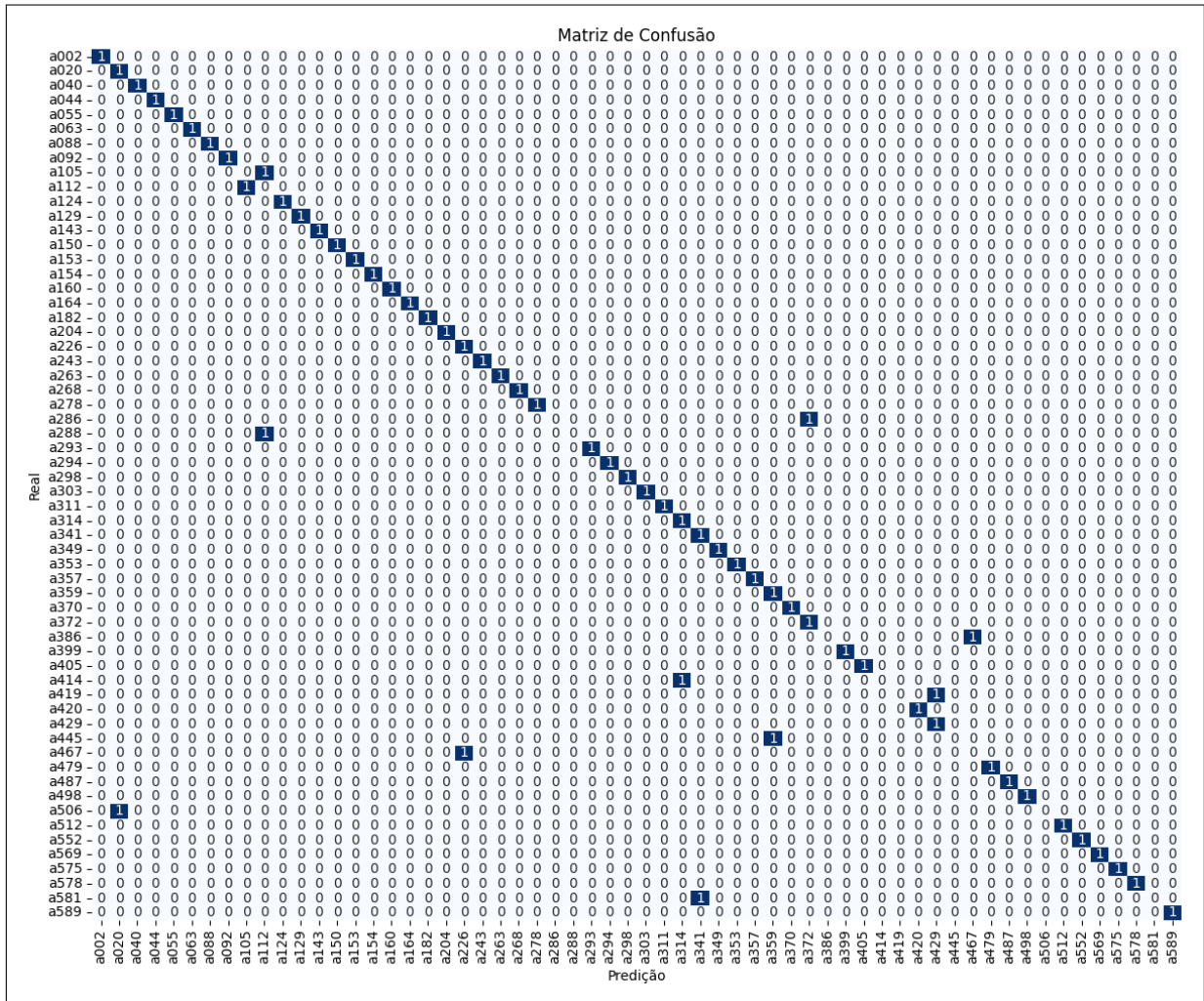


Figura 5.4: Matriz de confusão em uma análise de 60 autores.

5.2 Resultados

A Tabela - 5.1 apresenta as melhores taxas de acerto obtidas por cada modelo, variando-se a quantidade de autores presentes nos conjuntos de dados. As melhores taxas de acerto foram selecionadas porque o algoritmo realiza a escolha dos autores de forma aleatória, o que pode gerar variações nos resultados obtidos em cada execução. Portanto, destacar os melhores desempenhos permite identificar o potencial máximo de cada modelo, considerando a variabilidade inerente à seleção dos dados.

A fim de analisar a influência do número de autores no desempenho dos modelos e determinar a quantidade ideal de escritores para estabilizar os resultados, foi adotada uma metodologia similar à de Amaral (2014), incrementando o conjunto de dados em intervalos de 20 autores. Essa abordagem permite avaliar a robustez dos modelos em cenários

com diferentes níveis de complexidade, representativos de situações reais encontradas em perícias.

Uma análise aprofundada dos melhores resultados obtidos nos experimentos, com foco nas diferenças de desempenho entre os modelos e na influência da quantidade de autores na acurácia, será apresentada no próximo capítulo.

Tabela 5.1: Melhores resultados obtidos por cada modelo.

Número de Escritores	Taxa de Acerto (%)		
	SVM	SVM com Grid Search	Random Forest
20	95,00	95,00	90,00
40	80,00	92,50	82,50
60	78,33	81,67	78,33
80	68,75	76,25	76,25
100	67,00	71,00	68,00
120	70,83	76,66	71,66
140	59,28	69,28	62,14
160	61,87	71,25	65,00
180	56,11	67,77	67,22
200	55,00	65,50	59,50

Análise e discussão dos resultados

Os experimentos realizados, seguindo o protocolo descrito no capítulo 5, tiveram como objetivo principal avaliar o desempenho da ferramenta desenvolvida para identificação de autoria em manuscritos, utilizando características grafométricas e os classificadores SVM, SVM com *Grid Search* e *Random Forest*. A métrica utilizada para avaliação foi a acurácia, que representa a porcentagem de acertos do modelo na classificação dos autores.

A Tabela - 5.1 apresenta as melhores taxas de acerto obtidas por cada modelo, variando-se a quantidade de autores de 20 a 200, em intervalos de 20. Essa variação permitiu analisar a influência do número de autores no desempenho dos modelos e a robustez dos mesmos em cenários com diferentes níveis de complexidade.

Observando os resultados, destaca-se o desempenho superior do SVM com *Grid Search* em relação aos outros dois modelos, em todos os cenários testados. O *Grid Search*, ao otimizar os hiperparâmetros do SVM, permitiu uma melhor adaptação do modelo aos dados, resultando em maior acurácia. Para um número menor de autores (20 e 40), as taxas de acerto foram consideravelmente altas, 95% e acima de 92%, respectivamente para o SVM com *Grid Search*. Isso sugere que em cenários com um número reduzido de potenciais autores, a ferramenta apresenta alta eficiência na identificação de autoria.

À medida que o número de autores aumenta, observa-se uma queda gradual na acurácia para todos os modelos. Essa queda é esperada, uma vez que a complexidade do problema aumenta com o número de classes a serem classificadas. No entanto, mesmo com 200 autores, o SVM com *Grid Search* manteve uma taxa de acerto de 65,5%, demonstrando ainda uma capacidade razoável de discriminação entre autores. Comparativamente, o

SVM sem *Grid Search* e *Random Forest* apresentaram desempenho inferior, com acurácia de 55% e 59,5%, respectivamente, no cenário com 200 autores.

Conclusão

Este trabalho propôs uma ferramenta para identificação de autoria em manuscritos, baseado em características grafométricas extraídas de cartas forenses e utilizando algoritmos de aprendizado de máquina. Os resultados obtidos demonstram a viabilidade da abordagem proposta, com o SVM otimizado por *Grid Search* apresentando o melhor desempenho dentre os classificadores avaliados. A pesquisa contribui para a área de computação forense, oferecendo uma ferramenta que pode auxiliar peritos grafotécnicos na análise de manuscritos, automatizando parte do processo e reduzindo a subjetividade inerente à análise manual.

Apesar dos resultados promissores, identificou-se a necessidade de expandir o conjunto de características grafométricas analisadas para alcançar um desempenho comparável ao de outros trabalhos na literatura, como o de Amaral (2014). Como trabalhos futuros, sugere-se a investigação de novas características grafométricas, o desenvolvimento de técnicas de pré-processamento mais robustas, e a exploração de outros algoritmos de aprendizado de máquina, como redes neurais convolucionais, que têm se mostrado eficazes em tarefas de reconhecimento de padrões em imagens. Além disso, a expansão da base de dados com amostras de manuscritos que apresentem desafios específicos para a análise forense, como textos curtos, escritos disfarçados ou com diferentes graus de legibilidade e a avaliação do desempenho do método em cenários forenses reais são importantes passos para a consolidação da ferramenta como uma solução prática para a identificação de autoria em manuscritos.

REFERÊNCIAS

ALVES, Y. Manuscritus: Ferramenta baseada em características grafométricas para identificação de manuscritos. Disponível em: <https://manuscritus.vercel.app>, 2024a.

ALVES, Y. Repositório com o código da ferramenta manuscritus. Disponível em: <https://github.com/yuripiresalves/manuscritus>, 2024b.

AMARAL, A. M. M. M. *Identificação de autoria de documentos manuscritos utilizando características grafométricas*. Tese (doutorado), Pontifícia Universidade Católica do Paraná, 2014.

AMARAL, A. M. M. M.; FREITAS, C. O. A.; BORTOLOZZI, F. Combining multiple features based on graphometry for writer identification as part of forensic handwriting analysis. In: *Proceedings of International Document Image Processing*, Patras-Greece: International Association for Pattern Recognition (IAPR), 2013a, p. 23–30.

AMARAL, A. M. M. M.; FREITAS, C. O. A.; BORTOLOZZI, F. Feature selection for forensic handwriting identification. In: *Proceedings of International Conference on Document Analysis and Recognition (ICDAR)*, 2013b, p. 922–926.

BARANOSKI, F. *Verificação da autoria em documentos manuscritos usando svm*. Dissertação (mestrado em ciência da computação), Pontifícia Universidade Católica do Paraná, Paraná, 2005.

BATISTA, N.; ALVES, Y. Identificação de autoria em manuscritos utilizando a característica grafométrica inclinação axial. Projeto de Iniciação Científica — Universidade Estadual de Maringá, 2024.

BRUMASSIO, D. Identificação de autoria em manuscritos utilizando as características grafométricas: número de linhas e posicionamento de margens. Trabalho de Conclusão de Curso — Universidade Estadual de Maringá, 2024.

CHEN, J.; LOPRESTI, D.; KAVALLIERATOU, E. The impact of ruling lines on writer identification. In: *Proceedings of International Conference on Frontiers in Handwriting Recognition*, 2010, p. 439–444.

FREITAS, C. O. A.; OLIVEIRA, L. S.; BORTOLOZZI, F.; SABOURIN, R. Brazilian forensic letter database. In: *Proceedings of International Workshop on Frontiers on Handwriting Recognition*, 2008, p. 64–69.

SCIKIT LEARN scikit-learn: machine learning in python — scikit-learn 1.5.2 documentation. Disponível em: <https://scikit-learn.org/stable>, 2024.

LUNA, E. C. H.; RIVERON, E. M. F.; CALDERON, S. G. A supervised algorithm with a new differentiated-weighting scheme for identifying the author of a handwritten text. *Pattern Recognition Letters*, v. 32, p. 1139–1144, 2011.

MENDES, L. B. *Documentoscopia*. Campinas: Millennium, 344 p., 2003.

OTSU, N. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, v. SMC-9, n. 1, p. 62–66, 1979.

PERVOUCHINE, V.; LEEDHAM, G. Extraction and analysis of forensic document examiner features used for writer identification. *Pattern Recognition*, v. 40, p. 1004–1013, 2007.

RUSSELL, S.; NORVIG, P. *Artificial intelligence: A modern approach*. 4th edition ed. Pearson, 1152 p., 2020.

SAUDEK, R. *Experiments with handwriting*. Great Britain: George Allen & Unwin, 389 p., 1978.