UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
ESCOLA DE ENGENHARIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

Giovanni Gaiardo

Identificação e Cadastro de Medidores de Energia Elétrica Utilizando Técnicas de Aprendizagem de Máquina

Porto Alegre

2023

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
ESCOLA DE ENGENHARIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

Giovanni Gaiardo

Identificação e Cadastro de Medidores de Energia Elétrica Utilizando Técnicas de Aprendizagem de Máquina

Projeto de Diplomação, apresentado ao Departamento de Engenharia Elétrica da Escola de Engenharia da Universidade Federal do Rio Grande do Sul, como requisito parcial para a obtenção do grau de Engenheiro Eletricista

UFRGS

Orientador: Prof. Dr. Tiago Oliveira Weber

Porto Alegre
2023

Resumo

Laboratórios de metrologia legal são instalações responsáveis por garantir a conformidade e precisão das medições realizadas em diversos setores da sociedade, como comércio, indústria e saúde. O laboratório de Verificação Metrológica do LABELO - PUCRS recebe cerca de 3000 medidores de energia elétrica por mês, sendo o tempo médio para cadastro de seus dados de identificação, calculado ao longo de 2 meses, igual a 5 minutos e 25 segundos. Uma das etapas do cadastro é efetuar um registro fotográfico da amostra. Este projeto busca extrair da imagem os dados em texto para preencher 10 campos de cadastro e efetuar o preenchimento deste automaticamente. A principal ferramenta utilizada é o *software* de código aberto *Tesseract*, motor de reconhecimento óptico de caracteres (OCR) que será utilizado em série com algoritmos de aprendizagem de máquina como SVM, Naive Bayes e Random Forests, além de técnicas clássicas, para classificar os textos extraídos entre os campos de cadastro do medidor. A pesquisa culminou no desenvolvimento de três modelos finais (SVM, um modelo baseado em léxico e um modelo híbrido SVM que avaliava as saídas do baseado léxico) com 77,22%, 68,12% e 97,01% de taxa de acerto para classificação de textos extraídos de imagens de medidores de energia elétrica. Estes modelos foram comparados em termos de desempenho e complexidade, a fim de determinar se uma abordagem utilizando *machine learning* era de fato necessária. Por fim, concluiu-se que o modelo híbrido seria a melhor opção no contexto desta pesquisa para atuar na classificação de palavras para cadastro automático dos medidores.

Palavras-chave: Aprendizagem de Máquina, OCR, Metrologia Legal, Medidores de Energia Elétrica.

Abstract

Legal metrology laboratories are facilities responsible for ensuring the compliance and accuracy of measurements carried out in various sectors of society, such as commerce, industry and health. The LABELO - PUCRS Metrological Verification laboratory receives about 3000 electricity meters per month, with the average time to register their identification data, calculated over 2 months, equal to 5 minutes and 25 seconds. One of the registration steps is to carry out a photographic record of the sample. This project seeks to extract text data from the image to fill in 10 registration fields and fill them in automatically. The main tool used is the open source *software Tesseract*, an optical character recognition (OCR) engine that will be used in series with machine learning algorithms such as SVM, Naive Bayes and Random Forests, in addition to classic techniques, to classify the extracted texts between the meter registration fields. The research culminated in the development of three final models (SVM, a lexicon-based model and a hybrid SVM model that evaluated the lexicon-based outputs) with 77.22%, 68.12% and 97.01% rate of success for classifying texts extracted from images of electric energy meters. These models were compared in terms of performance and complexity, in order to determine if an approach using *machine learning* was indeed necessary. Finally, it was concluded that the hybrid model would be the best option in the context of this research to act in the classification of words for the automatic registration of meters.

Keywords: Machine Learning, OCR, Legal Metrology, Electrical Energy Meters.

Lista de Figuras

Figura 1 – Perdas sobre a energia injetada. 9
Figura 2 – Tela de cadastro de um medidor recebido. 10
Figura 3 – Número de Unidades Consumidoras no Brasil entre 2011 e 2017. 15
Figura 4 – Resultados da pesquisa bibliométrica agrupados em *clusters*. 16
Figura 5 – Uma abordagem para o reconhecimento óptico de caracteres. 22
Figura 6 – Tempo médio empregado em cada etapa do fluxo verificação. 25
Figura 7 – Fluxo de funcionamento da metodologia experimental. 26
Figura 8 – Técnico do laboratório fotografando uma amostra para cadastro em sua estação. 29
Figura 9 – Exemplo de imagem presente no banco de dados. 30
Figura 10 – Sinalização dos textos que deseja-se extrair da imagem. 31
Figura 11 – Exemplo de imagem com má legibilidade dos dados de identificação do instrumento. 32
Figura 12 – Ilustração da normalização das palavras. 35
Figura 13 – Imagem de amostra sem nenhum pré-processamento. 43
Figura 14 – Etapas de pré-processamento das imagens discriminadas. 43
Figura 15 – Imagem de amostra após emprego das técnicas de pré-processamento. . 44
Figura 16 – Matriz de confusão para o modelo SVM. 46
Figura 17 – Matriz de confusão para o modelo RF. 47
Figura 18 – Matriz de confusão para o modelo NB. 49
Figura 19 – Variação da distância mínima tolerada em função da taxa de acerto na classificação de palavras. 50
Figura 20 – Matriz de confusão para o modelo baseado em léxico. 51
Figura 21 – Matriz de confusão para o modelo híbrido. 59

Lista de Tabelas

Tabela 1 – Evolução do número de unidades consumidoras de 2011 a 2017. 15
Tabela 2 – Ferramentas Computacionais Utilizadas. 28
Tabela 3 – Distribuição da base de dados utilizada. 32
Tabela 4 – Trecho do banco de dados em texto. 34
Tabela 5 – Busca aleatorizada pelo valor ótimo dos hiper-parâmetros para cada modelo. 38
Tabela 6 – Exemplificação da aplicação da distância de Levenshtein. 39
Tabela 7 – Novo banco de dados para o modelo final. 40
Tabela 8 – Exemplificação da aplicação das técnicas de pré-processamento dos textos. 45
Tabela 9 – Desempenho do classificador SVM em cada classe. 46
Tabela 10 – Desempenho do classificador RF em cada classe. 48
Tabela 11 – Desempenho do classificador Naive Bayes em cada classe. 49
Tabela 12 – Desempenho do classificador baseado em léxico em cada classe. 52
Tabela 13 – Desempenho dos diferentes classificadores de texto testados 52
Tabela 14 – Exemplo do modelo SVM em operação 54
Tabela 15 – Exemplo do modelo baseado em léxico em operação. 56
Tabela 16 – Desempenho dos diferentes classificadores de texto testados. 57
Tabela 17 – Desempenho do classificador híbrido em cada classe. 59
Tabela 18 – Desempenho dos modelos inteligente, clássico e híbrido comparados. . . 60

Lista de abreviaturas

OCR	Optical Character Recognition
NB	Naive Bayes
SVM	Support Vector Machine
RF	Random Forests
INMETRO	Instituto Nacional de Metrologia, Qualidade e Tecnologia
ANEEL	Agência Nacional de Energia Elétrica
GPU	Graphics Processing Unit

Sumário

1 INTRODUÇÃO 9
1.1 Contextualização 9
1.2 Objetivo Geral 11
1.3 Objetivos Específicos 12
1.4 Trabalhos relacionados 12

2 FUNDAMENTAÇÃO TEÓRICA E REVISÃO BIBLIOGRÁFICA . . 14
2.1 Metrologia Legal - Medidores de Energia Elétrica 14
2.2 Pesquisa Bibliométrica 16
2.3 Aprendizado de Máquina - Classificação de textos utilizando modelos inteligentes 17
2.3.1 Naive Bayes 17
2.3.2 Máquina de Vetores de Suporte (SVM) 18
2.3.3 Florestas Aleatórias (RF) 19
2.4 Processamento de Linguagem Natural 20
2.5 Visão Computacional 20
2.5.1 Pré-processamento de Imagens 21
2.5.2 OCR - *Optical Character Recognition* 21
2.6 Análise de Desempenho dos Modelos 23
2.6.1 Métricas estatísticas 23
2.6.2 Comparação com o tempo médio de cadastro sem preenchimentos automáticos 24

3 METODOLOGIA 26
3.1 Materiais e Ferramentas 27
3.1.1 Ferramentas Computacionais 27
3.1.2 *Hardware* Utilizado 28
3.2 Estudo e Uso das Bases de Dados 28
3.2.1 Construção da Base de Dados de Imagens 28
3.2.1.1 Pré-processamento 33
3.2.2 Base de Dados de Textos da Carcaça dos Medidores 33
3.2.2.1 Pré-processamento 34
3.3 Procedimento 37
3.3.1 Separação de Conjuntos de Treinamento, Validação e Teste 37
3.3.2 Abordagem utilizando Modelo Inteligente 37
3.3.2.1 Otimização de hiper-parâmetros 37
3.3.3 Método Baseado em Léxico 39

3.3.4 Abordagem Híbrida 40
3.4 Tesseract 41

4 ANÁLISE DE RESULTADOS 42
4.1 Pré-processamento 42
4.1.1 Imagens 42
4.1.2 Textos 45
4.2 Seleção do modelo inteligente 45
4.2.1 SVM 45
4.2.2 RF 47
4.2.3 NB 48
4.2.4 Classificador baseado em Léxico 50
4.2.5 Comparação de desempenho e testes da Abordagem Clássica e da Abordagem com Modelos Inteligentes 52
4.3 Discussões acerca da Abordagem Clássica comparada à Abordagem com Modelos Inteligentes 58
4.4 Implementação Híbrida: SVM + Levenshtein 58

5 CONCLUSÃO 61
5.1 Considerações finais 61
5.2 Trabalhos futuros 61
5.3 Limitações da Pesquisa 62

REFERÊNCIAS BIBLIOGRÁFICAS 64

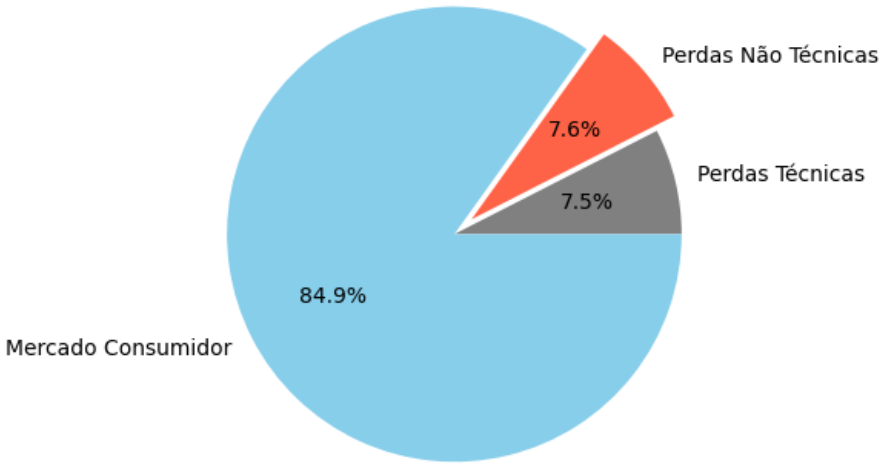
1 Introdução

1.1 Contextualização

O sistema elétrico brasileiro se divide em três partes: Geração, Transmissão e Distribuição. Na Distribuição existem perdas, definidas pela diferença entre a energia elétrica adquirida pelas distribuidoras e a faturada aos seus consumidores. Estas perdas categorizam-se entre perdas técnicas (inerentes à atividade de distribuição de energia elétrica) e perdas não técnicas, que estarão em foco neste estudo. Para estimar as perdas não técnicas, toma-se a diferença entre as perdas totais e as perdas técnicas. Tipicamente este tipo de perda tem origem em furtos, fraudes, erros de leitura, medição e faturamento.

Quantitativamente, as perdas não técnicas foram estimadas em torno de 37,9 TWh em 2020 (ANEEL, 2020). Este é um dado de grande relevância social, pois impacta diretamente no preço da energia elétrica faturada ao consumidor final, uma vez que os montantes de prejuízos por perdas não técnicas são divididos entre os consumidores do mercado de baixa tensão faturado. A Figura 1 ilustra a proporção dos tipos de perdas em relação à energia injetada.

Figura 1 – Perdas sobre a energia injetada.



Fonte: ANEEL (2020).

Assim, percebe-se a importância da área de Metrologia Legal para esse mercado, a fim de garantir a robustez e proteção dos instrumentos diante de tentativas de fraude e outras violações, seguindo os respectivos requisitos normativos das Portarias Inmetro 586 e 587 de 2012, por exemplo, para o caso específico de Medidores de Energia Elétrica.

No entanto, mesmo após processos rigorosos de homologação de um instrumento antes de seu lançamento no mercado, ainda ocorrem casos de fraude. Nesta situação, a distribuidora pode procurar um laboratório acreditado para realizar ensaios e incrementar a investigação da causa da perda. Se foi de fato um caso de perda não técnica; se sim, qual foi sua causa, como foi realizada a infração; estes são questionamentos que um Laboratório de Metrologia Legal deve ser capaz de sanar por meio de um Laudo Técnico.

O LABELO é o complexo de Laboratórios Especializados em Eletroeletrônica da PUCRS, onde será integrado um novo Laboratório de Metrologia Legal com foco em medidores de energia elétrica sob suspeita de fraude ou irregularidade na medição. A empresa estima que o laboratório receberá, por mês, uma média de 3 mil medidores de energia elétrica para inspeção. Assim, o laboratório foi projetado para seguir modelos de altíssima produtividade, onde o tempo é o mais importante dos recursos.

A primeira etapa do fluxo de um medidor de energia elétrica nesse laboratório é o cadastro do instrumento, onde será feito um registro fotográfico do estado do medidor para constar posteriormente no Laudo Técnico. Esta fotografia é capturada por um operador, que posteriormente preenche 10 campos de cadastro com especificações e dados de identificação impressos na carcaça do instrumento, como tensão nominal, modelo e fabricante. A Figura 2 mostra a íntegra da tela de cadastro.

Figura 2 – Tela de cadastro de um medidor recebido.

Fonte: Adaptado de LABELO, 2022.

Pretende-se implementar, por meio de técnicas de aprendizagem de máquina ou técnica clássica, um recurso computacional que extraia da imagem os dados em texto e preencha automaticamente os campos correspondentes no cadastro. O objetivo dessa implementação é abreviar a tarefa do operador, de forma que este precise apenas fotografar o

medidor de energia elétrica e conferir rapidamente se os dados preenchidos automaticamente estão de acordo. A expectativa é de que o funcionário responsável consiga cadastrar mais medidores por dia, alcançando melhores resultados em termos de produtividade e atingindo com mais facilidade as metas do laboratório em questão.

1.2 Objetivo Geral

Este trabalho tem como objetivo testar a hipótese de que um método envolvendo *machine learning* é a melhor forma para classificar dados extraídos de medidores de energia elétrica via Reconhecimento Óptico de Caracteres (OCR) a partir de suas imagens. Para tal, deseja-se implementar, por meio de técnicas de aprendizagem de máquina, um recurso computacional que extraia da imagem do medidor de energia elétrica os dados em texto (apresentados em diferentes regiões da carcaça do instrumento) e preencha automaticamente os campos correspondentes no cadastro. Pretende-se avaliar também o desempenho destas técnicas comparando-as com métodos baseados em léxico. Serão utilizados diferentes recursos de OCR e aprendizagem de máquina, com destaque para o motor *Tesseract*.

1.3 Objetivos Específicos

- Aquisição e análise das bases de dados;
- Avaliar os distintos efeitos de classificação das amostras com base em diferentes condições de captura das imagens;
- Estudar técnicas de pré-processamento de imagens para preparação destas para OCR;
- Estudar técnicas de pré-processamento de texto para preparar estes para alimentar modelos classificadores de textos (extraídos das imagens);
- Testar diferentes abordagens e algoritmos para classificar textos extraídos de imagens de medidores de energia elétrica via OCR;
- Avaliar a necessidade do emprego de modelos inteligentes na classificação de textos extraídos de imagens, comparando com técnicas clássicas.

1.4 Trabalhos relacionados

No trabalho de (AMALIA *et al.*, 2018) descreve um método para classificação automática de memes de acordo com sua categoria de opinião política usando Reconhecimento Óptico de Caracteres (OCR) e o algoritmo Naïve Bayes. O estudo se baseia em uma coleta de dados de 100 memes políticos em língua indonésia. O pré-processamento dos dados consistiu na extração de texto e imagem de cada meme e na remoção de *stopwords* e caracteres especiais. Em seguida, o texto de cada meme foi categorizado em duas classes de opinião sobre o governo: positiva e negativa, usando o classificador Naïve Bayes. Os resultados indicaram que o método proposto teve uma precisão de 75% na classificação dos memes políticos. O estudo sugere que o uso de OCR e técnicas de processamento de texto pode ser útil para a análise de memes políticos e pode ser aplicado em pesquisas de opinião pública e monitoramento de redes sociais.

Já o trabalho de (HUBERT *et al.*, 2021a) discute o desenvolvimento de um sistema para classificar imagens promocionais automaticamente, sem intervenção humana, utilizando Reconhecimento Óptico de Caracteres (OCR) e o algoritmo Naïve Bayes como classificador. O estudo busca resolver o problema de promoções de negócios estarem misturadas em meio a um grande volume de outras imagens nas redes sociais, dificultando o acesso do público a essas informações. Os pesquisadores compararam o desempenho do algoritmo Naïve Bayes com *Random Forest* e *K-Nearest Neighbor*, utilizando o método de validação cruzada com 158 imagens divididas em cinco grupos para treinar e testar o modelo. Os resultados mostraram que o modelo Naïve Bayes obteve uma precisão

média de 94,31%, recall de 94,33%, precisão de 94,11% e F1 score de 0,93, o que foi o melhor desempenho entre os três algoritmos testados. Os autores concluem que o OCR e o algoritmo Naïve Bayes são adequados para a classificação de imagens promocionais. O estudo contribui para a aplicação de técnicas de inteligência artificial na identificação automática de informações promocionais em imagens em redes sociais.

A pesquisa de (CâNDIDO, 2020) aborda a aplicação de redes neurais profundas na classificação automática de texto. A autora desenvolveu um estudo comparativo entre diversos métodos, incluindo diferentes arquiteturas de redes neurais como redes neurais de convolução, de atenção, e transformadores bidirecionais e as comparou com um dos algoritmos de aprendizado de máquina mais tradicionais, denominado Máquinas de Vetor de Suporte (SVM). Estes foram aplicados na classificação de textos em duas bases de dados, obtidas a partir de avaliações de filmes e de comentários em redes sociais. A pesquisa utilizou medidas de desempenho, como taxa de acerto, precisão, *recall* e F1 score, para comparar os resultados obtidos pelos diferentes modelos de redes neurais profundas. Os resultados indicaram que as redes neurais profundas apresentaram um desempenho superior a outros métodos de classificação de texto, alcançando uma taxa de acerto média de 93,5% na base de dados de avaliações de filmes e 84,3% na base de dados de comentários em redes sociais. A dissertação contribui para a aplicação de técnicas de inteligência artificial na classificação automática de texto e apresenta um estudo comparativo relevante entre diferentes modelos de redes neurais profundas para essa finalidade.

(ZHANG *et al.*, 2017) propuseram um modelo de classificação de produtos baseado em redes neurais convolucionais (CNN) de nível de caractere. O modelo foi treinado em um grande conjunto de dados de produtos e alcançou uma precisão de classificação de mais de 95%. Esse trabalho demonstrou a eficácia do uso de técnicas de deep learning para classificação de texto em grande escala.

(CHEN; LIU; ZHANG, 2020) também utilizaram uma abordagem baseada em redes neurais para classificação de produtos, mas dessa vez utilizando uma rede neural recorrente (RNN) com atenção para modelar a relação entre os atributos do produto e a categoria do produto. Os resultados mostraram que essa abordagem obteve uma precisão de classificação de mais de 96%.

2 Fundamentação Teórica e Revisão Bibliográfica

Neste capítulo são discutidos e revisados temas e trabalhos pertinentes e necessários para o entendimento da pesquisa. Primeiro é revisado o contexto dos medidores de energia elétrica na Metrologia Legal; após, o aprendizado de máquina e, por fim, o reconhecimento óptico de caracteres.

2.1 Metrologia Legal - Medidores de Energia Elétrica

Metrologia Legal é a parte da metrologia referente às atividades resultantes de exigências obrigatórias, relacionada às medições, unidades de medida, instrumentos e métodos de medição, desenvolvidas por organismos competentes (SAMPAIO *et al.*, 2009). A meta é, principalmente, garantir a qualidade das medições efetuadas em relações comerciais, provendo confiabilidade às medições, bem como aos instrumentos utilizados para definir quantidades envolvidas em transações.

A Metrologia Legal é o conjunto de atividades que compreendem a fiscalização e a verificação de instrumentos de medição, com o objetivo de assegurar que os resultados das medições realizadas com esses instrumentos sejam confiáveis e que os produtos comercializados estejam de acordo com as especificações estabelecidas (INMETRO, 2023).

Dentre as principais atividades realizadas neste campo de trabalho, destacam-se a calibração e a verificação dos instrumentos de medição, a aprovação de modelos de instrumentos de medição e a fiscalização do cumprimento das normas e regulamentações relacionadas à metrologia. Essas atividades são realizadas por organismos competentes, como o INMETRO e as agências reguladoras de cada setor.

A Metrologia Legal é importante para assegurar a justiça nas transações comerciais, uma vez que as medições corretas são fundamentais para determinar preços, quantidades e qualidade dos produtos. Além disso, a Metrologia Legal também é importante para garantir a segurança e a saúde pública, pois muitos equipamentos e instrumentos de medição são utilizados em áreas críticas, como saúde, meio ambiente, transporte e energia, área de interesse do trabalho em questão.

A ANEEL fornece base de dados que indica o crescimento de consumidores de energia elétrica no Brasil. Observar esta tendência permite entender que a cada ano, mais pessoas podem ser impactadas por violações metrológicas. A Tabela 1 mostra esta progressão quantitativamente.

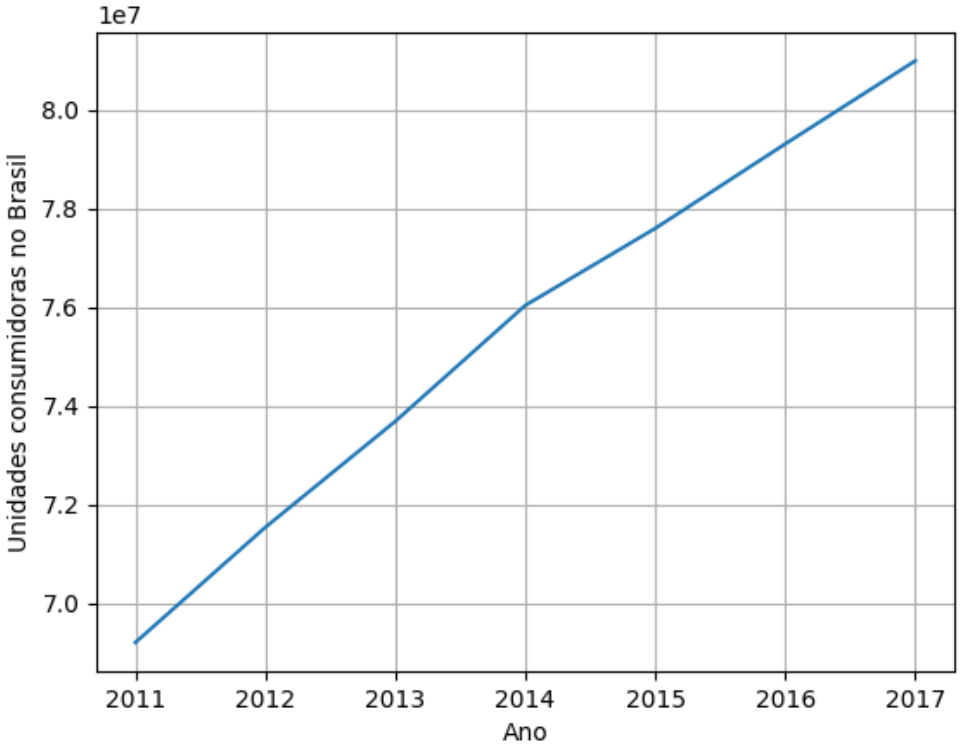
Tabela 1 – Evolução do número de unidades consumidoras de 2011 a 2017.

Ano	**Número de Unidades Consumidoras**
2011	69203952
2012	71540711
2013	73691206
2014	76042278
2015	77604858
2016	79318057
2017	81002827

Fonte: ANEEL, s/d.

A Figura 3 ilustra o crescimento do número (em dezenas de milhões) de unidades consumidoras no Brasil no período entre os anos de 2011 e 2017.

Figura 3 – Número de Unidades Consumidoras no Brasil entre 2011 e 2017.



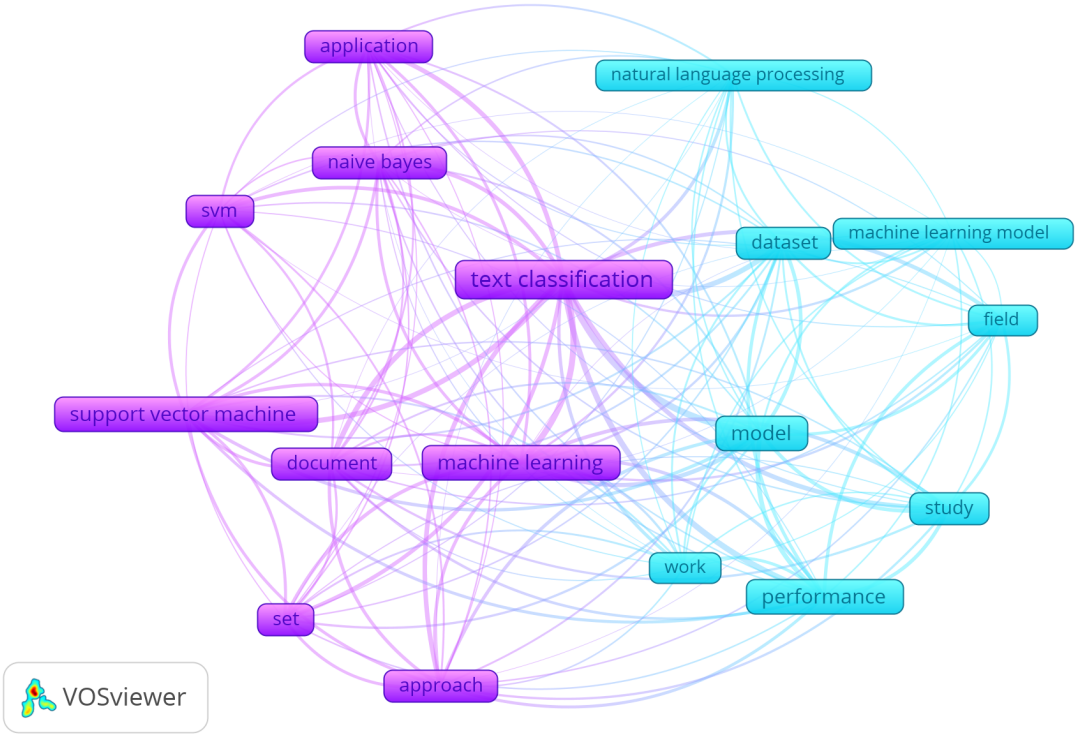
Fonte: O autor, com base em dados da ANEEL, 2023.

Esta visualização dos dados revela uma taxa de crescimento com característica linear, dentro da faixa de tempo observada, do número de unidades consumidoras no Brasil, destacando-se o crescimento médio de aproximadamente 2 milhões de unidades consumidoras ao ano. Isto reforça a importância deste estudo, uma vez que busca auxiliar um laboratório cujo objetivo é avaliar casos de fraudes neste grande volume de consumidores.

2.2 Pesquisa Bibliométrica

Foi realizada uma pesquisa bibliométrica acessando diversos bancos de dados por meio da plataforma *Web of Science*. A Figura 4 apresenta os resultados da pesquisa, buscando pelos termos mais recorrentes entre artigos sobre classificação de textos e modelos inteligentes, isto é, utilizando como parâmetros as palavras-chave "*Text Classification*"e "*Machine Learning*". A pesquisa limitou-se a resultados correspondentes ao período de 01/01/2018 a 31/12/2022.

Figura 4 – Resultados da pesquisa bibliométrica agrupados em *clusters*.



Fonte: O autor, 2023.

Na Figura 4 podemos perceber a formação de dois agrupamentos de palavras. Em lilás, podemos observar as palavras mais relacionadas à aplicação de classificação de texto em si, com algumas das principais técnicas utilizadas em destaque. Já o *cluster* correspondente à cor azul indica as palavras resultantes da pesquisa mais voltadas ao contexto geral de *machine learning*, sem envolver necessariamente uma dada aplicação. Desta forma, buscou-se analisar e compreender os pontos de conexão entre estes agrupamentos. Destacam-se as ligações fortes do termo "Text classification"com alguns modelos inteligentes clássicos, como Máquina de Vetores de Suporte e o classificador Naive Bayes. Por outro lado, também há uma ligação forte com a palavra "Document", sinalizando que esta abordagem é comumente utilizada para classificação de textos extensos e não tanto para palavras individualmente, como se perceberá futuramente nesta pesquisa. Pela recorrência observada nos mais de 7 mil artigos resultantes da pesquisa, os modelos SVM

e Naive Bayes foram escolhidos para os testes executados neste trabalho. Adicionalmente, foi testado o método de aprendizado por Florestas Aleatórias para comparação.

2.3 Aprendizado de Máquina - Classificação de textos utilizando modelos inteligentes

(TURING, 1950) discutiu em um artigo a possibilidade matemática de criar uma Inteligência Artificial em 1950. No entanto, a capacidade limitada e alto custo de computadores à época impediram a implementação do conceito (CHEN *et al.*, 2022). Hoje, Inteligência Artificial é um termo amplo que pode ser interpretado como desenvolvimento e programação de um computador projetado para treinar máquinas para executar tarefas (HARIKA *et al.*, 2022). É dentro deste amplo escopo que está inserido o conceito de aprendizado de máquina, técnica que engloba o aprendizado profundo futuramente discutido, por exemplo.

Métodos de aprendizado de máquina supervisionados podem compreender conteúdo personalizado com base em dados de texto rotulados manualmente e construir indutivamente classificadores com base em padrões observados sem exigir programação manual de regras de classificação (HARTMANN *et al.*, 2019). Em comparação, existem os métodos de classificação baseados em léxicos, que funcionam como correções por dicionário. Um especialista no domínio desejado é necessário para construir o vocabulário. Se a área de pesquisa não possuir dicionário previamente construído, é necessário criar um próprio, como no presente caso. Modelos inteligentes podem trazer maiores vantagens nesta aplicação por serem naturalmente mais flexíveis do que uma busca pela melhor correspondência em um léxico.

2.3.1 Naive Bayes

O algoritmo Naive Bayes é um algoritmo de classificação baseado na regra de Bayes e um conjunto de suposições de independência condicional. Ele prevê um valor de classe dado um conjunto de conjuntos de atributos (RISH, 2001). A probabilidade posterior, $P(c|x)$, pode ser calculada por meio do teorema de Bayes usando a probabilidade anterior de classe, $P(c)$, probabilidade anterior do preditor $P(x)$ e probabilidade, $P(x|c)$. O classificador "ingênuo"(em inglês, Naive) de Bayes assume que o impacto do valor de um preditor (x) em uma determinada classe (c) é independente dos valores de outros preditores. O termo "independência condicional de classe"refere-se a essa presunção.

O algoritmo Naive Bayes é amplamente utilizado na área de aprendizado de máquina devido à sua simplicidade e eficiência em classificar grandes conjuntos de dados com alta dimensionalidade. Ele é particularmente útil quando há muitos atributos, e a

interdependência entre eles é difícil de modelar. O algoritmo é baseado na suposição de que a probabilidade condicional de um atributo dada uma classe é independente dos outros atributos.

A Equação 1 ilustra o cálculo a probabilidade posterior, $P(c|x)$, usando o teorema de Bayes.

$$P(c|x) = \frac{P(x|c)P(c)}{P(x)} \tag{1}$$

Onde:

- $P(c|x)$ é a probabilidade posterior da classe c dado um vetor de atributos x;
- $P(c)$ é a probabilidade anterior da classe c;
- $P(x|c)$ é a probabilidade condicional de um vetor de atributos x dado uma classe c;
- $P(x)$ é a probabilidade marginal de um vetor de atributos x.

O classificador ingênuo de Bayes assume que todos os atributos são independentes, dado a classe c. Isso significa que a probabilidade condicional de um vetor de atributos x dado a classe c pode ser escrita como a multiplicação das probabilidades condicionais de cada atributo, dado a classe c, conforme apresentado na Equação 2.

$$P(x_1, x_2, ..., x_n|c) = \prod_{i=1}^{n} P(x_i|c) \tag{2}$$

Onde n é o número de atributos. Esta suposição de independência condicional pode nem sempre ser verdadeira na prática, mas muitas vezes é uma boa aproximação e permite que o algoritmo funcione bem em muitos casos.

Algumas variações do algoritmo Naive Bayes incluem o Naive Bayes gaussiano e o Naive Bayes multinomial, que são adaptados para diferentes tipos de dados. O Naive Bayes gaussiano é usado para dados contínuos, enquanto o Naive Bayes multinomial é usado para dados discretos ou de contagem.

2.3.2 Máquina de Vetores de Suporte (SVM)

A técnica de Máquina de Vetores de Suporte (SVM) é um algoritmo de aprendizado de máquina supervisionado capaz de realizar tanto regressão quanto classificação (CRISTIANINI; SHAWE-TAYLOR, 2000). O SVM funciona plotando cada item de dados como um ponto no espaço n-dimensional, onde n é o número de recursos disponíveis. O objetivo do SVM é traçar o hiperplano ótimo que separa os dados em suas classes conforme os rótulos fornecidos (BURGES, 1998).

O SVM é um classificador que divide um espaço vetorial entre zonas separadas por um hiperplano. Esse hiperplano é definido pela maximização da margem de separação entre as classes (CORTES; VAPNIK, 1995). O SVM é considerado uma técnica robusta e eficiente para lidar com dados de alta dimensionalidade e com problemas de classificação não-lineares (VAPNIK, 1995).

O treinamento do SVM envolve a otimização de um problema de programação quadrática que visa minimizar uma função custo que leva em consideração tanto a margem de separação quanto a ocorrência de pontos que estejam dentro da margem ou que tenham sido classificados erroneamente (SHALEV-SHWARTZ; BEN-DAVID, 2014).

O SVM é inerentemente um classificador binário, o que implica na adoção de abordagens como um contra um ou um contra todos, a fim de adaptar o SVM binário para tarefas de classificação multi-classe. O SVM linear é especialmente popular para problemas de classificação de texto, uma vez que é robusto para dados de alta dimensionalidade, sendo o melhor desempenho em tais cenários (CâNDIDO, 2020).

Em suma, a técnica de Máquina de Vetores de Suporte é uma ferramenta poderosa para lidar com problemas de classificação e regressão em diferentes áreas, como biologia, finanças e processamento de imagens (CRISTIANINI; SHAWE-TAYLOR, 2000; BURGES, 1998; CORTES; VAPNIK, 1995; VAPNIK, 1995).

2.3.3 Florestas Aleatórias (RF)

O algoritmo *Random Forests* (RF) é um dos algoritmos de aprendizado de máquina mais populares e é utilizado em problemas de classificação, regressão e outras tarefas de aprendizado supervisionado. Ele é composto por um conjunto de árvores de decisão, em que cada árvore é construída a partir de uma amostra aleatória de observações e variáveis preditoras (BREIMAN, 2001).

O processo de construção de uma árvore em um modelo *Random Forests* começa com a escolha de um subconjunto aleatório de observações e variáveis preditoras. Em seguida, o algoritmo encontra o melhor preditor para dividir o conjunto de dados em duas partes. Esse processo é repetido em cada subconjunto de dados, resultando em várias árvores de decisão.

A principal vantagem do algoritmo RF é sua capacidade de lidar com grandes conjuntos de dados e variáveis preditoras, além de ser menos suscetível a *overfitting* do que outros algoritmos de árvore de decisão (HO, 1995). Além disso, o RF pode ser usado para estimar a importância de cada variável preditora no modelo, o que é útil na seleção de variáveis em conjuntos de dados com muitas características (BREIMAN, 2001).

No entanto, assim como outros algoritmos de aprendizado de máquina, o RF também possui algumas limitações, como a dificuldade de interpretar a lógica por trás do

modelo e a necessidade de ajustar vários hiper-parâmetros para obter o melhor desempenho (BREIMAN, 2001).

O algoritmo RF tem sido aplicado em diversas áreas, como classificação de imagens (BANERJEE *et al.*, 2020), diagnóstico médico (LIU; WU; CHEN, 2020) e análise de séries temporais (CERQUEIRA *et al.*, 2020).

2.4 Processamento de Linguagem Natural

Para o uso adequado dos modelos estudados na Seção 2.3, foi necessária a compreensão dos fundamentos de Processamento de Linguagem Natural (PLN). PLN é uma subárea da Inteligência Artificial que lida com a compreensão e geração automática de linguagem humana. Um dos principais desafios do PLN é representar e processar a linguagem natural em formato computacional. A vetorização e tokenização de palavras são dois dos métodos mais importantes para representar o texto em formato computacional (CHEN; WANG; YANG, 2019).

A vetorização de palavras é o processo de representar palavras ou documentos em forma de vetores numéricos, a fim de permitir o uso de algoritmos de aprendizado de máquina (JURAFSKY; MARTIN, 2020). O *CountVectorizer* é uma técnica comum de vetorização de palavras que converte o texto em uma matriz de contagem de frequência de palavras. Cada linha da matriz representa um documento e cada coluna representa uma palavra única. O valor em cada célula representa o número de ocorrências da palavra correspondente no documento. O *CountVectorizer* é útil para encontrar a frequência de palavras em documentos e identificar palavras-chave.

A tokenização é o processo de dividir o texto em unidades significativas, chamadas *tokens*. Os *tokens* geralmente são palavras individuais ou pontuações, mas podem ser definidos de várias maneiras, dependendo do contexto do problema em questão (MANNING; SCHÜTZE, 1999). O *CountVectorizer* usa uma tokenização simples, que divide o texto em palavras individuais usando espaços em branco e pontuações como delimitadores. Existem outras técnicas de tokenização mais avançadas, como a tokenização baseada em regras e a tokenização fundamentada em aprendizado de máquina, que podem melhorar a qualidade da representação do texto em alguns casos.

2.5 Visão Computacional

Nesta seção, será feita uma breve revisão bibliográfica sobre os temas da área de visão computacional abordados pelo trabalho. Visão computacional é a área da ciência da computação responsável por analisar e interpretar informações relevantes de imagens e/ou vídeos para gerar dados para aplicações, por vezes demandando domínio de diferentes

técnicas de pré-processamento para tal. Outra atribuição comum da área é a extração de informações das imagens/vídeos, onde entra o conceito de OCR futuramente discutido.

2.5.1 Pré-processamento de Imagens

Uma das técnicas mais utilizadas para pré-processamento de imagens é a conversão das imagens para *Grayscale*, que é um processo que consiste em converter as imagens coloridas em tons de cinza, reduzindo a informação de cor para uma escala de valores entre 0 e 255. Essa técnica é amplamente utilizada em processamento de imagens para facilitar a análise e a aplicação de algoritmos de reconhecimento de padrões. A conversão para *Grayscale* é útil para reduzir a complexidade computacional, tornar as imagens mais fáceis de serem manipuladas e reduzir os efeitos de iluminação e sombra que podem interferir na análise (GONZÁLEZ; WOODS, 2007).

Outra importante técnica adotada neste trabalho é a limiarização das imagens, que é um processo de segmentação de imagens que consiste em converter os pixels da imagem em valores binários, utilizando um limiar definido previamente. Esse limiar é um valor que separa os pixels em duas classes: pixels que possuem intensidades maiores ou iguais ao limiar, que são convertidos para o valor máximo (geralmente 255), e pixels que possuem intensidades menores que o limiar, que são convertidos para o valor mínimo (geralmente 0). Essa técnica é útil para destacar objetos ou regiões de interesse em uma imagem e eliminar ruídos ou áreas irrelevantes. A limiarização é uma técnica de segmentação de imagens amplamente utilizada em processamento de imagens para extrair informações relevantes de uma imagem (JAIN; KASTURI; SCHUNCK, 1995).

Por fim, abertura das imagens é um processo morfológico que consiste em realizar uma dilação seguida de uma erosão na imagem. A dilação é um processo que expande as áreas brancas (objetos) da imagem, enquanto a erosão é um processo que reduz essas áreas. A abertura é útil para eliminar ruídos e pequenas manchas na imagem, mantendo as características principais dos objetos. A abertura é uma técnica de processamento morfológico que pode ser utilizada para reduzir o tamanho de objetos, separar objetos conectados ou eliminar áreas indesejadas em uma imagem (SERRA, 1982).

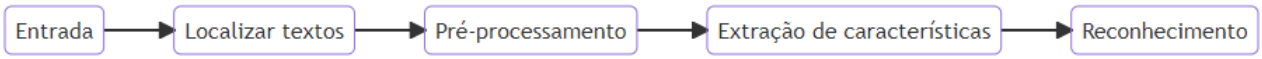
2.5.2 OCR - *Optical Character Recognition*

Em português, Reconhecimento Óptico de Caracteres é uma tecnologia capaz de reconhecer texto a partir de uma imagem digital (CHRISTENSSON, 2018). Atualmente, entre as diversas aplicações desta técnica, destacam-se a digitalização de documentos históricos manuscritos (KUMAR; KUMAR; PRATAP, 2018) e a conversão de dados em texto de uma imagem para dados manipuláveis (RADWAN; KHALIL; ABBAS, 2018).

Um sistema de reconhecimento óptico de caracteres depende principalmente da extração de recursos de uma imagem e da distinção destes recursos por meio de padrões.

Essencialmente, o OCR proporciona a obtenção de um arquivo editável de texto a partir de uma imagem digital ou mapa de bits. Uma abordagem típica de um sistema de reconhecimento óptico de caracteres divide-se em digitalização, determinar a posição dos textos, pré-processamento da imagem, extração de características e reconhecimento. Este processo está ilustrado na Figura 5.

Figura 5 – Uma abordagem para o reconhecimento óptico de caracteres.



Fonte: Adaptado de (BUI *et al.*, 2021).

O processo de digitalização é o primeiro passo no reconhecimento óptico de caracteres. Ele é responsável por converter uma imagem analógica em uma imagem digitalizada, a qual pode ser processada por um sistema computacional (ABBASI; ASLAM, 2017). Em seguida, é necessário determinar a posição dos textos na imagem digitalizada. Esse processo, conhecido como localização de texto, é crucial para a separação do texto de outros elementos presentes na imagem, como figuras e fundos (KIM; LEE, 2018).

Após a localização de texto, é necessário aplicar técnicas de pré-processamento na imagem para melhorar a qualidade do texto reconhecido. Essas técnicas incluem limiarização, suavização, binarização, segmentação e normalização (SETYAWAN; WIJAYA, 2019). A etapa seguinte é a extração de características, que envolve a identificação de padrões relevantes na imagem que possam ser utilizados para distinguir os caracteres. Entre as técnicas de extração de características mais utilizadas estão a análise de Fourier, transformada de Wavelet e Hough Transform (SHARMA; GUPTA; VOHRA, 2020).

Por fim, é realizado o reconhecimento óptico de caracteres propriamente dito, que consiste em classificar os caracteres extraídos por meio de um algoritmo de aprendizado de máquina. Os algoritmos mais comuns são o K-NN, SVM, Redes Neurais e Árvores de Decisão (SHARMA; GUPTA; VOHRA, 2020). Vale ressaltar que a precisão do reconhecimento óptico de caracteres depende da qualidade da imagem e do conjunto de treinamento utilizado para treinar o algoritmo.

Em resumo, o reconhecimento óptico de caracteres é uma técnica importante para converter textos em imagens em um formato editável de texto. Seu sucesso depende de um conjunto de técnicas que envolvem desde a digitalização até o reconhecimento propriamente dito por meio de algoritmos de aprendizado de máquina.

Na pesquisa em questão, a ferramenta de OCR utilizada será o *software* Tesseract, que utiliza uma rede neural LSTM (Long Short-Term Memory) focada em reconhecimento de linhas.

2.6 Análise de Desempenho dos Modelos

2.6.1 Métricas estatísticas

Para avaliar o desempenho dos modelos testados, foram utilizadas algumas métricas estatísticas que serão melhor descritas a seguir, sendo elas:

- Precisão;
- Sensibilidade;
- F1-score;
- Suporte;
- Taxa de acerto.

Os conceitos de precisão, recall, f1-score e suporte são amplamente utilizados em avaliação de modelos de classificação e recuperação de informações (POWERS, 2020).

A precisão (*precision*) é definida como a proporção de instâncias classificadas como positivas que são realmente positivas em relação ao total de instâncias classificadas como positivas (POWERS, 2020). Matematicamente, pode ser expressa conforme Equação 3.

$$precisão = \frac{verdadeiros\ positivos}{verdadeiros\ positivoss + falsos\ positivos} \tag{3}$$

onde $verdadeiros\ positivos$ são as instâncias positivas corretamente classificadas e $falsos\ positivos$ são as instâncias negativas incorretamente classificadas como positivas.

A sensibilidade (*recall*) é definida como a proporção de instâncias positivas que foram corretamente identificadas como positivas em relação ao total de instâncias positivas (POWERS, 2020). Matematicamente, pode ser expressa conforme Equação 4.

$$Sensibilidade = \frac{Verdadeiros\ positivos}{Verdadeiros\ positivos + Falsos\ negativos} \tag{4}$$

onde $falsos\ negativos$ são as instâncias positivas erroneamente classificadas como negativas.

O F1-score é uma medida que combina precisão e sensibilidade, dando um único número para avaliar o desempenho de um modelo (POWERS, 2020). É a média harmônica da precisão e da sensibilidade, dada pela Equação 5.

$$F1 = 2 \cdot \frac{precisão \cdot sensibilidade}{precisão + sensibilidade} \tag{5}$$

O suporte (*support*) é o número de ocorrências de cada classe no conjunto de dados (PEDREGOSA *et al.*, 2011).

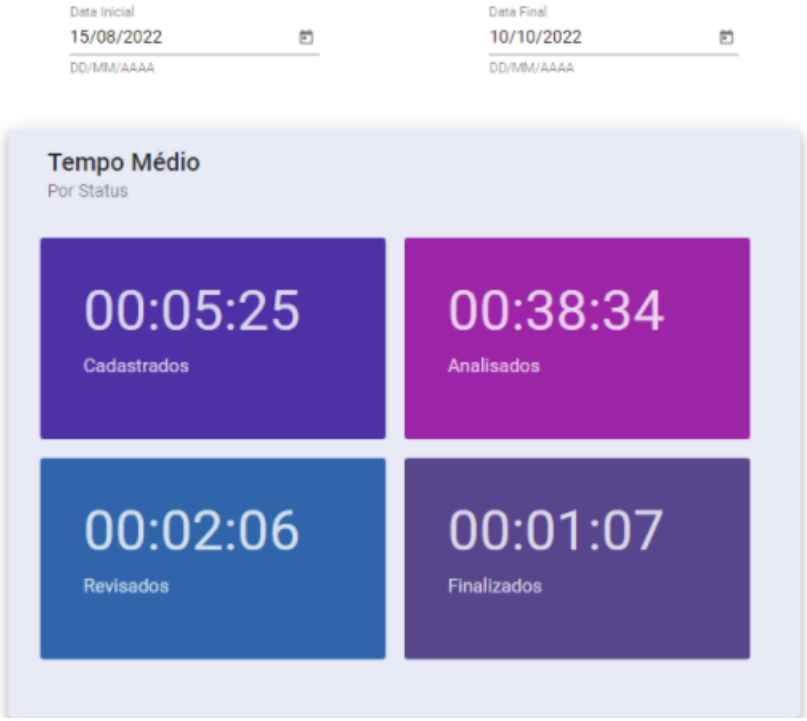
(GÉRON, 2019) define *Taxa de acerto* como a proporção de exemplos classificados corretamente em relação ao total de exemplos. É uma medida simples e intuitiva de avaliação de modelos de classificação binária, mas pode não ser adequada quando há classes desbalanceadas ou quando as classes têm custos diferentes de classificação incorreta. A Taxa de acerto pode ser calculada pela Equação 6.

$$\text{Taxa de acerto} = \frac{\text{verdadeiros pos.} + \text{verdadeiros neg.}}{\text{verdadeiros pos.} + \text{verdadeiros neg.} + \text{falsos pos.} + \text{falsos neg.}} \tag{6}$$

2.6.2 Comparação com o tempo médio de cadastro sem preenchimentos automáticos

Uma das métricas de avaliação do desempenho da ferramenta é a comparação com o tempo médio levado por um operador para fazer o preenchimento manual do cadastro. Na Figura 6 está apresentado o tempo médio calculado no período de 15/08/2022 a 10/10/2022 para executar verificação metrológica completa de um medidor.

Figura 6 – Tempo médio empregado em cada etapa do fluxo verificação.



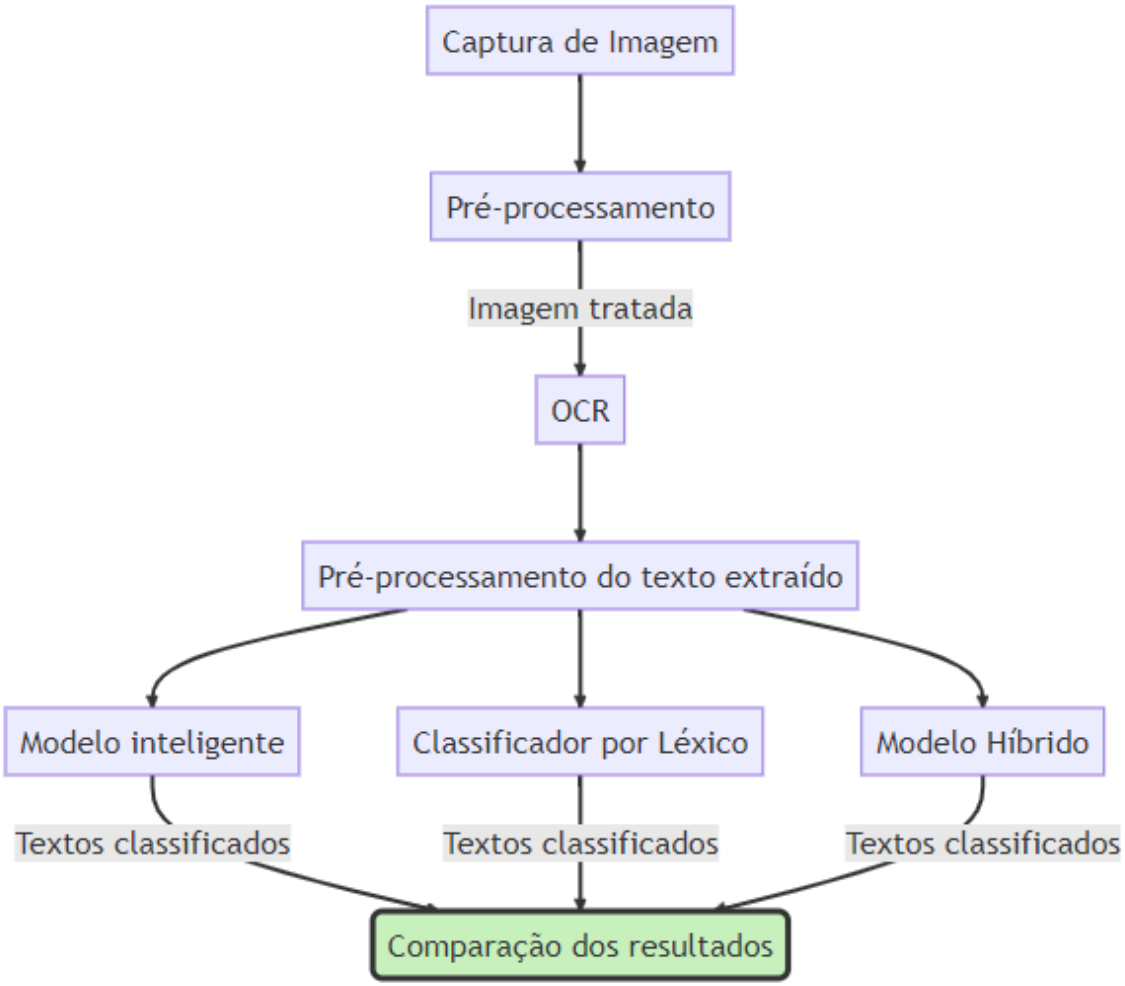
Fonte: O autor, 2023.

Destaca-se o tempo médio empregado no cadastro, que foi de 5 minutos e 25 segundos. Esta é a etapa inicial do procedimento e é a segunda que mais demanda tempo do operador. Considerando que este processo se repete para 3 mil medidores todo mês, serão feitos cálculos para obter métricas quantitativas de redução de custo anual que darão dimensão dos ganhos proporcionados ao laboratório.

3 Metodologia

Neste capítulo, são estudadas as soluções testadas para o problema de extração de dados em texto de uma fotografia de medidor de energia elétrica. Estas estão apresentadas, em resumo, na Figura 7.

Figura 7 – Fluxo de funcionamento da metodologia experimental.



Fonte: O autor, 2023.

Individualmente, as etapas que compõem o trabalho são:

- Captura de Imagem: este bloco trata sobre as condições de captura e coleta das imagens e é descrito em detalhe na Subseção 3.2.1;
- Pré-processamento: aqui é realizado o pré-processamento das imagens capturadas, com técnicas descritas em 3.2.1.1;

- OCR: este passo consiste em extrair os textos das imagens por meio de um *script* de OCR elaborado mediante uso do *software* Tesseract, descrito na Seção 3.4;
- Pré-processamento do texto extraído: os textos são, então, pré-processados para alimentar os modelos classificadores a fim de otimizar seus resultados, conforme técnicas descritas em 3.2.2.1;
- Modelo inteligente: esta é a etapa de testes dos modelos inteligentes para classificação de textos, descrita em maior detalhe nas Subseções 3.3.2 e 3.3.2.1;
- Classificador por léxico: este bloco corresponde à técnica clássica apresentada como alternativa aos modelos inteligentes e explicada em maior detalhe na Subseção 3.3.3;
- Híbrido: este bloco se refere ao modelo híbrido desenvolvido, que trata-se de uma abordagem onde um modelo inteligente é utilizado em série com o classificador por léxico, utilizando a classe da palavra mais próxima encontrada e a distância até esta como variáveis de entrada para classificar a palavra desejada. Este método é apresentado em detalhe na Subseção 3.3.4.
- Comparação dos resultados: a etapa final da pesquisa corresponde à comparação dos resultados obtidos por cada modelo e uma discussão acerca de seus desempenhos, conforme apresentado no Capítulo 4.

3.1 Materiais e Ferramentas

3.1.1 Ferramentas Computacionais

Foi utilizado o *Python* para escrever e executar códigos. Para a manipulação das imagens utilizadas durante a pesquisa foi utilizada a biblioteca *Open CV*. Para maior manipulação dos dados e resultados obtidos, serão utilizadas as bibliotecas *Numpy*, *Pandas* e *Matplotlib*. Por fim, para reconhecimento óptico de caracteres e manipulação da ferramenta *Tesseract*, foi utilizada a biblioteca *pytesseract*.

A lista completa de ferramentas computacionais utilizadas está apresentada na Tabela 2, juntamente de suas respectivas versões.

Tabela 2 – Ferramentas Computacionais Utilizadas.

Ferramenta Computacional	**Versão**
Python	3.10.6
opencv-python	4.7.0.72
scikit-learn	1.2.1
pytesseract	0.3.10
numpy	1.23.2
pandas	1.5.3
matplotlib	3.5.3
Levenshtein	0.20.9
Tesseract	v5.3.0.20221222

Fonte - O autor, 2023.

3.1.2 *Hardware* Utilizado

Para execução dos testes, foi utilizado o seguinte *hardware*:

- CPU Intel(R) Core(TM) i5-10500 3.10 GHz
- Memória RAM de 8,00 GB
- GPU Intel UHD Graphics Family 128 MB

A câmera utilizada para fotografar os medidores assim que chegam ao laboratório é de modelo Logitech *StreamCam*, que possui resolução máxima de 1920x1080 e 60 quadros por segundo em MJPEG. Possui foco automático (10 cm até o infinito), com abertura focal de $\frac{f}{2.0}$ e comprimento focal ajustável de 3,7 mm. Suas dimensões com suporte de monitor são 85 x 58 x 48 mm.

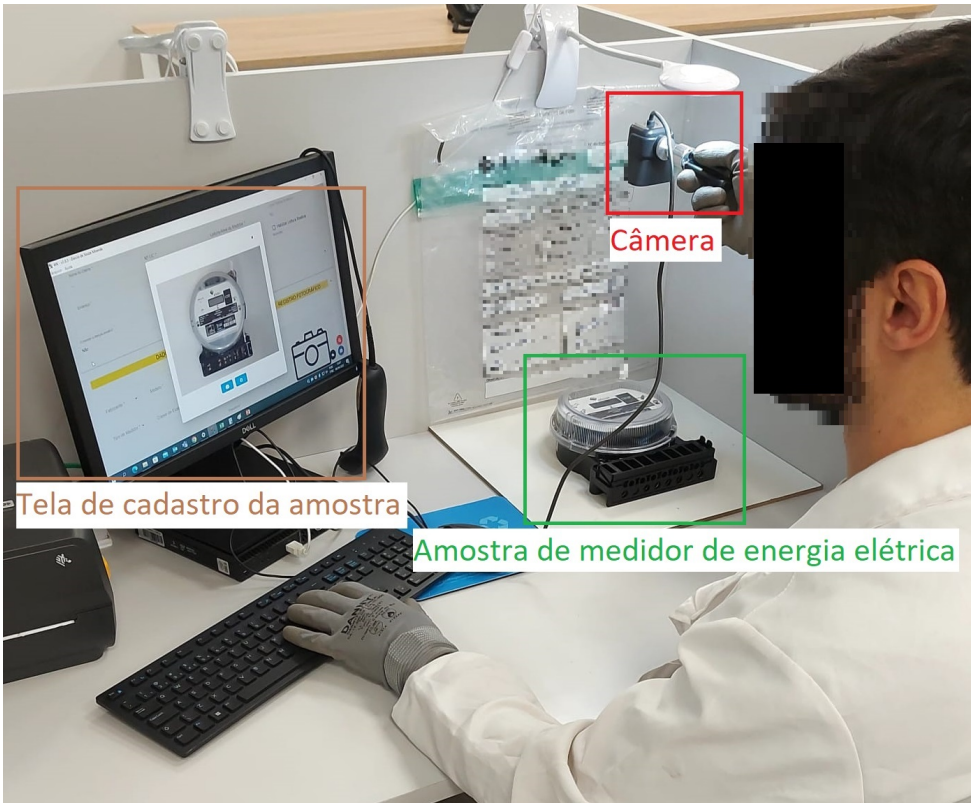
Apesar do potencial da câmera em termos de especificações, as fotografias obtidas do banco de imagens do laboratório contavam com resolução de 533x400, ou seja, abaixo da máxima suportada, e 96 dpi.

3.2 Estudo e Uso das Bases de Dados

3.2.1 Construção da Base de Dados de Imagens

Foi utilizado um banco de imagens de amostras de medidores de energia elétrica, fornecido por laboratório de Metrologia Legal. As imagens não foram submetidas a nenhum tipo de pré-processamento até o momento da execução deste projeto. A Figura 8 mostra a operação de cadastro sendo executada em laboratório.

Figura 8 – Técnico do laboratório fotografando uma amostra para cadastro em sua estação.



Fonte - O autor, 2023.

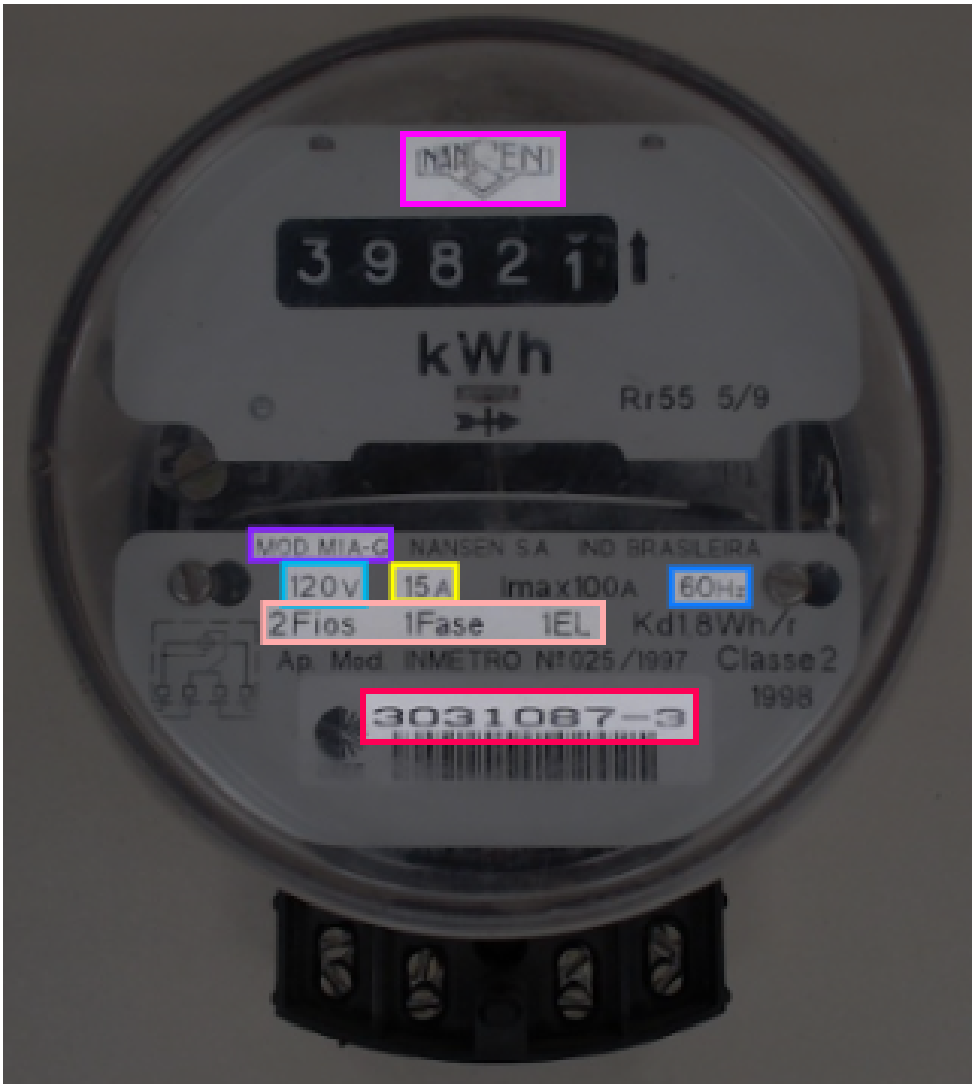
Em uma estação como esta, cerca de 3 mil medidores de energia elétrica são cadastrados por mês. O presente estudo contou com imagens de medidores cadastrados entre 17/08/2022 e 23/09/2022. As informações para cadastro dos medidores são lidas por cada operador e digitadas no sistema do laboratório (indicado na cor marrom), que se comunica com o site *Sharepoint*, que por sua vez armazena todos os cadastros em uma planilha. O técnico responsável pela captura fotográfica posiciona o medidor (indicado na cor verde) e seu envólucro na estação de trabalho de forma que possa ler com clareza os dados de cadastro. Então, tira uma fotografia frontal do instrumento, utilizando suas próprias mãos, sem ângulo padrão ou suporte para posicionar a câmera (indicada na cor vermelha). A única exigência feita pelo laboratório sobre as condições da fotografia é que esta capture a vista frontal do medidor inteiramente, ou seja, sem cortar nenhuma parte do instrumento na imagem. A Figura 9 ilustra uma amostra utilizada para testes nesta pesquisa.

Figura 9 – Exemplo de imagem presente no banco de dados.

Fonte - Banco de dados interno de laboratório privado, 2022.

A Figura 10 mostra a mesma imagem com alguns dos textos correspondentes aos campos de cadastros realçados.

Figura 10 – Sinalização dos textos que deseja-se extrair da imagem.



Fonte - O autor, 2023.

Destaca-se que a leitura inicial do medidor, denotada na imagem pelo número 39821, também é uma informação de interesse para os técnicos do laboratório. No entanto, para esta pesquisa foi necessário desconsiderar este campo para a tarefa de classificação dos textos, uma vez que é dificilmente diferenciável do número de patrimônio do instrumento.

Foram retiradas 409 fotografias de amostras de medidores de energia elétrica (residenciais) de 8 fabricantes diferentes. A distribuição de imagens de amostras por fabricante está apresentada na Tabela 3.

Tabela 3 – Distribuição da base de dados utilizada.

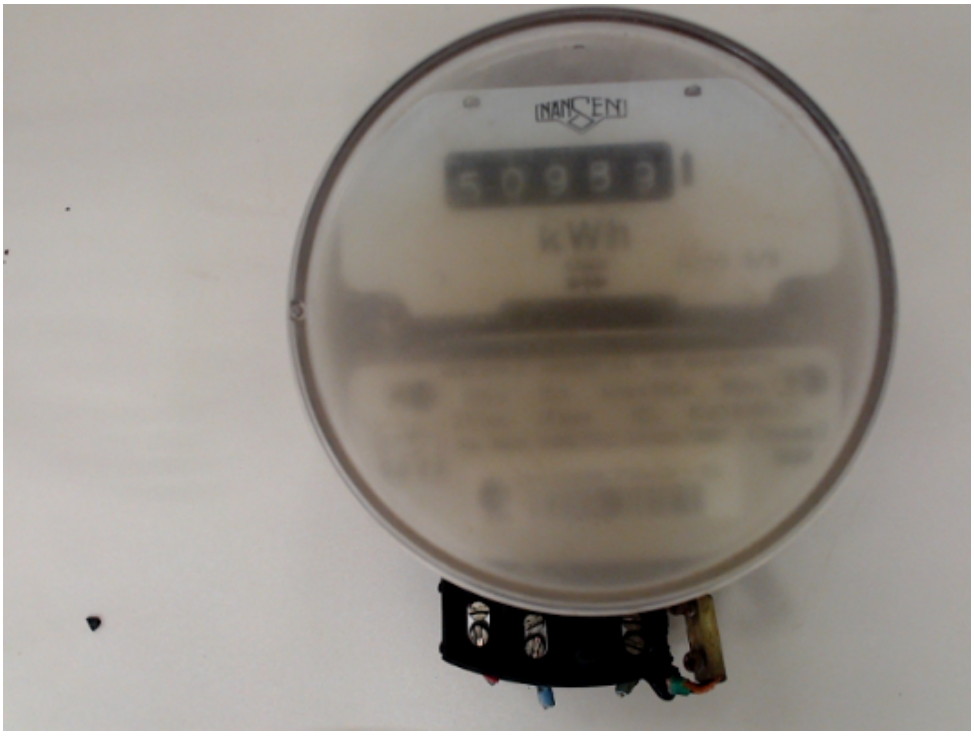
Fabricante	**Quantidade de imagens**
ABB	25
Aprel	29
Elster	40
Fae	102
GE	81
Landis	42
Nansen	64
Westinghouse	26

Fonte - O autor, 2023.

O rótulo tomado para avaliar a distribuição das imagens no banco de dados foi o de Fabricante, pois este é um dos principais dados a serem extraídos da amostra. A partir deste, o operador já reduz as possibilidades de modelos, por exemplo. Todas imagens foram submetidas ao *script* de OCR e os textos obtidos na saída do sistema foram rotulados para compor, assim, dados em texto extraídos das imagens para treinar os modelos utilizados.

Alguns dos medidores recebidos pelo laboratório já sofreram avarias ou estão com sua estrutura desgastada pelas condições de uso em que se encontravam antes de serem retirados para verificação metrológica. Estas condições adversas dificultam muito a extração de informações em texto, conforme exemplificado pela Figura 11.

Figura 11 – Exemplo de imagem com má legibilidade dos dados de identificação do instrumento.



Fonte - Banco de dados interno de laboratório privado, 2022.

Este tipo de cenário tornou necessário o uso de técnicas de pré-processamento das imagens, para que toda imagem fosse tratável pelo sistema, mesmo com poucos textos extraídos com sucesso.

3.2.1.1 Pré-processamento

Antes de alimentar o *script* de OCR, as imagens foram submetidas a pré-processamento, conforme as seguintes etapas:

- Padronização do tamanho das imagens em 1080x1200;
- Conversão das imagens para tons de cinza (*grayscale*);
- Limiarização das imagens;
- Abertura das imagens (dilação seguida de erosão).

As imagens foram redimensionadas para obter a saída com mais confiabilidade possível de acordo com os parâmetros da ferramenta *Tesseract* (SMITH; BEUSEKOM; CONTRIBUTORS, 2021). A conversão das imagens para tons de cinza é uma técnica amplamente utilizada para facilitar a detecção de bordas/objetos em imagens, que também é o objetivo da técnica de limiarização, onde se adota um limiar (nesta aplicação foi adotado o limiar de 128, numa escala de 0 a 255) e, após, percorre-se a imagem trocando os valores de intensidade de cada pixel maior que o limiar por 1 (branco) e menor que o limiar por 0 (preto). A limiarização de uma imagem serve para destacar suas características. Por fim, o processo de abertura (dilação seguida de erosão) é uma técnica amplamente utilizada para remover ruído e pequenos objetos da imagem preservando a forma e tamanho de objetos maiores.

3.2.2 Base de Dados de Textos da Carcaça dos Medidores

Também foi necessário manipular uma base de dados contendo os textos correspondentes aos cadastros já efetuados por operadores no período de operação do laboratório. Todos os textos receberam rótulos coerentes com seus respectivos campos de cadastro. Estavam disponíveis para análise nesta base de dados os cadastros de 20127 amostras. A Tabela 4 mostra um trecho deste banco de dados, onde podemos ver alguns campos de cadastro, que são variáveis de interesse.

Tabela 4 – Trecho do banco de dados em texto.

Patrimonio	Numero Serie	Tipo Medidor	Numero Elementos	Fabricante	Modelo
686498	1423786	Eletromecanico	1 elemento 2 fios	7	48
2330380	4191075	Eletromecanico	2 elementos 3 fios	1	2
38026985	-	Eletromecanico	2 elementos 3 fios	6	44
-	1739988	Eletromecanico	1 elemento 2 fios	7	49
46611522	-	Eletronico	3 elementos 4 fios	13	112
36012702	-	Eletromecanico	1 elemento 2 fios	6	46
1694637	4663005	Eletromecanico	1 elemento 2 fios	9	72
2404130	4310808	Eletromecanico	2 elementos 3 fios	1	2

Fonte - Banco de dados do laboratório, 2023.

É importante ressaltar que os textos não foram submetidos a nenhum tipo de pré-processamento até o momento da execução deste projeto. Além disso, destaca-se que as colunas "Fabricante"e "Modelo"foram estruturadas de acordo com identificadores numéricos unívocos atribuídos pela empresa. Foi necessário elaborar manualmente um dicionário em *Python* para converter os dados destas duas colunas para os textos correspondentes e gerar um novo *DataFrame*.

Esta base de dados original foi somada ao conjunto de dados obtidos a partir do OCR das imagens mencionadas em 3.2.1.

3.2.2.1 Pré-processamento

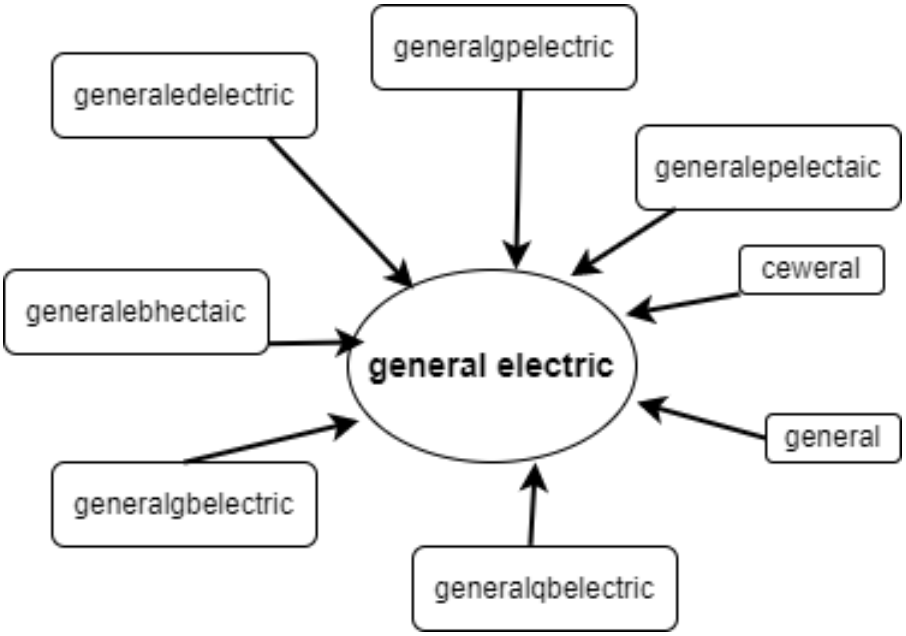
A etapa de pré-processamento do texto serve para prepará-lo para alimentar o classificador inteligente em condições que proporcionem a melhor saída possível. Em suma, foram seguidos os seguintes passos para preparar os textos da saída do OCR:

- Remover todo caractere que não números, letras ou espaço;
- Converter todas as letras maiúsculas para minúsculas;
- Remover acentos;
- Remover palavras contendo menos que 3 caracteres;
- Normalizar as palavras por meio de dicionário;
- Vetorizar as palavras.

Caracteres especiais foram removidos pois não fazem parte de nenhuma das palavras possíveis para preenchimento dos campos de cadastro. É necessário converter todas as letras maiúsculas para minúsculas porque esta conversão melhora a taxa de acerto e reduz a dimensionalidade em problemas de classificação textual (UYSAL; GUNAL, 2014). Este processo facilita a medida, por exemplo, da frequência que um texto é encontrado, pois

passamos a tratar ['Texto'], ['texto'] e ['TEXTO'] como palavras iguais. A normalização das palavras extraídas das imagens foi feita por meio de um dicionário em Python, elaborado manualmente pelo autor conforme ilustra a Figura 12.

Figura 12 – Ilustração da normalização das palavras.



Fonte: O autor, 2023.

Por fim, a vetorização de palavras se deu por meio do método *CountVectorizer*, transformando cada palavra no seu *token* correspondente. Este processo é importante pois é necessário que a entrada do modelo inteligente seja fornecida em formato numérico. Os parâmetros utilizados para este método estão listados na sequência:

- input: 'content';
- stop_words: None;
- ngram_range: (1, 1);
- analyzer: 'word';
- max_df: 1.0;
- min_df: 1;
- max_features: None.

Para ilustrar melhor o funcionamento do vetorizador, tomam-se os conjuntos de palavras abaixo elencados:

- "Refrigeradores Nansen 220V";
- "Medidores de gás Elster 10A";
- "Produtos elétricos Westinghouse 127V".

A partir desta lista de conjuntos de palavras, o vetorizador compõe uma lista com todas as palavras encontradas individualmente, por exemplo: ['10a', '127v', '220v', 'elétricos', 'elster', 'gás', 'medidores', 'nansen', 'produtos', 'refrigeradores', 'westinghouse'].

Por fim, o método percorre cada frase e monta o seu vetor correspondente baseado na comparação com a lista completa de palavras. Se for encontrada uma correspondência, o valor daquela posição no vetor sobe de 0 para 1. A lista inicial de conjuntos de palavras após vetorização ficaria assim:

- "0 0 1 0 0 0 0 1 0 1 0";
- "1 0 0 0 1 1 1 0 0 0 0";
- "0 1 0 1 0 0 0 0 1 0 1";

Nota-se que na terceira posição do primeiro vetor encontra-se um valor "1". Isto se dá porque na primeira frase há a palavra "220V", que ocupa a terceira posição na lista total de palavras. Da mesma forma, no segundo vetor temos o valor 1 na primeira posição e 0 na segunda, isto porque a primeira palavra da lista, "10a", consta na segunda frase, enquanto que a segunda palavra da lista, que é "127v", não.

3.3 Procedimento

O procedimento geral da metodologia da pesquisa é descrito em maior detalhe nesta seção.

3.3.1 Separação de Conjuntos de Treinamento, Validação e Teste

Para treinamento, validação e testes de todos os modelos testados nesta pesquisa, a base de dados foi separada em conjuntos de treinamento (80%) e teste (20%). Foram utilizadas as técnicas de otimização de hiper-parâmetros descritas na Subseção 3.3.2.1. Para o modelo clássico baseado em léxico, no entanto, o conjunto de teste foi utilizado para determinar a distância de Levenshtein ideal para qual se atribui uma palavra lida ao campo "Não identificado".

3.3.2 Abordagem utilizando Modelo Inteligente

Preliminarmente, foram testados 3 modelos diferentes como classificadores de texto, sendo eles:

- Máquina de vetores de suporte;
- Classificador Naive Bayes;
- Florestas aleatórias;

Nesta hipótese, o sistema de reconhecimento óptico de caracteres age nas imagens pré-processadas e retorna o maior número possível de dados extraídos da imagem em texto. Após, ocorre o pré-processamento destes dados extraídos, que posteriormente alimentam um classificador de textos que avalia os resultados e atribui a cada palavra um dos campos de cadastro contidos na Figura 2.

3.3.2.1 Otimização de hiper-parâmetros

A escolha dos hiper-parâmetros para cada um dos modelos inteligentes testados se deu por meio da técnica de otimização conhecida como busca aleatorizada. Em vez de testar todas as combinações possíveis de hiper-parâmetros, o que pode ser impraticável em modelos com muitos hiper-parâmetros, a busca aleatorizada seleciona um conjunto aleatório de valores para cada hiper-parâmetro e avalia o desempenho do modelo para cada combinação. O processo é repetido várias vezes, e a melhor combinação é selecionada com base no desempenho do modelo em um conjunto de dados de validação. Esta é uma técnica eficaz para otimização de modelos de aprendizado de máquina, pois permite explorar um espaço de hiper-parâmetros muito grande em um tempo razoável. Além disso, evita que o

modelo seja superajustado a um conjunto específico de hiper-parâmetros (*overfitting*), o que pode ocorrer com técnicas de busca exaustiva. Ademais, este método foi escolhido em vez do método de busca em grade ou busca exaustiva pelas seguintes vantagens encontradas:

- Nem todo hiper-parâmetro é igualmente crítico para otimização. A busca exaustiva aloca muitas tentativas para espaços exploratórios menos relevantes para o desempenho do modelo;
- Toda tentativa pode ser executada de forma assíncrona;
- Experimentos em grade com resolução adequada para otimização geralmente são proibitivamente mais custosos que experimentos aleatorizados (BERGSTRA; BENGIO, 2012).

Comparado com experimentos por busca em grade, a busca aleatorizada encontrou modelos melhores com menos custo computacional na maioria dos casos (LAROCHELLE *et al.*, 2007). Assim, a Tabela 5 mostra o espaço exploratório para testes com valores aleatórios para cada hiper-parâmetro de cada um dos modelos.

Tabela 5 – Busca aleatorizada pelo valor ótimo dos hiper-parâmetros para cada modelo.

Modelo	**Hiper-parâmetro**	**Intervalo de teste**
SVM	C	0,001:10000
	Gamma	0,00001:1
	Kernel	rbf, poly, linear
RF	Número de árvores	10:1000
	Características consideradas	auto,sqrt
	Profundidade máxima	10:110
	Número mínimo para dividir um nó	2, 5 e 10
	Número mínimo de amostras por nó	1, 2 e 4
NB	Alpha	0:2

Fonte - O autor, 2023.

Para treinamento e validação dos modelos obtidos, foi utilizada a técnica de validação cruzada chamada *k-fold*, que consiste em subdividir a base de dados em k subconjuntos (nesta pesquisa, foi tomado $k = 5$) mutuamente exclusivos e de mesma dimensão e tomar um destes para testes e os demais para treinamento do modelo. Este procedimento se repete k vezes e, ao final das iterações e de posse das taxas de acerto calculadas para cada rodada do experimento, a taxa de acerto é recalculada em função dos erros encontrados, proporcionando uma melhor noção da capacidade de generalização

do modelo, isto é, a capacidade de avaliar corretamente dados jamais vistos pelo modelo na fase de treinamento.

3.3.3 Método Baseado em Léxico

No intuito de avaliar a necessidade (ou não) do emprego de aprendizagem de máquina para solução do problema em questão, foi testada uma abordagem alternativa utilizando apenas o método matemático conhecido como distância de Levenshtein.

O algoritmo de distância de Levenshtein utiliza programação dinâmica *bottom-up*, que funciona envolvendo uma matriz contendo números que são os custos necessários para converter uma *string* em outra *string*. Este algoritmo retorna um número que é o custo mínimo necessário para converter a *string* inicial em uma *string* de destino que conhecemos junto com a distância Levenshtein (PORTER, 1980). Quanto mais baixo for este custo de conversão, mais parecidas são as palavras, e é por isto que este algoritmo é uma boa escolha como métrica de similaridade entre palavras.

O algoritmo utiliza, principalmente, 3 operações (YAHYA *et al.*, 2022):

- Inserção;
- Remoção;
- Substituição.

A quantidade de vezes que estas operações precisam ser executadas para que uma palavra se torne igual à palavra destino, indica a distância de Levenshtein. A Tabela 6 exemplifica o funcionamento desta técnica.

Tabela 6 – Exemplificação da aplicação da distância de Levenshtein.

Palavra Alvo	**Palavra Origem**	**Distância de Levenshtein**
medidor	medidor	0
medidor	m3didor	1
medidor	madid0rr	3

Fonte - O autor, 2023.

Neste exemplo, estamos comparando 3 palavras diferentes com a palavra meta "medidor". Pode-se perceber que a distância de Levenshtein é nula para palavras iguais. No caso abordado na terceira linha da tabela, foi necessária a substituição de dois caracteres e remoção de um, totalizando 3 operações para transformar a palavra origem na palavra alvo. Portanto, a distância calculada entre estas palavras é 3.

Como serão extraídos todos os textos da imagem e não apenas os 10 correspondentes aos campos de cadastro, foi necessário estabelecer um limiar para considerar a palavra como "Não identificada"quando esta não corresponde a nenhum campo válido. Este limiar foi calculado a partir da variação da taxa de acerto do classificador baseado em léxico em função do aumento da distância de Levenshtein considerada como limite para uma palavra não ser classificada como "Não identificado".

3.3.4 Abordagem Híbrida

Nesta pesquisa, foi contemplada a possibilidade de que os resultados encontrados indicassem que a tokenização de palavras pelo método *CountVectorizer* fosse inadequada para uma aplicação que busca classificar palavras isoladas, assim como poderia ser insuficiente o critério empírico adotado para definir a distância de Levenshtein para a qual uma palavra deve ser considerada como campo "Não identificado", pois algumas classes precisam de mais operações (inserção, remoção e substituição) para chegar na palavra correta e outras classes precisam de menos operações pelo padrão observado nos testes executados.

Dessa forma, testou-se uma hipótese final, onde um novo modelo SVM foi treinado para classificar as palavras a partir do índice da classe da palavra mais próxima encontrada pelo algoritmo de Levenshtein e da distância desta até a palavra que se deseja classificar. Foi montado então um banco de dados adaptado, contendo as duas variáveis de entrada e o respectivo rótulo de saída. Esta abordagem permite contornar o problema de rotular uma palavra válida como "Não identificado"em função da tolerância calculada para a distância de Levenshtein, ou seja, o modelo vai manter os rótulos atribuídos corretamente pelo modelo baseado em léxico e vai corrigir aqueles que foram atribuídos erroneamente em função do critério de tolerância. A Tabela 7 mostra um trecho do novo banco de dados elaborado para a abordagem híbrida, hipótese final da pesquisa.

Tabela 7 – Novo banco de dados para o modelo final.

Índice	**Distância de Levenshtein**	**Rótulo**
0	0	Fabricante
7	3	Nao identificado
7	6	Patrimonio
0	0	Fabricante
1	2	Nao identificado
4	2	Corrente

Fonte - O autor, 2023.

3.4 Tesseract

A ferramenta de OCR utilizada foi o *software* de código aberto *Tesseract*, programado principalmente em linguagem C++. Para aplicar as funcionalidades deste motor, foi utilizada a biblioteca *pytesseract*, que utiliza linguagem *Python* e serve para manipular e trazer mais funções ao *Tesseract*. O *Tesseract* foi adotado em detrimento do Keras OCR e Easy OCR, outras ferramentas populares no mercado, por apresentar as seguintes vantagens:

- Precisão para linguagens técnicas e suporte para múltiplos idiomas, inclusive línguas de escrita complexa;
- Desempenho para grandes volumes de palavras, sendo capaz de processar notável quantidade de documentos em um curto espaço de tempo, graças ao seu modelo altamente eficiente;
- Capacidade de personalização, oferecendo versatilidade para muitos testes de metodologia de trabalho, principalmente por meio da biblioteca *pytesseract*.

4 Análise de Resultados

Neste capítulo serão expostos os resultados preliminares obtidos com *scripts* elaborados pelo autor utilizando as tecnologias elencadas no capítulo anterior. Ao final, serão colocadas análises pertinentes considerando a aplicação destes resultados e considerações sobre resultados esperados.

4.1 Pré-processamento

Nesta seção serão apresentados os resultados obtidos no uso das técnicas de pré-processamento das imagens utilizadas para alimentar o *script* de OCR e dos textos utilizados para alimentar os modelos classificadores de texto.

4.1.1 Imagens

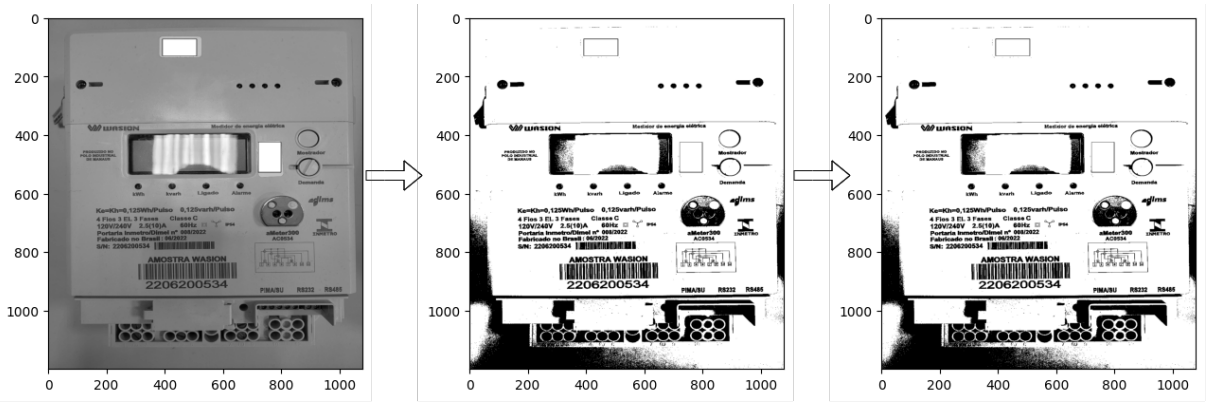
Para exemplificar as técnicas de pré-processamento de imagens empregadas neste projeto, foi escolhida a Figura 13.

Figura 13 – Imagem de amostra sem nenhum pré-processamento.

Fonte: O autor, 2023.

A Figura 14 mostra as etapas intermediárias de pré-processamento das imagens discriminadas, sendo a primeira a mudança nas dimensões da imagem e conversão para tons de cinza, a segunda corresponde à limiarização e a terceira é a etapa de dilação.

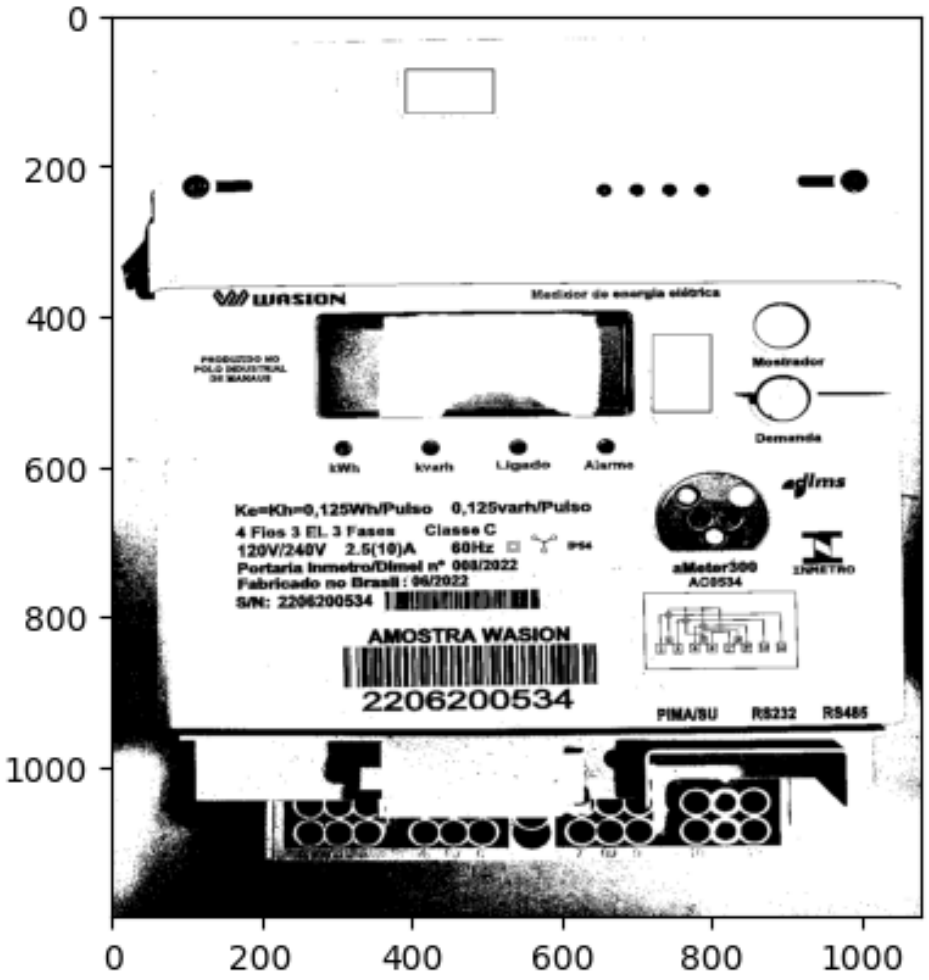
Figura 14 – Etapas de pré-processamento das imagens discriminadas.



Fonte: O autor, 2023.

A Figura 15 apresenta a imagem após aplicação das técnicas de pré-processamento, sendo a última a erosão da imagem.

Figura 15 – Imagem de amostra após emprego das técnicas de pré-processamento.



Fonte: O autor, 2023.

É possível observar que agora, até mesmo para o olho humano, as informações em texto presentes na carcaça do instrumento estão mais legíveis. Além disto, nota-se que a imagem está redimensionada conforme previsto para entrada no *script* de OCR. Outro ponto importante é que foi reduzida a dimensionalidade do problema ao utilizar a técnica de limiarização com limiar intermediário na escala de intensidade luminosa dos pixels. Por meio desta técnica, nossa imagem RGB inicial deixa de ser uma matriz tridimensional e passa a ser uma matriz bidimensional quadrada, o que se torna muito mais computacionalmente viável.

4.1.2 Textos

Para exemplificar a aplicação das técnicas de pré-processamento dos textos, foi utilizada a saída do OCR aplicado à imagem apresentada na Figura 15. A Tabela 8 mostra um trecho da saída antes e depois da aplicação das técnicas.

Tabela 8 – Exemplificação da aplicação das técnicas de pré-processamento dos textos.

Palavra antes do pré-processamento	**Palavra após pré-processamento**
o@ee88@	oee88
Ke=Kh=0,125WhiPulso	kekh0125whipulso
aMeter300	ameter300
WASION	wasion
elétrica	eletrica

Fonte - O autor, 2023.

É importante ressaltar que além das técnicas observadas na tabela, foi averiguado que as saídas do OCR que correspondiam a palavras vazias, espaços sozinhos ou palavras com menos de 3 caracteres foram removidas pelo filtro implementado.

4.2 Seleção do modelo inteligente

Foram testados diferentes modelos inteligentes para que fosse escolhido o mais apropriado para a aplicação de classificação de textos. Os resultados dos testes estão descritos nas subseções a seguir:

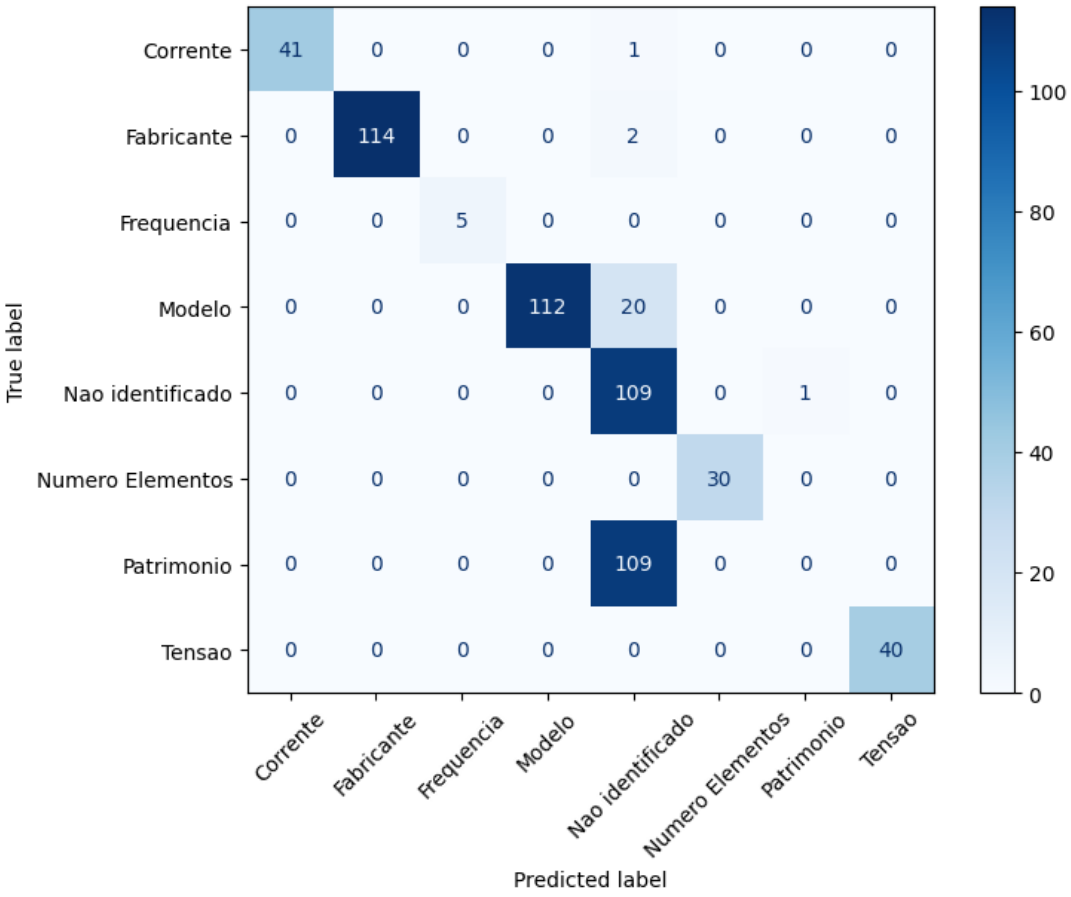
4.2.1 SVM

O resultado da otimização de hiper-parâmetros para a máquina de vetores de suporte trouxe os seguintes valores:

- C: 4,42;
- gamma: 0,64;
- kernel: rbf.

A Figura 16 mostra a matriz de confusão obtida com este modelo.

Figura 16 – Matriz de confusão para o modelo SVM.



Fonte: O autor, 2023.

A Tabela 9 apresenta detalhadamente o desempenho do classificador SVM para cada categoria, baseado em métricas estatísticas.

Tabela 9 – Desempenho do classificador SVM em cada classe.

Classe	Precisão	Sensibilidade	F1-score	Suporte
Corrente	1,00	0,98	0,99	42
Fabricante	1,00	0,98	0,99	116
Frequencia	1,00	1,00	1,00	5
Modelo	1,00	0,85	0,92	132
Nao identificado	0,45	0,99	0,62	110
Numero Elementos	1,00	1,00	1,00	30
Patrimonio	0,00	0,00	0,00	109
Tensao	1,00	1,00	1,00	40

Fonte - O autor, 2023.

É possível perceber que as classes que mais oferecem dificuldades ao modelo são "Patrimônio"e "Não identificado". Isto pode se explicar pela natureza unívoca do patrimônio de cada medidor, isto é, são números que jamais se repetem e não possuem similaridades

muito claras, tornando difícil traçar um hiperplano de decisão entre um número que de fato corresponde a um patrimônio e um número que é apenas uma saída do OCR correspondente a outro valor numérico encontrado na imagem.

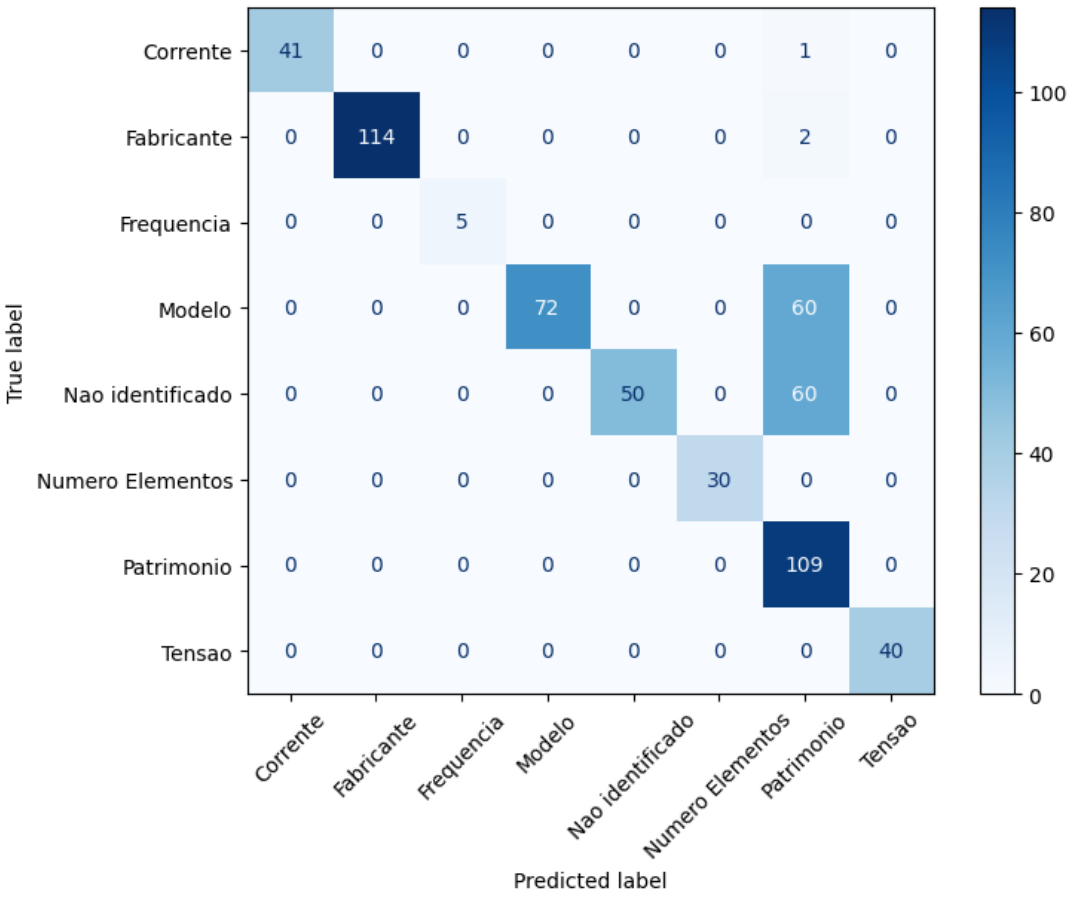
4.2.2 RF

O resultado da otimização de hiper-parâmetros para a abordagem por florestas aleatórias trouxe os seguintes valores:

- Profundidade máxima: 100;
- Características consideradas: auto;
- Número mínimo para dividir um nó: 5;
- Número mínimo de amostras por nó: 1;
- Número de árvores: 629.

A Figura 17 mostra a matriz de confusão obtida com este modelo.

Figura 17 – Matriz de confusão para o modelo RF.



Fonte: O autor, 2023.

A Tabela 10 apresenta detalhadamente o desempenho do classificador RF para cada categoria, baseado em métricas estatísticas.

Tabela 10 – Desempenho do classificador RF em cada classe.

Classe	Precisão	Sensibilidade	F1-score	Suporte
Corrente	1,00	0,98	0,99	42
Fabricante	1,00	0,98	0,99	116
Frequencia	1,00	1,00	1,00	5
Modelo	1,00	0,48	0,65	132
Nao identificado	1,00	0,44	0,61	110
Numero Elementos	1,00	1,00	1,00	30
Patrimonio	0,45	1,00	0,62	109
Tensao	1,00	1,00	1,00	40

Fonte - O autor, 2023.

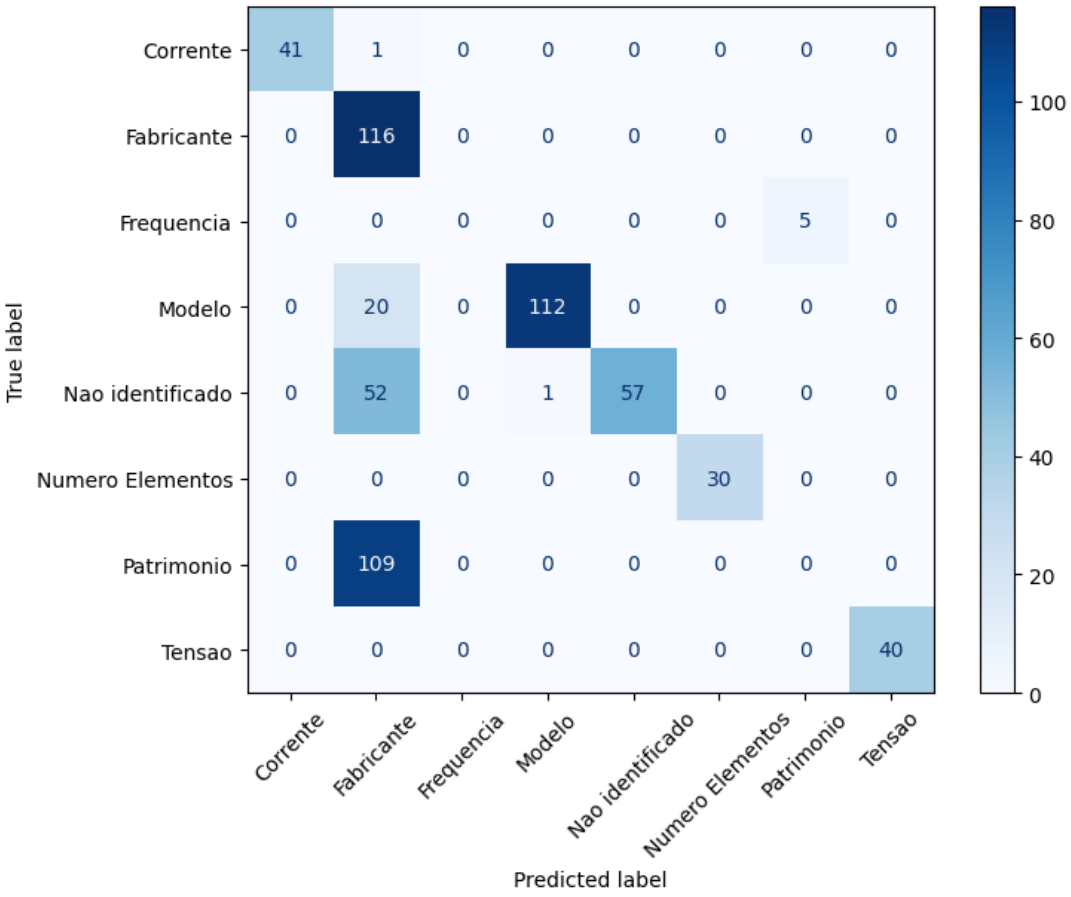
Mais uma vez, o modelo apresentou dificuldades na classificação do patrimônio, desta vez cometendo erros na classificação onde palavras que correspondiam a modelos ou textos não identificados foram erroneamente classificadas como patrimônios.

4.2.3 NB

O resultado da otimização de hiper-parâmetros para a abordagem por Naive Bayes trouxe o valor de $\alpha = 0,57$.

A Figura 18 mostra a matriz de confusão obtida com este modelo.

Figura 18 – Matriz de confusão para o modelo NB.



Fonte: O autor, 2023.

A Tabela 11 apresenta detalhadamente o desempenho do classificador Naive Bayes para cada categoria, baseado em métricas estatísticas.

Tabela 11 – Desempenho do classificador Naive Bayes em cada classe.

Classe	Precisão	Sensibilidade	F1-score	Suporte
Corrente	1,00	0,98	0,99	42
Fabricante	0,39	1,00	0,56	116
Frequencia	1,00	1,00	1,00	5
Modelo	0,99	0,85	0,91	132
Nao identificado	1,00	0,52	0,68	110
Numero Elementos	1,00	1,00	1,00	30
Patrimonio	0,00	0,00	0,00	109
Tensao	1,00	1,00	1,00	40

Fonte - O autor, 2023.

Desta vez, o modelo teve maior dificuldade na classificação de palavras que correspondiam a algum fabricante, atribuindo o rótulo corretamente em menos da metade

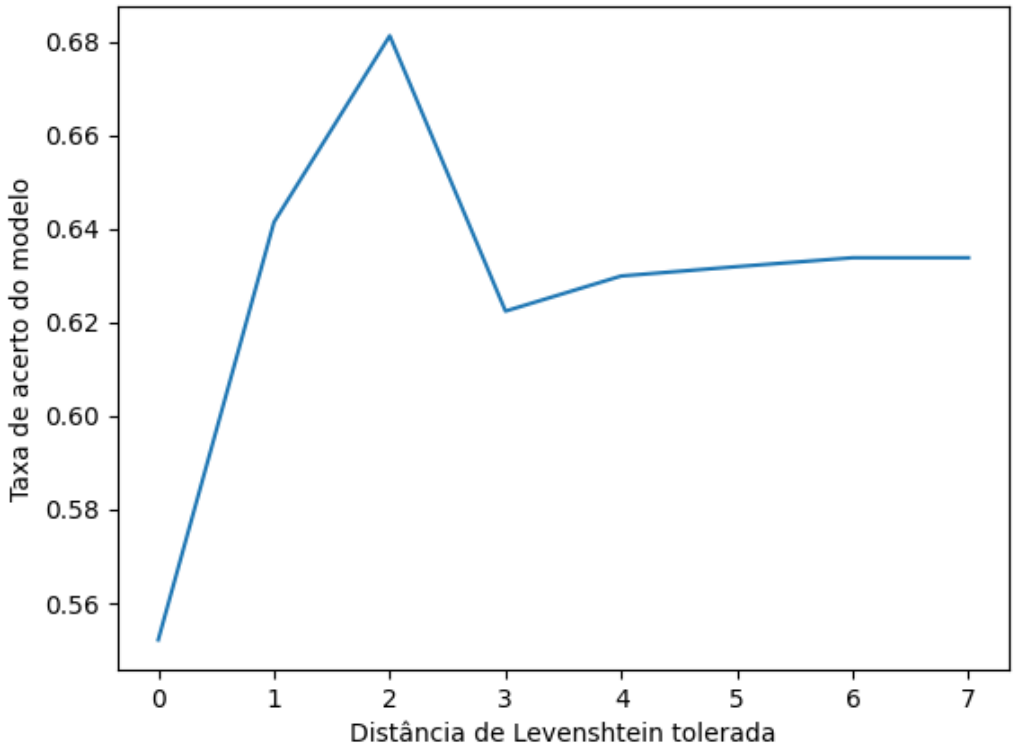
dos casos. Classificou como fabricante palavras que correspondiam na verdade a modelos, patrimônios e campos não identificados.

4.2.4 Classificador baseado em Léxico

A alternativa à abordagem por aprendizagem de máquina é resolver o problema utilizando apenas funções matemáticas para calcular a distância de *Levenshtein* entre a palavra obtida na etapa de OCR e o dicionário de palavras elaborado a partir do banco de dados em texto.

Uma espécie de "hiper-parâmetro"calculado para este modelo é a tolerância de distância para se classificar uma palavra com campo "Não identificado", pois muitos textos extraídos das imagens não correspondem necessariamente a algum dos campos de cadastro. Para determinar uma distância limite a partir da qual o uma palavra deveria ser considerada como não correspondente a nenhum dos campos, sem deixar de classificar corretamente palavras válidas, foi executada uma bateria de testes com diferentes distâncias. O resultado relacionado com o impacto na taxa de acerto do classificador baseado em léxico está apresentado na Figura 19.

Figura 19 – Variação da distância mínima tolerada em função da taxa de acerto na classificação de palavras.

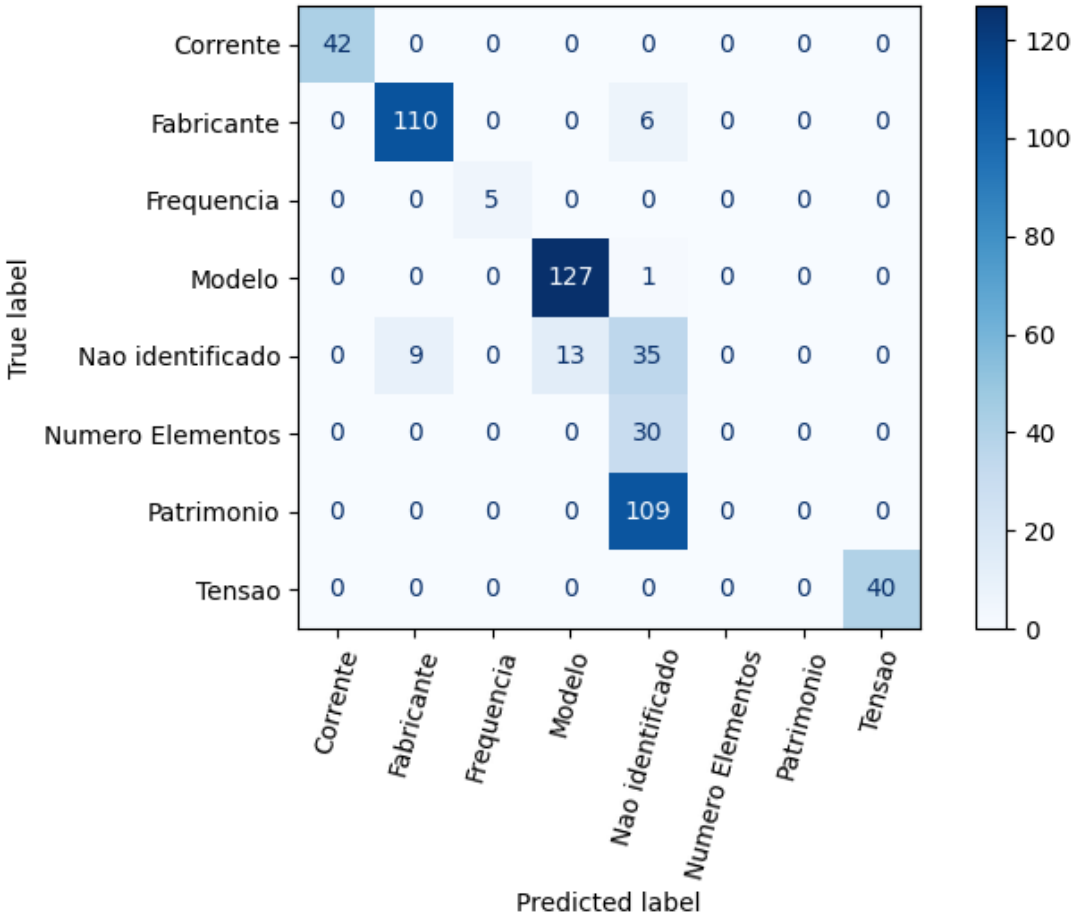


Fonte: O autor, 2023.

Calculando a influência da distância mínima tolerável para atribuir um campo válido a uma palavra para todo o conjunto de teste, verificou-se que a distância que proporciona a melhor taxa de acerto para o modelo é 2.

A matriz de confusão obtida para o modelo baseado em léxico, utilizando o mesmo conjunto de teste adotado para os outros modelos, está apresentada na Figura 20.

Figura 20 – Matriz de confusão para o modelo baseado em léxico.



Fonte: O autor, 2023.

A Tabela 12 apresenta detalhadamente o desempenho do classificador baseado em léxico para cada categoria, baseado em métricas estatísticas. O que se destaca negativamente nesta visualização dos resultados é a precisão na categoria "Não identificado", que alcançou apenas 23%. Isto indica que 77% das palavras que receberam o rótulo "Não identificado"pertenciam, na verdade, a outro campo de cadastro, o que é grave quando se considera a aplicação desta ferramenta na rotina do laboratório almejando o ganho de tempo do operador. Provavelmente grande parte destas palavras pertenciam à categoria "Patrimônio", que acabou não recebendo nenhuma classificação, uma vez que é uma classe que exige mais correções (operações da distância de Levenshtein) da palavra extraída do OCR para chegar a uma palavra semelhante, dado que é um número unívoco atribuído pela concessionária para identificação do instrumento.

Tabela 12 – Desempenho do classificador baseado em léxico em cada classe.

Classe	Precisão	Sensibilidade	F1-score	Suporte
Corrente	1.00	1.00	1.00	42
Fabricante	0.92	0.95	0.94	116
Frequencia	1.00	1.00	1.00	5
Modelo	0.91	0.98	0.94	128
Nao identificado	0.23	0.61	0.33	57
Numero Elementos	1.00	1.00	1.00	30
Patrimonio	0.00	0.00	0.00	109
Tensao	1.00	1.00	1.00	40

Fonte - O autor, 2023.

Esta abordagem se mostrou insuficiente para classificação das palavras que correspondiam ao campo "Não identificado", atribuindo erroneamente este rótulo a 109 palavras que na verdade correspondiam ao campo "Patrimônio"e 30 palavras que correspondiam ao campo "Número de elementos", por exemplo. Isto se deve ao critério adotado para classificar uma palavra como "Não identificado", que considera uma distância de Levenshtein atribuída de forma empírica baseada na relação desta com a taxa de acerto geral do modelo sem avaliar, portanto, as necessidades individuais de cada classe.

4.2.5 Comparação de desempenho e testes da Abordagem Clássica e da Abordagem com Modelos Inteligentes

Por fim, as informações descritas nas subseções anteriores foram reunidas e estão apresentadas em resumo na Tabela 13.

Tabela 13 – Desempenho dos diferentes classificadores de texto testados

Modelo testado	Taxa de acerto	Tempo de treinamento (s)
SVM	77,22%	389,26
NB	66,15%	0,39
RF	77,05%	105,45
Léxico	68,12%	-

Fonte - O autor, 2023.

É possível concluir a partir destes resultados que o modelo mais promissor para classificação dos textos é, mesmo com o tempo de treinamento mais extenso entre os modelos, o SVM, uma vez a etapa de treinamento tem execução única e desta forma seu tempo de duração é menos impactante no funcionamento diário da ferramenta se comparado com a taxa de acerto, que neste caso é o melhor critério para seleção do

algoritmo no qual se baseará o modelo inteligente pois indica menor necessidade de ajuste humano na hora de preencher os campos de cadastro.

Após os resultados obtidos nos testes, o modelo inteligente escolhido para classificar os textos fornecidos pela saída do *script* de OCR foi o SVM, resultado análogo ao obtido por (CâNDIDO, 2020), que indica que, para conjuntos de dados menores, o método mais tradicional e barato (máquina de vetores de suporte) está entre os melhores desempenhos no geral, superando significativamente abordagens neurais muito mais sofisticadas e caras quando o custo-benefício é considerado. Este modelo foi salvo em um arquivo *.p*, extensão criada por meio do uso da biblioteca *pickle* para converter o modelo para uma representação em *bytes* e facilitar a transferência e uso por outros programas. Assim, o modelo foi colocado em série com a rotina de efetua o reconhecimento óptico de caracteres. A Tabela 14 mostra um exemplo de aplicação do sistema desenvolvido em operação.

Tabela 14 – Exemplo do modelo SVM em operação

Imagem	Texto extraído	Campo atribuído
	oee88	Nao identificado
	medidor	Nao identificado
	enargia	Nao identificado
	eletrica	Nao identificado
	wasion	Fabricante
	produzido	Nao identificado
	polo	Nao identificado
	industiual	Nao identificado
	mostrador	Nao identificado
	manaus	Nao identificado
	demanda	Nao identificado
	kvarh	Nao identificado
	ligade	Nao identificado
	alarme	Nao identificado
	agims	Nao identificado
	kekh0125whipulso	Nao identificado
	0125varhpulso	Nao identificado
	classe	Nao identificado
	fios	Nao identificado
	fases	Nao identificado
	120v240v	Tensao
	2510a	Nao identificado
	60hz	Frequencia
	portaria	Nao identificado
	line	Nao identificado
	982022	Nao identificado
	ameter300	Modelo
	inmetro	Nao identificado
	fabricado	Nao identificado
	ac0534	Nao identificado
	sin	Nao identificado
	2206200834	Nao identificado
	wasion	Fabricante
	itera	Nao identificado
	2206200534	Nao identificado
	pimasu	Nao identificado
	rs232	Nao identificado
	rs485	Nao identificado
	weenie	Nao identificado

Fonte - O autor, 2023.

Inicialmente, uma observação importante a ser feita é que muitas das palavras extraídas, apesar de estarem corretas, não correspondem a campos presentes na tela de cadastro dos medidores. Por este motivo, a estas foi atribuído o rótulo "Não identificado". Destaca-se neste exemplo que foi possível extrair corretamente 3 dos 10 campos de cadastro:

Fabricante, Modelo e Frequência, em um tempo de execução de 2,32 segundos. Este valor está muito abaixo do tempo médio de 5 minutos e 25 segundos empregados por cadastro, principal métrica de comparação temporal adotada nesta pesquisa. Em outras palavras, foi possível extrair automaticamente 30% das informações de cadastro tomando um tempo correspondente a 0,714% da duração média de um cadastro.

Tabela 15 – Exemplo do modelo baseado em léxico em operação.

Imagem	Texto extraído	Campo atribuído
	oee88	Não identificado
	medidor	Não identificado
	enargia	Não identificado
	eletrica	Não identificado
	wasion	Fabricante
	produzido	Não identificado
	polo	Fabricante
	industiual	Não identificado
	mostrador	Não identificado
	manaus	Não identificado
	demanda	Não identificado
	kvarh	Não identificado
	ligade	Não identificado
	alarme	Não identificado
	agims	Não identificado
	kekh0125whipulso	Não identificado
	0125varhpulso	Não identificado
	classe	Não identificado
	fios	Não identificado
	fases	Não identificado
	120v240v	Tensao
	2510a	Modelo
	60hz	Frequencia
	portaria	Não identificado
	line	Não identificado
	982022	Não identificado
	ameter300	Modelo
	inmetro	Não identificado
	fabricado	Não identificado
	ac0534	Não identificado
	sin	Não identificado
	2206200834	Tensao
	wasion	Fabricante
	itera	Não identificado
	2206200534	Não identificado
	pimasu	Não identificado
	rs232	Não identificado
	rs485	Não identificado
	weenie	Não identificado

Fonte - O autor, 2023.

Observa-se que há mais de uma palavra com o rótulo de "Fabricante", por exemplo. Foi elaborada uma sub-rotina para tomar a palavra com a menor distância de Levenshtein entre as palavras com o mesmo rótulo. Todas as palavras entendidas pelo programa como "Modelo", por exemplo, são armazenadas pareadas com sua respectiva distância. Então, seleciona-se a menor distância e a função retorna a palavra correspondente e associa esta ao rótulo "Modelo".

Foi calculado o tempo médio de execução dos códigos para cada implementação ao longo de 111 imagens. A Tabela 16 ilustra os resultados obtidos neste experimento:

Tabela 16 – Desempenho dos diferentes classificadores de texto testados.

Método	**Tempo de OCR (s)**	**Tempo de predições (s)**	**Tempo total (s)**
SVM	0,414	0,041	0,455
NB	0,414	0,039	0,453
RF	0,414	0,062	0,476
Léxico	0,414	0,045	0,459

Fonte - O autor, 2023.

Observa-se que o tempo de execução médio calculado para uma série de imagens não indica uma escolha evidente entre os dois modelos. O tempo médio calculado sofre influência de alguns valores atípicos ou *outliers*, que são na verdade imagens que obtiveram pouquíssimo sucesso no reconhecimento óptico de caracteres, como no caso do exemplo apresentado anteriormente na Figura 11. Uma imagem neste nível de má legibilidade não permite que o Tesseract extraia muitos textos, logo, o tempo de execução do programa é extremamente menor, pois são analisadas pouquíssimas palavras, quiçá nenhuma.

De forma geral, apesar de oferecerem classificações corretas em alguns casos, tanto a abordagem clássica quanto a que utiliza *machine learning* tiveram taxas de acerto consideradas baixas para aplicação em campo do sistema, isto é, para aplicação no contexto do laboratório. Na abordagem utilizando apenas o modelo inteligente, encontrou-se uma limitação ao passar o resultado do método *CountVectorizer* diretamente como entrada para este, pois o método é mais adequado para classificação de documentos de texto e não é satisfatório para classificar palavras individualmente. Da mesma forma, na abordagem baseada em léxico foi encontrada uma vulnerabilidade ao determinar uma distância de Levenshtein absoluta como critério para atribuir o rótulo de "Não identificado", pois foi observado ao longo dos experimentos que algumas classes precisam de mais correções e outras precisam de menos. Ou seja, uma distância de Levenshtein maior ou menor não garantiria maior taxa de acerto ao modelo como um todo, mas sim para algumas classes de acordo com suas individualidades e, ainda assim, poderia atribuir o rótulo "Não identificado"à uma palavra que correspondia sim a algum campo de cadastro, mas estava apenas sujeita a ruídos no processo de OCR.

4.3 Discussões acerca da Abordagem Clássica comparada à Abordagem com Modelos Inteligentes

Em pesquisa com objetivo similar, (AMALIA *et al.*, 2018) obtiveram um sistema classificador de imagens com base nos textos extraídos destas. Pretendia-se determinar se um meme trazia sentimentos positivos ou negativos baseado nos textos obtidos após submeter a imagem ao reconhecimento óptico de caracteres. O modelo inteligente adotado foi um classificador baseado no algoritmo de Naive Bayes. Foi obtido um modelo com taxa de acerto de 75%, sendo que a base de dados continha apenas 100 amostras e foi dividida entre 60% reservado para treinamento e 40% separado para teste. Em comparação com o desenvolvido pelo autor nesta pesquisa, pode-se considerar que a base de dados limitada em quantidade realmente prejudica o desempenho dos modelos inteligentes, enquanto que o método Naive Bayes foi mais assertivo no caso de (AMALIA *et al.*, 2018) por tratar-se de uma classificação baseada em um conjunto de palavras, não de uma palavra em específico. Outra diferença importante está no número de classes, onde no caso do autor haviam 8 classes enquanto que no caso de comparação haviam apenas duas classes diferenciáveis.

Outro projeto semelhante buscou determinar se imagens são promocionais ou não, isto é, buscou identificar imagens que correspondiam à anúncios promocionais, baseando-se nos textos extraídos destas imagens (HUBERT *et al.*, 2021b). O resultado obtido indicou que usando o Reconhecimento Óptico de Caracteres (OCR) e o Algoritmo Naive Bayes é possível criar um sistema que pode classificar automaticamente se uma imagem contém uma oferta promocional ou não. O algoritmo classificador Naive Bayes foi a melhor escolha a ser usada para o sistema em questão quando comparado com o Random Forests, apresentando 94,31% de taxa de acerto. Para alcançar esta taxa, foram utilizados métodos de pré-processamento como a aplicação *stemming*, remoção de caracteres específicos e outros processos menores. É possível observar semelhanças entre as técnicas de pré-processamento adotadas por (HUBERT *et al.*, 2021b) e as adotadas pelo autor, pois tratam-se de boas práticas no tratamento de textos. Também é interessante perceber que em ambos os casos desta comparação, buscava-se integrar o classificador de textos a uma outra aplicação.

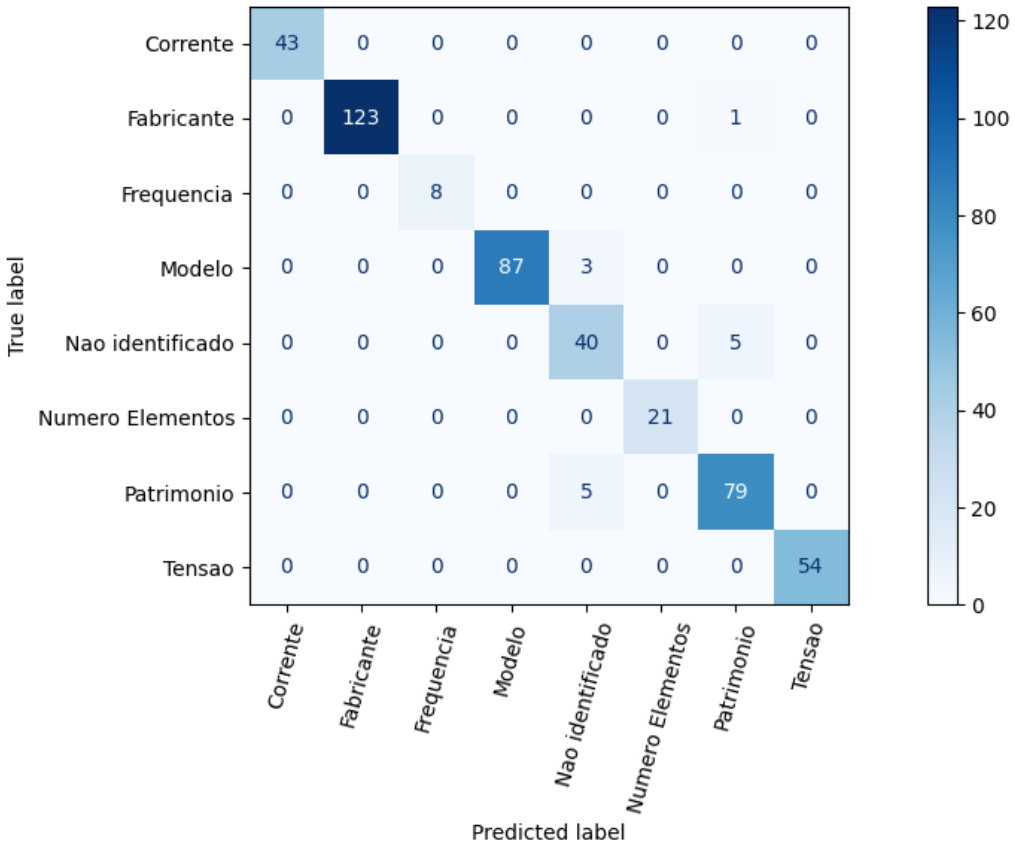
4.4 Implementação Híbrida: SVM + Levenshtein

O resultado da otimização de hiper-parâmetros para a máquina de vetores de suporte treinada para a abordagem híbrida trouxe os seguintes valores:

- C: 0,47;
- gamma: 0,82;
- kernel: rbf.

A Figura 21 mostra a matriz de confusão obtida com este modelo.

Figura 21 – Matriz de confusão para o modelo híbrido.



Fonte: O autor, 2023.

A Tabela 17 detalha o desempenho do modelo híbrido para cada classe separadamente, considerando os critérios estatísticos adotados pela pesquisa.

Tabela 17 – Desempenho do classificador híbrido em cada classe.

Classe	Precisão	Sensibilidade	F1-score	Suporte
Corrente	1,00	1,00	1,00	43
Fabricante	1,00	0,96	0,98	124
Frequencia	1,00	1,00	1,00	8
Modelo	1,00	0,98	0,99	90
Nao identificado	0,93	0,95	0,94	45
Numero Elementos	1,00	1,00	1,00	21
Patrimonio	0,94	0,99	0,96	84
Tensao	1,00	1,00	1,00	54

Fonte - O autor, 2023.

Destaca-se que para todas as classes, todas as métricas apresentaram resultados acima de 0,9. Ao observar a coluna referente à precisão, com exceção das classes "Não identificado"e "Número de Elementos", toda vez que o modelo atribuiu uma palavra a uma classe, a rotulação foi feita corretamente, ou seja, sempre que o modelo fez uma predição em qualquer uma das outras 6 classes, a predição estava correta. Agora, se for observada a sensibilidade, nota-se que o modelo híbrido deixou de atribuir algumas palavras que pertenciam às classes "Fabricante"e "Modelo"aos seus devidos campos, ou seja, nem toda amostra pertencente a estes grupos recebeu seu rótulo corretamente. De uma forma geral, os números indicam desempenho superior às abordagens descritas nas seções anteriores deste capítulo e

Os resultados obtidos para o modelo híbrido desenvolvido apresentam grande potencial para classificar corretamente as palavras extraídas das imagens de medidores de energia elétrica por meio de OCR, utilizando apenas a distância de Levenshtein calculada até a palavra mais próxima encontrada e o índice desta no dicionário elaborado pelo autor. A taxa de acerto obtida para este modelo foi de 97,01%. Como já era esperado pelos resultados observados nos testes das outras técnicas, as classes que mais ofereceram problemas foram "Patrimônio"e "Não identificado", pelo seu grau de aleatoriedade maior que o dos demais campos.

A Tabela 18 apresenta uma comparação de desempenho entre o modelo inteligente atuando sozinho, o classificador por léxico também atuando por conta e, por fim, a abordagem híbrida que utiliza um modelo inteligente para classificar as palavras baseado nas informações obtidas a partir dos resultados do método baseado em léxico.

Tabela 18 – Desempenho dos modelos inteligente, clássico e híbrido comparados.

Modelo testado	**Taxa de acerto**	**Tempo de treinamento (s)**
SVM	77,22%	389,26
Léxico	68,12%	-
Híbrido	97,01%	256,11

Fonte - O autor, 2023.

5 Conclusão

5.1 Considerações finais

A pesquisa culminou no desenvolvimento de três modelos finais para classificação de textos extraídos de imagens de medidores de energia elétrica. Estes modelos foram comparados em termos de desempenho e complexidade, a fim de determinar se uma abordagem utilizando *machine learning* era de fato necessária.

Após pesquisa e testes, concluiu-se que o modelo inteligente com melhor desempenho dentre as técnicas testadas foi a Máquina de Vetores de Suporte, com taxa de acerto de 77,22%. Esta taxa de acerto supera a obtida em alguns trabalhos similares como (AMALIA *et al.*, 2018), mas fica aquém dos resultados obtidos em (HUBERT *et al.*, 2021b), por exemplo, muito em função de pequenas divergências entre as aplicações em questão.

A alternativa testada para avaliar a dispensabilidade do emprego de um modelo inteligente foi a elaboração de um algoritmo que calcula a distância de Levenshtein entre as palavras saídas do OCR e a lista de palavras presentes no banco de dados de texto (Figura 4) e retorna a classe da palavra mais próxima. Foi possível aperfeiçoar o algoritmo para que se houvesse mais de uma palavra atribuída a uma determinada classe na mesma foto, apenas a palavra com a menor distância de Levenshtein fosse retornada. Também foi implementado um recurso para atribuir a classe "Não identificado"às palavras cuja distância superasse um limiar calculado de tolerância.

Comparando os resultados obtidos entre as duas implementações citadas neste capítulo até agora, inclinou-se à conclusão de que ambas eram insuficientes para solução do problema em questão, restando então a alternativa que aproveita as saídas do modelo baseado em Léxico e, por meio de um novo classificador SVM treinado, busca atribuir corretamente as palavras aos seus respectivos rótulos no campo de cadastro. O resultado final deste modelo apresentou 97,01% de taxa de acerto e, portanto, mostrou potencial para trazer resultados satisfatórios ao preenchimento automático do cadastro almejado no início deste projeto, principalmente pelo desempenho superior em comparação aos outros dois métodos testados neste estudo.

5.2 Trabalhos futuros

Os resultados obtidos nesta pesquisa indicam possibilidade de melhorias, especialmente se houver aprimoramentos na metodologia adotada para os modelos inteligentes. Uma possibilidade ventilada durante a pesquisa foi de testar outro método para *tokenização*

das palavras ao invés do *CountVectorizer*, de forma que seja possível converter as strings em número unívocos sem trabalhar com o número de aparições da palavra em um dado documento, recurso utilizado pelo *CountVectorizer* para classificar textos e não palavras individualmente.

Também notou-se que pode ser promissor investir em uma mudança de procedimento para captura das fotografias para que estas possuam melhor qualidade e, portanto, melhores resultados na extração de caracteres. Desta forma, estima-se que uma posição padrão para as fotografias e um suporte firme e fixo para a câmera fotográfica possam ser fatores contribuintes para melhores resultados em trabalhos futuros.

A pesquisa também mostra potencial de melhoria na parte de visão computacional, a começar pela definição de métricas de qualidade da imagem para avaliar as entradas oferecidas ao *Tesseract*. Após, também é válido aprofundar a pesquisa na área de OCR baseando-se nos métodos utilizados por medidores de velocidade, que capturam imagens de veículos automotores e extraem destas as respectivas placas de identificação destes, lidando com questões como iluminação e perspectiva da câmera.

Um projeto futuro também contempla a combinação do algoritmo classificador de texto com um modelo inteligente aplicado diretamente nas imagens, a fim de reconhecer na fotografia as regiões onde estão localizados os textos correspondentes a cada campo de cadastro. As coordenadas indicando a posição das informações em texto na imagem ajudarão no reconhecimento de padrões para classificação dos textos, bem como auxiliaria na solução do problema de diferenciar dois textos muito parecidos pertencentes a campos diferentes (patrimônio e leitura inicial podem ser diferenciados pela sua posição na carcaça do medidor, por exemplo).

Por fim, há a missão de integrar a melhor ferramenta dentre as desenvolvidas ao software de operações do laboratório, de forma a proporcionar o preenchimento automático dos campos numa operação de cadastro de fato. O desafio desta integração envolve mais conhecimentos de engenharia da computação e programação, pois trata-se de adicionar os *scripts* em Python ao *back-end* da aplicação já existente, que foi desenvolvida em TypeScript.

5.3 Limitações da Pesquisa

Os resultados obtidos neste trabalho estão sujeitos a alguns limitadores. Os fatores considerados como limitações para este estudo são apresentados a seguir:

- Estado da arte na área de classificação de textos é muito mais rico na rotulação de textos corridos do que na classificação de palavras isoladas, como ocorre no caso aqui estudado;

- Balanceamento das bases de dados sujeito ao mercado de medidores de energia elétrica (Fabricantes e modelos menos populares tendem a aparecer com menos frequência nas bases utilizadas);

- Para cumprimento dos Procedimentos da Qualidade do laboratório (STANDARDIZATION, 2017), especialmente quanto à confidencialidade, o *setup* de fotografia teve que ser mantido em seu ambiente de trabalho e quaisquer testes que envolvessem acesso às amostras de medidores de energia elétrica só poderiam ser feitos durante sua jornada de trabalho;

- O trabalho partiu da premissa de não alterar a forma de trabalho já existente no laboratório, apenas facilitá-la. Portanto, a condição de captura das imagens e o procedimento geral de cadastro deviam ser mantidos;

- Uma das maiores limitações na extração de caracteres das imagens se deu pela qualidade das imagens contidas no banco de dados, bem como da falta de uma posição padrão para captura das fotografias;

Referências Bibliográficas

ABBASI, W. A.; ASLAM, M. Performance comparison of ocr systems on different platforms. *International Journal of Computer Science and Network Security*, v. 17, n. 8, p. 72–77, 2017.

AMALIA, A.; SHARIF, A.; HAISAR, F.; GUNAWAN, D.; NASUTION, B. B. Meme opinion categorization by using optical character recognition (OCR) and naïve bayes algorithm. p. 1–5, 2018.

BANERJEE, A.; MUKHOPADHYAY, S.; MUKHERJEE, A.; MITRA, P. Random forests for the classification of hyperspectral images: A review. *Journal of Applied Remote Sensing*, International Society for Optics and Photonics, v. 14, n. 1, p. 012204, 2020.

BERGSTRA, J.; BENGIO, Y. Random search for hyper-parameter optimization. *Journal of machine learning research*, v. 13, n. 2, 2012.

BREIMAN, L. Random forests. *Machine learning*, Springer, v. 45, n. 1, p. 5–32, 2001.

BUI, D. C.; TRUONG, D.; VO, N. D.; NGUYEN, K. Mc-ocr challenge 2021: Deep learning approach for vietnamese receipts ocr. In: IEEE. *2021 RIVF International Conference on Computing and Communication Technologies (RIVF)*. [S.l.], 2021. p. 1–6.

BURGES, C. J. A tutorial on support vector machines for pattern recognition. *Data mining and knowledge discovery*, Springer, v. 2, n. 2, p. 121–167, 1998.

CERQUEIRA, F. S.; OLIVEIRA, D. N.; JÚNIOR, S. W.; HONORIO, K. M. Random forest algorithm for feature selection in qspr models: A comparative study with linear regression models. *Computational and Theoretical Chemistry*, Elsevier, v. 1161, p. 112848, 2020.

CHEN, M. K.; LIU, X.; SUN, Y.; TSAI, D. P. Artificial intelligence in meta-optics. *Chemical Reviews*, ACS Publications, 2022.

CHEN, W.; WANG, Y.; YANG, S. Natural language processing: A systematic review. *International Journal of Computational Linguistics Research*, Science Publishing Group, v. 10, n. 1, p. 1–18, 2019.

CHEN, Y.; LIU, Z.; ZHANG, J. Product classification using attention-based recurrent neural network. *IEEE Access*, IEEE, v. 8, p. 52610–52619, 2020.

CHRISTENSSON. *OCR definition*. 2018. Disponível em: <https://techterms.com/definition/ocr>.

CORTES, C.; VAPNIK, V. Support-vector networks. *Machine Learning*, Springer, v. 20, n. 3, p. 273–297, 1995.

CRISTIANINI, N.; SHAWE-TAYLOR, J. *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge: Cambridge University Press, 2000.

CâNDIDO, E. C. R. *Um estudo comparativo de redes neurais profundas para classificação automática de texto*. Dissertação (Dissertação de Mestrado) — Universidade Federal de Minas Gerais, Belo Horizonte, 2020.

GÉRON, A. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. [S.l.]: O'Reilly Media, Inc., 2019.

GONZÁLEZ, R. C.; WOODS, R. E. *Processamento de Imagens Digitais*. [S.l.]: Pearson, 2007.

HARIKA, J.; BALEESHWAR, P.; NAVYA, K.; SHANMUGASUNDARAM, H. A review on artificial intelligence with deep human reasoning. In: IEEE. *2022 International Conference on Applied Artificial Intelligence and Computing (ICAAIC)*. [S.l.], 2022. p. 81–84.

HARTMANN, J.; HUPPERTZ, J.; SCHAMP, C.; HEITMANN, M. Comparing automated text classification methods. *International Journal of Research in Marketing*, v. 36, n. 1, p. 20–38, 2019. ISSN 0167-8116. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0167811618300545>.

HO, T. K. Random decision forests. *Proceedings of 3rd International Conference on Document Analysis and Recognition*, p. 278–282, 1995.

HUBERT; PHOENIX, P.; SUDARYONO, R.; SUHARTONO, D. Classifying promotion images using optical character recognition and naïve bayes classifier. *Procedia Computer Science*, v. 179, p. 498–506, 2021. ISSN 1877-0509. 5th International Conference on Computer Science and Computational Intelligence 2020. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1877050921000387>.

HUBERT; PHOENIX, P.; SUDARYONO, R.; SUHARTONO, D. Classifying promotion images using optical character recognition and naïve bayes classifier. *Procedia Computer Science*, v. 179, p. 498–506, 2021. ISSN 1877-0509. 5th International Conference on Computer Science and Computational Intelligence 2020. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1877050921000387>.

INMETRO. *Metrologia Legal*. 2023. <https://www.inmetro.gov.br/metrologia-legal/>.

JAIN, A. K.; KASTURI, R.; SCHUNCK, B. G. *Machine Vision*. [S.l.]: McGraw-Hill, 1995.

JURAFSKY, D.; MARTIN, J. H. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. [S.l.]: Pearson Education, 2020.

KIM, J. H.; LEE, D. H. Text recognition from the web images using deep learning. *Multimedia Tools and Applications*, Springer, v. 77, n. 16, p. 21565–21587, 2018.

KUMAR, P.; KUMAR, V.; PRATAP, R. Prototyping and hardware-in-loop verification of OCR. *IET Generation, Transmission & Distribution*, Wiley Online Library, v. 12, n. 12, p. 2837–2845, 2018.

LAROCHELLE, H.; ERHAN, D.; COURVILLE, A.; BERGSTRA, J.; BENGIO, Y. An empirical evaluation of deep architectures on problems with many factors of variation. In: *Proceedings of the 24th international conference on Machine learning*. [S.l.: s.n.], 2007. p. 473–480.

LIU, X.-L.; WU, Y.-L.; CHEN, Y.-J. Random forest algorithm and its applications. *Journal of Mechanical Engineering Research and Developments*, v. 43, n. 1, p. 1–12, 2020.

MANNING, C. D.; SCHÜTZE, H. *Foundations of statistical natural language processing*. [S.l.]: MIT press, 1999.

PEDREGOSA, F.; VAROQUAUX, G.; GRAMFORT, A.; MICHEL, V.; THIRION, B.; GRISEL, O.; BLONDEL, M.; PRETTENHOFER, P.; WEISS, R.; DUBOURG, V.; VANDERPLAS, J.; PASSOS, A.; COURNAPEAU, D.; BRUCHER, M.; PERROT, M.; DUCHESNAY, Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, v. 12, p. 2825–2830, 2011.

PORTER, M. F. An algorithm for suffix stripping. *Program*, MCB UP Ltd, v. 14, n. 3, p. 130–137, 1980.

POWERS, D. M. Evaluation: From precision, recall and f-factor to roc, informedness, markedness & correlation. *Journal of Machine Learning Technologies*, v. 2, n. 1, p. 37–63, 2020.

RADWAN, M. A.; KHALIL, M. I.; ABBAS, H. M. Neural networks pipeline for offline machine printed arabic OCR. *Neural Processing Letters*, Springer, v. 48, n. 2, p. 769–787, 2018.

RISH, I. An empirical study of the naive bayes classifier. *ICML*, v. 1, p. 41–48, 2001.

SAMPAIO, F.; PINTO, M. A. C.; ASSIS, A.; RÉCHE, M. M. O papel da metrologia legal no inmetro como ferramenta de política industrial. In: *Resumos do V Congresso Brasileiro de Metrologia*. [S.l.: s.n.], 2009. p. 1–7.

SERRA, J. *Image Analysis and Mathematical Morphology*. [S.l.]: Academic Press, 1982.

SETYAWAN, I. B.; WIJAYA, A. Y. Preprocessing of image in character recognition system using morphological operations. *Journal of Physics: Conference Series*, v. 1214, n. 1, p. 012045, 2019.

SHALEV-SHWARTZ, S.; BEN-DAVID, S. Understanding machine learning: From theory to algorithms. In: *Foundations and Trends® in Machine Learning*. [S.l.]: Now Publishers Inc., 2014. v. 4, n. 2, p. 143–223.

SHARMA, A.; GUPTA, A.; VOHRA, R. Optical character recognition: Techniques and its application. *IETE Technical Review*, Taylor & Francis, v. 37, n. 1, p. 22–33, 2020.

SMITH, R.; BEUSEKOM, S. van; CONTRIBUTORS other. *Tesseract Documentation*. Mountain View, CA, 2021. Disponível em: <https://tesseract-ocr.github.io/tessdoc/Home.html>.

STANDARDIZATION, I. O. for. *ISO/IEC 17025:2017 General requirements for the competence of testing and calibration laboratories*. 2017. <https://www.iso.org/standard/66912.html>.

TURING, A. M. Computing machinery and intelligence. *Mind*, 1950.

UYSAL, A. K.; GUNAL, S. The impact of preprocessing on text classification. *Information processing & management*, Elsevier, v. 50, n. 1, p. 104–112, 2014.

VAPNIK, V. The nature of statistical learning theory. *Springer Science & Business Media*, New York, 1995.

YAHYA, M. I.; ARINI; AMRIZAL, V.; MATIN, I. M. M.; KHAIRANI, D. Spelling correction using the levenshtein distance and nazief and adriani algorithm for keyword search process indonesian qur'an translation. In: *2022 Seventh International Conference on Informatics and Computing (ICIC)*. [S.l.: s.n.], 2022. p. 01–06.

ZHANG, J.; LIU, Q.; CHEN, H.; ZHANG, D. A product classification method based on character-level convolutional neural network. *Journal of Computational Information Systems*, v. 13, n. 4, p. 1075–1081, 2017.