UNIVERSIDADE FEDERAL DE VIÇOSA
PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

YAGO JOSÉ ARAÚJO DOS SANTOS

AVALIAÇÃO DE TÉCNICAS DE RECONHECIMENTO ÓPTICO DE CARACTERES (OCR) PARA ANÁLISE DE DADOS EM PORTUGUÊS DISSEMINADOS EM PLATAFORMAS DE MÍDIAS SOCIAIS

VIÇOSA - MINAS GERAIS

2024

YAGO JOSÉ ARAÚJO DOS SANTOS

AVALIAÇÃO DE TÉCNICAS DE RECONHECIMENTO ÓPTICO DE CARACTERES (OCR) PARA ANÁLISE DE DADOS EM PORTUGUÊS DISSEMINADOS EM PLATAFORMAS DE MÍDIAS SOCIAIS

Dissertação apresentada à Universidade Federal de Viçosa, como parte das exigências do Programa de Pós-Graduação em Ciência da Computação, para obtenção do título de *Magister Scientiae*.

Orientador: Julio Cesar Soares dos Reis

Coorientador: Michel Melo da Silva

VIÇOSA - MINAS GERAIS
2024

Ficha catalográfica elaborada pela Biblioteca Central da Universidade Federal de Viçosa - Campus Viçosa

T

S237a
2024

Santos, Yago José Araújo dos, 1996-

Avaliação de técnicas de reconhecimento óptico de caracteres (OCR) para análise de dados em português disseminados em plataformas de mídias sociais / Yago José Araújo dos Santos. – Viçosa, MG, 2024.

1 dissertação eletrônica (86 f.): il. (algumas color.).

Inclui apêndices.

Orientador: Julio Cesar Soares dos Reis.

Dissertação (mestrado) - Universidade Federal de Viçosa, Departamento de Informática, 2024.

Referências bibliográficas: f. 73-79.

DOI: https://doi.org/10.47328/ufvbbt.2024.310

Modo de acesso: World Wide Web.

1. Redes sociais on-line. 2. Desinformação. 3. Análise de conteúdo (Comunicação). I. Reis, Julio Cesar Soares dos, 1988-. II. Universidade Federal de Viçosa. Departamento de Informática. Programa de Pós-Graduação em Ciência da Computação. III. Título.

CDD 23. ed. 004.678

Bibliotecário(a) responsável: Euzébio Luiz Pinto CRB-6/3317

YAGO JOSÉ ARAÚJO DOS SANTOS

AVALIAÇÃO DE TÉCNICAS DE RECONHECIMENTO ÓPTICO DE CARACTERES (OCR) PARA ANÁLISE DE DADOS EM PORTUGUÊS DISSEMINADOS EM PLATAFORMAS DE MÍDIAS SOCIAIS

Dissertação apresentada à Universidade Federal de Viçosa, como parte das exigências do Programa de Pós-Graduação em Ciência da Computação, para obtenção do título de *Magister Scientiae.*

APROVADA: 03 de maio de 2024.

Assentimento:

Documento assinado digitalmente
gov.br
YAGO JOSE ARAUJO DOS SANTOS
Data: 17/07/2024 17:50:37-0300
Verifique em https://validar.iti.gov.br

Yago José Araújo dos Santos
Autor

Documento assinado digitalmente
gov.br
JULIO CESAR SOARES DOS REIS
Data: 17/07/2024 18:08:57-0300
Verifique em https://validar.iti.gov.br

Júlio Cesar Soares dos Reis
Orientador

Agradecimentos

Agradeço primeiramente a Deus por me guiar e me confortar nessa trajetória. Agradeço aos meus pais Andréa de Castro e Arnaldo Santos por tudo o que fizeram e ainda fazem por mim. Nunca conseguirei compensar devidamente a dedicação que sempre manifestaram. Aos meus orientadores Julio Cesar e Michel Melo pelo conhecimento transmitido e pela paciência durante esse processo. Aos meus padrinhos Geraldo de Castro, João Nepomuceno e Celiana Hipólita e, à minha avó Bárbara de Castro pelas orações. Agradeço também, à Cláudia de Castro e Ricardo Sampaio por todo o suporte e acolhimento durante a minha vinda para Viçosa. Aos meus colegas de laboratório pela parceria e amizade. O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Código de Financiamento 001.

"Que nossas filosofias sigam no mesmo passo das nossas tecnologias. Que nossa compaixão siga no mesmo passo dos nossos poderes. E que o amor e não o medo, seja o motor da mudança"

(BROWN, 2017) (Tradução: Alves Calado)

Resumo

Ao mesmo tempo em que as plataformas de mídias sociais facilitaram as interações e ajudaram a democratizar o acesso à informação, estas também são exploradas para disseminação de desinformação em diferentes contextos, como saúde, política, dentre outros. Fatores como: a velocidade de disseminação, a demora na verificação de fatos e a complexidade de análise de mídias como imagens e vídeos, fazem com que o combate a essa prática seja cada vez mais desafiador. Esforços anteriores revelaram que as imagens representam o tipo de mídia mais explorado nas plataformas sociais. Neste contexto, uma abordagem para combater a desinformação em imagens é extrair o conteúdo textual para processamento posterior. Assim, o objetivo deste trabalho é investigar o desempenho de ferramentas de OCR na recuperação de informações textuais em Português do Brasil, a fim de contribuir para o desenvolvimento de sistemas de moderação e combate à desinformação cada vez mais eficientes. Este estudo apresenta uma metodologia para avaliar ferramentas de OCR considerando variações em 7 aspectos de imagem que são comumente encontrados nos recursos de edição das plataformas de mídias sociais, a saber: o ângulo de rotação do texto, as dimensões da imagem, a cor e o estilo da fonte, o tamanho da fonte, a presença de sombras no texto e o plano de fundo. Nossos resultados revelam a influência dos aspectos da imagem analisada na precisão do OCR, destacando o plano de fundo, o ângulo de rotação do texto e o estilo da fonte como os aspectos que produzem o maior impacto. Além disso, relatamos uma variação considerável entre os sistemas de OCR avaliados em termos de desempenho. Nossos experimentos demonstram que, dentre as ferramentas avaliadas, o Microsoft OCR apresenta os melhores resultados de CER em todos os aspectos analisados com valores médios variando entre 0,14% e 0,71%. Já os piores resultados são do Easy OCR, com valores médios de CER variando entre 1,5% e 57,8%, e do PyTesseract, com valores variando entre 3,9% e 35,6%. Por fim, além de realizarmos um experimento para avaliar como o desempenho das ferramentas de OCR impactam na detecção de desinformação, disponibilizamos um conjunto de imagens com desinformação em Português do Brasil que poderá ser utilizado pela comunidade acadêmica para diferentes fins.

Palavras-chave: Reconhecimento Óptico de Caracteres. Desinformação. Dados sintéticos.

Abstract

At the same time that social media platforms facilitate interactions and help democratize access to information, they are also exploited to spread misinformation in different contexts, such as health, politics, among others. Factors such as: the speed of dissemination, the delay in verifying facts and the complexity of analyzing media such as images and videos, make combating this practice increasingly challenging. Previous efforts have revealed that images represent the most explored type of media on social platforms. In this context, one approach to combating misinformation in images is to extract the textual content for further processing. Therefore, the objective of this work is to investigate the performance of OCR tools in retrieving textual information in Brazilian Portuguese, in order to contribute to the development of increasingly efficient moderation and combating disinformation systems. This study presents a methodology to evaluate OCR tools considering variations in 7 image aspects that are commonly found in the editing features of social media platforms, namely: text rotation angle, image dimensions, color and font style, font size, the presence of shadows in the text and the background. Our results reveal the influence of aspects of the analyzed image on OCR accuracy, highlighting the background, text rotation angle and font style as the aspects that produce the greatest impact. Furthermore, we report considerable variation between the OCR systems evaluated in terms of performance. Our experiments demonstrate that, among the tools evaluated, Microsoft OCR presents the best CER results in all aspects analyzed with average values varying between 0.14% and 0.71%. The worst results are from Easy OCR, with average CER values varying between 1.5% and 57.8%, and from PyTesseract, with values varying between 3.9% and 35.6%. Finally, in addition to carrying out an experiment to evaluate how the performance of OCR tools impacts the detection of misinformation, we provide a set of images with misinformation in Brazilian Portuguese that can be used by the academic community for different purposes.

Keywords: Optical Character Recognition (OCR). Misinformation. Synthetic Data.

Lista de ilustrações

Lista de tabelas

Lista de Algoritmos

Lista de abreviaturas e siglas

AFC	Verificação de Fatos Automatizada (do inglês *Automated Fact-Checking*)
ASR	Reconhecimento Automático de Fala (do inglês *Automatic Speech Recognition*)
AUC	Área Sob a Curva (do inglês *Area Under the Curve*)
CCD	Dispositivo de Carga Acoplada (do inglês *Charge-Coupled Device*)
CER	Taxa de Erro de Caracteres (do inglês *Character Error Rate*)
CPU	Unidade Central de Processamento (do inglês *Central Process Unit*)
GANs	Redes Adverárias Generativas Condicionais (do inglês *Conditional Generative Adversarial Networks*)
GPU	Unidade de Processamento Gráfico (do inglês *Graphics Processing Unit*)
IA	Inteligência Artificial
IBM	*International Business Machines Corporation*
MPF	Ministério Público Federal
OCR	Reconhecimento Óptico de Caracteres (do inglês *Optical Character Recognition*)
Ptbr	Português do Brasil
ROC	Curva Característica de Operação do Receptor (do inglês *Receiver Operating Characteristic Curve*)
TA	Tradução Automatizada
TTS	Conversão de Texto para Fala (do inglês *Text-To-Speech*)
WER	Taxa de Erro de Palavras (do inglês *Word Error Rate*)

Sumário

1 Introdução

O uso das plataformas de mídias sociais têm expandido a capacidade das pessoas de compartilharem informações a qualquer momento, em qualquer lugar, e sobre qualquer assunto (*e.g.*, sentimentos, experiências, críticas, etc.) (WANG et al., 2023). A facilidade de acesso aliada ao tempo de imersão dos usuários em plataformas sociais como X[1] (antigo Twitter), Facebook[2], WhatsApp[3] e Instagram[4] fazem desses sistemas a principal fonte para consumo de informações, incluindo notícias (SHU et al., 2017; ZACHLOD et al., 2022; KAZEMI et al., 2022). No entanto, a comodidade proporcionada por essas plataformas também as torna ambientes extremamente propícios à ampla disseminação de desinformação, que se refere a notícias fabricadas com conteúdo intencionalmente falso/inverídico (LI et al., 2023). Esforços anteriores demonstraram que a propagação da desinformação representa ameaças aos indivíduos e à sociedade, com impactos diversos que vão desde o comprometimento da reputação de personalidades bem como ao descrédito de instituições diversas (REIS et al., 2019; GALHARDI et al., 2020; BIANCOVILLI; MAKSZIN; JURBERG, 2021). Porém, a contenção desse tipo de conteúdo em plataformas de mídias sociais ainda é um desafio e muitos países, incluindo o Brasil (SOUSA; FREITAS, 2022), estão discutindo políticas de regulamentação da Internet como alternativa para minimização dos impactos ocasionados pela disseminação de desinformação no ambiente on-line.

De forma geral, usuários de plataformas de mídias sociais podem compartilhar conteúdo multimídia que viabilizam o fornecimento de informações por meio da combinação de texto, imagem, áudio e vídeo (WANG et al., 2023). Para extração e análise de conteúdo a partir desses diferentes formatos de mídia, diversas linhas de pesquisa inseridas nos campos de Reconhecimento de Padrões e Inteligência Artificial ganharam impulso para serem aprimoradas, tais como: Reconhecimento Automático de Fala (do inglês, *Automatic Speech Recognition* - ASR), Reconhecimento Óptico de Caracteres (do inglês, *Optical Character Recognition* - OCR), Tradução Automatizada (TA), Conversão de Texto para Fala (do inglês, *Text-To-Speech* - TTS), entre outras (CHAUDHURI et al., 2017). Especificamente, no contexto de detecção de desinformação, estas técnicas podem ser exploradas como ferramentas de suporte para identificação de desinformação em diferentes cenários.

Particularmente, este trabalho está focado na avaliação do desempenho de técnicas de OCR para extração de dados textuais de imagens compartilhadas em mídias sociais. Primeiro, este é um pré-processamento bastante comum em abordagens para detecção de desinformação (REIS et al., 2019), que realizam a extração de conteúdo textual de determinado tipo de mídia para processamento posterior. Além disso, um levantamento estatístico referente ao primeiro trimestre de 2021 reportado em (NANDY, 2022) e realizado pela Socialbakers[5] (empresa de *marketing* focada em mídia social) (BAKERS, 2021), apontou os tipos de mídia mais usados em postagens em duas das maiores redes sociais do mundo: o Facebook e o Instagram. Por meio dos gráficos apresentados na Figura 1,

[1] <https://twitter.com/>
[2] <https://www.facebook.com/>
[3] <https://www.whatsapp.com/>
[4] <https://www.instagram.com/>
[5] <https://www.digitalinformationworld.com/2021/05/images-videos-or-links-study-shows-most.html>

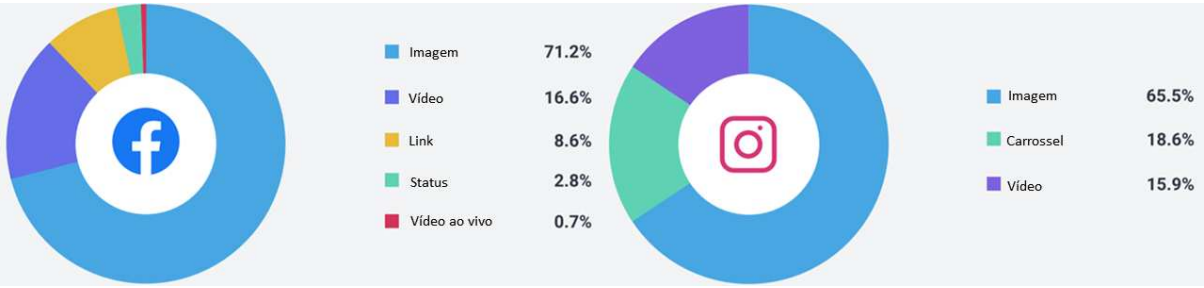


Figura 1 – Levantamento estatístico dos tipos de mídia mais compartilhados no Facebook e Instagram no primeiro trimestre de 2021.
Fonte: Adaptado de (AHMED, 2021).

podemos notar que as imagens representam o tipo de mídia mais compartilhada tanto no Facebook quanto no Instagram, sendo 71,2% e 65,5% respectivamente. Já na Figura 2, os dados revelam que as publicações de imagem são as que geraram os maiores níveis de engajamento entre os usuários de páginas do Facebook. Especificamente no contexto de detecção de desinformação, esforços anteriores já demonstraram que imagens são o tipo de mídia mais explorado para a disseminação deste tipo de conteúdo em plataformas de mídias sociais (SABAT; FERRER; NIETO, 2019; RESENDE et al., 2019).

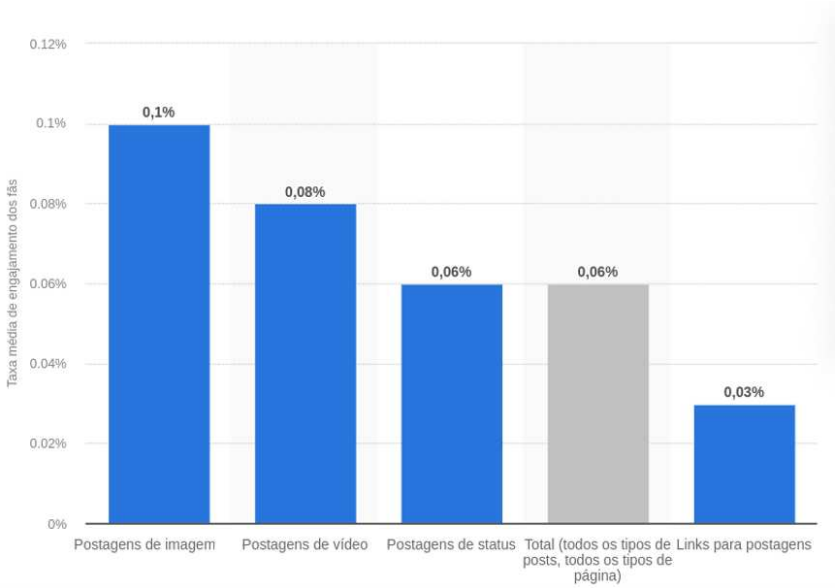


Figura 2 – Engajamento médio dos fãs da página do Facebook com tipos de postagem selecionados em todo o mundo em novembro de 2023.
Fonte: Adaptado de (DIXON, 2024).

Imagens contendo texto representam um desafio para análise em vários cenários, incluindo o combate à desinformação e a moderação de conteúdo (KAZEMI et al., 2022). Isso porque as informações textuais presentes nas imagens não se encontram de forma facilmente processável pelo computador e isso dificulta a análise do valor semântico das sentenças presentes no texto (AWEL; ABIDI, 2019). A Figura 3 exemplifica o compartilhamento de desinformação por meio de imagens. Além disso, existem muitas fontes possíveis de variação que dificultam a extração do texto de um plano de fundo, como: o sombreamento, a textura, o contraste, a cor, o estilo e tamanho de fonte, a orientação e o alinhamento (JUNG; KIM; JAIN, 2004; YE; DOERMANN, 2014; YIN et al., 2016). Em uma análise aprofundada das mídias propagadas durante os processos eleitorais indiano em

2019, foi verificado que imagens contendo texto representavam 40% do conteúdo propagado no WhatsApp, 35% eram "memes"(*i.e.*, sátiras com teor cômico) (KAZEMI et al., 2022). Durante as eleições presidenciais brasileiras de 2018, houve uma grande volume de imagens contendo desinformação sendo propagadas nas plataformas de mídias sociais (BARRETO JR; VENTURI JR, 2020). A associação entre as imagens e o fenômeno da desinformação é uma forma de comover as massas e promover uma distorção da realidade por meio da concatenação entre aspectos gerais da política com elementos culturais e comportamentais capazes de atribuir uma aura de credibilidade e amplificar seus efeitos (BARRETO JR; VENTURI JR, 2020). Neste cenário, faz-se necessária a busca por soluções que auxiliem na análise de tais dados, facilitando a captura da informação contida nas imagens e viabilizando uma vasta gama de análises futuras.

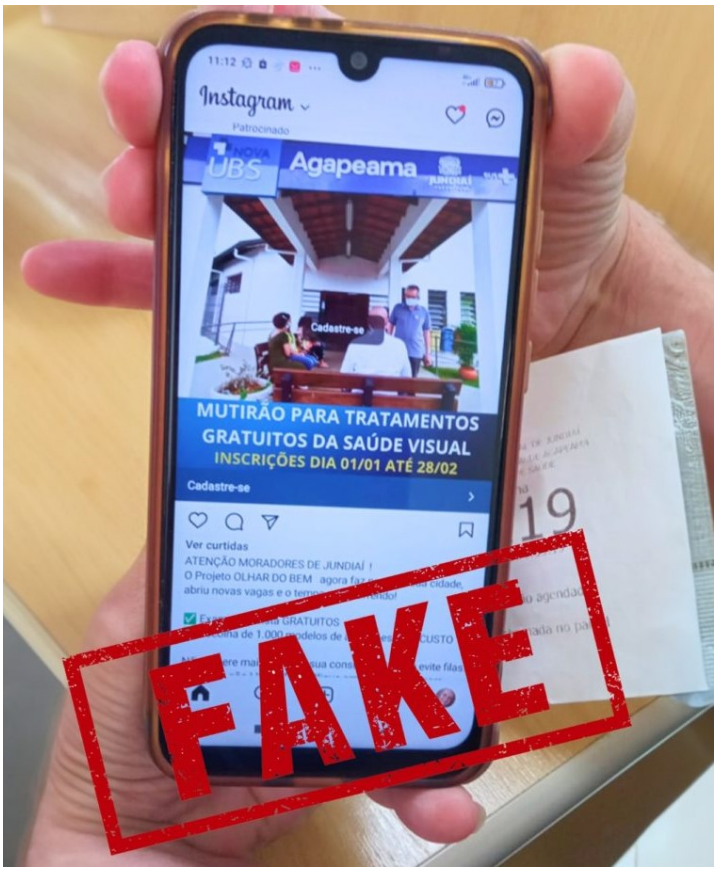


Figura 3 – Exemplo de desinformação propagada por meio de imagem.
Fonte: (JUNDIAí, 2024).

Diante do exposto, a proposta deste trabalho é avaliar o desempenho de um conjunto de ferramentas de recuperação textual de imagens, quando aplicadas em imagens com características comumente encontradas em mídias sociais e no idioma Português do Brasil. Para isso, foi realizado um levantamento das principais ferramentas de OCR disponíveis online, onde cinco foram selecionadas para análise com base em seus requisitos e capacidades, são elas: Microsoft OCR[6], Google Vision[7], Google Drive OCR[8], PyTesseract[9] e EasyOCR[10]. A fim de apresentar uma análise mais consistente, que permeie todas as etapas, desde o

[6] <https://learn.microsoft.com/pt-br/azure/ai-services/computer-vision/overview-ocr>
[7] <https://cloud.google.com/vision/docs/apis?authuser=2&hl=pt-br>
[8] <https://google-drive-ocr.readthedocs.io/en/latest/readme.html#features>
[9] <https://pypi.org/project/pytesseract/>
[10] <https://github.com/JaidedAI/EasyOCR>

tratamento dos dados, até a recuperação da informação textual, este trabalho apresenta a construção de um repositório de imagens sintéticas, contendo desinformação. As imagens produzidas são então inseridas nas ferramentas de OCR e os dados retornados por estas são avaliados por meio da aplicação de um conjunto de métricas específicas, de forma a apurar a qualidade, o desempenho, o tempo de processamento e identificar os pontos de deficiênica de cada ferramenta. Com isso, neste trabalho foi realizada uma discussão sobre a eficiência dos sistemas de OCR e sobre a viabilidade de aplicação da metodologia proposta. Por fim, foi conduzido um experimento que avalia como a qualidade dos dados retornados pelas ferramentas de OCR pode afetar no processo de detecção de desinformação.

1.1 Justificativa

Este trabalho respalda-se pelo fato de que o desempenho das ferramentas de OCR apresentam resultados distintos para diferentes contextos, conforme analisado por: Romanello, Najem-Meyer e Robertson (2021) que revelam que a ferramenta de OCR Tesseract[11] foi analisada na extração de textos de documentos digitalizados do século XIX e a ferramenta não apresentou bons resultados de precisão. Já em (PATEL; PATEL; PATEL, 2012) a mesma ferramenta foi utilizada para extrair informações textuais de fotografias de placas de veículos e os valores de precisão foram satisfatórios. Assim, a seleção da melhor ferramenta está à mercê do contexto de aplicação. Conforme observado por Castro et al. (2021), além de se considerar o contexto prático de aplicação, é preciso também considerar o idioma em que a ferramenta será aplicada, e nesse caso, existe uma grande escassez de trabalhos que se propõem a analisar essas ferramentas no idioma Português do Brasil. Dessa forma, com a finalidade de apresentar uma análise detalhada do desempenho das ferramentas para a temática abordada e que permeie todas as etapas, desde a recuperação da informação textual, até à detecção da desinformação, este trabalho propõe a construção de um conjunto de dados e a realização de um estudo sobre os principais sistemas de OCR no contexto de extração de textos, que se apresentam no idioma Português do Brasil, e de como os sistemas de OCR impactam na detecção da desinformação.

1.2 Objetivos

A tarefa de extrair a informação textual de uma imagem, e deixá-la em um formato processável por computador, representa um desafio para diversos problemas, incluindo o combate à desinformação e a moderação de conteúdo em plataformas de mídias sociais. Por exemplo, trabalhos anteriores mostraram que este foi um tipo de mídia bastante explorado por campanhas de desinformação durante as eleições presidenciais no Brasil (RESENDE et al., 2019). Dessa forma, o objetivo geral deste trabalho é investigar o desempenho de um conjunto de ferramentas de OCR na extração de dados textuais de imagens, com padrões comumente identificados em plataformas digitais, que se apresentam exclusivamente no idioma Português do Brasil.

1.2.1 Objetivos Específicos

O OCR permite que uma máquina reconheça caracteres através de mecanismos ópticos. O resultado produzido pelo OCR deve, em um cenário ideal, ser idêntico à entrada

[11] <https://github.com/tesseract-ocr/tesseract>

em relação à formatação. Dessa forma, o objetivo geral deste trabalho pode ser divido nos seguintes objetivos específicos:

- Construir e disponibilizar um conjunto de dados, contendo imagens com diferentes configurações comumente identificadas em plataformas de mídias sociais e com informações textuais no idioma Português do Brasil, para avaliar técnicas de OCR e, assim, incentivar pesquisas futuras;
- Realizar um levantamento das técnicas de OCR amplamente exploradas e que fornecem suporte ao idioma Português do Brasil;
- Comparar técnicas de OCR em termos de desempenho, qualidade e tempo, na recuperação de informação textual apresentada no idioma Português do Brasil, a partir de imagens com diferentes configurações;
- Verificar os impactos da aplicação das ferramentas de OCR na detecção de desinformação por abordagens baseadas em aprendizado de máquina.

1.3 Delimitação do Trabalho

Não faz parte do escopo deste trabalho a análise de procedimentos legais para acesso e manipulação de dados contendo desinformação, uma vez que, os dados utilizados se encontram em domínio público. O estudo apresentado diz respeito à manipulação de informações públicas fornecidas por repositórios textuais de terceiros e que foram devidamente referenciados neste trabalho.

Além disso, neste trabalho é apresentada uma metodologia para a construção de um conjunto de imagens sintéticas e outra para a execução da avaliação das ferramentas de OCR. Como estudo adicional, será realizado um breve experimento de classificação, onde um algoritmo classificador será aplicado no conjunto de dados retornado pelos OCRs de forma a verificar o impacto dessas ferramentas na detecção de desinformação. Não faz parte deste trabalho a aplicação de algoritmos de aprendizado de máquina mais avançados dado que o volume de dados disponível é baixo e também porque o foco deste trabalho se encontra na avaliação dos sistemas de OCR.

1.4 Contribuições

A principal contribuição deste trabalho consiste no desenvolvimento de uma metodologia para avaliação das ferramentas de OCR. Além disso, criamos e disponibilizamos um conjunto de imagens sintéticas contendo desinformação em Português e realizamos um estudo com um algoritmo de classificação para determinar se, e como, a qualidade dos dados recuperados pelas ferramentas de OCR impacta na tarefa de detecção de desinformação. As contribuições serão detalhadas a seguir.

Para avaliar as ferramentas de OCR, foi construído e disponibilizado um conjunto de dados composto por imagens sintéticas e com características específicas para identificar os quesitos problemáticos das ferramentas. As imagens produzidas além de possuírem informações textuais incorporadas (no idioma Ptbr), possuem também variações de aspectos básicos de edição como: variações no ângulo de rotação do texto, plano de fundo, estilo da fonte, entre outros. As informações textuais presentes nas imagens são referentes à

manchetes de notícias e possuem às respectivas verificações de fatos, o que permite à classificação de desinformação. Em suma, o conjunto de dados criado representa um recurso robusto cuja aplicação não se limita apenas à avaliação de ferramentas de OCR mas também em novos estudos nas áreas de Aprendizagem de Máquina e Processamento Digital de Imagens.

Além disso, neste trabalho apresentamos uma metodologia para avaliação de ferramentas de OCR no contexto de mídias sociais. Com essa metodologia, é possível conduzir uma avaliação univariada das ferramentas e identificar seus pontos de deficiência, expondo os problemas para que os desenvolvedores possam aprimorar suas ferramentas. Embora tenha sido originalmente desenvolvida para o contexto de mídias sociais, a metodologia apresentada por este trabalho pode ser facilmente adaptada para aplicação em outros contextos.

Como contribuição final, conduzimos um estudo para verificar como a qualidade dos dados retornados pelas ferramentas de OCR impacta na detecção da desinformação. Para isso, um algoritmo classificador foi selecionado para realizar inferências sobre os dados textuais retornados pelos OCRs, comparando-as com as inferências realizadas sobre os textos originais (ou verdade fundamental). Com isso, foi possível comprovar que a qualidade dos dados retornados pelos OCRs impacta diretamente na detecção da desinformação. Assim, podemos concluir que a seleção da ferramenta de OCR para integração com sistemas de detecção de desinformação e moderação de conteúdo influenciará diretamente na qualidade da detecção e contenção da desinformação.

1.5 Organização do Trabalho

O Capítulo 2 apresenta conceitos que fundamentam este trabalho, incluindo um levantamento dos trabalhos realizados na área, as métricas utilizadas e o funcionamento das técnicas abordadas neste trabalho, nesse capítulo também é apresentado um levantamento dos trabalhos relacionados, onde são mostrados os contextos de aplicação e os resultados obtidos pelos pesquisadores. No Capítulo 3, descrevemos a construção do conjunto de imagens, os critérios de seleção das ferramentas de OCR para análise e os preparativos para a condução dos experimentos. O Capítulo 4 apresenta os resultados e as discussões sobre os experimentos realizados com as ferramentas de OCR. O Capítulo 5 concentra informações relativas aos experimentos conduzidos com o algoritmo classificador para detecção de desinformação. Por fim, o Capítulo 6 conclui o estudo, relaciona os trabalhos futuros e as limitações deste trabalho.

2 Fundamentação Teórica e Trabalhos Relacionados

Conforme ilustrado pela Figura 4, esta pesquisa pode ser estruturada em três etapas: i) Construção do conjunto de dados para avaliar as ferramentas de OCR; ii) Avaliação das ferramentas de OCR com a metodologia desenvolvida (apresentada no Capítulo 3) e; iii) Estudo dos impactos de integração das ferramentas de OCR em sistemas de detecção de desinformação. Dessa forma, este capítulo apresenta um conjunto de conceitos que tem como finalidade prover um embasamento teórico para entender as principais tecnologias e metodologias utilizadas em cada uma das etapas desta pesquisa. Além disso, são apresentados trabalhos relacionados que ajudam a esclarecer o objeto de estudo. Inicialmente, é apresentado o campo de estudo com trabalhos que mostram o histórico da área de Processamento de Imagens dedicada ao OCR desde o seu surgimento. Em seguida, são apresentados conceitos para geração de repositórios de dados sintéticos, destacando suas vantagens e aplicações. Também são apresentadas as métricas utilizadas para avaliar as técnicas de OCR, bem como os tipos de erro capturados por essas métricas. Depois, são descritos os repositórios de dados relacionados à desinformação existentes na literatura. Em seguida, são apresentados os estudos em OCR. Por fim, são relacionadas as lacunas de pesquisa onde são discutidos alguns problemas em aberto e é realizada uma correlação entre os trabalhos selecionados.

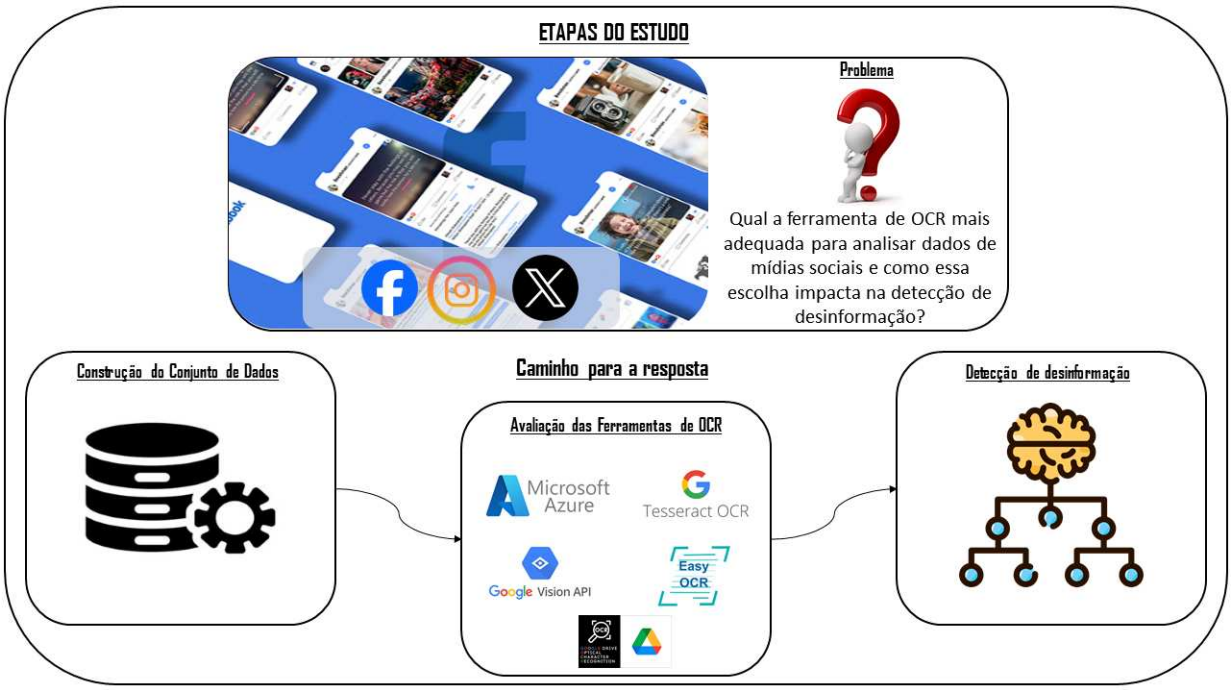


Figura 4 – Visão geral das etapas do estudo.

2.1 Reconhecimento Óptico de Caracteres (OCR)

Os primeiros esforços em OCR advém de 1914, quando Emanuel Goldberg desenvolveu uma máquina que recebia como entrada conjuntos de caracteres e os convertia em

código telegráfico padrão (SCHANTZ, 1982). Entre as décadas de 1920 e 1930, Emanuel Goldberg desenvolveu a chamada "Máquina Estatística", cujo objetivo era pesquisar arquivos de microfilmes através de um sistema de reconhecimento óptico. Em 1931, ela foi patenteada e, posteriormente, a patente foi adquirida pela *International Business Machines Corporation (IBM)* (SCHANTZ, 1982).

Outro importante fator de influência no desenvolvimento da área surgiu em 1974 com Ray Kurzweil, que desenvolveu o OCR Omnifonte, um sistema capaz de reconhecer texto impresso em qualquer fonte (SCHANTZ, 1982). Kurzweil decidiu que a melhor aplicação dessa tecnologia era na acessibilidade, permitindo a criação de uma máquina de leitura para cegos. A ideia era que pessoas com deficiência visual tivessem um computador com a capacidade de ler textos para elas em voz alta, o que demandou o desenvolvimento de duas outras tecnologias: o *scanner* de mesa CCD (Dispositivo de Carga Acoplada, do inglês *Charge-Coupled Device - CCD*) e o sintetizador de conversão de texto em fala.

Chaudhuri et al. (2017) descrevem que o OCR é obtido através de imprescindíveis etapas de segmentação, extração de características e classificação. Os autores descrevem ainda que os sistemas de OCR se tornaram uma das aplicações mais bem sucedidas nos campos de Reconhecimento de Padrões e Inteligência Artificial (IA). De acordo com o estudo, embora ainda não sejam capazes de competir com as capacidades de leitura humana, existe uma grande variedade de aplicações para os sistemas de OCR, como a automação de faturas (CASTRO, 2022), análise de dados armazenados em nuvem (DASH, 2021), moderação em redes sociais (SABAT; FERRER; NIETO, 2019), automação de formulários (NOVELIANTO, 2021), entre outras. Após isso, informações contextuais são usadas para reconstruir palavras e números do texto original. O aprendizado da máquina é feito submetendo exemplos de todas as classes. Durante o reconhecimento, os caracteres desconhecidos são comparados com as descrições obtidas em etapas anteriores e atribuídos à classe de melhor correspondência. Os autores descrevem ainda que na maioria dos sistemas comerciais para reconhecimento de caracteres, o processo de treinamento é realizado com antecedência. Alguns sistemas, no entanto, incluem facilidades para treinamento no caso de inclusão de novas classes de caracteres. Um sistema de OCR típico consiste em vários componentes, sendo o primeiro digitalizar o documento analógico usando um *scanner* óptico (CHAUDHURI et al., 2017). Quando as regiões do texto são localizadas, a segmentação cuida de extrair os símbolos. Os símbolos são então pré-processados de forma a eliminar o ruído e a facilitar a extração de recursos. A seguir, é feita uma correspondência entre os símbolos extraídos e as descrições de classes obtidas na etapa de aprendizagem anterior.

Por fim, esforços anteriores revelam que o OCR fornece resultados excepcionais apenas em casos de uso específicos (ANTONELLO, 2017; ROMANELLO; NAJEM-MEYER; ROBERTSON, 2021). Na maioria das aplicações práticas, ainda está muito abaixo do nível humano. Os aplicativos OCR modernos são pouco eficientes no processamento de documentos com baixa qualidade de imagem e em alguns alfabetos como fontes árabes, caligrafia e letra cursiva (ROUCHDI; ELLOTF; BAYOUMI, 2020; ROMANELLO; NAJEM-MEYER; ROBERTSON, 2021). Isso evidencia a necessidade de novas pesquisas para avaliar as capacidades desses sistemas.

2.2 Métricas de Avaliação Para Ferramentas de OCR

Para avaliar a qualidade do desempenho de sistemas de OCR, várias métricas podem ser aplicadas para verificação de diferentes aspectos (KANAI et al., 1993; SABER; AHMED; HADHOUD, 2014). Em resumo, essas métricas ajudam a estimar o quão próximo o texto predito está do original. As métricas mais utilizadas buscam levantar as seguintes informações: precisão de caracteres, a similaridade de *strings* (onde algoritmos de busca aproximada podem ser empregados para avaliar a similaridade entre o texto original e o transcrito), e a taxa de acerto. Algumas métricas são: Distância de Levenshtein (ENE; ENE, 2017); CER (Taxa de Erro de Caracteres, do inglês *Character Error Rate*) e WER (Taxa de Erro de Palavras, do inglês *Word Error Rate*) (DETLEFSEN et al., 2022); Distância Episódica, Smith-Waterman, Coeficiente de Dice, *Q-Gram* e Similaridade do Cosseno (GOMES, 2022). Neste trabalho, foram consideradas métricas que avaliam a qualidade, o tempo e a capacidade de inferência, as quais serão detalhadas a seguir.

2.2.1 Qualidade

Para avaliar a qualidade da recuperação de informação textual pelos sistemas de OCR, existem três tipos de erro a serem considerados (AWEL; ABIDI, 2019; ROMANELLO; NAJEM-MEYER; ROBERTSON, 2021). A Figura 5 apresenta um exemplo:

- Erro de substituição: caracteres/palavras com erros ortográficos;
- Erro de exclusão: caracteres/palavras perdidos ou ausentes;
- Erro de inserção: Inclusão incorreta de caracteres/palavras.

Figura 5 – Exemplos de erro considerados na avaliação de qualidade.
Fonte: Adaptado de Leung (2021).

Com isso, para o critério de qualidade, existem três métricas (algumas métricas podem sofrer variações devido à normalização), que são comumente utilizadas em estudos anteriores (HAMDI et al., 2023; NGUYEN; LE; ZELINKA, 2019), e que serão exploradas neste estudo:

- Distância de Levenshtein: Essa é uma métrica de distância que mede a diferença entre duas sequências de *strings* (ENE; ENE, 2017). É o número mínimo de edições de um único caractere (ou palavra) (isto é, inserções, exclusões ou substituições) necessárias para transformar uma palavra (ou frase) em outra. Quanto menor a distância, mais semelhantes são as *strings*. Por exemplo, a distância de Levenshtein entre “comovida e “comprida” é 2 , pois são necessárias no mínimo 2 edições para

transformar uma palavra na outra. (Ver Fórmula 2.1).

$$lev_{a,b}(i,j) = \begin{cases} max(i,j) & if\ min(i,j) = 0, \\ min \begin{cases} lev_{a,b}(i-1,j) + 1 \\ lev_{a,b}(i,j-1) + 1 \\ lev_{a,b}(i-1,j-1) + 1_{(a_i \neq b_j)} \end{cases} & otherwise, \end{cases} \tag{2.1}$$

em que:

- lev = Distância de Levenshtein;
- a = *String* 1;
- b = *String* 2;
- i = a posição do caractere terminal da *string* 1;
- j = a posição do caractere terminal da *string* 2.

- *Character Error Rate* (CER) (Taxa de Erro de Caractere): De acordo com Detlefsen et al. (2022), o cálculo do CER é baseado no conceito de distância de Levenshtein, em que contamos o número mínimo de operações em nível de caractere necessárias para transformar o texto da verdade básica (também conhecido como texto de referência) na saída do OCR. A saída dessa equação representa a porcentagem de caracteres no texto de referência que foram previstos incorretamente na saída do OCR. Quanto menor o valor de CER (com 0 sendo uma pontuação perfeita), melhor o desempenho do modelo de OCR (ver Fórmula 2.2 apresentada em (DETLEFSEN et al., 2022)).

$$CER = \frac{S + D + I}{N} = \frac{S + D + I}{S + D + C}, \tag{2.2}$$

em que:

- S = Número de Substituições;
- D = Número de elementos deletados;
- I = Número de inserções;
- C = Número de caracteres corretos;
- N = Número de caracteres no texto de referência (também conhecido como verdade fundamental).

- *Normalização de CER*: Um caso a se considerar, é que os valores obtidos pelo CER podem exceder 100%. Isso acontece principalmente quando há muitas inserções, por exemplo, o valor de CER entre as strings "XYZ"e "XYZ95861"é de 166,67%. Uma forma de deixar os valores na escala 0-100% é dividindo o número de erros pela soma do número de operações de edição e o número de símbolos corretos, que é sempre maior que o numerador, para isso, basta fazer a divisão pela soma S + D + C + I. Dessa forma, o valor de CER ficará sempre na faixa de 0 a 100% (ver Fórmula 2.3 descrita em (DETLEFSEN et al., 2022)).

$$CER_{normalizado} = \frac{S + D + I}{S + D + I + C}, \tag{2.3}$$

em que:

  - S = Número de Substituições;
  - D = Número de elementos deletados;
  - I = Número de inserções;
  - C = Número de caracteres corretos.

- *Word Error Rate* (WER) (Taxa de Erro de Palavras): Pode ser mais aplicável se envolver a transcrição de parágrafos e frases. A fórmula do WER é a mesma do CER, mas o WER opera no nível da palavra . Representa o número de substituições, exclusões ou inserções de palavras necessárias para transformar uma frase em outra. Quanto menor o valor de WER, melhor é a precisão do OCR. Assim como o CER, os valores de WER podem extrapolar os 100%, isso acontece quando há muitas inserções indevidas de palavras (ver Fórmula 2.4 extraída de (DETLEFSEN et al., 2022)).

$$WER = \frac{S_w + D_w + I_w}{N_w} = \frac{S_w + D_w + I_w}{S_w + D_w + C_w}, \tag{2.4}$$

  em que:

  - S_w = Número de palavras substituídas;
  - D_w = Número de palavras deletadas;
  - I_w = Número de palavras inseridas;
  - C_w = Número de palavras corretas;
  - N_w = Número de palavras no texto de referência (também conhecido como verdade fundamental).

2.2.2 Avaliação binária

A avaliação binária é um processo de classificação que envolve duas categorias distintas e mutuamente exclusivas, ou seja, cada item avaliado pode ser classificado em apenas uma das categorias possíveis (NETO et al., 2020). Como métricas binárias temos as seguintes:

- Acertos: Verifica se o mecanismo de OCR foi capaz de produzir uma inferência 100% correta, ou seja, se foi capaz de acertar todo o conteúdo textual presente na imagem;
- Erros: Número de instâncias que o OCR conseguiu produzir algum resultado (independente se estiver certo ou errado, excluindo as inferências 100% corretas);
- Sem Resultados: Número de instâncias que o OCR não conseguiu produzir nenhum resultado.

2.3 Estudos em OCR

Em (REUL et al., 2018), é apresentada uma avaliação de sistemas de OCR mistos, sem treinamento específico, como: OCRopus, Tesseract e ABBYY na recuperação de textos de uma coleção variada de dados de livros, periódicos e um dicionário do século XIX. Os dados utilizados se apresentam nos idiomas inglês, alemão e espanhol. Os resultados obtidos mostraram que o OCR Calamari supera consideravelmente os outros mecanismos,

reduzindo em média a taxa de erro de caracteres (CER) do ABBYY em mais de 70% resultando em um CER médio abaixo de 1%.

Já Antonello (2017), faz uso do OCR do OpenCV para realizar a identificação de placas de veículos. Em (ROMANELLO; NAJEM-MEYER; ROBERTSON, 2021), são utilizados os mecanismos de OCR Kraken + Ciaconna e o Google Tesseract, para o reconhecimento de documentos clássicos do século XIX que foram digitalizados. Os documentos se apresentavam nos idiomas Grego, Inglês e Alemão. Algumas das ferramentas analisadas são comuns aos dois trabalhos, no entanto, em uma análise inicial, percebe-se a divergência de desempenho entre elas. Isso se deve ao contexto de aplicação que muda de um trabalho para o outro, ou seja, os dados textuais explorados em (ANTONELLO, 2017) possuem padrões e propósitos diferente dos utilizados em (ROMANELLO; NAJEM-MEYER; ROBERTSON, 2021). Esses impactos de desempenho observados nos dois trabalhos, provocados pela variação de contexto de aplicação, indicam a necessidade de uma pesquisa mais aprofundada.

Sabat, Ferrer e Nieto (2019) abordaram o problema da detecção de discurso de ódio em memes da internet. Os autores descrevem os memes como ilustrações e frases que, quando combinadas, geralmente adotam um caráter cômico. No entanto, os memes também podem ser usados para propagar ódio pelas redes sociais. Os autores defendem que a detecção automática de memes contendo discurso de ódio ajudaria a reduzir o impacto social prejudicial. Com isso eles propuseram um sistema baseado no *Multilayer Perceptron*, treinado em um corpus de 5.020 memes, para realizar essa tarefa. Durante a condução dos experimentos, os autores utilizaram o mecanismo de OCR Tesseract em sua versão 4.0.0 para extrair as informações textuais das imagens e analisá-las à parte. Os autores descrevem que o melhor valor de acurácia obtido em seus experimentos foi de 83% e, embora o modelo produzido tenha sido capaz de identificar discurso de ódio em algumas imagens, seu resultado está longe de ser o ideal, seja pela arquitetura utilizada ou pelo tamanho do conjunto de dados. Em sua conclusão, os autores apontam a necessidade de uma avaliação dos sistemas de OCR, haja vista que o mecanismo de extração de dados textuais das imagens implicam diretamente na precisão do sistema de detecção de discurso de ódio como um todo.

Uma metodologia baseada em redes adverárias generativas condicionais (do inglês *Conditional Generative Adversarial Networks* - GANs) para o aprimoramento das técnicas de OCR é apresentada em (CASTRO et al., 2021). Essas GANs são aplicadas de forma a melhorar a qualidade da renderização do texto nas imagens, maximizando assim o desempenho do OCR. O mecanismo de OCR utilizado foi o Tesseract. Para os experimentos, os autores descrevem que foi criado um conjunto de dados sintético capaz de emular os problemas mais comuns encontrados em textos digitalizados (como distorções de imagens, caracteres borrados e fundos irregulares) e que esse conjunto foi utilizado para treinar a rede. Além do conjunto de dados sintético, também foram utilizadas imagens reais extraídas de teses brasileiras e periódicos americanos, ou seja, os textos se apresentavam em dois idiomas distintos o português e o inglês. Com isso, a eficácia da abordagem adotada pôde ser comprovada em diferentes idiomas. Os resultados obtidos demonstram que a metodologia proposta foi capaz de superar a maioria dos desafios que afetam a qualidade do OCR utilizado. As métricas CER e WER que indicam as taxas de erro apresentaram uma redução de até 50% em comparação com os resultados obtidos sem a metodologia.

Drobac e Lindén (2020) destacam que a qualidade do OCR aplicados na recuperação de informação, em um corpus de jornais e revistas finlandesas, é muito baixa e pouco

confiável para pesquisas científicas. De acordo com os autores, *softwares* comerciais atingem uma taxa de CER estimada entre 8 e 13%. Os autores descrevem ainda que houve tentativas de treinar modelos de OCR como o Ocropy e o Tesseract para o reconhecimento multilingual eficiente, mas que nenhuma dessas soluções foi capaz de produzir bons resultados. Os autores explicam que a dificuldade reside no fato de que o corpus se apresenta nas duas pricipais línguas da Finlândia, o finlandês e o sueco, e em duas famílias de fontes, a Blackletter e a Antiqua. Os experimentos conduzidos pelos autores, têm o objetivo de melhorar a qualidade da recuperação de informação textual dos OCRs por meio da pós-correção. Para isso, foram selecionadas duas ferramentas de OCR para trabalharem em conjunto, o Ocropy e o Calamari, e adicionado um modelo de reconhecimento de linguagem mista, baseado em redes neurais profundas, para ser aplicado na etapa de pós-correção. Feito isso, foi realizada uma análise de erros que indicou um aumento significativo na precisão, apresentando um CER de 1,7% no conjunto de testes finlandês e de 2,7% no conjunto de testes sueco. Por fim, os experimentos realizados em (DROBAC; LINDÉN, 2020), indicam pontos de melhoria nos procedimentos de recuperação de informação textual multilingual.

Embora grande parte dos materiais recentes disponíveis na literatura estejam dedicados à estudarem as aplicações das ferramentas de OCR no reconhecimento de placas de veículos e na recuperação de dados textuais de documentos históricos digitalizados, conforme observado em estudos anteriores (DROBAC; LINDÉN, 2020; ANTONELLO, 2017; REUL et al., 2018), as aplicações dos sistemas de OCR não se limitam a isso. Outras possibilidades de aplicação são, entre outras:

- Acessibilidade para deficientes visuais, viabilizando a legibilidade de textos em imagens por meio de leitores de tela (SHEELA et al., 2023);
- Reconhecimento de informação textual de pôsteres (GEORGIEVA; ZHANG, 2020);
- Reconhecimento de sinais de trânsito (GREENHALGH; MIRMEHDI, 2014);
- Extração de informações de recibos e faturas (KUMAR et al., 2020);
- Automação de processos empresariais, como no processamento de formulários preenchidos à mão (CHAUDHURI et al., 2017).

2.4 Construção de Repositórios de Dados Sintéticos

Os dados representam o componente principal de qualquer projeto de Aprendizagem de Máquina (GOLLAPUDI, 2016). Em (SILVA JR et al., 2021; ALZUBAIDI et al., 2023; WEI et al., 2024) é descrito que aplicar técnicas de aprendizagem de máquina em uma base de dados pequena é um desafio que permanece em aberto. Isso se deve ao fato de que os algoritmos desenvolvidos possuem melhor desempenho ao lidarem com grandes volumes de dados durante a etapa de treinamento (SILVA JR et al., 2021). Pesquisadores que se dedicam a desenvolver projetos de Aprendizagem de Máquina em campos de estudos novos, como é o caso da Detecção Automatizada de Desinformação, acabam sofrendo contratempos pela escassez de conjuntos de dados bem definidos, isso pode ser observado em (PÉREZ-ROSAS et al., 2019) e (SHAIKH; PATIL, 2020).

Para os casos em que a coleta de novos dados é inviável, uma boa alternativa é utilizar a técnica de geração sintética de dados, isso ajuda a aumentar o volume do

conjunto e a reduzir a variância obtida pelos algoritmos de aprendizagem de máquina (SILVA JR et al., 2021). Os dados sintéticos podem ser definidos como quaisquer dados que não foram coletados de eventos reais, ou seja, são produzidos por um sistema capaz de reproduzir características reais essenciais (NIKOLENKO, 2021).

2.5 Repositórios de Dados Relacionados

Em (HEYBURN et al., 2018), os dados sintéticos são definidos como os gerados a partir de dados reais, isso é feito usando as propriedades estatísticas subjacentes dos reais para produzir conjuntos de dados sintéticos que exibem essas mesmas propriedades estatísticas. Os autores realizaram um estudo comparativo entre um conjunto de dados sintético e um conjunto real. Eles buscavam entender se os dados gerados sinteticamente eram capazes de substituir os reais nas tarefas de aprendizagem de máquina. Com isso, Heyburn et al. (2018) conduziram testes com os dois tipos de conjunto e avaliaram pelas métricas de aprendizagem de máquina. Assim, os autores chegaram à conclusão de que os experimentos com o conjunto sintético são capazes de produzir resultados semelhantes ao conjunto real para as medições de pontuação F1, acurácia, precisão e revocação. Os resultados identificados, embora os autores deixem explícito se tratar de um estudo inicial, fornecem indicativos de que a avaliação de modelos construídos usando dados sintéticos pode refletir os resultados que seriam alcançados por meio de conjuntos de dados reais.

Esforços anteriores se dedicaram a disponibilizar repositórios de dados de imagens para impulsionar pesquisas que seguem a linha de entender e mitigar o problema ocasionado pela difusão da desinformação por meio deste tipo de mídia nas plataformas digitais (SHARMA; GARG, 2021; KRSTOVSKI; RYU; KOGUT, 2022). Os repositórios apresentados em (SHARMA; GARG, 2021) e (KRSTOVSKI; RYU; KOGUT, 2022), são o `IFND` e o `Evons`, respectivamente. A composição de ambos repositórios, se dá por meio de imagens manipuladas em editores como o *Photoshop*, confeccionadas com o objetivo de prejudicar a reputação de personalidades ou organizações importantes. Já em (BOIDIDOU et al., 2014), foram coletados dados da plataforma Twitter para a construção de um repositório contendo imagens (classificadas entre falsas e reais) com associação à mensagens, em que a confiabilidade pudesse ser verificada por fontes on-line independentes.

Especialmente em países como Brasil e Índia, onde o WhatsApp se tornou uma plataforma bastante explorada para propagação de campanhas de desinformação, estão emergindo algumas propostas de criação de repositórios de dados públicos para pesquisas neste contexto. Em (REIS et al., 2020), por exemplo, os autores disponibilizaram um repositório de dados contendo imagens verificadas por agências de checagem de fatos durante as eleições presidenciais brasileira de 2018 e indiana de 2019. No total, o repositório é composto por 135 e 837 imagens rotuladas como desinformação (ou *"misinformation"*) propagadas no Brasil e Índia, respectivamente, durante eventos políticos. Este repositório tem sido explorado por trabalhos nos mais diferenciados contextos, que vão desde à investigação como também no desenvolvimente de técnicas para detecção e contenção de desinformação (AKBAR et al., 2021; HAO et al., 2021; SHAHID et al., 2022).

Moreno e Bressan (2019) defendem que a verificação de fatos automatizada (do inglês *Automated Fact-Checking - AFC*) é o ápice da verificação de fatos. Com isso, os autores defendem o uso de técnicas de aprendizado de máquina na classificação de notícias. No entanto, de acordo com os autores, para que essas técnicas funcionem bem, elas demandam um grande conjunto de dados para o treinamento. Assim, a fim de viabilizar

trabalhos de classificação, os autores apresentam a criação do `FACTCK.BR` [1], um repositório baseado em alegações em português verificadas por agências de checagem de fatos. Os autores descrevem que as *Fake News* podem ser divididas em três grupos principais: i) *Hoaxes* (Boatos): Destinam-se a enganar o público, apresentando-se como notícias genuínas. Pode causar danos materiais ou danos à vítima; ii) Fabricações Sérias: São artigos escritos pela chamada imprensa amarela. Eles podem usar "*clickbaits*", manchetes mentirosas que não condizem com o conteúdo, ou "*hype*"*e.g.*, promoção extrema de uma ideia, tendência. para obter tráfego e ganho financeiro; e iii) *Fakes* humorísticos: Distinguem-se das notícias fabricadas, pois um leitor ciente da intenção satírica do conteúdo não estará disposto a acreditar na informação. Além de descrever os tipos de "*Fake News*", os autores descrevem ainda a forma em que os dados se encontram estruturados. No entanto, não é relatado nenhuma execução de experimentos de classificação, isso fica pendente como proposta de trabalhos futuros.

Um método linguístico que combina polaridade, emoção e características gramaticais para detectar notícias falsas em Português é apresentado em (SOUZA et al., 2020). O método apresentado revelou resultados promissores nos dados experimentais, obtendo acurácia superior a 92% na detecção de notícias falsas. Em média, o método proposto superou os métodos baseados em polaridade (mais conhecido na literatura como "Análise de Sentimentos", essa tarefa indica se as opiniões expressas no texto são positivas, neutras ou negativas em relação ao assunto abordado) e classificação gramatical (procedimento que rotula os componentes textuais com base em sua função gramatical para obter medidas de sua utilização) em 1,4 ponto percentual. Para a execução dos experimentos foi utilizado o repositório `Fake.BR` (SANTOS; MONTEIRO; PARDO, 2018), esse repositório possui uma base balanceada com rotulação binária *fake* e *não fake* bem definida. 7.200 notícias, onde 3.600 são falsas e as 3.600 restantes são reais.

2.6 Lacuna de Pesquisa

OCR é um tópico em evidência em Visão Computacional (CHENG et al., 2018). Apesar de várias décadas de pesquisa nesse tópico, reconhecer textos de imagens naturais ainda é uma tarefa árdua. A dificuldade nesse tópico acontece porque os textos de cena se apresentam frequentemente em arranjos irregulares, isto é, curvos ou distorcidos, e esses casos ainda não foram bem tratados na literatura. Além disso, os métodos existentes de reconhecimento de texto foram projetados para textos regulares (*e.g.,* horizontais e frontais) e que não podem ser facilmente generalizados para tratar textos irregulares (CHENG et al., 2018).

As ferramentas de OCR desempenham um trabalho fundamental no estágio de pré-processamento para a detecção de desinformação em dados visuais como imagens e vídeos que contêm texto (LI et al., 2022). Isso porque, ao converter o texto das imagens em dados editáveis, as ferramentas de OCR permitem que sistemas automatizados de checagem de fatos analisem e identifiquem conteúdos desinformativos de uma forma mais eficiente. Com isso, existe uma demanda por estudos que avaliem o desempenho das ferramentas de OCR a fim de verificar como essas ferramentas podem impactar no processo de detecção de desinformação.

Ademais, o desempenho das técnicas de OCR varia de acordo com o contexto de aplicação (REUL et al., 2018; ANTONELLO, 2017; ROMANELLO; NAJEM-MEYER;

[1] <https://github.com/jghm-f/FACTCK.BR>

ROBERTSON, 2021), o que ressalta a importância deste estudo. Até o momento, ainda não foi identificada uma técnica que apresente o melhor desempenho para todos os cenários de uso e, além disso, análises de ferramentas de OCR com suporte ao idioma Português do Brasil ainda são limitadas. Considerando o contexto deste estudo, estas são as lacunas de pesquisa preenchidas por este trabalho que tem como objetivo principal a investigação do desempenho de ferramentas de OCR para extração de textos escritos no idioma Português do Brasil, a partir de imagens com padrões (*e.g.,* diferentes tipos de fontes, cores, etc) comumente identificados em postagens realizadas em plataformas de mídia sociais, como alternativa para suportar as tarefas de análise, detecção e moderação de dados contendo desinformação.

3 Metodologia

Este capítulo descreve a metologia proposta para a condução das avaliações. Inicialmente são apresentados detalhes relativos ao processo de construção do conjunto de imagens (Seção 3.1). Depois, descrevemos o processo de seleção das ferramentas de OCR (Seção 3.2). Por fim, detalhamos o processo de avaliação das ferramentas de OCR. A Figura 6 apresenta uma visão geral da metodologia proposta para o trabalho.

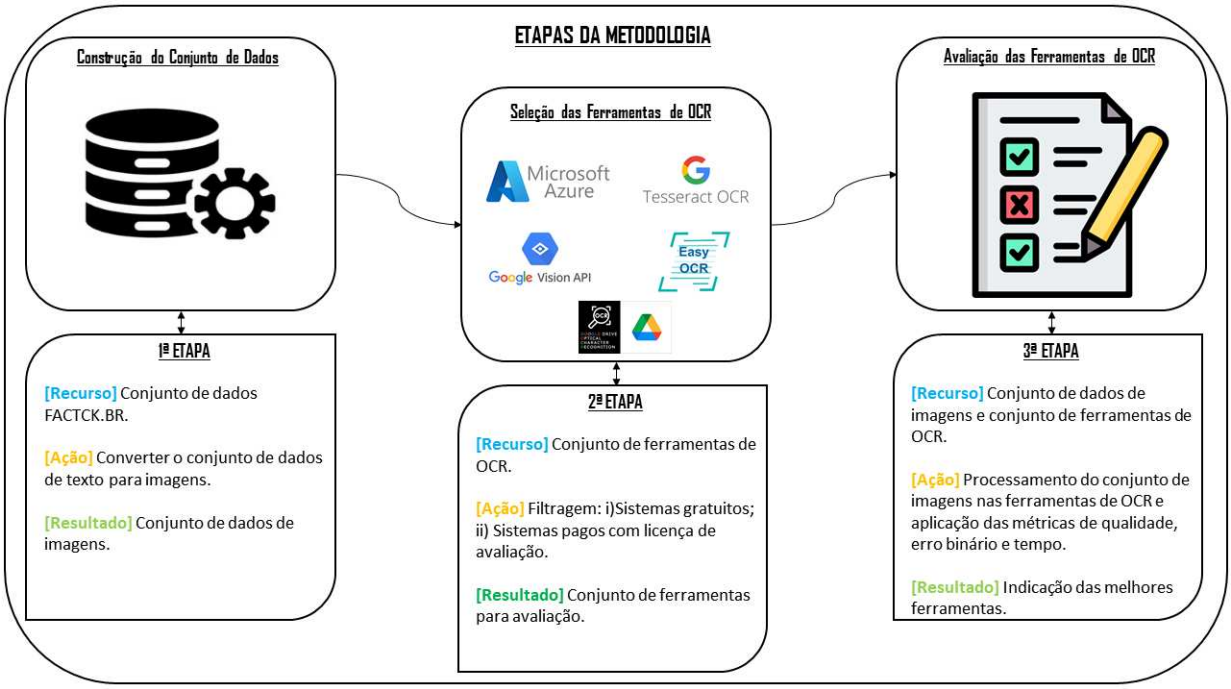


Figura 6 – Visão geral das etapas do trabalho.

3.1 Construção do Conjunto de Imagens

Para avaliar as técnicas de OCR, foi preciso criar um conjunto de imagens com características próprias para identificação dos pontos adversos às ferramentas. Essa etapa de criação do conjunto de dados se fez necessária pois não foi identificado um conjunto que atenda às necessidades deste estudo. A ideia é que as imagens geradas de forma sintética, para compor o conjunto, contenham informações textuais no idioma Português do Brasil checadas por especialistas e com variações de características como a rotação do texto, o tamanho e o estilo da fonte, o contraste entre a cor do texto e o plano de fundo, as dimensões da imagem e a presença de sombras no texto, comumente identificadas em postagens realizadas em plataformas de mídias sociais.

Durante o processo de construção do conjunto de imagens, exploramos um conjunto de dados textual contendo desinformação chamado `FACTCK.BR` (MORENO; BRESSAN, 2019). Esse conjunto de dados é composto por 1.314 notícias com suas respectivas verificações de fatos e classificações. Conforme descrito pelos autores, esses dados foram coletados do *ClaimReview*[1], um esquema de dados estruturados usado por agências de verificação de

[1] <https://www.claimreviewproject.com/>

fatos e que permite a coleta de dados em tempo real. O `FACTCK.BR` foi selecionado por conter manchetes de notícias, ou seja, dados textuais mais curtos comumente propagados em plataformas sociais, e que podem ser melhor aproveitados para a geração das imagens. Por fim, é importante mencionar que a metodologia proposta é robusta para variação de qualquer outro dado fornecido como entrada.

Uma vez que se tem os dados textuais, o próximo passo é a inserção desse conteúdo nas imagens. Para isso, foi implementado um gerador que cria imagens sintéticas, com os textos informados, variando aspectos como dimensão, fonte, visual e formatação do texto. Em resumo, as escolhas das variações de aspectos foram realizadas com base em padrões comumente identificados no conteúdo disseminado em plataformas digitais. Especificamente, as escolhas dos estilos de fonte bem como das dimensões, foram baseadas nos padrões utilizados pelo Instagram[2]. Como as fontes do Instagram não são de código aberto, foi realizada uma busca por fontes de código aberto que se assemelham às utilizadas pela plataforma em questão. As fontes utilizadas foram obtidas do DaFont[3]. Quanto aos tamanhos de fonte, sua seleção se deu baseada nos padrões praticados pelo Gmail[4]. As cores de fonte e de plano de fundo foram selecionadas de forma a ter um conjunto com baixo, médio e alto contraste quando relacionadas. Finalmente, o intervalo dos ângulos de rotação foi definido com base em avaliações experimentais.

O gerador proposto trabalha basicamente variando sete aspectos, detalhados a seguir:

- Fontes: Neste trabalho exploramos nove estilos de fontes, conforme apresentado na Tabela 1.

Tabela 1 – Estilos de fonte.

Estilo	Amostra
ComicSansMS3	Estilo da fonte
ModernSerif	ESTILO DA FONTE
KGPartofME	Estilo da fonte
AppleGaramond	Estilo da fonte
Timeless	Estilo da fonte
TypeMachine	Estilo da fonte
NixieOne	Estilo da fonte
KGLegoHouse	Estilo da fonte
KGSorryNotSorry	ESTILO DA FONTE

[2] <https://developers.facebook.com/products/instagram/?locale=pt_BR>
[3] <https://www.dafont.com/pt/>
[4] <https://developers.google.com/gmail/api/guides?hl=pt-br>

- Para cada uma das fontes selecionadas, foram exploradas seis cores sólidas, conforme ilustrado na Tabela 2.

Tabela 2 – Cores sólidas selecionadas.

Cores					
Branco	Preto	Azul	Azul Médio	Azul Ardósia	Azul Escuro

- Também consideramos três tamanhos de fonte distintos, sendo: pequeno = 40 pts, médio = 60 pts e grande = 80 pts. A Tabela 3 apresenta um exemplo.

Tabela 3 – Tamanhos de Fonte.

Tamanhos de Fonte
Tamanho 40
Tamanho 60
Tamanho 80

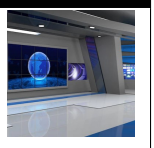
- Sombra: Para o repositório de dados (*i.e.*, imagens) construído neste trabalho, sombra é uma característica binária para textos sem, ou com sombra, respectivamente, conforme apresentado na Tabela 4.

Tabela 4 – Exemplo de aplicação da sombra no texto.

Sem Sombra	Com Sombra
Estilo da fonte	Estilo da fonte

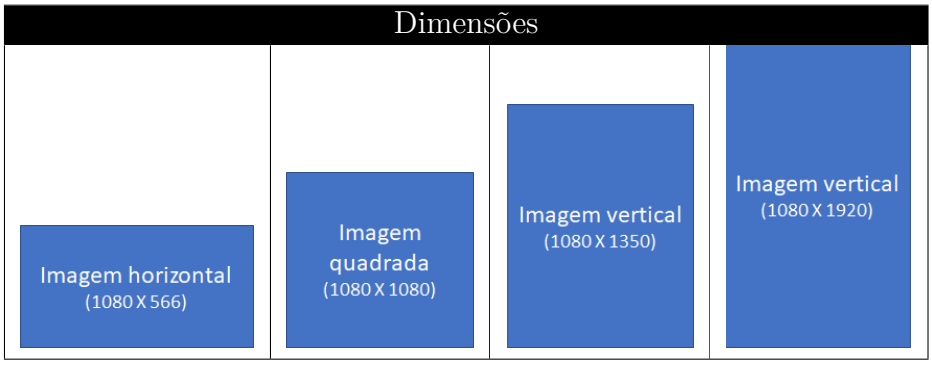
- Cor do plano de fundo: Esta característica foi variada com seis cores sólidas, e duas imagens utilizadas como *background*. A seleção de imagens de *background* é importante para que sejam abrangidas plataformas digitais que permitem a inserção de texto, em uma foto, por exemplo. No total, oito é o número de configurações possíveis. A Tabela 5 apresenta um exemplo.

Tabela 5 – Planos de fundo.

Planos de fundo							
Branco	Preto	Azul	Azul médio	Azul ardósia	Azul escuro		

- Dimensões: Nesta característica, foram selecionadas quatro dimensões: (1080 × 1080)px, (1080 × 1350)px, (1080 × 566)px e (1080 × 1920)px. Para fins de ilustração das proporções, relacionamos exemplos na Tabela 6.

Tabela 6 – Dimensões de imagem selecionadas.



- Ângulos de rotação do texto: Selecionamos treze ângulos para variação da rotação do texto (de $-30°$ à $30°$, a um passo de $5°$), conforme apresentado na Tabela 7.

Tabela 7 – Ângulos de rotação do texto.

Ângulos de rotação do texto												
-30	-25	-20	-15	-10	-5	0	5	10	15	20	25	30
T	T	T	T	T	T	T	T	T	T	T	T	T

Ao analisar o número de opções por cada aspecto variado e fazendo uma análise combinatória, temos um total de 134.784 possibilidades de combinações por amostra de texto. Esse número pode ser menor devido às restrições de tamanho do texto, limitado a 280 caracteres; Tamanho da fonte; Tamanho das margens e das dimensões da imagem.

3.1.1 Ajuste do Texto

Dado um texto de entrada, é necessário realizar um ajuste do mesmo para acomodação nas imagens. Para isso, deve-se considerar limitações como as dimensões da imagem, o tamanho das margens, o ângulo de rotação, o tamanho e o estilo da fonte e, a presença, ou não, do efeito sombra. Todos esses fatores influenciam no ajuste do texto para acomodação em uma imagem. O ajuste do texto é realizado a partir do cálculo das coordenadas dos pontos da caixa de texto.

Para fins de ilustração deste processo, considere o seguinte trecho do primeiro canto do poema "Os Lusíadas", de Luíz Vaz de Camões (CAMÕES, 1818) (ver Figura 7):

> "As armas e os Barões assinalados,
> Que da ocidental praia Lusitana
> Por mares nunca de antes navegados"(...)
> (CAMÕES, 1818, Canto Primeiro, p.1).

Para as marcações de 1 a 6 na Figura 7, é inserida uma palavra por vez. A cada palavra inserida as dimensões da caixa de texto são calculadas. No momento em que a caixa de texto ultrapassa os limites da imagem, o texto é então reajustado inserindo-se uma quebra de linha antes da palavra que ultrapassou os limites da imagem. Para a linha seguinte, ao invés de se inserir uma palavra de cada vez, o que demandaria mais processamento, a função de ajuste foi otimizada de forma a considerar a quantidade de

caracteres que couberam na linha anterior. Deste modo, as palavras são acumuladas em um conjunto de caracteres (*i.e.*, *string*) e enquanto este número de caracteres não exceder o valor da linha anterior, a palavra não será inserida na imagem, a menos que esta seja a última palavra do texto. Em outras palavras, o referido processo pode ser resumido nas seguintes etapas (ver Algoritmo 3.1):

1. Comece inserindo uma palavra por vez para acomodação na imagem;
2. Se a palavra exceder o limite da imagem, insira uma quebra de linha antes dela e descarte a última imagem produzida (ver marcações 7 e 9 na Figura 7);
3. Para a linha seguinte, considere a quantidade de caracteres da linha anterior, acumule a palavra em uma *string* e realize a inserção apenas se o número de caracteres for igual ou ultrapassar o limite de caracteres suportado na linha anterior;
4. Se após inserir uma *string* ainda couber uma palavra na linha, realize a inserção e recalcule a quantidade caracteres suportados por linha. Caso seja superior ao limite anterior, atualize o limite;
5. Caso chegue na última palavra do texto e as palavras que estão sendo acumuladas na *string* ainda não tiverem sido inseridas na imagem, faça: Descarte a última imagem produzida, adicione a palavra à *string* e realize a inserção na imagem (ver marcações 9 e 10 na Figura 7).

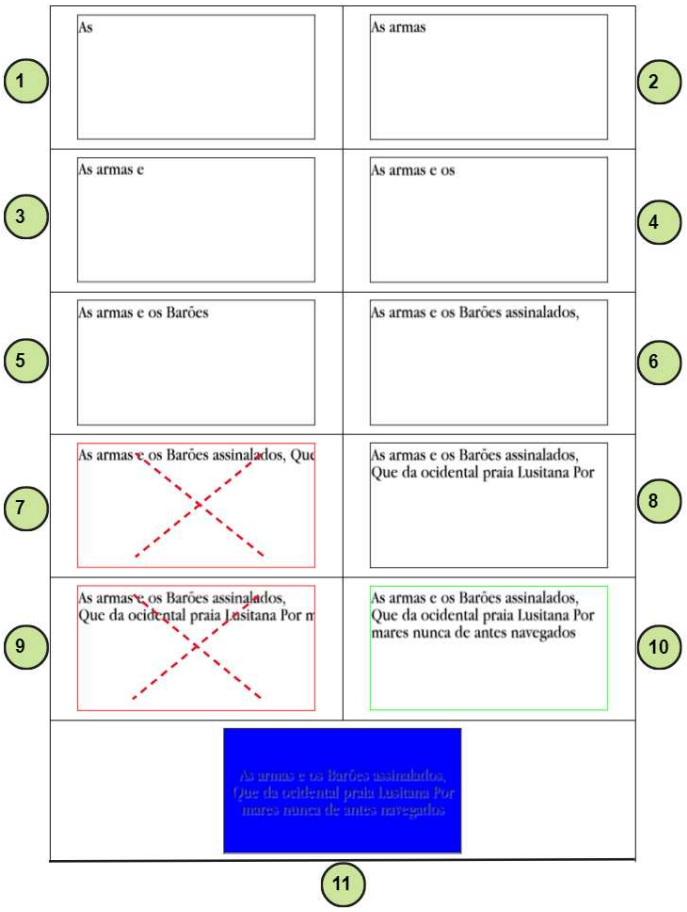


Figura 7 – Ilustração do ajuste de texto.

Algoritmo 3.1: Pseudocódigo da função de ajuste de texto.

```
início
    FUNÇÃO: ajustaTexto
    input: fonte: String, tamanhoDaFonte: Inteiro, dimensoes: Tupla, margem: Tupla, texto: Lista
    qtdPalavras ← 0: Inteiro
    limiteDeCaracteres ← 0: Inteiro
    text: String
    n ← qtdDePalavrasNoTexto: Inteiro
    para i ← 1 até n faça
        palavra ← texto[i]: String
        qtdPalavras ← +1
        aux ← text
        // Se a string text for vazia, faça text ← palavra.
        se tamanhoDoTexto == 0 então
            text ← qtdPalavras
            frase ← qtdPalavras
            aux2 ← frase
        // Senão, concatene a palavra à string text e à string frase.
        senão
            text ← text + palavra
            aux2 ← frase
            frase ← frase + palavra
        se qtdCaracteresNaFrase > limiteDeCaracteres ou qtdPalavras == n então
            imagemAuxiliar ← criaImagem(dimensoes)
            draw ← insereTextoNaImagem(text, imagemAuxiliar)
            larguraDaCaixaDeTexto ← dimensoesDaCaixaDeTexto[2] − dimensoesDaCaixaDeTexto[0]
            alturaDaCaixaDeTexto ← dimensoesDaCaixaDeTexto[3] − dimensoesDaCaixaDeTexto[1]
            // Se a largura da caixa de texto exceder os limites da imagem, reajuste o texto de
               forma a quebrá-lo em mais linhas.
            se larguraDaCaixaDeTexto > limitesDaImagem então
                text ← aux
                text ← text +' \n' + palavra
                limiteDeCaracteres ← len(aux2)
                frase ← palavra
                aux ← text
                draw ← insereTextoNaImagem(text, imagemAuxiliar)
                // Se mesmo reajustando o texto ele ainda exceder os limites da imagem, Retorne
                   Falso.
                se larguraDaCaixaDeTexto or alturaDaCaixaDeTexto > limitesDaImagem então
                    retorna False
        se larguraDaCaixaDeTexto or alturaDaCaixaDeTexto > limitesDaImagem então
            retorna False
    retorna text
    Fim
```

Quando a configuração exige que o texto seja rotacionado, a função que ajusta o texto deverá ser executada considerando margens mais espessas, com 150 pixels nos lados direito e esquerdo, isso é feito para ter um melhor aproveitamento do espaço no centro da imagem e permitir que sentenças muito longas sejam melhor acomodadas nas imagens. Quando a rotação não é exigida, as margens são de 5 pixels em todos os lados (superior, esquerda, inferior, direita). A Figura 8 mostra um exemplo do texto ajustado para a rotação e seu resultado final.

Por fim, para verificar se o procedimento de rotação não irá eliminar a informação textual, é importante verificar se as coordenadas dos pontos da caixa de texto não ultrapassarão os limites da imagem. Para isso, calculamos as coordenadas em que os pontos irão ficar antes de aplicar a rotação. Depois, precisamos centralizar a caixa de texto na origem, aplicar a fórmula da equação abaixo para, em seguida, reposicionar a caixa de texto na imagem. A partir da referida equação, é possível rotacionar uma figura em qualquer ângulo, em torno da origem $(0, 0)$.

$$
\begin{aligned}
x' &= x * \cos(\theta) - y * \sin(\theta) \\
y' &= x * \sin(\theta) - y * \cos(\theta)
\end{aligned}
\tag{3.1}
$$

Figura 8 – Exemplo do ajuste de texto para rotação.

3.1.2 Configuração do Gerador de Imagens

A Figura 9 exemplifica algumas imagens produzidas com a variação dos aspectos descritos na Seção 3.1. Após a execução do gerador de imagens, os dados ficam organizados em um conjunto principal, que por sua vez possui subconjuntos onde as imagens referentes a cada aspecto variado ficam agrupadas. O conjunto de imagens foi criado considerando todas as notícias presentes no `FACTCK.BR`, sendo que, para cada notícia, limitada a 280 caracteres (limite definido a partir do padrão de postagem da plataforma Twitter (agora X), com essa limitação, o número de notícias válidas passou de 1.314 para 1.208) foi variado um atributo de cada aspecto da imagem, com isso foi possível produzir um conjunto de dados contendo 52.525 imagens totais.

Figura 9 – Representação visual de imagens de amostra do ImageFactCk.BR.

Conforme mencionado no parágrafo anterior, os subconjuntos criados são referentes a cada aspecto que será avaliado, ou seja, ângulo, rotação, plano de fundo, etc. Para a criação desses subconjuntos, foi estabelecida uma configuração padrão para as imagens (ver Tabela 8) e a partir dessa configuração, começamos a manipular as variáveis de forma isolada. Assim, foram criados sete subconjuntos (ver Tabela 9):

Tabela 8 – Configuração padrão do gerador de imagens.

ASPECTO	VALOR
Ângulo de rotação do texto	0º
Dimensões	(1080 X 1920)px.
Estilo da fonte	Timeless
Cor da fonte	Preto
Tamanho da fonte	40pts.
Sombra	Desativada
Cor do plano de fundo	Branco

Tabela 9 – Conjuntos de dados gerados

NOME DO CONJUNTO DE DADOS	QUANTIDADE DE IMAGENS
PADRÃO VARIANDO ÂNGULO	15.296
PADRÃO VARIANDO COR DA FONTE	6.040
PADRÃO VARIANDO DIMENSÕES	4.832
PADRÃO VARIANDO ESTILO DA FONTE	10.754
PADRÃO VARIANDO PLANO DE FUNDO	9.664
PADRÃO VARIANDO SOMBRA	2.416
PADRÃO VARIANDO TAMANHO DA FONTE	3.523

Esses subconjuntos foram criados de forma que, para cada amostra de texto todas as possibilidades de um aspecto fossem aplicadas, caso não fosse possível produzir a imagem com algum valor assumido pelo aspecto, o texto seria descartado e o algoritmo gerador passaria para o próximo dado textual.

Por fim, de forma a organizar melhor os dados, o nome do arquivo contendo cada uma das imagens é constituído pelo seguinte padrão (*i.e.*, `regex`): Cada característica da imagem é separada pelo '-' na `regex`, assim, o 1º elemento representa o número da linha da planilha que contém a notícia no repositório de dados do `FACTCK.BR`, o 2º elemento representa o estilo da fonte, o 3º representa o tamanho da fonte, o 4º a cor da fonte, o 5º representa a sombra, o 6º a cor do plano de fundo, o 7º as dimensões da imagem, o 8º o ângulo de rotação do texto e, por fim, o 9º representa se a notícia presente na imagem contém ou não "desinformação", *i.e.*, é *fake* ou não (*True* para *fake* e *False* para representar as outras classes de notícia do FACTCK.BR). Um exemplo de nome de figura seria: `(8)-Timeless-40-(0,0,0)-False-(255,255,255)-(1080,1920)-5-True.jpg`. Por fim, é válido ressaltar que todas as imagens geradas estão no formato `".jpg"`. O conjunto de dados criado está disponível em: <https://zenodo.org/doi/10.5281/zenodo.10993043>.

3.2 Seleção das Ferramentas de Reconhecimento Óptico de Caracteres

Foi realizado um levantamento das ferramentas de OCR disponíveis on-line e acessíveis para serem incluídas nas análises. Para cada ferramenta foi buscada sua licença de uso, linguagem de programação, se é processamento em *cloud* ou aplicação local, os sistemas operacionais suportados, suporte ao Ptbr e as limitações conhecidas. A Tabela 10 apresenta os detalhes de cada ferramenta.

Tabela 10 – Seleção das ferramentas de OCR.

Selecionado	Mecanismo	Licença	Suporte ao Português	Local/ Cloud	Linguagem de Programação	Sistema Operacional	Limitações
✓	PyTesseract v. 5.2.0	Livre	Sim	Local	Python /C/C++	Windows/ Linux/ Android/ MAC OS	Python >= 3.7
X	Calamari v.2.2.2	Livre	Não	Local	Python	Linux/ Windows/ Mac OS	Downgrade para o Python 3.7
X	OCRopus	Livre	Não	Local	Python	MAC OS/ Linux/ BSD	—
X	OpenCV EAST	Livre	Independente de linguagem	Local	Python	Linux/ MAC OS	As imagens devem ter dimensões múltiplas de 32.
X	ABBYY v.0.3	Paga	Sim	Cloud	Python/Java/ PHP/ JavaScript/ Ruby	Cloud service	Python>=3.8<=3.11. Sendo 90 dias de teste com um limite de 500 imagens.
✓	Microsoft OCR v.3.2	Paga	Sim	Local/ Cloud	Python/C#/ Java/Ruby	Cloud service	30 dias de avaliação com USD $200 de crédito Azure cedidos para o primeiro cadastro. Sendo 1.000 consultas por $1.
✓	Google Vision v.3.4.2	Paga	Sim	Cloud	PythonGo/ Java/Node.js/ Python/Ruby/ PHP/C#/C++	Cloud service	90 dias de avaliação. Python>=3.6
X	Amazon v.1.3.1	Paga	Sim	Cloud	.NET/Python/ Java/PHP/ JavaScript	Cloud service	Sem versão de avaliação. Free Container com um limite de 5.000 requisições por mês, limitado a 20 requisições/min.
✓	EasyOCR v.1.7.0	Livre	Sim	Local	Python	Linux/ Windows	Python>=3.8<=3.11
	Online OCR	Paga	Sim	Cloud	Java/C#/PHP	Linux/ Windows/ Mac OS	30 dias de avaliação com um limite de 25 imagens/dia.
X	Adobe OCR	Paga	Não	Cloud	Java/.NET/ Node.js/ Rest API	Linux/ Windows/ Mac OS	6 meses de avaliação limitado a aproximadamente 10.000 requisições. Apenas arquivos PDF.
✓	Google Drive OCR v.0.2.6	Livre	Sim	Cloud	Go/Java/ Node.js/C# Python/C++/ Ruby/PHP	CLoud service	Python>=3.6

Analisando o resultado da pesquisa, a seleção das ferramentas se iniciou verificando os critérios Licença e Suporte ao Português. As ferramentas que oferecem suporte à Língua Portuguesa do Brasil (Ptbr) são nove, no entanto, ao analisar as suas licenças, selecionamos as que permitem processar todo o conjunto de dados (52.525 imagens) sem custo, dessa forma, o número de ferramentas foi reduzido para seis. Após isso, voltamos nossa atenção para os sistemas operacionais suportados por cada ferramenta, priorizamos as ferramentas que podem ser executadas no Linux por ser um sistema de código aberto, felizmente todas as ferramentas levantadas oferecem suporte a essse sistema operacional. Em seguida, procuramos avaliar as ferramentas em uma linguagem de programação que fosse comum a todas, com isso, verificamos que a linguagem *Python* era a linguagem que todas ofereciam

suporte. Executar os experimentos em uma linguagem de programação que seja comum a todas as ferramentas é parte essencial na execução dos testes, pois ao se analisar o tempo de execução das ferramentas o comparativo fica mais justo, haja vista que linguagens de baixo nível fornecem desempenho extremamente rápido se comparadas às de alto nível (ALI; QAYYUM, 2021). Em sequência, verificamos as limitações de cada ferramenta, das seis analisadas apenas uma foi descartada, que é o OpenCV EAST[5], as dimensões de imagem suportadas por essa ferramenta devem ser múltiplas de 32 (tanto na largura como na altura) e isso a inviabiliza de ser aplicada no nosso conjunto de dados, com isso temos que cinco ferramentas foram selecionadas nesta etapa. Por fim, foi avaliado o tipo de processamento de cada ferramenta, se era local ou em nuvem (*cloud*), isso foi levantado pelo motivo de executarmos medições de tempo de processamento em nosssas análises e o tipo de processamento influencia nesse quesito. Com isso, selecionamos cinco ferramentas conforme apresentadas na Tabela 10.

3.3 Avaliação das Ferramentas de OCR

Para a avaliação, foram considerados critérios como tempo de processamento, qualidade dos resultados e capacidade do OCR de produzir algum resultado para a instância a qual foi submetido. Como o objetivo desta análise é identificar os pontos de deficiência das ferramentas por aspecto de forma isolada, o tipo de análise empregada foi univariada. Esse tipo de análise é aplicada quando se precisa sintetizar a distribuição de uma única variável (COIMBRA et al., 2007). No contexto deste trabalho, buscamos entender quais aspectos de imagem produzem influência negativa nos OCRs e quais configurações desses aspectos são mais impactantes.

Cada conjunto de dados, descrito na Tabela 9, foi submetido nas ferramentas de OCR, e estas, por sua vez, foram avaliadas com as seguintes métricas:

- Qualidade: CER, WER, Distância de Levenshtein (Descritas na Seção 2.2.1 do Capítulo 2);
- Erro binário: Acertos, Erros e Sem Resultados (Descritas na Seção 2.2.2 do Capítulo 2);
- Tempo de execução (Descrito a seguir).

Para o cálculo do CER e WER, foi utilizada a biblioteca *"torchmetrics"* [6] também explorada em esforços anteriores (DETLEFSEN et al., 2022), as métricas de CER e WER dessa biblioteca seguem a implementação convencional, ou seja, não são normalizadas. Já para o cálculo da Distância de Levenshtein foi utilizada a biblioteca *"python-Levenshtein"*[7] (BACHMANN, 2022).

A medição do tempo de processamento das ferramentas foi executado observando os seguintes casos:

- i) Medindo o tempo geral de execução do processo e, ii) o tempo de execução do processo no núcleo do processador;

[5] <https://docs.opencv.org/4.x/d8/ddc/classcv_1_1dnn_1_1TextDetectionModel__EAST.html>
[6] <https://lightning.ai/docs/torchmetrics/stable/>
[7] <https://pypi.org/project/python-Levenshtein/>

- Considerando a etapa de pós-processamento. O resultado obtido por algumas ferramentas de OCR, além de conterem marcadores inseridos pela ferramenta, se encontram em diferentes formatos de arquivo. Por isso uma etapa de pós-processamento se faz necessária, para padronizar o formato dos resultados das ferramentas de OCR e inseri-los no algoritmo de avaliação que vai aplicar as métricas de qualidade (veja na Figura 10 um exemplo de dado retornado por um sistema de OCR e sua respectiva padronização). Com isso, a medição é feita da seguinte forma: i) considerando o tempo geral de execução do processo, incluindo o pós-processamento e ii) considerando o tempo de execução do processo no núcleo do processador, também incluindo o pós-processamento.

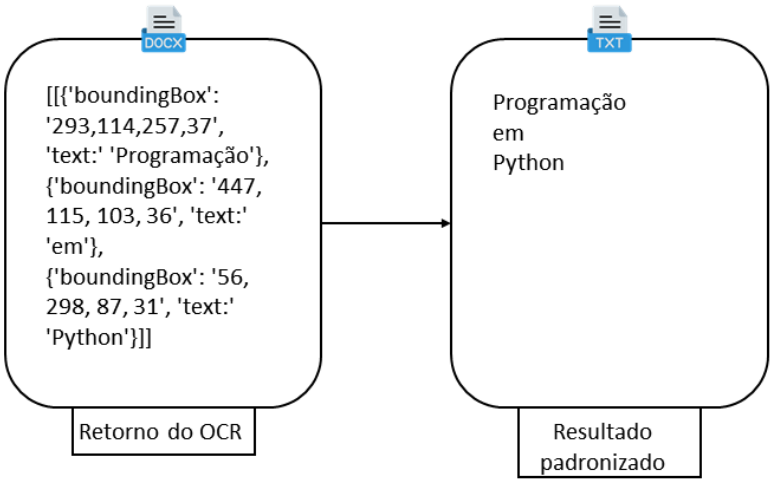


Figura 10 – Exemplo de padronização dos dados.

Com a execução da análise univariada, podemos identificar as variáveis que produzem maior impacto na qualidade dos resultados dos mecanismos de OCR. A partir daí, é possível discutir os pontos de melhoria para essas ferramentas e definir qual, ou quais, ferramentas são mais adequadas para lidar com determinados tipos de dados.

4 Avaliação Experimental das Ferramentas de OCR

Neste capítulo são relatadas as configurações e os resultados obtidos pelos experimentos conduzidos através da metodologia descrita. Inicialmente são relatadas as especificações da máquina de testes. Por fim, são descritos os resultados de avaliação das ferramentas de OCR.

4.1 Especificações da Máquina de Execução dos Experimentos

Os testes com as ferramentas de OCR foram executados em uma máquina com a seguinte configuração:

- RAM: 32,0 GiB. DIMM DDR4 2133 MHz;
- Armazenamento: 490,6 GB. HDD;
- Processador: Intel® Core™ i7-6700 CPU @ 3.40GHz × 8;
- GPU: Mesa Intel® HD Graphics 530 (SKL GT2) / NVE7;
- Sistema Operacional: Ubuntu 22.04.3 LTS de 64 bits.

4.2 Experimentos com as Ferramentas de OCR

O conjunto de dados descrito no Capítulo 3 e detalhado na Tabela 9 foi utilizado para avaliação de desempenho das ferramentas de OCR selecionadas (Tabela 10). Para uma melhor organização das informações, os resultados de cada aspecto de imagem investigado nos OCRs estão descritos nas seções seguintes.

4.2.1 Métrica CER

Para essa métrica, quanto menor o valor, melhor é o desempenho do OCR. A Figura 11 apresenta a taxa média de CER por aspecto, cada barra representa a taxa média de caracteres errôneos produzidos por um sistema de OCR medido em comparação com a verdade fundamental (conforme observado em (REUL et al., 2019; SPRINGMANN; LUEDELING, 2017), OCRs que apresentam o valor de CER igual ou inferior a 10% são considerados muito eficientes e os que apresentam o valor igual ou inferior a 5% são excelentes). Já a Figura 12 apresenta o desvio de cada aspecto da imagem na métrica CER, que podemos interpretar como o efeito de cada aspecto da imagem na precisão do OCR. Cada barra representa o desvio entre o CER médio de um grupo de imagens com a configuração padrão dos aspectos da imagem variando do aspecto selecionado.

Analisando os valores apresentados nas Figuras 11 e 12, destacamos que o ângulo de rotação do texto, o plano de fundo da imagem e o estilo da fonte são os aspectos da imagem com maior impacto no desempenho geral dos sistemas OCR avaliados. Explorando ainda mais essas variáveis, apresentamos mapas de calor nas Figuras 13, 14 e 15 para

demonstrar o efeito de cada aspecto da imagem na precisão do OCR. Através dos mapas de calor, podemos observar especificamente quais configurações de aspecto estão impactando negativamente no desempenho das ferramentas, o que nos permite detectar padrões. Para os mapas de calor, cada célula representa o CER médio calculado no grupo de imagens com configuração padrão dos aspectos da imagem variando o aspecto analisado (ângulo, plano de fundo e estilo da fonte), e as cores variam do vermelho (pior) ao azul (melhor).

A partir dos mapas de calor, notamos que, com exceção do Microsoft OCR, quanto maior o ângulo de rotação, tanto no sentido horário quanto no sentido anti-horário, piores serão os resultados. Ao analisar a variação do fundo, notamos que imagens com fundos não sólidos aumentam as taxas de CER, mesmo para o Microsoft OCR, indicando outra característica que tem impacto significativo na qualidade dos resultados do OCR. Para o PyTesseract, em uma das imagens de fundo, o mecanismo não conseguiu produzir nenhum resultado. Este foi o único sistema avaliado que apresentou falha crítica.

Ao realizar uma análise abrangente, voltamos nosso foco para uma compreensão mais profunda dos resultados apresentados na Figura 12. Nesta discussão, investigamos a influência de cada aspecto da imagem na precisão do OCR.

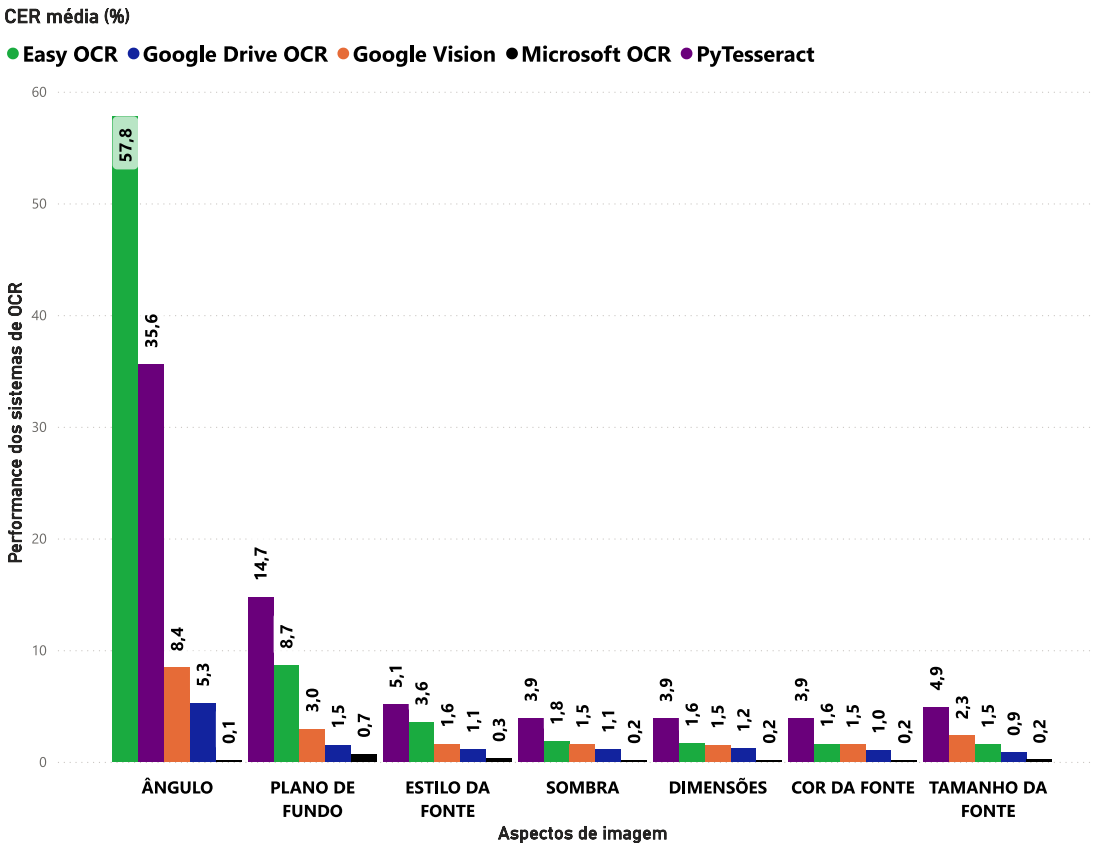


Figura 11 – Gráfico da média de CER por aspecto de imagem.

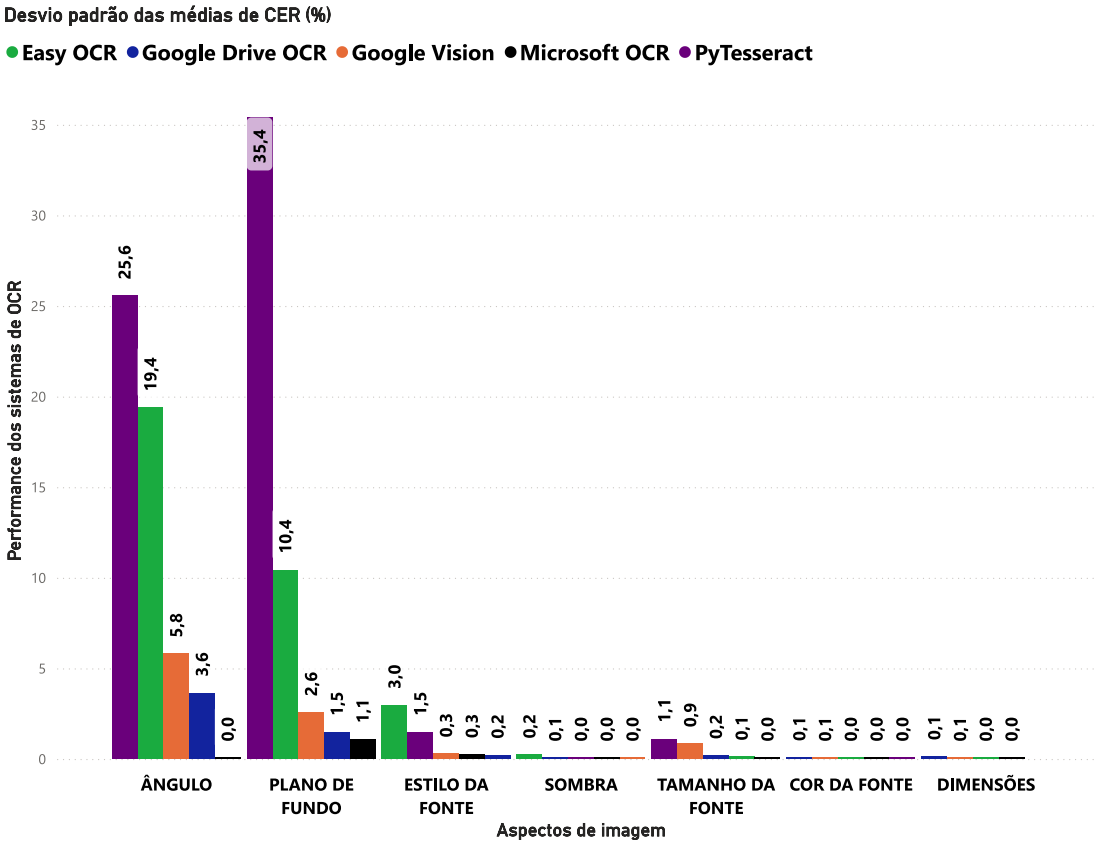


Figura 12 – Gráfico do desvio de CER causado pelos aspectos da imagem analisada.

- **Aspecto Ângulo:** o PyTesseract e o Easy OCR obtiveram os maiores desafios no tratamento da rotação de texto entre os sistemas de OCR avaliados, com CER médio de 35,6% e 57,8% respectivamente e com um desvio das taxas CER de 25,6% e 19,4%, respectivamente. Explorando ainda mais esta variável com as informações apresentadas na Figura 13, pode-se observar que quanto mais atenuado for o ângulo de rotação do texto, pior será o desempenho dessas ferramentas. O OCR do Google Drive, o Google Vision e o OCR da Microsoft lidaram bem com a rotação de texto, com as médias e desvios de CER abaixo de 10%. Dentre essas ferramentas, destacou-se o OCR da Microsoft, com desvio de CER de 0,01% e média de CER de 0,1%, essa foi a única ferramenta analisada que não sofreu impacto com a variação do ângulo.

	-30	-25	-20	-15	-10	-5	0	5	10	15	20	25	30
PyTesseract	85.6	76.3	62.6	52.7	44.5	23.6	3.9	37.7	55.4	68.4	78.8	82.7	86.8
Microsoft OCR	0.1	0.2	0.2	0.1	0.1	0.1	0.2	0.1	0.2	0.2	0.1	0.1	0.2
Google Vision	11.9	9.2	6.6	5.0	4.2	3.4	1.5	4.2	4.6	8.6	12.3	16.8	21.4
Google Drive OCR	10.9	8.0	4.9	3.6	2.9	2.1	1.1	2.1	2.4	3.2	6.1	9.5	11.7
Easy OCR	62.9	62.0	61.8	58.8	54.6	40.0	1.7	55.4	65.8	70.6	72.5	73.9	74.0

100
50
25
10
2
0

Figura 13 – Mapa de calor da média de CER que exemplifica o impacto da rotação do texto na precisão do OCR.

- **Aspectos Dimensões e Cor da Fonte:** todos os sistemas de OCR avaliados lidaram bem com a mudança no tamanho da imagem e na mudança da cor da fonte, com um desvio das taxas de CER próximo de 0% para todos os sistemas. Os sistemas Google Vision e Google Drive, foram os únicos que apresentaram um ligeiro desvio de 0,1% em ambos os aspectos. Ao analisar a média de CER, todas as ferramentas

apresentaram bons resultados, se mantendo abaixo de 4%. O Microsoft OCR foi a ferramenta que apresentou o melhor resultado de média de CER sendo 0,2% para os dois aspectos analisados, o pior desempenho foi do PyTesseract com média de CER de 3,9% para os dois aspectos analisados.

- **Aspecto Plano de Fundo:** Pytesseract e Easy OCR alcançaram um desvio de CER de 35,4% e 10,4% e um CER médio de 14,7% e 8,7% ao alterar o fundo da imagem. O Google Vision obteve 2,6% de desvio e 3% de CER médio, o Google Drive OCR obteve um desvio de 1,5% e um CER médio de 1,5% e o Microsoft OCR obteve os melhores resultados, com um desvio das taxas de CER de 1,1% e um CER médio de 0,7%. Na Figura 14, pode-se observar que as ferramentas que apresentaram alta variação no desvio do CER, como PyTesseract e Easy OCR, não tratam bem o caso de imagens com tons escuros combinados com fonte escura.

	Branco	Azul	Azul medio	Azul claro	Azul escuro	im1	im2	100
PyTesseract	3.9	37.8	3.9	5.5	46.8	Sem resultados	93.1	50
Microsoft OCR	0.2	0.2	0.2	0.2	0.2	1.1	3.0	25
Google Vision	1.5	1.5	1.4	1.4	1.5	7.1	6.3	10
Google Drive OCR	1.1	0.9	0.7	0.7	0.9	1.3	4.8	2
Easy OCR	1.7	3.9	1.5	1.5	25.1	4.4	22.4	0

Figura 14 – Mapa de calor da média de CER que exemplifica o impacto do plano de fundo na precisão do OCR.

- **Aspecto Estilo da Fonte:** alguns sistemas de OCR, como o PyTessarct e o Easy OCR obtiveram um desvio superior a 1%, enquanto os outros sistemas alcançaram um desvio menor. O Easy OCR foi o mais impactado pela mudança no estilo da fonte, com desvio de 3,0% e CER médio de 3,6%. O Microsoft OCR, o Google Vision e o Google Drive OCR tiveram um bom desempenho, com um desvio inferior a 0,4%. O melhor desempenho de média de CER foi do Microsoft OCR, com 0,3%. Explorando mais esse aspecto e, com base no mapa de calor apresentado na Figura 15, é possível notar que fontes com um padrão de design comumente usadas pela imprensa e trabalhos científicos, como APpleGaramond, TypeMachine e Timeless produzem pouco impacto no desempenho dos OCRs. Fontes com um traço de design fino como KGPartofME e NixieOne produziram um grande impacto no PyTesseract, sendo que destas, especificamente a KGPartofME produziu maior impacto. Já as fontes estilizadas como KGLegoHouse e estilizadas com letras exclusivamente em caixa alta como ModernSerif e KGSorryNotSorry foram as responsáveis pelos maiores impactos nas ferramentas Microsoft OCR, Google Vision e EasyOCR e, embora esse estilo de fonte não seja o responsável pelo pior impacto do PyTesseract e do Google Drive OCR, elas também tiveram interferência significativa para esses OCRs.

	ComicSansMS3	Modern Serif	KGPartofMe	AppleGaramond	Timeless	Type Machine	NixieOne	KGLegoHouse	KGSorryNotSorry	100
PyTesseract	3.97	4.76	8.1	3.49	3.88	4.64	6.17	6.33	4.87	50
Microsoft OCR	0.15	0.21	0.65	0.15	0.15	0.14	0.15	0.74	0.59	25
Google Vision	1.48	1.72	1.75	1.24	1.54	1.77	1.56	2.11	1.19	10
Google Drive OCR	1.06	0.82	1.21	1.43	1.06	1.34	0.75	1.11	1.03	2
Easy OCR	1.54	2.03	4.51	1.85	1.65	1.88	2.52	5.37	10.58	0

Figura 15 – Mapa de calor da média de CER que exemplifica o impacto do estilo da fonte na precisão do OCR.

- **Aspecto Sombra:** todos os sistemas avaliados foram capazes de lidar bem com a variação na sombra do texto, alcançando um desvio das taxas de CER de até

0,23%. Nesta análise, os sistemas que apresentaram menor desvio na precisão ao alterar a sombra do texto foi o Microsoft OCR e o Google Vision, ambos com 0,01%. Google Drive OCR e PyTesseract obtiveram valores próximos, sendo 0,05% e 0,03% respectivamente. A maior taxa de desvio foi obtida pelo Easy OCR, com desvio de 0,23%. De forma geral, a sombra produziu pouco impacto na qualidade dos OCRs.

- **Aspecto Tamanho da Fonte:** os sistemas avaliados demonstraram manipulação eficaz das variações no tamanho da fonte dos textos incorporados. Nesta análise, o melhor desempenho foi alcançado pelo Microsoft OCR, com desvio de 0,04% e CER médio de 0,2%. O Google Vision alcançou um desvio de 0,85% com CER médio de 2,3% e o PyTesseract obteve um desvio de 1,10% com CER médio de 4,9%, enquanto os demais ficaram próximos de um desvio de 0% e CER médio abaixo de 2%. Explorando mais a fundo esse aspecto e, com base na Figura 16, percebemos que, com exceção do Google Drive OCR e do Easy OCR que adotaram comportamento oposto, a medida em que o tamanho da fonte aumentava, o desempenho do reconhecimento de texto diminuía.

	40	60	80
PyTesseract	3.88	4.89	6.08
Microsoft OCR	0.15	0.17	0.22
Google Vision	1.54	2.27	3.24
Google Drive OCR	1.06	0.83	0.67
Easy OCR	1.65	1.42	1.54

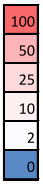


Figura 16 – Mapa de calor da média de CER que exemplifica o impacto do tamanho da fonte na precisão do OCR.

4.2.2 Métrica WER

Os valores WER obtidos não representam impacto significativo na qualidade das análises, isso porque diferentemente dos caracteres, as palavras têm comprimentos diferentes e não permitem uma comparação clara. Em outras palavras, se uma única letra estiver incorreta, a palavra inteira será classificada como incorreta. No entanto, o WER fornece indicações da taxa de precisão esperada para sistemas OCR. Para essa métrica, quanto menor o valor, melhor é o desempenho do OCR.

Ao analisar a média e o desvio do WER, conforme mostrado nas Figuras 17 e 18, notamos que além do ângulo e do fundo, outra variável que impacta no desempenho dos mecanismos de OCR é o estilo da fonte. O impacto desse aspecto não fica tão evidente na análise do CER. Além disso, ao examinar a métrica WER, notamos os seguintes pontos [1]:

[1] * Observação: Conforme mencionado no Capítulo 2, os valores de WER, apresentados a seguir, podem exceder os 100%. isso acontece quando há muitas inserções indevidas de palavras.

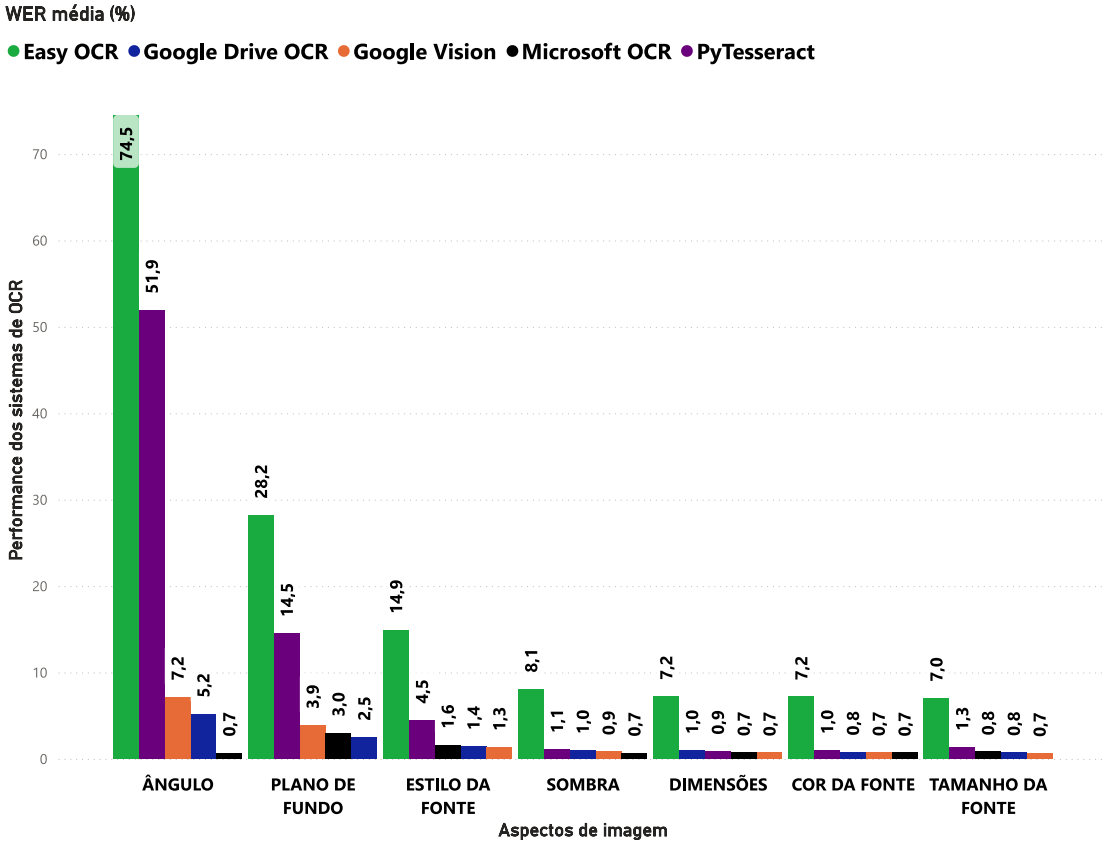


Figura 17 – Gráfico da média de WER por aspecto de imagem.

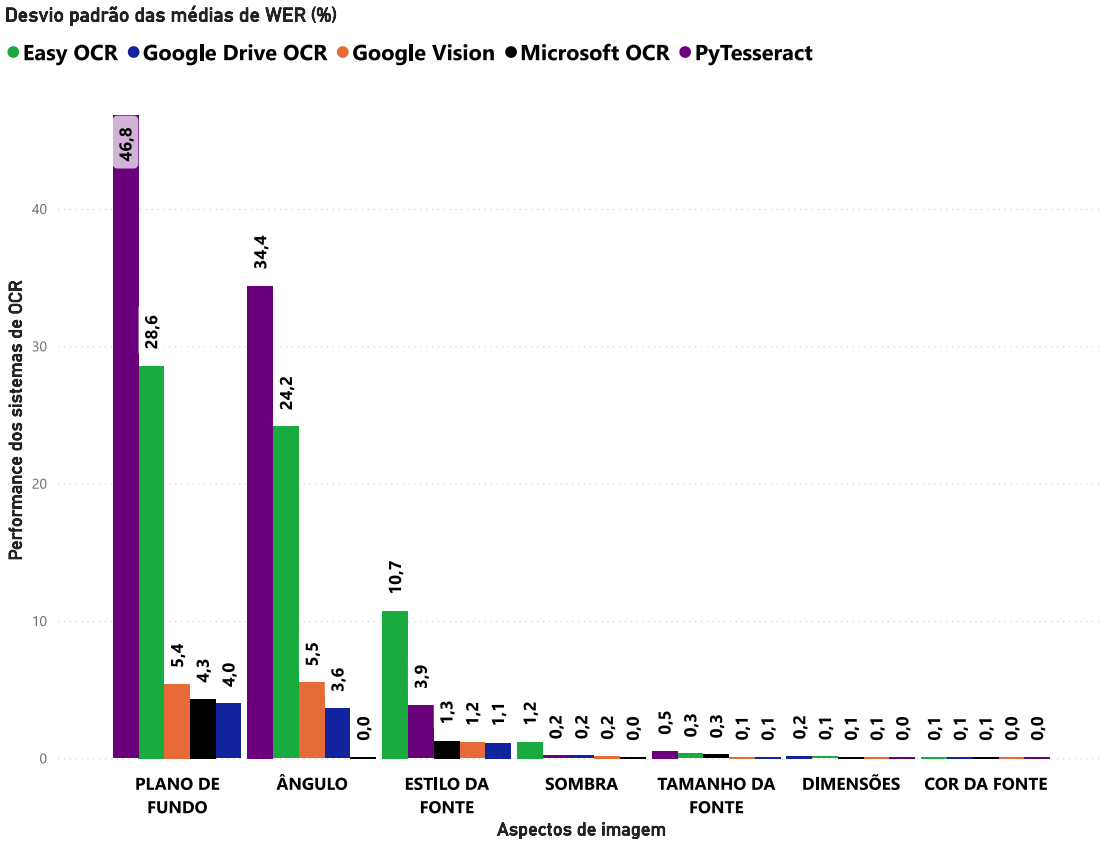


Figura 18 – Gráfico do desvio de WER causado pelos aspectos da imagem analisada.

- **Aspecto Ângulo:** Os piores valores de desvio WER foram obtidos pelo PyTesseract 34, 4% e pelo Easy OCR 24, 2%, com médias de WER de 51,9% e 74,5% respectivamente. O Microsoft OCR foi a ferramenta que apresentou os melhores resultados, com valores de média e desvio de WER de 0,7% e 0,04% respectivamente.

	-30	-25	-20	-15	-10	-5	0	5	10	15	20	25	30
PyTesseract	115	111	103	95	77	36	0.9	48	81	104	106	102	102
Microsoft OCR	0.8	0.7	0.7	0.7	0.6	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.7
Google Vision	12	8.5	5.5	3.8	2.9	2.1	0.7	3	3.5	7.3	11	15	19
Google Drive OCR	12	8.5	5.1	3.7	2.8	2.2	0.8	2.2	2.4	3	5.7	9	11
Easy OCR	82	80	78	74	68	51	7.2	68	84	90	94	98	97

100
50
25
10
2
0

Figura 19 – Mapa de calor da média de WER que exemplifica o impacto da rotação do texto na precisão do OCR.

- **Aspecto Plano de Fundo:** Nesta análise, assim como no ângulo, apenas Easy OCR e PyTesseract produziram resultados ruins, com 28,6% e 46,8% de desvio respectivamente e, 28,2% e 14,5% de média de WER. O melhor resultado foi do Google Drive OCR, com desvio de WER de 4,0% e média de 2,5%.

	Branco	Azul	Azul medio	Azul claro	Azul escuro	im1	im2
PyTesseract	0.93	43.4	0.92	3.58	53.01	Sem resultados	120.25
Microsoft OCR	0.68	1.09	0.78	0.79	0.88	4.41	12.28
Google Vision	0.74	0.92	0.61	0.62	0.82	10.74	12.69
Google Drive OCR	0.82	0.72	0.44	0.43	0.86	3.05	11.34
Easy OCR	7.24	20.69	6.64	6.65	79.05	20.26	56.96

100
50
25
10
2
0

Figura 20 – Mapa de calor da média de WER que exemplifica o impacto do plano de fundo na precisão do OCR.

- **Aspecto Estilo da Fonte:** Apenas o Easy OCR teve desempenho insatisfatório nesta análise, com desvio de WER de 10,7% e média de WER de 14,9%.

	ComicSansMS3	Modern Serif	KGPartofMe	AppleGaramond	Timeless	Type Machine	NixieOne	KGLegoHouse	KGSorryNotSorry
PyTesseract	0.82	3.18	8.91	0.98	0.93	3.74	2.76	10.56	8.74
Microsoft OCR	0.83	0.96	3.36	0.73	0.68	0.71	0.54	3.44	2.9
Google Vision	0.52	0.4	2.63	0.54	0.74	0.57	0.6	3.09	2.78
Google Drive OCR	0.63	0.37	2.39	0.93	0.82	0.97	0.37	2.87	3.32
Easy OCR	7.63	9	18.72	8.92	7.24	7.98	9.91	27.76	36.9

100
50
25
10
2
0

Figura 21 – Mapa de calor da média de WER que exemplifica o impacto do estilo da fonte na precisão do OCR.

- **Outros aspectos:** Os valores de média e de desvio do WER ficaram abaixo de 10%. Para sombra, o melhor resultado foi obtido pelo Microsoft OCR, com 0,02% de desvio de WER e média de 0,7%. Google Vision, Google Drive OCR e PyTesseract também apresentaram excelentes resultados, girando em torno de 0,20% e valores de média entre 0,9% e 1,1%. O Easy OCR apresentou o pior resultado para sombra comparado aos demais sistemas avaliados, mas permaneceu abaixo de 2% no desvio do WER e com média de WER de 8,1%. Para as demais variáveis, todas as ferramentas tiveram bom desempenho, com valores médios de desvio do WER variando de 0 a 0,52% e valores de média de WER inferiores à 10%.

4.2.3 Métrica Distância de Levenshtein

Para essa métrica, quanto menor o valor, melhor é o desempenho do OCR. O aspecto ângulo foi o que produziu os maiores valores para a Distância de Levenshtein, sendo que nesse aspecto, o Easy OCR foi a ferramenta que produziu o maior valor médio de distância de edição. Ainda com relação ao ângulo, o melhor resultado pôde ser obtido pelo Microsoft OCR com distância de edição média igual a 0,1. Analisando o impacto do plano de fundo, o PyTesseract e o Easy OCR foram os sistemas que apresentaram maiores dificuldades ao lidarem com essa variação de aspecto, apresentando uma pontuação média de 16,4 e 8,7, respectivamente. As demais ferramentas obtiveram excelentes resultados ao lidarem com o plano de fundo, com uma média inferior a 5. Com relação ao aspecto cor da fonte e para os demais, o PyTesseract apresentou o pior desempenho com valores médios oscilando entre 3,5 e 4,5. As demais ferramentas lidaram bem com os demais aspectos, com pontuações médias inferiores a 3 (Ver Figura 22).

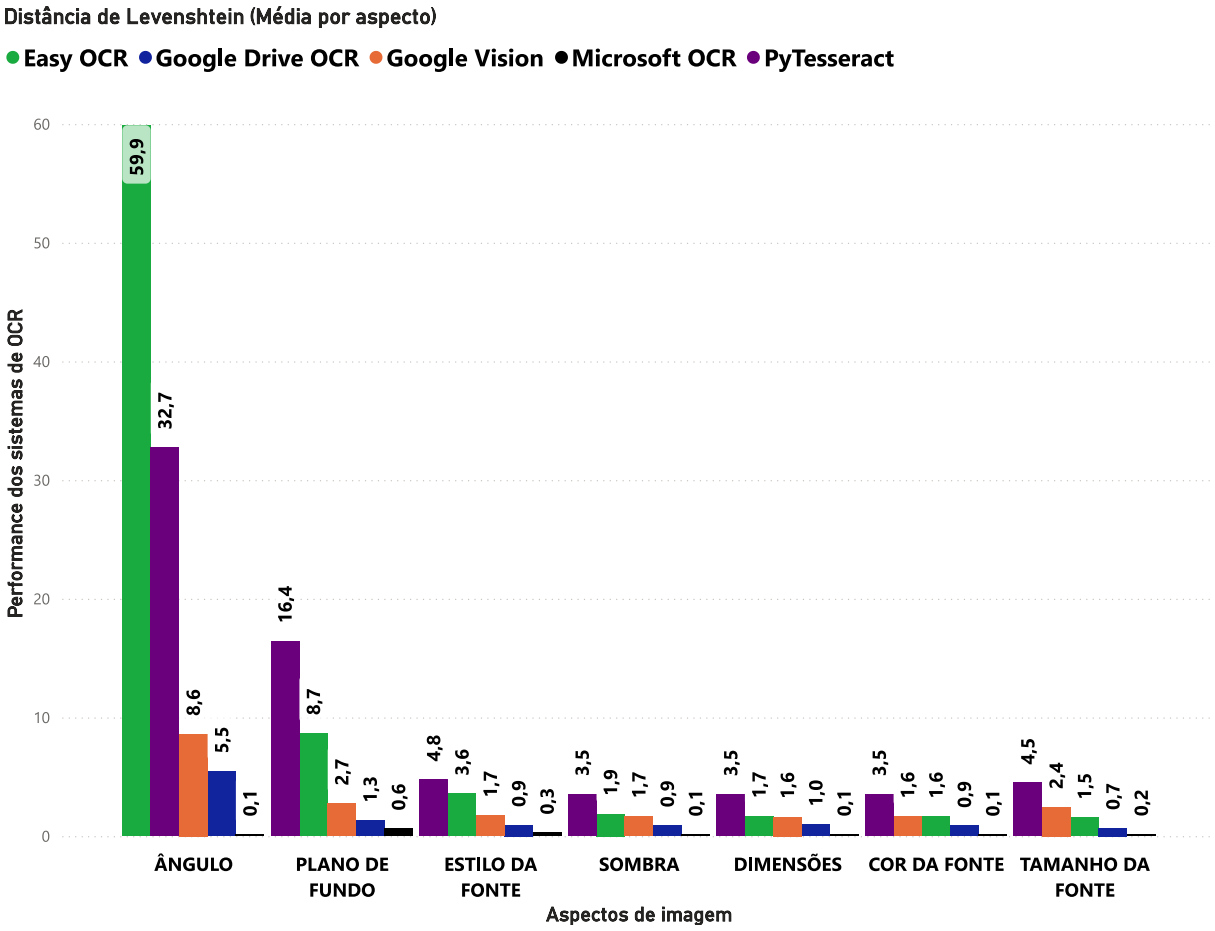


Figura 22 – Gráfico dos resultados de média da Distância de Levenshtein por aspecto.

4.2.4 Métricas de Avaliação Binárias

A condução dos experimentos com as métricas binárias e os valores apresentados nas Figuras 23, 24, 25, 26 e 27, procuram responder as seguintes perguntas: Qual a taxa de imagens que o OCR não conseguiu produzir nenhum resultado por variação de aspecto? Das imagens que o OCR conseguiu produzir algum resultado, quantas ele acertou 100%?

A taxa de acertos apresentada nas Figuras 23, 24, 25, 26 e 27, indicam a proporção de imagens de cada aspecto em que foi possível obter um resultado 100% correto, a taxa de erros e sem resultado, capturam os demais casos.

Com isso, temos que: O aspecto plano de fundo foi o aspecto comum de falhas na produção de resultados no Google Drive OCR (0,01%) e no PyTesseract (41,41%). O PyTesseract foi o que mais falhou, sendo que suas deficiências são mais agravantes nos aspectos ângulo e plano de fundo. As maiores taxas de acerto puderam ser observadas pelo Microsoft OCR, Google Drive OCR e Google Vision, enquanto que as maiores taxas de erro foram do Easy OCR e PyTesseract respectivamente. Em suma, com exceção do PyTesseract e do Google Drive OCR, nenhum sistema de OCR deixou de produzir resultados para as imagens a que foram submetidos.

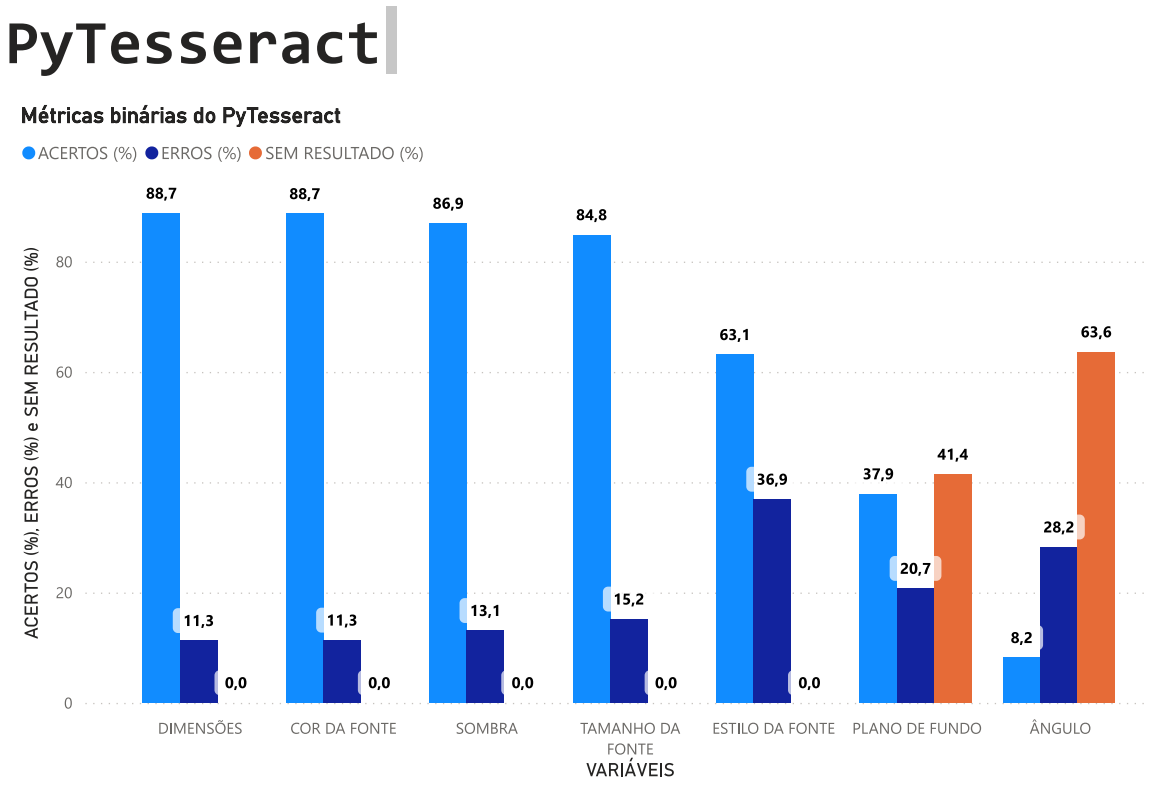


Figura 23 – Resultados das métricas binárias para o PyTesseract por variação de aspecto.

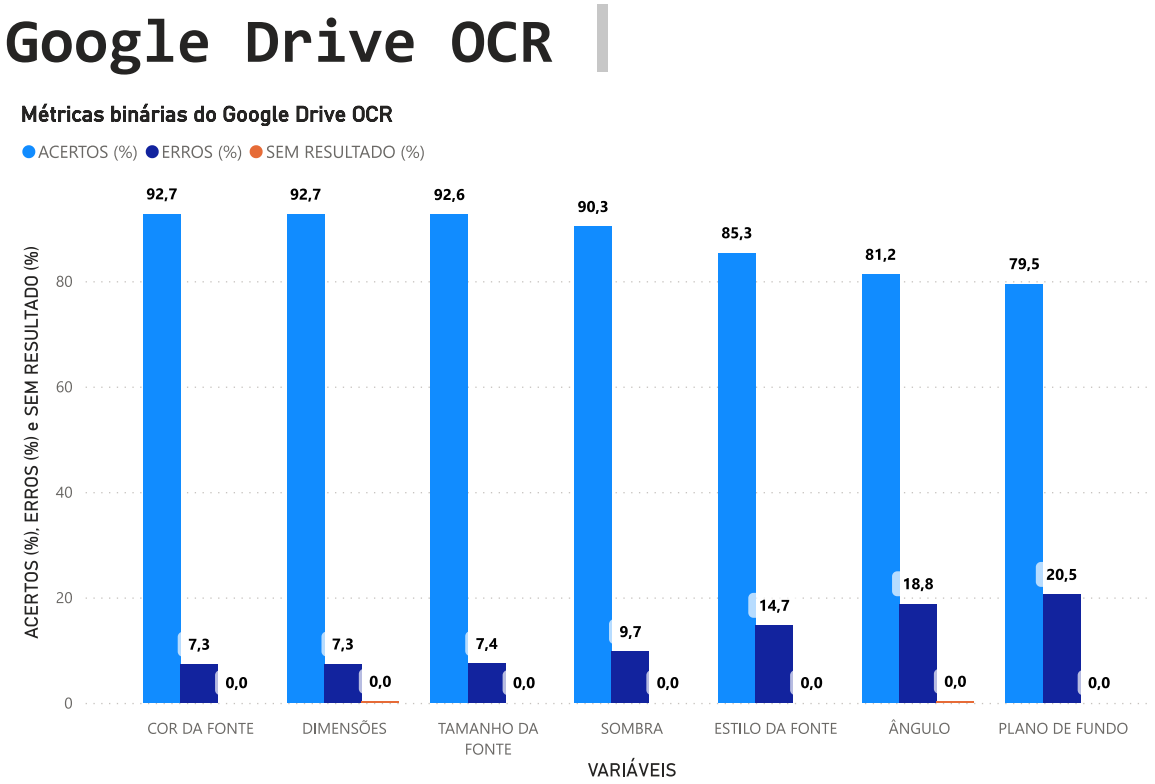


Figura 24 – Resultados das métricas binárias para o Google Drive OCR por variação de aspecto.

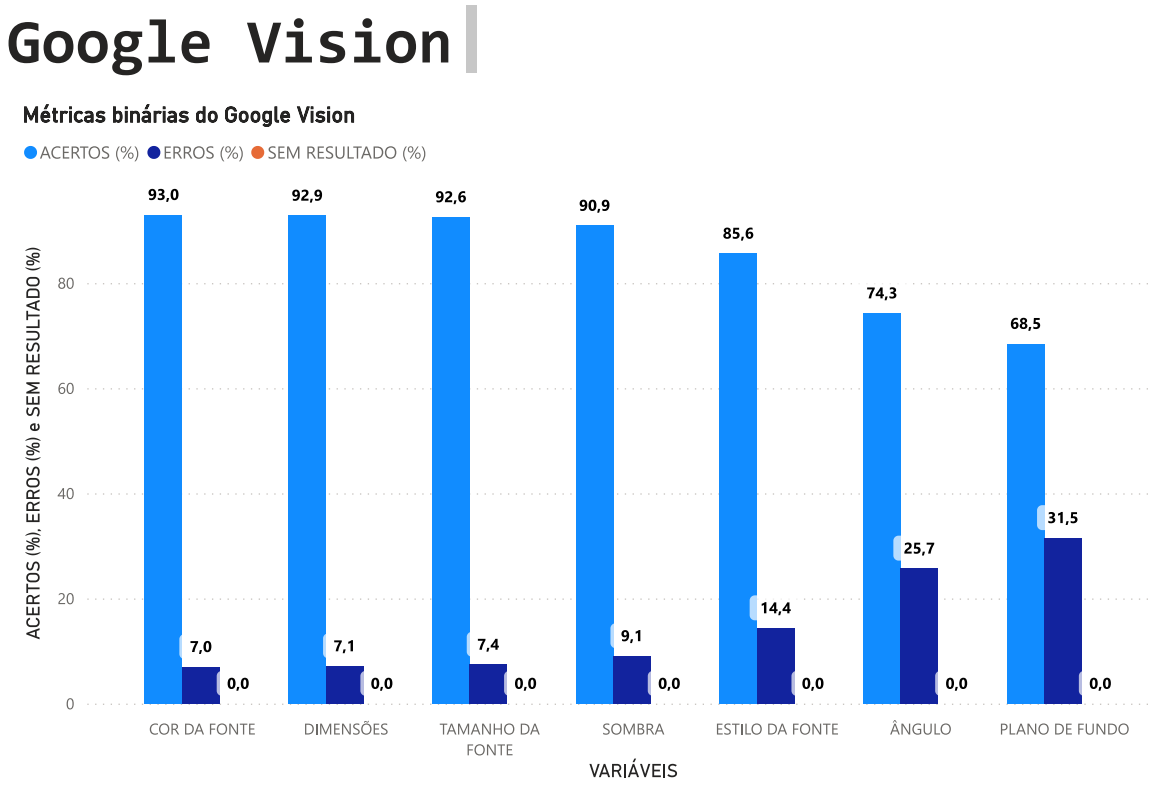


Figura 25 – Resultados das métricas binárias para o Google Vision por variação de aspecto.

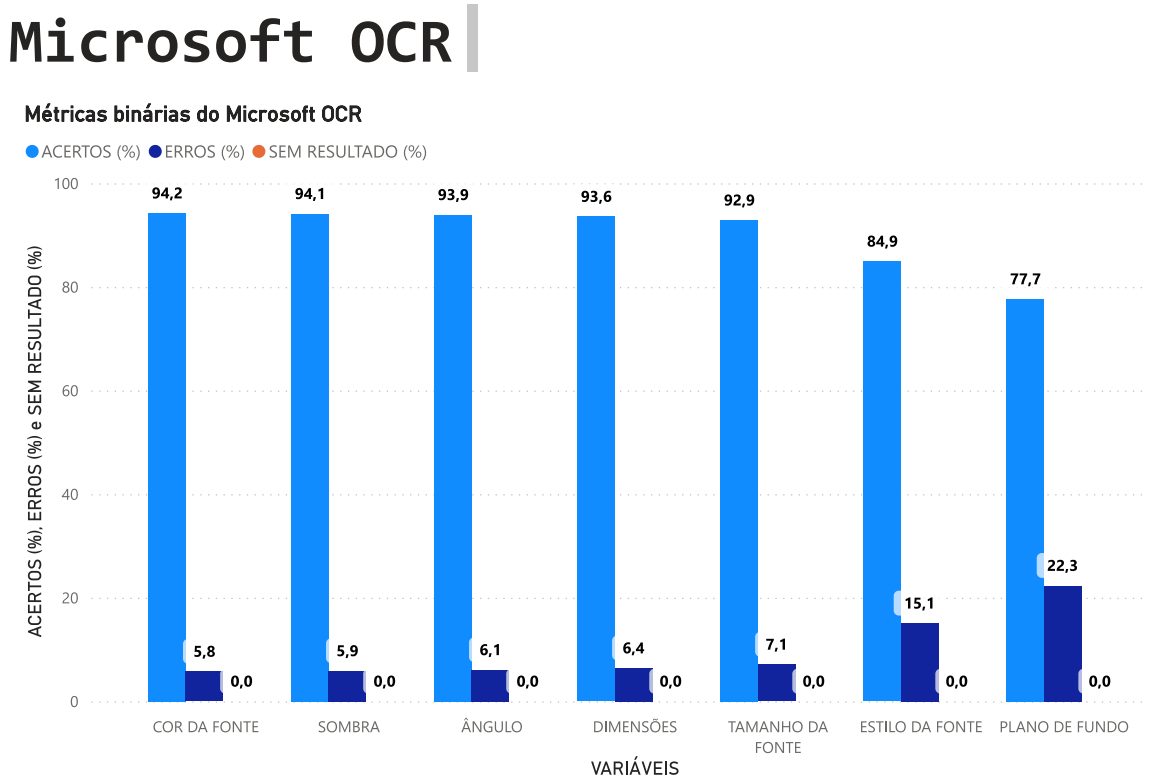


Figura 26 – Resultados das métricas binárias para o Microsoft OCR por variação de aspecto.

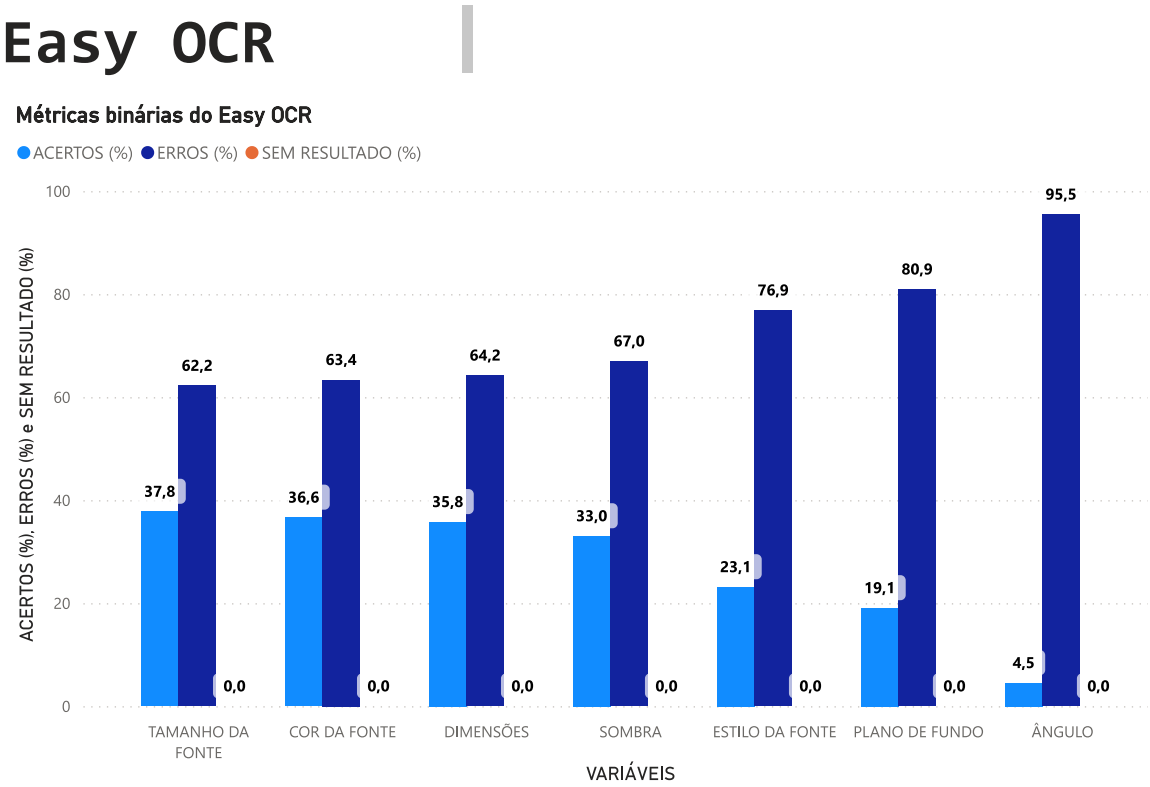


Figura 27 – Resultados das métricas binárias para o Easy OCR por variação de aspecto.

4.2.5 Tempo

Para a análise do tempo, foram realizadas 4 medições principais: 1ª Soma de tempo da CPU (Unidade Central de Processamento, do inglês *Central Process Unit*), que representa o tempo de processamento de todos os conjuntos de imagens no núcleo do processador; 2ª Soma de tempo da CPU com pós-processamento, que além do tempo obtido na 1ª medição, considera também a etapa de padronização dos arquivos de saída dos OCRs; 3ª representa o tempo total de execução dos experimentos, não apenas o tempo de núcleo; 4ª Soma do tempo total com a etapa de pós-processamento. Todas as medições retornaram os valores na escala de segundos.

Pelos resultados apresentados na Tabela 11, é possível notar que o Pytesseract possui o melhor desempenho em três das quatro medições (só não obteve o melhor desempenho na 1ª medição, que retorna a soma de tempo da CPU, o Google Drive OCR obteve o melhor desempenho nessa medição), isso se deve ao fato de que sua execução é realizada localmente. O Easy OCR foi a ferramenta que apresentou o pior desempenho em todas as medições, tanto que pela Tabela 11 é possível notar uma variação na escala (resultados destacados pela cor vermelho), sendo esta a única ferramenta que atingiu à casa dos milhares de segundos. Embora o Easy OCR seja executado localmente, seu desempenho baixo pode estar relacionado ao fato dessa ferramenta estar otimizada para execução em GPUs da NVDIA [2], em sua execução existe um aviso que indica que a execução em uma GPU (Unidade de Processamento Gráfico, do inglês *Graphics Processing Unit*) é consideravelmente mais rápida, nossos testes com as ferramentas de execução local foram todos realizados na CPU, isso permite uma comparação mais justa entre as ferramentas. Para as demais ferramentas, com execução em nuvem, as medições foram referentes ao tempo de chamada e retorno da API pela máquina local, assim, temos que o Google Vision possui o melhor desempenho geral (observando a 3ª e 4ª medição descritas anteriormente), seguido pelo Microsoft OCR. O Google Drive OCR é a ferramenta que possui o melhor desempenho de processamento (considerando a 1ª medição que mede a soma de tempo da CPU), no entanto, devido às limitações do volume de consultas por minuto por parte da desenvolvedora, seu tempo de execução geral acabou ficando elevado.

Tabela 11 – Medições de tempo das ferramentas de OCR em segundos.

Medição	PyTessract	MicrosoftOCR	Google Vision	Google Drive OCR	EasyOCR
Tempo da CPU	1,56 Mil	1,63 Mil	2,05 Mil	1,37 Mil	**2,23 Mi**
Tempo da CPU + Pós-Processamento	1,60 Mil	1,65 Mil	2,11 Mil	3,36 Mil	**2,23 Mi**
Tempo total	16,91 Mil	72,53 Mil	61,99 Mil	316,84 Mil	**1,06 Mi**
Tempo total + Pós-Processamento	147,92 Mil	73,11 Mil	62,37 Mil	394,08 Mil	**1,06 Mi**

[2] <https://www.nvidia.com/pt-br/>

5 Aplicação Prática

Este capítulo concentra todos os experimentos realizados com o algoritmo classificador para detecção de desinformação. O objetivo desses experimentos é verificar se e como a utilização de ferramentas de OCR pode impactar na construção de sistemas de detecção, contenção e moderação de conteúdo para plataformas de mídias sociais. Para isso, foi selecionado um algoritmo classificador para produzir inferências sobre os dados processados pelas ferramentas de OCR e também sobre os dados que representam a verdade fundamental, ou seja, a base original de notícias `FACTCK.BR` (que contém os textos inseridos nas imagens) a fim de compará-los. O resultado das inferências são então avaliados por um conjunto de métricas específicas que revelam os impactos desse procedimento.

5.1 Introdução

O grande volume de dados produzido e disseminado diariamente nas plataforams de mídias sociais, representam um desafio para a detecção de desinformação e a moderação de conteúdo nessas plataformas (GONGANE; MUNOT; ANUSE, 2022). O uso de algoritmos de aprendizagem de máquina na construção de sistemas de detecção de desinformação e moderação de conteúdo em plataformas de mídias sociais oferece muitas vantagens, isso porque esses algoritmos possuem a capacidade de processar e analisar grandes volumes de dados de forma rápida e eficiente (SANTOS, 2023).

A construção de sistemas de moderação de conteúdo, evolve a aplicação de diversas tecnologias para tratamento dos dados. Uma das tecnologias empregadas neste processo é o OCR, que captura as informações textuais de arquivos de imagens e as deixa em formato processável por máquina (MEMON et al., 2020). Com as informações textuais em formato adequado para processamento, são empregados algoritmos de aprendizagem de máquina para realizar a classificação do conteúdo (GONGANE; MUNOT; ANUSE, 2022).

A execução dos experimentos com as ferramentas de OCR, relatada no Capítulo 4, produz conjuntos de dados textuais referentes às informações capturadas pelo OCR de cada imagem processada. Com isso, podemos avaliar como a implantação das tecnologias de OCR em sistemas de moderação e combate à desinformação impactam nessas tarefas.

5.2 Trabalhos Relacionados

Conforme observado em (AHMED; TRAORE; SAAD, 2017), notícias contendo desinformação estão produzindo um impacto social significativo na vida das pessoas e também, em particular, no que se refere ao mundo político. Um exemplo de um grave problema social suscitado pela propagação da desinformação, e discutido em (SILVA, 2014), é o aumento do discurso de ódio nas plataformas de mídias sociais. De acordo com o autor em (SILVA, 2014), o Ministério Público Federal (MPF) vêm registrando um crescente número de denúncias referentes à xenofobia, homofobia, neonazismo, racismo e intolerância religiosa. De acordo com os autores em (AHMED; TRAORE; SAAD, 2017; SANTOS, 2023), a detecção de desinformação é uma área de pesquisa emergente que está recebendo cada vez mais interesse da comunidade acadêmica, mas que envolve alguns desafios que vão desde à quantidade limitada de recursos, como a baixa disponiblidade

de conjuntos de dados bem definidos, até o baixo volume de informações disponível na literatura publicada. De acordo com Pérez-Rosas et al. (2019), a proliferação de notícias enganosas nas plataformas de mídias sociais dificultou a identificação de fontes de notícias confiáveis, o que demanda a necessidade de ferramentas computacionais capazes de fornecer informações sobre a confiabilidade do conteúdo disponível na rede.

Em (PÉREZ-ROSAS et al., 2019), os autores conduziram experimentos utilizando técnicas de aprendizagem de máquina na tentativa de construir detectores precisos de notícias falsas. O algoritmo classificador utilizado pelos autores, foi a Máquina de Vetores de Suporte (do inglês *Support-Vector Machine - SVM*), os testes foram realizados criando classificadores SVMs com diferentes configurações. Os classificadores foram avaliados em seis domínios de notícias: tecnologia, educação, negócios, esportes, política e entretenimento. Os modelos de classificação produzidos se baseiam em uma combinação de informações lexicais, sintáticas e semânticas, bem como em recursos que representam propriedades de legibilidade do texto. Dessa forma, temos que o sistema automático criado em (PÉREZ-ROSAS et al., 2019), apresenta uma acurácia de 0,74 na detecção de notícias falsas provenientes de fontes mais sérias e diversas, enquanto que os seres humanos atingiram uma acurácia entre 0,70 e 0,71. Já para a categoria das celebridades, no domínio de entretenimento, a classificação humana se saiu melhor, com valores de acurácia entre 0,77 e 0,80 frente à 0,73 do sistema automatizado. Os resultados sugerem que os seres humanos são melhores em identificar notícias falsas na categoria das celebridades, no domínio do entretenimento, do que as notícias falsas nos outros domínios. Com isso, os autores concluíram que os melhores modelos produzidos foram capazes de atingir taxas de precisão semelhantes às humanas na detecção de conteúdo falso.

Em (SHAIKH; PATIL, 2020), é apresentada uma solução para o problema de detecção de notícias falsas, através da implementação de um modelo de detecção que faz uso de diferentes técnicas de classificação como: SVMs, Naïve Bayes e Classificadores Passivo Agressivos (do inglês *Passive Agressive Classifier - PA*). Com isso, através da análise dessas técnicas de classificação, os autores puderam observar que o modelo que utiliza técnicas de extração de recursos como Frequência do Termo–Inverso da Frequência nos Documentos (do inglês *Term Frequency-Inverse Document Frequency - TF-IDF*) e o SVM como classificador foi capaz de atingir uma precisão de 95,05% na detecção de desinformação. Além disso, os autores afirmam que a principal fonte de notícias falsas são as plataformas de mídias sociais.

Reis et al. (2019) apresentaram e avaliaram um novo conjunto de recursos para detecção automática de notícias falsas. Os autores realizaram um levantamento sobre os estudos dedicados à detecção de desinformação e selecionaram as principais características dessa tarefa para implementação. O estudo utiliza um conjunto diversificado de características extraídas das notícias e de suas fontes, como características psicolínguísticas, lexicais e semânticas, além de dados sobre a credibilidade e o domínio das notícias. Os autores descrevem ainda que para notícias incorporadas em mídias como imagens e vídeos, foram utilizadas técnicas de processamento de imagens para extrair as informações textuais presentes nelas. Com isso, foi possível testar a eficácia de uma variedade de classificadores de aprendizagem supervisionada ao distinguir histórias falsas de histórias reais em um conjunto de dados rotulado. Foram utilizados algoritmos classificadores como *k-Nearest Neighbors*, *Naive Bayes*, *Random Forests*, *Support Vector Machine with RBF kernel* e *XGBoost*. Os melhores resultados puderam ser alcançados pelos algoritmos *Random Forests* e *XGBoost* com AUC de 0,85 e 0,86 respectivamente.

Os trabalhos apresentados nesta seção, corroboram a ideia de que o desenvolvimento de técnicas computacionais para identificação e contenção de desinformação são de extrema importância para mitigar os impactos negativos advindos dessa prática. Trabalhos como os de Pérez-Rosas et al. (2019) e Shaikh e Patil (2020) demonstram que os sistemas automatizados são capazes de atingir uma precisão de detecção igual e até mesmo superior a dos seres humanos, o que indica o caminho para a realização de novos estudos e consequentemente para a implantação de sistemas mais eficientes.

5.3 Metodologia

Esta seção fornece uma descrição dos métodos e técnicas utilizadas para a condução dos experimentos, bem como a descrição da estrutura das etapas do estudo. A fim de verificar se (e como) as ferramentas de OCR influenciam na detecção de desinformação, foram conduzidos experimentos de classificação, onde um algoritmo classificador treinado com as notícias do `FACTCK.BR` foi designado para produzir inferências sobre os dados retornados pelas ferramentas de OCR. A Figura 28 fornece uma visão geral dos experimentos conduzidos durante esta etapa.

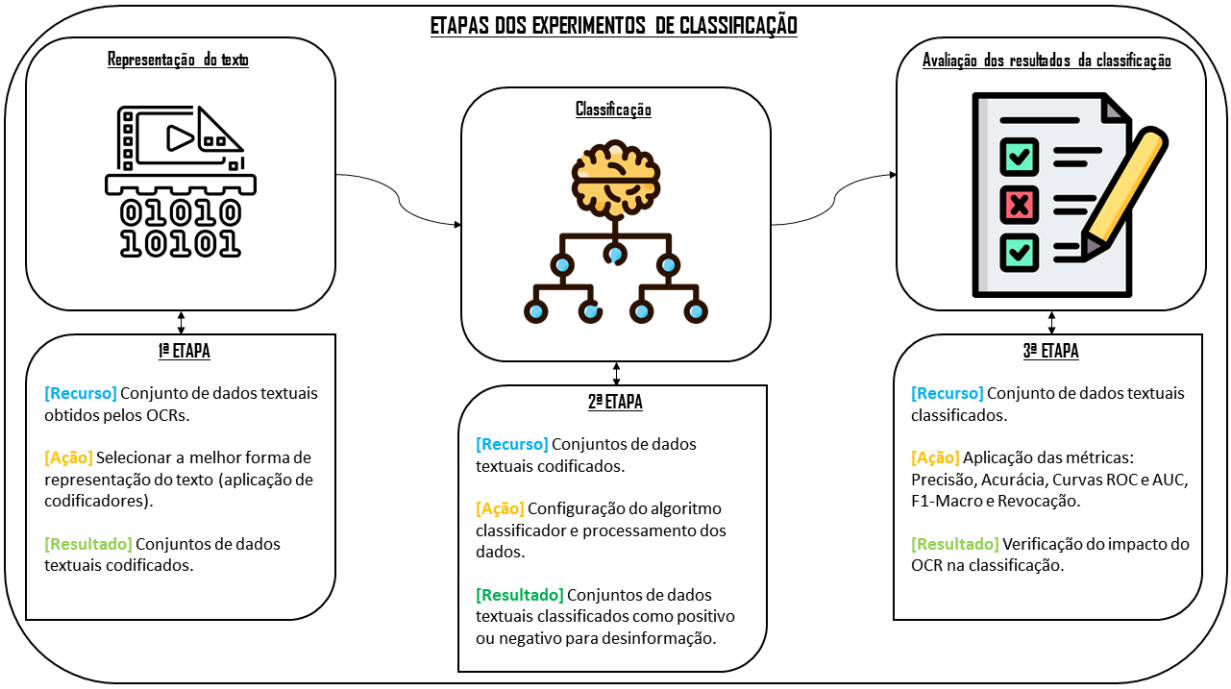


Figura 28 – Visão geral das etapas de classificação.

Para realizar tal avaliação, os dados retornados pelas ferramentas de OCR foram codificados para se adequarem aos padrões de entrada do algoritmo. Essa etapa de codificação também serve para avaliar a melhor representação dos dados textuais dentre um conjunto de sistemas codificadores. Após isso, os dados são inseridos em um algoritmo de aprendizado de máquina para que, assim, este possa realizar a classificação das notícias.

O algoritmo classificador selecionado foi o XGBoost. Para essa escolha, foi realizada uma busca na literatura por algoritmos que oferecessem bons resultados com um conjunto de dados pequeno e que fossem eficientes em relação ao tempo de treino e de experimentação ao lidarem com dados textuais, considerando a tarefa objetivo, ou seja, detecção de desinformação. Assim, os resultados apresentados em (HENDRAWAN; UTAMI; HARTANTO,

2022; HAUMAHU; PERMANA; YADDARABULLAH, 2021; CHEN; GUESTRIN, 2016; REIS et al., 2019), ofereceram indicativos de que o XGBoost é uma boa alternativa para a tarefa de interesse.

5.3.1 Abordagem de Aprendizado de Máquina

De acordo com o exposto em (CHEN; GUESTRIN, 2016), o XGBoost é uma implementação de árvores de decisão do tipo *Gradient Boosting*. Em resumo, nesse algoritmo, várias árvores de decisão são criadas de forma sequencial, e os pesos são então atribuídos a todas as variáveis independentes. Após esse procedimento, a primeira árvore de decisão é alimentada e prevê os resultados (ver Figura 29). Depois, o peso das variáveis previstas de forma errônea pela primeira árvore é aumentado e essas variáveis são então alimentadas para a segunda árvore de decisão. Esses preditores isolados são então agrupados formando assim um modelo mais robusto e preciso.

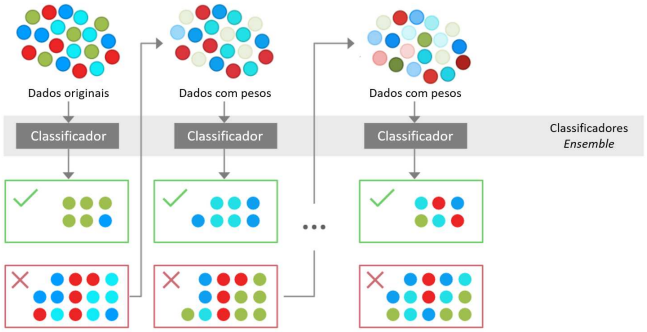


Figura 29 – Ilustração do precedimento de *Boosting*.
Fonte: (GEEKS, 2023).

5.3.2 Estratégias para Representação do Texto

De forma a viabilizar a condução dos experimentos com o algoritmo de classificação, os dados processados pelas ferramentas de OCR precisaram ser adequados para o padrão de entrada esperado por tal. Para realizar essa tarefa, foram testados 3 sistemas codificadores (ou estratégias para representação de texto), muito utilizadas no contexto de processamento de linguagem natural. Os sistemas codificadores são: *One Hot Encoder*, TF-IDF e BERTimbau, que serão descritos a seguir:

1. ***One Hot Encoder***: Na representação de textos, uma codificação *One Hot* é um vetor de 1 X N (onde N é o número de palavras no vetor) usado para diferenciar cada palavra em um vocabulário de todas as demais. O vetor consiste de valor 0 em todas as células, com exceção de uma que deverá conter o valor 1 para identificar exclusivamente a palavra. Esse tipo de codificação garante que o aprendizado de máquina não infira que valores altos possuem maior importância. Veja um exemplo de codificação *One Hot* na Figura 30, onde ao invés de palavras isoladas, são codificados nomes de cidades.

Figura 30 – Exemplo de transformação de dados categóricos por meio do *One Hot Encoder*. Fonte: (VIEIRA et al., 2021).

2. **TF-IDF**: O TF-IDF (Frequência do Termo–Inverso da Frequência nos Documentos, do inglês *Term Frequency-Inverse Document Frequency - TF-IDF*), é um codificador que calcula a importância de uma palavra em um corpus com base na frequência em que ela aparece. O TF-IDF é o produto de duas medições estatísticas: (i) frequência de termo e (ii) frequência inversa de documento.

   A frequência de termo, é a frequência relativa do termo t no documento d (MANNING; RAGHAVAN; SCHUTZE, 2008), dada pela fórmula:

$$tf(t,d) = \frac{f_{t,d}}{\sum_{t' \in d} f_{t',d}}, \tag{5.1}$$

   em que: $f_{t,d}$ é a contagem de um termo em um documento, ou seja, o número de ocorrências do termo t no documento d.

   Já a frequência inversa de documento, é a fração inversa logaritmicamente dimensionada dos documentos que contém a palavra, obtida pelo logaritmo da divisão do total de documentos pela quantidade de documentos que possui o termo (JONES, 1972).

$$idf(t,D) = log\frac{N}{|\{d \in D : t \in d\}|}, \tag{5.2}$$

   em que:

   - N: número total de documentos no corpus N = |D|;
   - $|\{d \in D : t \in d\}|$: número de documentos em que o termo t aparece. (Se o termo t não aparecer, a divisão por zero é um caso a ser tratado, normalmente isso é feito somando 1 ao numerador e também ao denominador).

   Com isso, o TF-IDF pode ser calculado da seguinte forma (ROBERTSON, 2004):

$$tfidf(t,d,D) = tf(t,d) * idf(t,D). \tag{5.3}$$

   Um alto valor de peso em TF-IDF descreve que o termo possui uma frequência elevada no documento dado e uma baixa frequência em relação a toda a coleção de documentos. Assim, os pesos tendem a filtrar termos comuns (ROBERTSON, 2004).

3. **BERTimbau**: O BERTimbau (SOUZA, 2020) é a versão para Português do BERT (*Bidirecional Encoder Representations from Transformers*), que por sua vez é um modelo de representação de linguagem projetado para pré-treinar representações

bidirecionais profundas a partir de texto não rotulado. O BERT condiciona conjuntamente o texto presente nas orientações esquerda e direita em todas as camadas (DEVLIN et al., 2018). A implementação inicial do BERT foi utilizando a Língua Inglesa em dois tamanhos de modelo: BERT_{base}: 12 *layers* com 12 *bidirectional self-attention heads* totalizando 110 milhões de parâmetros, e BERT_{large}: 24 *layers* com 16 *bidirectional self-attention heads* totalizando 340 milhões de parâmetros. Para a criação do BERTimbau, também foram utilizados dois tamanhos: BERTimbau_{base}: com 12 *layers* 768 dimensões ocultas e 12 *bidirectional self-attention heads* totalizando 110 milhões de parâmetros e BERTimbau_{large}: com 24 *layers*, 1024 dimensões ocultas e 16 *bidirectional self-attention heads* totalizando 330 milhões de parâmetros.

Conforme descrito anteriormente, o BERT treina os modelos de linguagem com base no conjunto completo de palavras de forma bidirecional, enquanto que outros modelos de linguagem fazem uso apenas de uma direção (da esquerda para a direita ou vice-versa), isso faz com que o BERT analise o contexto de uma palavra com base nas palavras presentes no seu entorno (DEVLIN et al., 2018). O uso do Transformer, um mecanismo de atenção composto por um codificador que lê a entrada de texto e um decodificador que produz uma previsão para a tarefa, permite que o BERT aprenda as relações contextuais entre as palavras (DEVLIN et al., 2018). A entrada do codificador Transformer é uma sequência de *tokens*, que são primeiro introduzidos em vetores e depois processados na rede neural, a saída é uma sequência de vetores de tamanho padronizado, onde cada vetor corresponde a um token de entrada com o mesmo índice (DEVLIN et al., 2018; SOUZA, 2020).

5.3.3 Métricas de Avaliação para Classificação

Para avaliação das tarefas de classificação, foram utilizadas métricas amplamente utilizadas em tarefas de recuperação de informação (BAEZA-YATES; RIBEIRO-NETO et al., 1999), que aplicam os seguintes indicadores i) **Verdadeiro positivo (VP)**, este indicador demonstra a quantidade de dados que foram classificados corretamente como positivos; ii) **Falso positivo (FP)**, que demonstra o total de dados que foram classificados como positivo incorretamente; iii) **Verdadeiro negativo (VN)**, que demonstra o total de dados que foram classificados corretamente como negativos; e **Falso negativo (FN)**, que demonstra o total de dados classificados incorretamente como negativos. Assim, foram utilizados as métricas de precisão, dada pela fórmula: $PRECISAO = \frac{VP}{VP+FP}$ (SANTOS, 2021); acurácia, dada por: $ACURACIA = \frac{VP+VN}{VP+FP+VN+FN}$ (INTERNATIONAL ORGANIZATION FOR STANDARDIZATION, 1994); curva ROC (Curva Característica de Operação do Receptor, do inglês *Receiver Operating Characteristic Curve*): essa curva representa os parâmetros i) Taxa de verdadeiro positivo $TVP = \frac{VP}{VP+FN}$; e ii) Taxa de falso negativo $TFP = \frac{FP}{FP+VN}$ (MARIANO et al., 2021) (veja um exemplo de curva ROC na Figura 31, onde quanto mais próximo do topo do eixo Y, melhor o classificador); curva AUC (Área Sob a Curva, do inglês *Area Under the Curve*): AUC mede toda a área bidimensional abaixo da curva ROC, essa métrica indica a probabilidade de duas previsões serem corretamente classificadas (MARIANO et al., 2021) (veja um exemplo na Figura 32); revocação (do inglês *recall*): $REVOCACAO = \frac{VP}{VP+FN}$ (ALLEN et al., 1955); pontuação F1 (do inglês *F1-Score*): $PONTUACAO\_F1 = \frac{2*PRECISAO*RECALL}{PRECISAO+RECALL}$ (SORENSEN, 1948); pontuação F1 Macro: $F1\_MACRO = \frac{\sum(PONTUACOES\_F1)}{N}$, em que N representa o número de classes (PILLAI; FUMERA; ROLI, 2017); e a matriz de confusão: essa matriz fornece uma visualização do quão bem o modelo de aprendizado de máquina classificou os dados,

isso é feito comparando suas previsões com os valores reais (NAVIN; PANKAJA, 2016) (a matriz exemplificada pela Tabela 12, ilustra o tipo de classificação binária, assim, a matriz possui duas dimensões: a de rótulos reais e a de rótulos previstos).

Tabela 12 – Exemplo de uma matriz de confusão.

		Rótulos previstos	
Rótulos Reais			
		VP	FN
		FP	VN

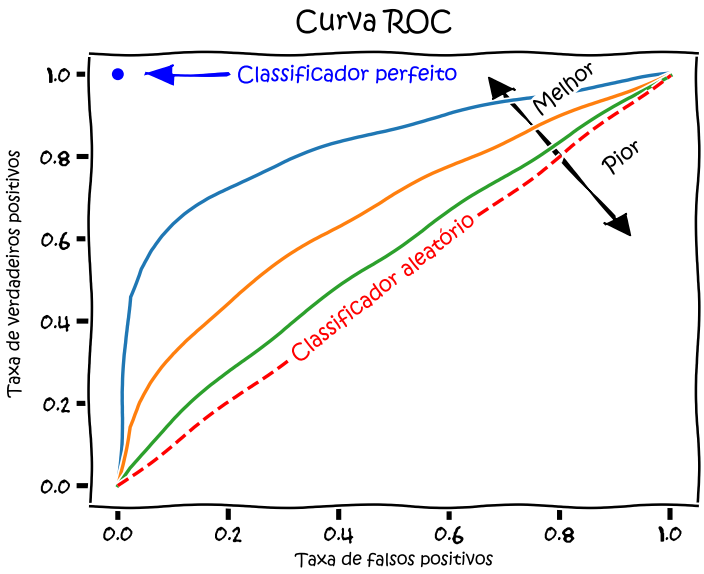


Figura 31 – Ilustração de uma curva ROC.
Fonte: (MARIANO et al., 2021).

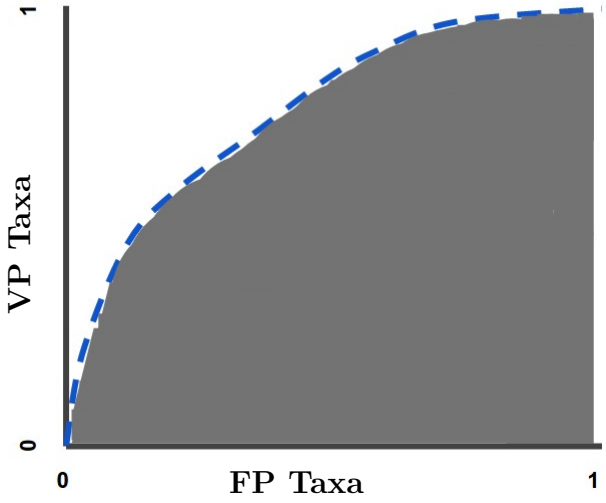


Figura 32 – Ilustração de um gráfico AUC.
Fonte: Adaptado de (DEVELOPERS, 2022).

5.4 Experimentos

Os experimentos de classificação, foram executados em uma máquina com a seguinte configuração:

- RAM: 8,0 GiB. Memória de 8 GB DDR4 (8 GB x 1) 2 SODIMM;
- Armazenamento: SSD NVMe de 256 GB;
- Processador: 11th Gen Intel(R) Core(TM) i5-1135G7 @2.40GHz 2.42 GHz;
- GPU: Gráficos Intel® Iris® Xe;
- Sistema Operacional: WSL Ubuntu 22.04.3 LTS de 64 bits executado no Windows 11 Home Single Language Versão 23H2. Compilação do SO 22631.2861.

A fim de encontrar uma configuração que produzisse bons resultados para o classificador, foi realizada uma busca exaustiva sobre valores de hiperparâmetros para o classificador. Esse procedimento foi realizado por meio da execução do `GridSearchCV`, um módulo do `scikit-learn`[1] onde dada uma grade de parâmetros, definida pelo usuário, ele percorre todos os parâmetros a fim de produzir a melhor combinação com base em uma métrica de pontuação.

Para este experimento, a grade de parâmetros foi definida da seguinte forma:

- ***objective:*** *"binary:logistic"*. Classificação binária, o alvo contém apenas duas classes.
- ***nthread:*** 4. Este parâmetro represetna o número de processos paralelos.
- ***seed:*** 413. Parâmetro de geração aleatória.
- ***max_depth:*** 4 valores escolhidos aleatoriamente no intervalo de [3, 10). Profundidade máxima de uma árvore. Aumentar esse valor tornará o modelo mais complexo e mais propenso a *overfit*. 0 indica que não há limite de profundidade.
- ***n_estimators:*** 4 valores múltiplos de 15, escolhidos aleatoriamente no intervalo de [100, 1000). Número de árvores geradas.
- ***learning_rate:*** 4 valores escolhidos aleatoriamente no intervalo de [0.01, 0.1] aumentado a um passo de 0.01. Encolhimento do tamanho do passo usado na atualização para evitar o sobreajuste. Após cada etapa de *boosting*, podemos obter diretamente os pesos dos novos recursos e reduzir os pesos dos recursos para tornar o processo de *boosting* mais conservador.
- ***colsample_bytree:*** 3 valores escolhidos aleatoriamente no intervalo de [0.3, 1] aumentado a um passo de 0.01. É a razão de subamostra de colunas ao construir cada árvore. A subamostragem ocorre uma vez para cada árvore construída.
- ***subsample:*** 3 valores escolhidos aleatoriamente no intervalo de (0.1, 1] aumentado a um passo de 0.01. Proporção de subamostras das instâncias de treinamento. Defini-lo como 0,5 significa que o XGBoost amostraria aleatoriamente metade dos dados de treinamento antes do crescimento das árvores. e isso evitará o *overfitting*. A subamostragem ocorrerá uma vez em cada iteração de *boosting*.

[1] <https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html>

- ***gamma:*** [0, 1, 5]. Redução mínima de perda necessária para fazer uma partição adicional em um nó de folha da árvore. Quanto maior, mais conservador será o algoritmo.

Com essa grade de parâmetros definidas para cada codificador, a validação cruzada definida em 5 partições e a `ROC_AUC` definida como métrica de avaliação, o `GridSearchCV` foi capaz de produzir as seguintes configurações de parâmetros apresentada na Tabela 13:

Tabela 13 – Seleção dos hiperparâmetros.

PARÂMETROS	BERTIMBAU	TF-IDF	ONE HOT ENCODER
objective	binary:logistic	binary:logistic	binary:logistic
nthread	4	4	4
seed	10	10	10
max_depth	6	7	9
n_estimators	100	100	640
learning_rate	0.02	0.05	0.02
colsample_bytree	0.3	0.3	0.6
subsample	0.8	0.8	0.6
gamma	0	1	0

Para a condução dos experimentos de classificação, foi necessário tratar os dados do `FACTCK.BR`. Esse repositório de notícias não se encontra balanceado, sendo que das 1.314 notícias presentes, 943 são rotuladas como *fake* (ou "desinformação") e 371 como pertencentes à outras classes. Destas 371 notícias pertencentes à outras classes, temos que 333 são confirmadas como verídicas e 38 possuem informações insuficientes para serem rotuladas como verdadeiras ou falsas. Dessa forma, a fim de balancear o conjunto de dados, foram selecionadas de forma aleatória 333 notícias, rotuladas como *fake* (ou "desinformação"), e criado um conjunto de dados composto por 666 dados (333 positivo para "desinformação"e 333 negativo). Com isso, temos um conjunto de dados balanceado para avaliar o desempenho do modelo de classificação.

Uma vez tendo a base balanceada, foi realizada uma avaliação dos sistemas codificadores, de forma a selecionar o mais adequado para o nosso conjunto de dados. Assim, o experimento foi realizado com validação cruzada (*cross-validation*) de cinco partições. Para cada execução do experimento, foram computadas as seguintes métricas: F1-Macro, ROC_AUC, Acurácia, Precisão e *Recall* (Revocação). Os resultados são apresentados na Tabela 14.

Pode-se perceber que o TF-IDF possui os melhores indicativos gerais para representação dos dados do `FACTCK.BR`, superando o *One Hot Encoder* em quatro dos indicadores, F1_Macro, ROC_AUC, Acurácia e Precisão e apresentando os menores desvios para essas métricas. No caso do *Recall*, que é uma métrica que tenta responder a seguinte pergunta: De todos as notícias que realmente são positivas para desinformação, qual percentual é identificado corretamente pelo modelo? O TF-IDF apresentou valores próximos ao One Hot Encoder (que obteve a melhor pontuação nessa métrica). Com isso, o codificador selecionado para os testes seguintes foi o TF-IDF.

Tabela 14 – Resultados de classificação.

CROSS VALIDATION - Bertimbau		
METRICAS	**MEDIA**	**DESVIO PADRAO**
F1_MACRO	54.26	10.21
ROC_AUC	56.92	13.14
ACURÁCIA	55.13	10.14
PRECISÃO	56.61	11.47
RECALL	59.13	10.69
CROSS VALIDATION - TFiDF		
METRICAS	**MEDIA**	**DESVIO PADRAO**
F1_MACRO	61.61	7.65
ROC_AUC	69.55	10.84
ACURÁCIA	62.33	7.53
PRECISÃO	64.55	11.09
RECALL	62.13	11.61
CROSS VALIDATION - One Hot Encoder		
METRICAS	**MEDIA**	**DESVIO PADRAO**
F1_MACRO	56.88	13.56
ROC_AUC	60.11	12.95
ACURÁCIA	58.56	11.26
PRECISÃO	60.63	12.39
RECALL	65.12	7.44

5.4.1 Resultados

Foram realizados experimentos com a base original de notícias `FACTCK.BR` e com os dados textuais recuperados pelos OCRs. Os resultados são apresentados e discutidos a seguir.

Experimentos com a base `FACTCK.BR`: O primeiro experimento realizado foi com o repositório original de notícias, ou seja, o `FACTCK.BR`. Para isso, considerando o balanceamento desse conjunto de dados, descrito anteriormente, o repositório balanceado foi divido em 80% treino e 20% teste de forma estratificada, ou seja, tanto o conjunto de treino quanto o de teste permaneceram balanceados. Para oferecer um melhor comparativo com os experimentos envolvendo os dados dos OCRs, que serão descritos posteriormente, esse experimento foi executado 3 vezes, considerando os dados de notícias equivalentes às usadas pelas ferramentas de OCR em cada aspecto analisado. Assim, os melhores hiperparâmetros obtidos com o codificador TF-IDF foram carregados para o modelo de classificação XGBoost, que após o treinamento, foi capaz de produzir os seguintes resultados apresentados nas Tabelas 15 e 16.

Como cada aspecto de imagem pode fazer uso de diferentes quantidades textuais e assim produzir diferentes quantidades de imagens, as bases de testes podem apresentar tamanhos diferentes, assim temos:

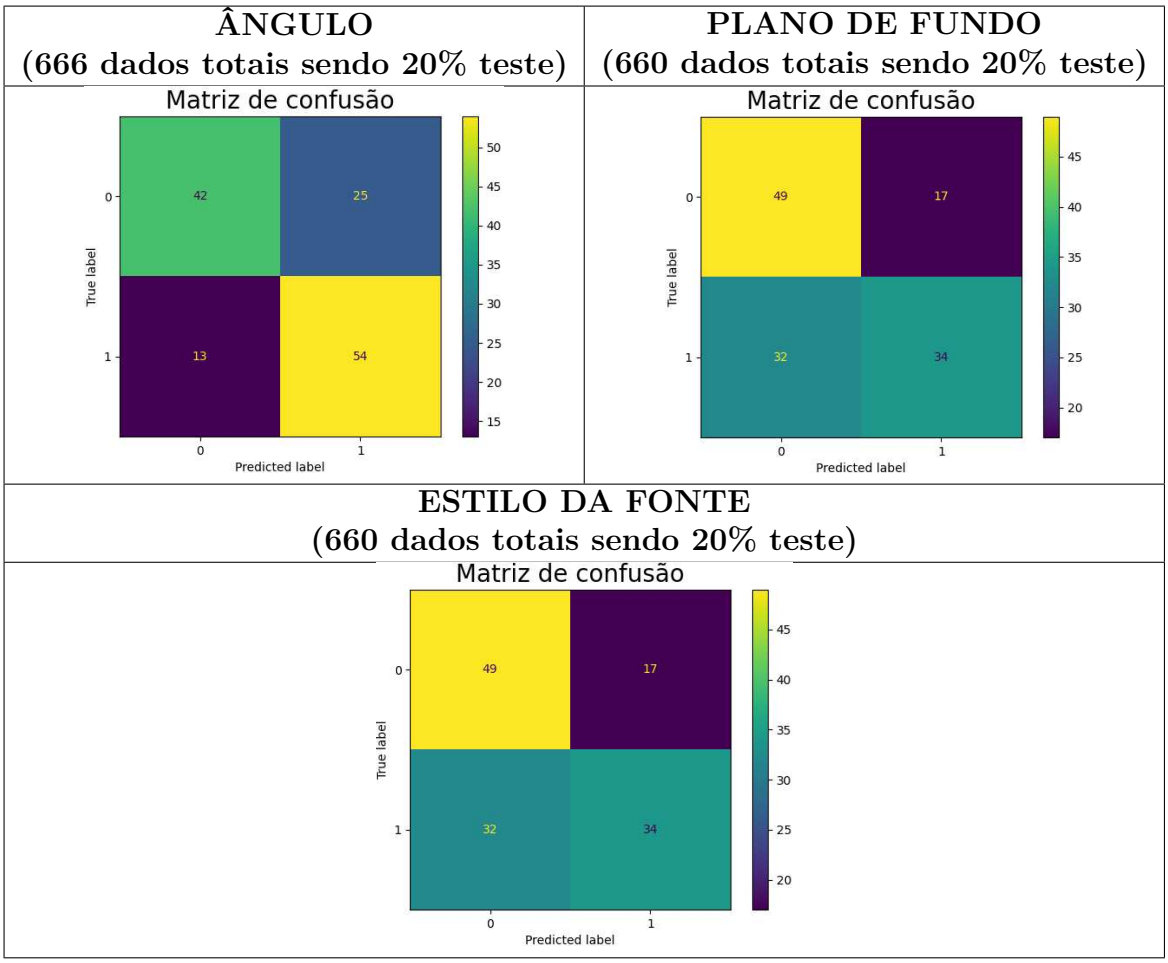
- **Ângulo:** 666 dados totais, sendo 333 postivo para desinformação e 333 negativo;
- **Plano de fundo:** 660 dados totais, sendo 330 postivo para desinformação e 330 negativo;

- **Estilo da fonte:** 660 dados totais, sendo 330 postivo para desinformação e 330 negativo.

Tabela 15 – Resultados da classificação com o repositório `FACTCK.BR` com dados balanceados e equivalentes a cada aspecto analisado

FACTCK.BR (Balanceado)	F1 MACRO	AUC	ACURACIA	PRECISAO	RECALL
Ângulo	71.41	75.08	71.64	68.35	80.60
Plano de fundo	62.39	67.13	62.88	66.67	51.52
Estilo da fonte	62.39	67.13	62.88	66.67	51.52

Tabela 16 – Matrizes de confusão para cada aspecto avaliado com a base de dados balanceada.



O resultado de 75,08% de AUC apresentado no Gráfico da Figura 33 sugere que o modelo tem um bom desempenho geral na distinção entre as classes positivas e negativas. De forma geral, um AUC de 50% indica que o modelo apresenta um desempenho aleatório, enquanto que um AUC de 100% indica que o modelo atingiu um desempenho perfeito na classificação. Já o valor de 67,13% de AUC indicado nos gráficos das Figuras 34 e 35 indicam que o modelo têm um desempenho aceitável, mas não particularmente forte. O que indica que o modelo está fazendo uma classificação razoável mas não altamente eficaz sobre os dados.

Os valores apresentados nas Tabelas 15 e 16 indicam que para o aspecto Ângulo, o modelo possui um desempenho razoável, cometendo menos erros ao classificar positivos,

mas perdendo alguns verdadeiros positivos e gerando alguns falsos negativos. Já para os aspectos Plano de Fundo e Estilo da Fonte, os resultados indicam que o modelo é bom em identificar a maioria dos casos positivos reais, no entanto, ele apresenta dificuldade em identificar corretamente os casos negativos, resultando em muitos falsos positivos.

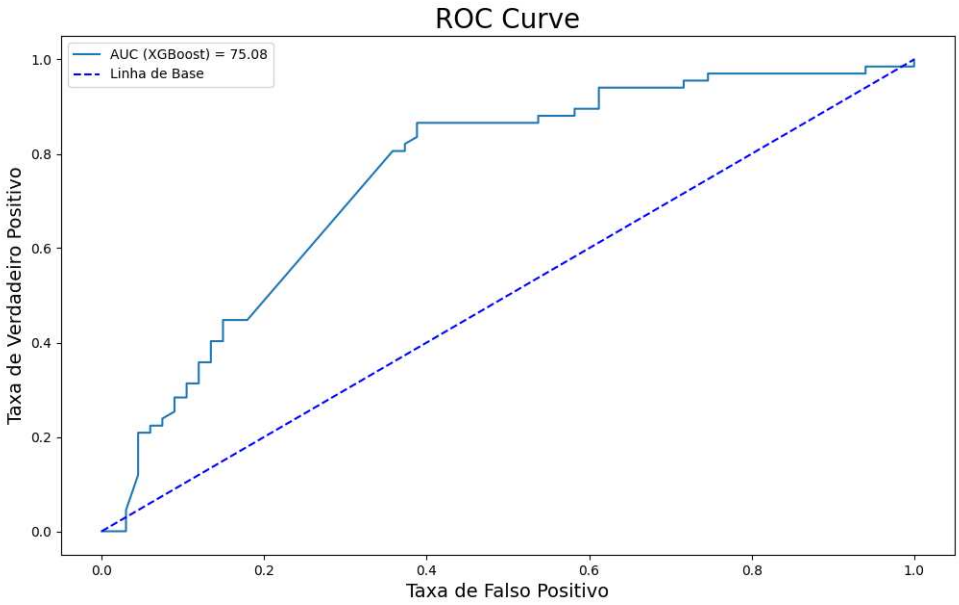


Figura 33 – Gráfico de AUC e Curva ROC para os experimentos com o aspecto Ângulo.

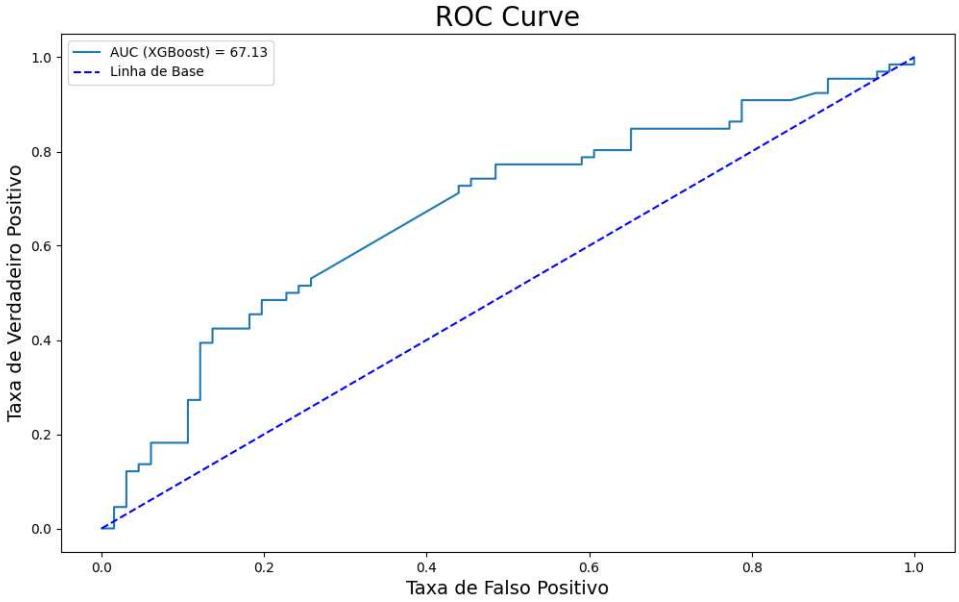


Figura 34 – Gráfico de AUC e Curva ROC para os experimentos com o aspecto Plano de Fundo.

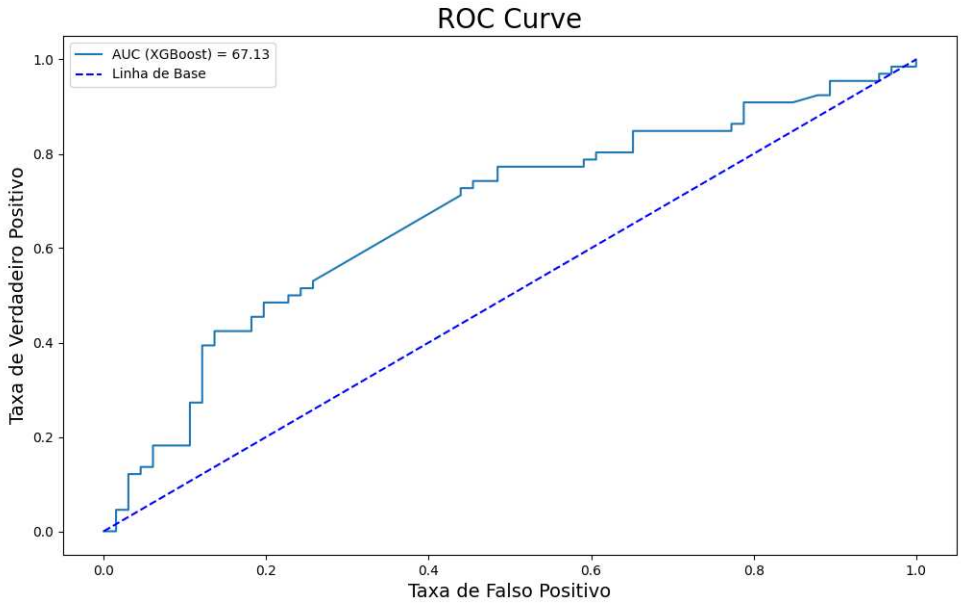


Figura 35 – Gráfico de AUC e Curva ROC para os experimentos com o aspecto Estilo da Fonte.

Experimentos com os resultados dos OCRs: Para a condução dos experimentos com os resultados dos OCRs, foi utilizado o mesmo modelo de classificação da etapa anterior, ou seja, o XGBoost foi treinado com a base `FACTCK.BR` de forma balanceada (80% treino e 20% teste com divisão estratificada da base de dados). Para os testes, foram selecionadas as imagens que utilizaram os mesmos textos da base de testes, ou seja, para cada aspecto de imagem processado pelos OCRs, selecionamos as imagens que continham as mesmas informações textuais da base de testes do modelo de classificação. Com isso, podemos fazer um estudo comparativo entre a base textual original e a base textual retornada pelos OCRs para determinar se a qualidade dos OCRs impactam no processo de classificação.

Como cada configuração de cada aspecto de imagem pode produzir diferentes quantidades de imagens, as bases de testes podem aparecer de forma desbalanceada, dessa forma temos a seguinte configuração para cada base de dados retornada pelos OCRs:

- **Ângulo:** 258 frases equivalentes à base de teste do modelo de classificação. 1682 imagens produzidas utilizando essas frases sendo que, 847 são positivas para desinformação e 835 são negativas;
- **Plano de fundo:** 132 frases equivalentes à base de teste do modelo de classificação. 924 imagens produzidas utilizando essas frases sendo que, 462 são positivas para desinformação e 462 são negativas.
- **Estilo da fonte:** 132 frases equivalentes à base de teste do modelo de classificação. 1163 imagens produzidas utilizando essas frases sendo que, 578 são positivas para desinformação e 585 são negativas.

Os resultados apresentados nas tabelas seguintes estão organizados da seguinte forma: A primeira linha de cada tabela exibe os resultados dos experimentos conduzidos com a base de dados textual `FACTCK.BR`. Já as linhas seguintes, demonstram os resultados

da classificação com as informações textuais extraídas de cada imagem por cada mecanismo de OCR. De forma a permitir um estudo comparativo justo, a base `FACTCK.BR` possui o mesmo tamanho das bases retornadas pelos OCRs. Para isso, para cada texto retornado pelo OCR, a base `FACTCK.BR` contém o texto original.

Testes com o aspecto ângulo: Com os resultados de acurácia, AUC, F1 macro, precisão e *recall* apresentados pela Tabela 17 (as tabelas completas podem ser consultadas no Apêndice A), é possível perceber que a qualidade de retorno das ferramentas de OCR influenciam diretamente na qualidade da detecção de desinformação (os melhores resultados estão destacados pela cor azul e os piores pela cor vermelha). O OCR da Microsoft alcançou os melhores resultados com as menores taxas de desvio em todas as métricas avaliadas. Para o indicador de precisão, que busca informar, de todos os dados classificados como positivo para desinformação, o percentual que realmente é positivo, o Microsoft OCR atingiu um percentual médio de 68,38%, chegando muito próximo do teste realizado com a base original 68,41% e apresentado na Tabela 15, o Google Drive e o Google Vision também conseguiram fazer o classificador produzir ótimos resultados (se comparado à classificação com a base de dados original `FACTCK.BR`). Ao analisar os valores de AUC, o PyTesseract produziu o pior resultado, ficando muito próximo do limiar entre classes com AUC médio de 58,67%, enquanto que os outros OCRs conseguiram valores superiores à 73%. Para a F1-Macro, que é uma média não ponderada da relação entre a precisão e o *recall*, os valores médios ficaram acima de 70% para o Microsoft OCR, Google Drive e Google Vision, o Easy OCR atingiu uma pontuação média de F1 macro de 69,72% enquanto que o PyTesseract, que foi o que obteve o pior resultado com média de 47,82% na pontuação F1 macro.

Tabela 17 – Resultados da classificação com o repositório produzido pelos OCRs com o aspecto ângulo.

BASE	ACURÁCIA	AUC	F1 MACRO	PRECISÃO	RECALL
FACTCK.BR	71.36 ± 0.12	74.65 ± 0.19	71.12 ± 0.13	68.41 ± 0.03	80.09 ± 0.23
PyTesseract	57.41 ± 8.6	58.67 ± 10.53	47.82 ± 15.96	55.87 ± 7.33	92.28 ± 9.78
Microsoft OCR	71.31 ± 0.21	74.68 ± 0.22	71.06 ± 0.21	68.38 ± 0.08	79.98 ± 0.4
Google Drive OCR	71.25 ± 0.25	74.65 ± 0.26	71.01 ± 0.25	68.35 ± 0.1	79.87 ± 0.49
Google Vision	71.14 ± 0.29	74.57 ± 0.28	70.91 ± 0.28	68.29 ± 0.1	79.65 ± 0.55
Easy OCR	69.87 ± 1.63	73.65 ± 0.91	69.72 ± 1.65	67.92 ± 1.57	76.13 ± 2.54

Testes com o aspecto estilo da fonte: Com os resultados de acurácia, AUC, F1 macro, precisão e *recall* apresentados pela Tabela 18 (as tabelas completas podem ser consultadas no Apêndice B), notamos que o desempenho dos OCRs com a variação do estilo da fonte não impactou a classificação, sendo que os resultados ficaram muito próximos dos testes realizados com a base `FACTCK.BR` original (os melhores resultados estão destacados pela cor azul e os piores pela cor vermelha). Se voltarmos ao gráfico apresentado na Figura 12 Seção 5.2.1, podemos notar que dentre os aspectos de imagem problemáticos selecionados para serem estudados nesta etapa de classificação, o aspecto estilo da fonte foi o que deu menos trabalho para os sistemas de OCR, a maior taxa média de desvio de CER registrada foi de 3% do Easy OCR, valor muito inferior se comparados aos registrados pelo ângulo e plano de fundo. Isso fornece mais um indicativo de que a qualidade dos sistemas de OCR influenciam diretamente no processo de classificação.

Tabela 18 – Resultados da classificação com o repositório produzido pelos OCRs com o aspecto estilo da fonte.

BASE	ACURÁCIA	AUC	F1 MACRO	PRECISÃO	RECALL
FACTCK.BR	63.29 ± 0.94	67.27 ± 0.67	62.78 ± 0.96	66.81 ± 1.00	51.92 ± 1.41
PyTesseract	62.86 ± 1.99	66.51 ± 1.83	62.30 ± 2.01	66.44 ± 2.56	51.05 ± 2.13
Microsoft OCR	63.20 ± 0.89	67.17 ± 0.72	62.68 ± 0.92	66.74 ± 1.14	51.73 ± 1.53
Google Drive OCR	62.69 ± 1.33	66.78 ± 1.09	62.12 ± 1.41	66.20 ± 1.34	50.87 ± 2.29
Google Vision	63.11 ± 1.17	67.21 ± 1.04	62.59 ± 1.28	66.56 ± 1.34	51.72 ± 2.31
Easy OCR	62.60 ± 2.21	66.73 ± 1.54	62.03 ± 2.27	66.13 ± 2.76	50.71 ± 2.73

Testes com o aspecto plano de fundo: Os valores de acurácia, AUC, F1 macro, precisão e *recall* apresentados pela Tabela 19 (as tabelas completas podem ser consultadas no Apêndice C), indicam que os dados textuais recuperados pelas ferramentas de OCR, com variação do aspecto plano de fundo, produziram influência na detecção de desinformação pelo classificador (os melhores resultados estão destacados pela cor azul e os piores pela cor vermelha). Alguns OCRs como o Easy OCR e o PyTesseract apresentaram dificuldades em situações onde o contraste entre a cor da fonte e o plano de fundo era baixo, representado pelos tons mais escuros de plano de fonte com a cor padrão do texto que é preto. Planos de fundo compostos por outras imagens, como fotografias ou arte digital (ou seja, cores não sólidas), também representam um desafio para recuperação de informação textual das imagens e isso impacta no processo de classificação. Assim, de forma geral, os melhores resultados com a variação do aspecto plano de fundo, foram obtidos pelos dados recuperados pelo Microsoft OCR e Google Vision. Dos OCRs gratuitos, o melhor desempenho foi obtido pelo Google Drive OCR, ficando muito próximo dos resultados das alternativas pagas, assim como o Easy OCR. O PyTesseract novamente foi a ferramenta com o pior resultado dentre as analisadas.

Explorando um pouco mais o resultado obtido pelo Microsoft OCR, pode-se perceber que seu valor médio de acurácia foi levemente superior ao da base `FACTCK.BR`, que contém os dados originais. No entanto, embora tenha produzido excelentes valores, se subtrairmos o seu valor de desvio, a média ficará em 62,85%, apenas 0,03% abaixo do `FACTCK.BR` o que a destaca como a melhor ferramenta para lidar com esse aspecto de imagem.

Tabela 19 – Resultados da classificação com o repositório produzido pelos OCRs com o aspecto plano de fundo.

BASE	ACURÁCIA	AUC	F1 MACRO	PRECISÃO	RECALL
FACTCK.BR	62.88 ± 0.00	67.13 ± 0.00	62.39 ± 0.00	66.67 ± 0.00	51.52 ± 0.00
PyTesseract	57.58 ± 6.20	59.86 ± 7.84	50.44 ± 13.50	51.66 ± 36.09	27.27 ± 23.84
Microsoft OCR	63.53 ± 0.68	67.17 ± 0.88	62.97 ± 0.76	67.90 ± 0.59	51.30 ± 1.36
Google Drive OCR	62.66 ± 1.29	66.74 ± 1.51	62.13 ± 1.35	66.56 ± 1.58	50.87 ± 1.72
Google Vision	62.99 ± 0.81	67.24 ± 0.34	62.47 ± 0.79	66.96 ± 1.20	51.30 ± 0.57
Easy OCR	62.01 ± 4.25	66.39 ± 3.76	60.60 ± 5.99	67.28 ± 3.30	45.89 ± 11.99

5.5 Conclusão

Com os resultados de classificação apresentados nas seções anteriores e pelo exposto na Tabela 20, é possível perceber que a qualidade de retorno das ferramentas de OCR influenciam diretamente na qualidade da detecção de desinformação. As ferramentas que apresentaram os melhores resultados na tarefa de recuperação de conteúdo textual, conforme discutido no Capítulo 4, foram as que obtiveram os melhores desempenhos de classificação. Dessa forma, podemos concluir que a seleção da ferramenta de OCR, para

aplicação na construção de sistemas de detecção, contenção e moderação de conteúdo para plataformas de mídias sociais, terá impacto direto na qualidade do sistema.

Tabela 20 – Relação entre os resultados de avaliação dos OCRs com os de classificação.

ASPECTO ANALISADO	BOA RECUPERAÇÃO DE TEXTO PELOS OCRs (CER <= 10%)	IMPACTO POSITIVO NA CLASSIFICAÇÃO
Ângulo	X	X
Estilo da Fonte	✓	✓
Plano de Fundo	X	X

6 Conclusão

Neste trabalho, apresentamos uma investigação de ferramentas de OCR, com suporte ao idioma Português Brasileiro, para avaliar seu desempenho no contexto de ferramenta de suporte ao processo de detecção de desinformação considerando imagens divulgadas em plataformas de mídias sociais. Tendo em conta os aspectos de imagens problemáticos, apontados na etapa de avaliação dos sitemas de OCR e analisados na etapa de classificação, podemos concluir que a qualidade da implementação de sistemas para detecção de desinformação em imagens está a mercê da seleção do sistema de OCR. Os sistemas que apresentaram os melhores resultados, na etapa de classificação, foram os que conseguiram extrair os textos das imagens com as menores taxas de erro.

Em suma, o sistema de OCR que obteve o melhor desempenho, neste estudo, foi o Microsoft OCR, sendo este o sistema que mais se destacou na maior parte dos aspectos de imagem analisados e também na etapa de classificação. Dentre as alternativas gratuitas, a ferramenta que apresentou o melhor desempenho foi o Google Drive OCR, sendo este capaz de superar até mesmo o Microsoft OCR na análise de desvio médio de CER para o aspecto estilo da fonte e apresentando resultados satisfatórios na etapa de classificação.

Os resultados mostram que os atributos da imagem que têm o impacto mais negativo no desempenho de alguns dos métodos avaliados são: o plano de fundo, o ângulo de rotação do texto e o estilo da fonte. Acreditamos que ao indicar pontos de melhoria nos sistemas, contribuímos para a construção de mecanismos cada vez mais eficazes em diferentes contextos. Além disso, destacamos também sistemas com desempenho satisfatório no contexto analisado, incluindo as melhores alternativas comerciais e de código aberto.

As contribuições científicas deste trabalho se dão através dos seguintes itens:

- Construção e disponibilização de um conjunto de imagens sintéticas, com informações textuais escritas em Português do Brasil, geradas considerando padrões comumente identificados em plataformas de mídias sociais. Esta contribuição pode ser extremamente útil para a condução de novas pesquisas não apenas nos campos de Aprendizagem de Máquina e Processamento Digital de Imagens mas também em outros campos, alguns exemplos são:
  - Segurança e Detecção de Fraudes, conjuntos de dados que reproduzem padrões de plataformas de mídias sociais são úteis para treinar modelos que detectam anomalias e aspectos fraudulentos nesses ambientes;
  - Pesquisa Linguística e Sociolinguística, estudos de padrões de comunicação e comportamentos em plataformas de mídias sociais podem ser realizados com esses dados;
  - Simulações e Testagem, conjuntos de dados sintéticos permitem a condução de novos experimentos e testes para novos aplicativos sem a exposição de dados reais, o que garante privacidade e segurança.
- Estabelecimento de uma metodologia para geração de dados sintéticos, que permite a geração de imagens com outros dados de entrada. Por meio da metodologia desenvolvida, além de ser possível conduzir experimentos com dados textuais multilinguais, também é possível adaptá-la a diferentes contextos de aplicação;

- Levantamento e investigação do desempenho das ferramentas de OCR no contexto de plataformas de mídias sociais com informações apresentadas no idioma Português do Brasil. Com essa contribuição, é possível identificar os pontos de deficiência apresentados pelas ferramentas de OCR ao lidarem com a Língua Portuguesa, com isso, é possível indicar para os desenvolvedores os pontos de otimização para a construção de ferramentas cada vez mais sofisticadas;

- Identificação dos aspectos de imagem que impactam no desempenho das ferramentas de OCR. As plataformas de mídias sociais apresentam uma grande variedade de fontes, estilos e outros recursos de personalização que prejudicam o reconhecimento do texto. O estudo apresentado neste trabalho ajuda a indicar os pontos de melhoria para que as ferramentas de OCR possam lidar bem com a diversidade de recursos de edição disponíveis para a criação de imagens;

- Verificação dos impactos da integração das ferramentas de OCR em sistemas de detecção, moderação e combate à desinformação, baseados em aprendizado de máquina. Com essa contribuição, foi possível provar que a seleção da ferramenta de OCR produz impacto direto na qualidade dos sistemas de detecção, contenção e moderação de conteúdo. Esta contribuição ajuda a indicar os caminhos para a construção de sistemas de moderação e combate à desinformação mais robustos.

De forma geral, este trabalho fornece descobertas valiosas que podem ser úteis como guia para profissionais e pesquisadores interessados na tarefa de extrair texto de imagens divulgadas em plataformas digitais com desinformação para a construção de mecanismos eficazes para identificar e conter esse tipo de conteúdo nesses ambientes. Além disso, a comunidade interessada no desempenho de técnicas de OCR para Português também pode se beneficiar dos resultados. Os testes de classificação apresentados, embora realizados com um algoritmo classificador modesto, comprovam que, de fato, a qualidade dos dados retornados pelas ferramentas de OCR impactam diretamente no processo de detecção de desinformação. Assim, a escolha da melhor ferramenta para aplicação neste contexto, impactará diretamente no desempenho do sistema de moderação ou contenção de desinformação.

Como produções científicas oriundas deste estudo, foram publicados dois artigos: O primeiro se trata da construção de um repositório[1] de dados, contendo 12.209 imagens sintéticas para o estudo de desinformação. O artigo aborda desde a construção do repositório até a disponibilização do mesmo e de um *script*[2] de geração de dados para a comunidade acadêmica e para o público em geral (SANTOS; SILVA; REIS, 2023b). Já o segundo artigo, apresenta uma avaliação comparativa entre as ferramentas de OCR em que são indicados os aspectos de imagem que impactam negativamente no desempenho de cada ferramenta (SANTOS; SILVA; REIS, 2023a). Além disso, durante o mestrado houve colaboração em outro artigo que se trata de uma avaliação de desempenho de ferramentas de transcrição de áudio em Português do Brasil utilizadas para análise de dados da Internet (SANTOS et al., 2023).

[1] <https://zenodo.org/records/8111884>
[2] <https://github.com/MaVILab-UFV/ImageFactCk.br-dataset-SBBD-DSW-2023>

6.1 Trabalhos Futuros

No que tange aos trabalhos futuros e de forma a aprimorar os resultados, planejamos conduzir novos testes de classificação com aplicação de algoritmos mais avançados, como os inseridos no campo de aprendizado profundo (do inglês, *Deep Learning*). Novos experimentos para geração de dados sintéticos também estão sendo planejados, como a adição de realismo nas imagens. A construção de conjuntos de dados bem estruturados é fundamental para a condução de experimentos com técnicas de *Machine Learning* e, isso se faz ainda mais necessário em contextos regionais, como é o caso do idioma. Atualmente existem poucos repositórios disponíveis para estudos em desinformação no idioma Português do Brasil. No que diz respeito a novos experimentos com ferramentas de OCR, estão sendo planejadas análises multivariadas, onde serão estudadas a qualidade da recuperação de informação textual em cenários com variações de múltiplos aspectos de imagem.

6.2 Limitações

Devido às limitações de licença dos mecanismos de OCR pagos e do tempo de consulta por algumas ferramentas gratuitas, não foi possível realizar experimentos mais extensos. A execução dos experimentos de classificação descritos neste trabalho, foi uma etapa extra de avaliação. Devido ao volume de dados disponível, não foi possível conduzir experimentos com métodos mais avançados, como as técnicas de aprendizado profundo.

Referências

AHMED, A. Images, videos or links? study shows most common social media post types on facebook and instagram. 2021. Disponível em: <https://www.digitalinformationworld.com/2021/05/images-videos-or-links-study-shows-most.html>. Acesso em: 12 abr. 2021.

AHMED, H.; TRAORE, I.; SAAD, S. Detection of online fake news using n-gram analysis and machine learning techniques. In: SPRINGER. **Intelligent, Secure, and Dependable Systems in Distributed and Cloud Environments: First International Conference, ISDDC 2017, Vancouver, BC, Canada, October 26-28, 2017, Proceedings 1**. [S.l.], 2017. p. 127–138.

AKBAR, S. Z. et al. Misinformation as a window into prejudice: Covid-19 and the information environment in india. **Proc. of the ACM on Human-Computer Interaction**, v. 4, n. CSCW3, p. 1–28, 2021.

ALI, S.; QAYYUM, S. A pragmatic comparison of four different programming languages. **University of Management And Technology**, 2021.

ALLEN, K. et al. *Machine literature searching VIII.* ***Operational criteria for designing information retrieval systems.*** **American Documentation (pre-1986)**, v. 6, n. 2, p. 93, 1955.

ALZUBAIDI, L. et al. A survey on deep learning tools dealing with data scarcity: definitions, challenges, solutions, tips, and applications. **Journal of Big Data**, Springer, v. 10, n. 1, p. 46, 2023.

ANTONELLO, L. L. R. Identificação automática de placa de veículos através de processamento de imagem e visão computacional. 2017.

AWEL, M. A.; ABIDI, A. I. Review on optical character recognition. **International Research Journal of Engineering and Technology (IRJET)**, v. 6, n. 6, p. 3666–3669, 2019.

BACHMANN, M. **Levenshtein**. [S.l.], 2022. Disponível em: <https://pypi.org/project/python-Levenshtein/>. Acesso em: 22 fev. 2023.

BAEZA-YATES, R.; RIBEIRO-NETO, B. et al. **Modern information retrieval**. [S.l.]: ACM press New York, 1999. v. 463.

BAKERS, S. **Social media trends report**. 2021.

BARRETO JR, I. F.; VENTURI JR, G. Fake news em imagens: um esforço de compreensão da estratégia comunicacional exitosa na eleição presidencial brasileira de 2018. **Revista Debates**, v. 14, n. 1, 2020.

BIANCOVILLI, P.; MAKSZIN, L.; JURBERG, C. Misinformation on social networks during the novel coronavirus pandemic: a quali-quantitative case study of brazil. **BMC Public Health**, BioMed Central, v. 21, n. 1, p. 1–10, 2021.

BOIDIDOU, C. et al. Challenges of computational verification in social multimedia. In: **Proc. of the Int'l ACM Conference on World Wide Web (WWW) Companion**. [S.l.: s.n.], 2014. p. 743–748.

BROWN, D. **Origem**. [S.l.]: Editora Arqueiro, 2017. v. 5.

CAMÕES, L. V. **Os Lusíadas, poema épico**. [S.l.]: Didot, 1818.

CASTRO, J. D. B. et al. Improvement optical character recognition for structured documents using generative adversarial networks. In: IEEE. **2021 21st International Conference on Computational Science and Its Applications (ICCSA)**. [S.l.], 2021. p. 285–292.

CASTRO, S. P. B. **Digitalização e automação de processos de controlo interno na TRIDEC**. Tese (Doutorado), 2022.

CHAUDHURI, A. et al. **Optical character recognition systems**. [S.l.]: Springer, 2017.

CHEN, T.; GUESTRIN, C. Xgboost: A scalable tree boosting system. In: **Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining**. [S.l.: s.n.], 2016. p. 785–794.

CHENG, Z. et al. Aon: Towards arbitrarily-oriented text recognition. In: **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition**. [S.l.: s.n.], 2018. p. 5571–5579.

COIMBRA, J. L. M. et al. Técnicas multivariadas aplicadas ao estudo da fauna do solo: contrastes multivariados e análise canônica discriminante. **Revista Ceres**, Universidade Federal de Viçosa, v. 54, n. 313, p. 271–277, 2007.

DASH, B. A hybrid solution for extracting information from unstructured data using optical character recognition (ocr) with natural language processing (nlp). 2021.

DETLEFSEN, N. S. et al. Torchmetrics-measuring reproducibility in pytorch. **Journal of Open Source Software**, v. 7, n. 70, p. 4101, 2022.

DEVELOPERS, G. Classificação: curva roc e auc | google developers. **Google Developers**, 2022. Disponível em: <https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc?hl=pt-br>. Acesso em: 10 nov. 2023.

DEVLIN, J. et al. Bert: Pre-training of deep bidirectional transformers for language understanding. **arXiv preprint arXiv:1810.04805**, 2018.

DIXON, S. J. **Global Facebook pages fan engagement rate 2023, by type of post**. 2024.

DROBAC, S.; LINDÉN, K. Optical character recognition with neural networks and post-correction with finite state methods. **International Journal on Document Analysis and Recognition (IJDAR)**, Springer, v. 23, n. 4, p. 279–295, 2020.

ENE, A.; ENE, A. An application of levenshtein algorithm in vocabulary learning. In: IEEE. **2017 9th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)**. [S.l.], 2017. p. 1–4.

GALHARDI, C. P. et al. Fact or fake? an analysis of disinformation regarding the covid-19 pandemic in brazil. **Ciência & Saúde Coletiva**, SciELO Public Health, v. 25, p. 4201–4210, 2020.

GEEKS, G. Xgboost. **Geeks for Geeks**, 2023. Disponível em: <https://www.geeksforgeeks.org/xgboost/>. Acesso em: 10 nov. 2023.

GEORGIEVA, P.; ZHANG, P. Optical character recognition for autonomous stores. In: IEEE. **2020 IEEE 10th International Conference on Intelligent Systems (IS)**. [S.l.], 2020. p. 69–75.

GOLLAPUDI, S. **Practical machine learning**. [S.l.]: Packt Publishing Ltd, 2016.

GOMES, L. e. S. **Um modelo para avaliação de similaridade de strings alfanuméricas resultantes do processo de reconhecimento óptico de caracteres**. Dissertação (Mestrado), 2022.

GONGANE, V. U.; MUNOT, M. V.; ANUSE, A. D. Detection and moderation of detrimental content on social media platforms: Current status and future directions. **Social Network Analysis and Mining**, Springer, v. 12, n. 1, p. 129, 2022.

GREENHALGH, J.; MIRMEHDI, M. Recognizing text-based traffic signs. **IEEE Transactions on Intelligent Transportation Systems**, IEEE, v. 16, n. 3, p. 1360–1369, 2014.

HAMDI, A. et al. In-depth analysis of the impact of ocr errors on named entity recognition and linking. **Natural Language Engineering**, Cambridge University Press, v. 29, n. 2, p. 425–448, 2023.

HAO, Q. et al. It's not what it looks like: Manipulating perceptual hashing based applications. In: **Proc. of the ACM Conference on Computer and Communications Security (SIGSAC)**. [S.l.: s.n.], 2021. p. 69–85.

HAUMAHU, J.; PERMANA, S.; YADDARABULLAH, Y. Fake news classification for indonesian news using extreme gradient boosting (xgboost). In: IOP PUBLISHING. **IOP Conference Series: Materials Science and Engineering**. [S.l.], 2021. v. 1098, n. 5, p. 052081.

HENDRAWAN, I. R.; UTAMI, E.; HARTANTO, A. D. Comparison of naïve bayes algorithm and xgboost on local product review text classification. **Edumatic: Jurnal Pendidikan Informatika**, v. 6, n. 1, p. 143–149, 2022.

HEYBURN, R. et al. Machine learning using synthetic and real data: similarity of evaluation metrics for different healthcare datasets and for different algorithms. In: WORLD SCIENTIFIC. **Data Science and Knowledge Engineering for Sensing Decision Support: Proceedings of the 13th International FLINS Conference (FLINS 2018)**. [S.l.], 2018. p. 1281–1291.

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION. **ISO 5725-1: 1994: Accuracy (Trueness and Precision) of Measurement Methods and Results-Part 1: General Principles and Definitions. International Organization for Standardization, 1994**. [S.l.], 1994.

JONES, K. S. A statistical interpretation of term specificity and its application in retrieval. **Journal of documentation**, MCB UP Ltd, v. 28, n. 1, p. 11–21, 1972.

JUNDIAí, P. M. d. Ação de cadastro para mutirão de atendimento visual é falsa. 2024. Disponível em: <https://jundiai.sp.gov.br/noticias/2024/02/16/acao-de-cadastro-para-mutirao-de-atendimento-visual-e-falsa/>. Acesso em: 21 mai. 2024.

JUNG, K.; KIM, K. I.; JAIN, A. K. Text information extraction in images and video: a survey. **Pattern recognition**, Elsevier, v. 37, n. 5, p. 977–997, 2004.

KANAI, J. et al. Performance metrics for document understanding systems. In: IEEE. **Proceedings of 2nd International Conference on Document Analysis and Recognition (ICDAR'93)**. [S.l.], 1993. p. 424–427.

KAZEMI, A. et al. Research note: Tiplines to uncover misinformation on encrypted platforms: A case study of the 2019 indian general election on whatsapp. **Harvard Kennedy School Misinformation Review**, 2022.

KRSTOVSKI, K.; RYU, A.; KOGUT, B. Evons: A dataset for fake and real news virality analysis and prediction. In: **Proc. of the Int'l Conference on Computational Linguistics (COLING)**. [S.l.: s.n.], 2022.

KUMAR, V. et al. Extraction of information from bill receipts using optical character recognition. In: IEEE. **2020 international conference on smart electronics and communication (ICOSEC)**. [S.l.], 2020. p. 72–77.

LEUNG, K. Evaluate ocr output quality with character error rate (cer) and word error rate (wer) - key concepts, examples, and python implementation of measuring optical character recognition output quality. 2021. Disponível em: <https://towardsdatascience.com/evaluating-ocr-output-quality-with-character-error-rate-cer-and-word-error-rate-wer-853175297510>. Acesso em: 25 nov. 2022.

LI, Y. et al. Fake news detection based on the correlation extension of multimodal information. In: SPRINGER. **Asia-Pacific Web (APWeb) and Web-Age Information Management (WAIM) Joint International Conference on Web and Big Data**. [S.l.], 2022. p. 443–450.

LI, Y. et al. Fake news detection based on the correlation extension of multimodal information. In: **Web and Big Data: International Joint Conference**. [S.l.: s.n.], 2023. p. 443–450.

MANNING, C.; RAGHAVAN, P.; SCHUTZE, H. Term weighting, and the vector space model. **Introduction to information retrieval**, Cambridge University Press Cambridge, p. 109–133, 2008.

MARIANO, D. et al. Revista brasileira de bioinformática e biologia computacional. 2021.

MEMON, J. et al. Handwritten optical character recognition (ocr): A comprehensive systematic literature review (slr). **IEEE access**, IEEE, v. 8, p. 142642–142668, 2020.

MORENO, J.; BRESSAN, G. Factck. br: a new dataset to study fake news. In: **Proceedings of the 25th Brazillian Symposium on Multimedia and the Web**. [S.l.: s.n.], 2019. p. 525–527.

NANDY, R. Facebook and the covid-19 crisis: building solidarity through community feeling. **Human Arenas**, Springer, v. 5, n. 4, p. 609–619, 2022.

NAVIN, J. M.; PANKAJA, R. Performance analysis of text classification algorithms using confusion matrix. **International Journal of Engineering and Technical Research (IJETR)**, v. 6, n. 4, p. 75–8, 2016.

NETO, F. et al. Computação em nuvem e aprendizado de máquina para análise de grandes volumes de dados educacionais. In: SBC. **Anais do XVII Encontro Nacional de Inteligência Artificial e Computacional**. [S.l.], 2020. p. 58–69.

NGUYEN, Q.-D.; LE, D.-A.; ZELINKA, I. Ocr error correction for unconstrained vietnamese handwritten text. In: **Proceedings of the 10th International Symposium on Information and Communication Technology**. [S.l.: s.n.], 2019. p. 132–138.

NIKOLENKO, S. I. **Synthetic data for deep learning**. [S.l.]: Springer, 2021. v. 174.

NOVELIANTO, K. T. **FORM PROCESSING AUTOMATION FRAMEWORK USING ZONAL OPTICAL CHARACTER RECOGNITION AND WORKFLOW METHODOLOGY**. Tese (Doutorado) — PRESIDENT UNIVERSITY, 2021.

PATEL, C.; PATEL, A.; PATEL, D. Optical character recognition by open source ocr tool tesseract: A case study. **International Journal of Computer Applications**, Electronic Publication: Digital Object Identifiers (DOIs), v. 55, n. 10, p. 50–56, 2012.

PÉREZ-ROSAS, V. et al. Automatic detection of fake news. In: ASSOCIATION FOR COMPUTATIONAL LINGUISTICS. [S.l.], 2019.

PILLAI, I.; FUMERA, G.; ROLI, F. Designing multi-label classifiers that maximize f measures: State of the art. **Pattern Recognition**, Elsevier, v. 61, p. 394–404, 2017.

REIS, J. C. et al. Supervised learning for fake news detection. **IEEE Intelligent Systems**, IEEE, v. 34, n. 2, p. 76–81, 2019.

REIS, J. C. et al. A dataset of fact-checked images shared on whatsapp during the brazilian and indian elections. In: **Proc. of the Int'l AAAI Conference on Web and Social Media (ICWSM)**. [S.l.: s.n.], 2020. p. 903–908.

RESENDE, G. et al. (mis)information dissemination in whatsapp: Gathering, analyzing and countermeasures. In: **Proc. of the ACM Web Conference (WWW)**. [S.l.: s.n.], 2019. p. 818–828.

REUL, C. et al. Ocr4all—an open-source tool providing a (semi-) automatic ocr workflow for historical printings. **Applied Sciences**, MDPI, v. 9, n. 22, p. 4853, 2019.

REUL, C. et al. State of the art optical character recognition of 19th century fraktur scripts using open source engines. **arXiv preprint arXiv:1810.03436**, 2018.

ROBERTSON, S. Understanding inverse document frequency: on theoretical arguments for idf. **Journal of documentation**, Emerald Group Publishing Limited, v. 60, n. 5, p. 503–520, 2004.

ROMANELLO, M.; NAJEM-MEYER, S.; ROBERTSON, B. Optical character recognition of 19th century classical commentaries: the current state of affairs. In: **The 6th International Workshop on Historical Document Imaging and Processing**. [S.l.: s.n.], 2021. p. 1–6.

ROUCHDI, R. K.; ELLOTF, M. A.; BAYOUMI, H. Egyptian press archive of cedej. a challenging case study of arabic ocr. **Égypte/Monde arabe**, Cairn, p. 41–55, 2020.

SABAT, B. O.; FERRER, C. C.; NIETO, X. Giro-i. Hate speech in pixels: Detection of offensive memes towards automatic moderation. **arXiv preprint arXiv:1910.02334**, 2019.

SABER, S.; AHMED, A.; HADHOUD, M. Robust metrics for evaluating arabic ocr systems. In: IEEE. **International Image Processing, Applications and Systems Conference**. [S.l.], 2014. p. 1–6.

SANTOS, F. C. C. Artificial intelligence in automated detection of disinformation: a thematic analysis. **Journalism and Media**, MDPI, v. 4, n. 2, p. 679–687, 2023.

SANTOS, J. et al. Avaliação do desempenho de ferramentas de transcrição de Áudio em português para análise de dados da web. In: **Anais do XII Brazilian Workshop on Social Network Analysis and Mining**. Porto Alegre, RS, Brasil: SBC, 2023. p. 228–233. ISSN 2595-6094. Disponível em: <https://sol.sbc.org.br/index.php/brasnam/article/view/24804>.

SANTOS, R. L.; MONTEIRO, R. A.; PARDO, T. A. The fake. br corpus-a corpus of fake news for brazilian portuguese. In: . [S.l.: s.n.], 2018.

SANTOS, Y.; SILVA, M.; REIS, J. C. Evaluation of optical character recognition (ocr) systems dealing with misinformation in portuguese. In: IEEE. **2023 36th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)**. [S.l.], 2023. p. 223–228.

SANTOS, Y.; SILVA, M.; REIS, J. C. S. Imagefactck.br: Repositório de imagens para a detecção de desinformação disseminada em plataformas digitais. In: **Anais do V Dataset Showcase Workshop**. Porto Alegre, RS, Brasil: SBC, 2023. p. 87–98. ISSN 0000-0000. Disponível em: <https://sol.sbc.org.br/index.php/dsw/article/view/25508>.

SANTOS, Y. J. A. d. Avaliação de técnicas de aprendizado supervisionado no auxílio a diagnósticos médicos. 2021.

SCHANTZ, H. F. **History of OCR, optical character recognition**. [S.l.]: Recognition Technologies Users Association, 1982.

SHAHID, W. et al. Detecting and mitigating the dissemination of fake news: Challenges and future research opportunities. **IEEE Transactions on Computational Social Systems**, IEEE, 2022.

SHAIKH, J.; PATIL, R. Fake news detection using machine learning. In: IEEE. **2020 IEEE International Symposium on Sustainable Energy, Signal Processing and Cyber Security (iSSSC)**. [S.l.], 2020. p. 1–5.

SHARMA, D. K.; GARG, S. Ifnd: a benchmark dataset for fake news detection. **Complex & Intelligent Systems**, Springer, p. 1–21, 2021.

SHEELA, S. et al. Enhancing accessibility: object detection and optical character recognition for empowering visually impaired individuals. IET, 2023.

SHU, K. et al. Fake news detection on social media: A data mining perspective. **ACM SIGKDD explorations newsletter**, ACM New York, NY, USA, v. 19, n. 1, p. 22–36, 2017.

SILVA JR, A. C. et al. Geração de dados sintéticos para classificação de disléxicos por meio de aprendizado de máquina. **Journal of Health Informatics**, v. 13, n. 1, 2021.

SILVA, R. L. da. A exposição de crianças e adolescentes aos discursos de ódio na internet: alternativas de enfrentamento dessa forma de violência. **DIREITOS HUMANOS E JUSTIÇA SOCIAL**, p. 229, 2014.

SORENSEN, T. A. *A method of establishing groups of equal amplitude in plant sociology based on similarity of species content and its application to analyses of the vegetation on Danish commons*. **Biol. Skar., v. 5, p. 1-34**, 1948.

SOUSA, D. da R.; FREITAS, C. O. de A. Os desafios e as perspectivas para a regulamentação da internet das coisas no brasil: The challenges and perspectives for regulating the internet of things in brazil. **International Journal of Digital Law**, v. 3, n. 2, p. 51–68, 2022.

SOUZA, F. C. de. **BERTimbau: pretrained BERT models for brazilian portuguese= BERTimbau: modelos BERT pré-treinados para português brasileiro**. Tese (Doutorado) — [sn], 2020.

SOUZA, M. P. de et al. A linguistic-based method that combines polarity, emotion and grammatical characteristics to detect fake news in portuguese. In: **Proceedings of the Brazilian Symposium on Multimedia and the Web**. [S.l.: s.n.], 2020. p. 217–224.

SPRINGMANN, U.; LUEDELING, A. Ocr of historical printings with an application to building diachronic corpora: A case study using the ridges herbal corpus. **Digital Humanities Quarterly**, Alliance Of Digital Humanities Organizations, n. 2, 2017.

VIEIRA, V. G. et al. Avaliaçao de abordagens baseadas em software para melhoria no desempenho e consumo de energia de algoritmos de aprendizado de máquina. 2021.

WANG, W. et al. Validating multimedia content moderation software via semantic fusion. In: **Proceedings of the 32nd ACM SIGSOFT International Symposium on Software Testing and Analysis**. [S.l.: s.n.], 2023. p. 576–588.

WEI, X. et al. Machine learning insights in predicting heavy metals interaction with biochar. **Biochar**, Springer, v. 6, n. 1, p. 1–11, 2024.

YE, Q.; DOERMANN, D. Text detection and recognition in imagery: A survey. **IEEE transactions on pattern analysis and machine intelligence**, IEEE, v. 37, n. 7, p. 1480–1500, 2014.

YIN, X.-C. et al. Text detection, tracking and recognition in video: a comprehensive survey. **IEEE Transactions on Image Processing**, IEEE, v. 25, n. 6, p. 2752–2773, 2016.

ZACHLOD, C. et al. Analytics of social media data–state of characteristics and application. **Journal of Business Research**, Elsevier, v. 144, p. 1064–1076, 2022.

Apêndices

APÊNDICE A – Resultados dos Experimentos de Classificação com o Aspecto Ângulo

As Tabelas 21, 22, 23, 24 e 25 apresentam os resultados completos obtidos através da execução dos experimentos de classificação com o XGBoost com o aspecto Ângulo. As Tabelas exibem os valores da classificação com dados processados pelos OCRs em comparação com os dados textuais originais `FACTCK.BR`. Em destaque nas colunas de média, estão os maiores valores de média registrados entre as ferramentas de OCR em cada métrica.

Tabela 21 – Resultados de acurácia com o repositório produzido pelos OCRs com o aspecto ângulo.

Base	0	(-5 e 5)	(-10 e 10)	(-15 e 15)	(-20 e 20)	(-25 e 25)	(-30 e 30)	Media	Desvio Padrao
FACTCK.BR	71.64	71.32	71.32	71.32	71.32	71.32	71.32	71.36	0.12
PyTesseract	71.64	66.67	57.75	54.65	50.78	50.39	50.00	57.41	8.60
Microsoft OCR	71.64	71.32	71.32	71.32	71.32	70.93	71.32	**71.31**	0.21
Google Drive	71.64	71.32	71.32	71.32	70.93	71.32	70.93	71.25	0.25
Google Vision	71.64	71.32	70.93	70.93	71.32	70.93	70.93	71.14	0.29
EasyOCR	71.64	66.67	69.38	70.93	69.77	69.77	70.93	69.87	1.63

Tabela 22 – Resultados de AUC com o repositório produzido pelos OCRs com o aspecto ângulo.

Base	0	(-5 e 5)	(-10 e 10)	(-15 e 15)	(-20 e 20)	(-25 e 25)	(-30 e 30)	Media	Desvio Padrao
FACTCK.BR	75.08	74.58	74.58	74.58	74.58	74.58	74.58	74.65	0.19
PyTesseract	74.81	71.00	60.70	54.56	50.02	50.00	49.62	58.67	10.53
Microsoft OCR	75.13	74.65	74.65	74.65	74.65	74.39	74.65	**74.68**	0.22
Google Drive	75.15	74.65	74.65	74.65	74.39	74.65	74.37	74.65	0.26
Google Vision	75.13	74.65	74.39	74.39	74.65	74.39	74.37	74.57	0.28
EasyOCR	75.19	72.90	72.82	74.01	73.16	73.01	74.43	73.65	0.91

Tabela 23 – Resultados de F1 macro com o repositório produzido pelos OCRs com o aspecto ângulo.

Base	0	(-5 e 5)	(-10 e 10)	(-15 e 15)	(-20 e 20)	(-25 e 25)	(-30 e 30)	Media	Desvio Padrao
FACTCK.BR	71.41	71.07	71.07	71.07	71.07	71.07	71.07	71.12	0.13
PyTesseract	71.41	66.21	52.87	42.40	35.03	33.51	33.33	47.82	15.96
Microsoft OCR	71.41	71.07	71.07	71.07	71.07	70.70	71.07	**71.06**	0.21
Google Drive	71.41	71.07	71.07	71.07	70.70	71.07	70.70	71.01	0.25
Google Vision	71.41	71.07	70.70	70.70	71.07	70.70	70.70	70.91	0.28
EasyOCR	71.41	66.42	69.21	70.74	69.70	69.70	70.86	69.72	1.65

Tabela 24 – Resultados de Precisão com o repositório produzido pelos OCRs com o aspecto ângulo.

Mecanismo	0	(-5 e 5)	(-10 e 10)	(-15 e 15)	(-20 e 20)	(-25 e 25)	(-30 e 30)	Media	Desvio Padrao
FACTCK.BR	68.35	68.42	68.42	68.42	68.42	68.42	68.42	68.41	0.03
PyTesseract	68.35	63.92	54.98	52.63	50.59	50.39	50.19	55.87	7.33
Microsoft OCR	68.35	68.42	68.42	68.42	68.42	68.21	68.42	**68.38**	0.08
Google Drive	68.35	68.42	68.42	68.42	68.21	68.42	68.21	68.35	0.10
Google Vision	68.35	68.42	68.21	68.21	68.42	68.21	68.21	68.29	0.10
EasyOCR	68.35	64.67	67.35	68.46	68.57	68.57	69.50	67.92	1.57

Tabela 25 – Resultados de *Recall* com o repositório produzido pelos OCRs com o aspecto ângulo.

Mecanismo	0	(-5 e 5)	(-10 e 10)	(-15 e 15)	(-20 e 20)	(-25 e 25)	(-30 e 30)	Media	Desvio Padrao
FACTCK.BR	80.60	80.00	80.00	80.00	80.00	80.00	80.00	80.09	0.23
PyTesseract	80.60	77.69	89.23	100.00	99.23	100.00	99.23	**92.28**	9.78
Microsoft OCR	80.60	80.00	80.00	80.00	80.00	79.23	80.00	79.98	0.40
Google Drive	80.60	80.00	80.00	80.00	79.23	80.00	79.23	79.87	0.49
Google Vision	80.60	80.00	79.23	79.23	80.00	79.23	79.23	79.65	0.55
EasyOCR	80.60	74.62	76.15	78.46	73.85	73.85	75.38	76.13	2.54

APÊNDICE B – Resultados dos Experimentos de Classificação com o Aspecto Estilo da Fonte

As Tabelas 26, 27, 28, 29 e 30 apresentam os resultados completos obtidos através da execução dos experimentos de classificação com o XGBoost com o aspecto Estilo da Fonte. As Tabelas exibem os valores da classificação com dados processados pelos OCRs em comparação com os dados textuais originais `FACTCK.BR`. Em destaque nas colunas de média, estão os maiores valores de média registrados entre as ferramentas de OCR em cada métrica.

Tabela 26 – Resultados de Acurácia com o repositório produzido pelos OCRs com o aspecto estilo da fonte.

Base	**Comic Sans MS3**	**Modern Serif**	**KG Part-of Me**	**Apple Garamond**	**Timeless**	**Type Machine**	**Nixie One**	**KG Lego House**	**KG Sorry Not Sorry**	Media	Desvio Padrão
FACTCK.BR	63.08	63.28	63.08	63.36	62.88	61.72	65.08	64.29	62.88	63.29	0.94
PyTesseract	63.08	63.28	63.85	63.36	63.64	58.59	65.87	62.70	61.36	62.86	1.99
Microsoft OCR	63.08	63.28	63.08	63.36	63.64	61.72	65.08	62.70	62.88	**63.20**	0.89
Google Drive	63.08	63.28	62.31	63.36	62.88	60.16	65.08	61.90	62.12	62.69	1.33
Google Vision	63.08	63.28	63.85	63.36	62.88	60.94	65.08	61.90	63.64	63.11	1.17
Easy OCR	63.08	64.06	63.85	63.36	64.39	60.94	64.29	61.90	57.58	62.60	2.21

Tabela 27 – Resultados de AUC com o repositório produzido pelos OCRs com o aspecto estilo da fonte.

Base	**Comic Sans MS3**	**Modern Serif**	**KG Part-of Me**	**Apple Garamond**	**Timeless**	**Type Machine**	**Nixie One**	**KG Lego House**	**KG Sorry Not Sorry**	Media	Desvio Padrão
FACTCK.BR	66.98	67.11	67.28	67.30	67.13	66.09	68.59	67.84	67.13	67.27	0.67
PyTesseract	66.93	66.84	66.17	67.30	67.44	62.24	69.05	66.05	66.53	66.51	1.83
Microsoft OCR	66.98	67.11	67.33	67.30	67.69	66.04	68.59	66.47	67.03	67.17	0.72
Google Drive	66.98	66.79	66.96	67.30	67.36	64.98	68.59	65.25	66.83	66.78	1.09
Google Vision	66.98	67.11	68.16	67.30	67.47	65.71	68.59	65.52	68.02	**67.21**	1.04
Easy OCR	67.62	67.33	67.99	67.20	67.76	65.34	67.96	65.90	63.44	66.73	1.54

Tabela 28 – Resultados de F1 Macro com o repositório produzido pelos OCRs com o aspecto estilo da fonte.

Base	**Comic Sans MS3**	**Modern Serif**	**KG Part-of Me**	**Apple Garamond**	**Timeless**	**Type Machine**	**Nixie One**	**KG Lego House**	**KG Sorry Not Sorry**	Media	Desvio Padrão
FACTCK.BR	62.51	62.62	62.64	62.87	62.39	61.19	64.64	63.77	62.39	62.78	0.96
PyTesseract	62.51	62.62	63.36	62.87	63.09	58.02	65.38	62.01	60.86	62.30	2.01
Microsoft OCR	62.51	62.62	62.64	62.87	63.09	61.19	64.64	62.01	62.52	**62.68**	0.92
Google Drive	62.51	62.62	61.80	62.87	62.39	59.44	64.64	61.11	61.69	62.12	1.41
Google Vision	62.51	62.62	63.48	62.87	62.39	60.32	64.64	61.11	63.33	62.59	1.28
Easy OCR	62.51	63.49	63.36	62.87	63.93	60.32	63.77	61.11	56.94	62.03	2.27

Tabela 29 – Resultados de Precisão com o repositório produzido pelos OCRs com o aspecto estilo da fonte.

Base	**Comic Sans MS3**	**Modern Serif**	**KG Part-of Me**	**Apple Garamond**	**Timeless**	**Type Machine**	**Nixie One**	**KG Lego House**	**KG Sorry Not Sorry**	Media	Desvio Padrão
FACTCK.BR	67.35	65.31	68.00	66.67	66.67	65.31	68.00	67.35	66.67	66.81	1.00
PyTesseract	67.35	65.31	69.39	66.67	68.00	61.22	69.39	65.96	64.71	66.44	2.56
Microsoft OCR	67.35	65.31	68.00	66.67	68.00	65.31	68.00	65.96	66.04	**66.74**	1.14
Google Drive	67.35	65.31	67.35	66.67	66.67	63.83	68.00	65.22	65.38	66.20	1.34
Google Vision	67.35	65.31	68.63	66.67	66.67	64.58	68.00	65.22	66.67	66.56	1.34
Easy OCR	67.35	66.00	69.39	66.67	68.63	64.58	67.35	65.22	60.00	66.13	2.76

Tabela 30 – Resultados de *Recall* com o repositório produzido pelos OCRs com o aspecto estilo da fonte.

Base	**Comic Sans MS3**	**Modern Serif**	**KG Part-of Me**	**Apple Garamond**	**Timeless**	**Type Machine**	**Nixie One**	**KG Lego House**	**KG Sorry Not Sorry**	Media	Desvio Padrão
FACTCK.BR	50.77	51.61	51.52	52.31	51.52	50.00	54.84	53.23	51.52	51.92	1.41
PyTesseract	50.77	51.61	51.52	52.31	51.52	46.88	54.84	50.00	50.00	51.05	2.13
Microsoft OCR	50.77	51.61	51.52	52.31	51.52	50.00	54.84	50.00	53.03	**51.73**	1.53
Google Drive	50.77	51.61	50.00	52.31	51.52	46.88	54.84	48.39	51.52	50.87	2.29
Google Vision	50.77	51.61	53.03	52.31	51.52	48.44	54.84	48.39	54.55	51.72	2.31
Easy OCR	50.77	53.23	51.52	52.31	53.03	48.44	53.23	48.39	45.45	50.71	2.73

APÊNDICE C – Resultados dos Experimentos de Classificação com o Aspecto Plano de Fundo

As Tabelas 31, 32, 33, 34 e 35 apresentam os resultados completos obtidos através da execução dos experimentos de classificação com o XGBoost com o aspecto Plano de Fundo. As Tabelas exibem os valores da classificação com dados processados pelos OCRs em comparação com os dados textuais originais `FACTCK.BR`. Em destaque nas colunas de média, estão os maiores valores de média registrados entre as ferramentas de OCR em cada métrica.

Tabela 31 – Resultados de Acurácia com o repositório produzido pelos OCRs com o aspecto plano de fundo.

Mecanismo	**Branco**	**Azul**	**Azul Médio**	**Azul Claro**	**Azul Escuro**	**im1**	**im2**	Media	Desvio Padrão
FACTCK.BR	62.88	62.88	62.88	62.88	62.88	62.88	62.88	62.88	0.00
PyTesseract	63.64	56.82	63.64	63.64	55.30	50.00	50.00	57.58	6.20
Microsoft OCR	63.64	63.64	63.64	63.64	63.64	64.39	62.12	**63.53**	0.68
Google Drive	62.88	62.88	62.88	62.88	63.64	63.64	59.85	62.66	1.29
Google Vision	62.88	63.64	62.88	62.88	63.64	63.64	61.36	62.99	0.81
Easy OCR	64.39	62.88	64.39	64.39	54.55	65.91	57.58	62.01	4.25

Tabela 32 – Resultados de AUC com o repositório produzido pelos OCRs com o aspecto plano de fundo.

Mecanismo	**Branco**	**Azul**	**Azul Médio**	**Azul Claro**	**Azul Escuro**	**im1**	**im2**	Media	Desvio Padrão
FACTCK.BR	67.13	67.13	67.13	67.13	67.13	67.13	67.13	67.13	0.00
PyTesseract	67.44	58.12	67.50	67.25	58.75	50.00	50.00	59.86	7.84
Microsoft OCR	67.69	67.50	67.28	67.28	67.50	67.73	65.21	67.17	0.88
Google Drive	67.36	66.97	66.97	67.36	67.50	67.69	63.37	66.74	1.51
Google Vision	67.47	67.50	66.97	66.97	67.50	67.53	66.72	**67.24**	0.34
Easy OCR	67.76	67.96	68.50	67.83	59.00	69.89	63.77	66.39	3.76

Tabela 33 – Resultados de F1 Macro com o repositório produzido pelos OCRs com o aspecto plano de fundo.

Mecanismo	**Branco**	**Azul**	**Azul Médio**	**Azul Claro**	**Azul Escuro**	**im1**	**im2**	Media	Desvio Padrão
FACTCK.BR	62.39	62.39	62.39	62.39	62.39	62.39	62.39	62.39	0.00
PyTesseract	63.09	52.21	63.09	62.95	45.06	33.33	33.33	50.44	13.50
Microsoft OCR	63.09	63.09	63.09	63.09	63.09	63.93	61.40	**62.97**	0.76
Google Drive	62.39	62.39	62.39	62.39	63.09	63.09	59.17	62.13	1.35
Google Vision	62.39	63.09	62.39	62.39	63.09	63.09	60.86	62.47	0.79
Easy OCR	63.93	62.09	63.93	63.93	48.86	65.33	56.13	60.60	5.99

Tabela 34 – Resultados de Precisão com o repositório produzido pelos OCRs com o aspecto plano de fundo.

Mecanismo	**Branco**	**Azul**	**Azul Médio**	**Azul Claro**	**Azul Escuro**	**im1**	**im2**	Media	Desvio Padrão
FACTCK.BR	66.67	66.67	66.67	66.67	66.67	66.67	66.67	66.67	0.00
PyTesseract	68.00	68.00	68.00	68.75	88.89	0.00	0.00	51.66	36.09
Microsoft OCR	68.00	68.00	68.00	68.00	68.00	68.63	66.67	**67.90**	0.59
Google Drive	66.67	66.67	66.67	66.67	68.00	68.00	63.27	66.56	1.58
Google Vision	66.67	68.00	66.67	66.67	68.00	68.00	64.71	66.96	1.20
Easy OCR	68.63	68.09	68.63	68.63	63.64	71.43	61.90	67.28	3.30

Tabela 35 – Resultados de *Recall* com o repositório produzido pelos OCRs com o aspecto plano de fundo.

Mecanismo	**Branco**	**Azul**	**Azul Médio**	**Azul Claro**	**Azul Escuro**	**im1**	**im2**	Media	Desvio Padrão
FACTCK.BR	51.52	51.52	51.52	51.52	51.52	51.52	51.52	51.52	0.00
PyTesseract	51.52	25.76	51.52	50.00	12.12	0.00	0.00	27.27	23.84
Microsoft OCR	51.52	51.52	51.52	51.52	51.52	53.03	48.48	**51.30**	1.36
Google Drive	51.52	51.52	51.52	51.52	51.52	51.52	46.97	50.87	1.72
Google Vision	51.52	51.52	51.52	51.52	51.52	51.52	50.00	**51.30**	0.57
Easy OCR	53.03	48.48	53.03	53.03	21.21	53.03	39.39	45.89	11.99