

**CENTRO ESTADUAL DE EDUCAÇÃO TECNOLÓGICA PAULA
SOUZA**

**FACULDADE DE TECNOLOGIA DE LINS PROF. ANTÔNIO SEABRA
CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE
SISTEMAS**

**FELIPE MORAIS CUNHA
GUILHERME HENRIQUE CURIEL CAETANO**

**TECNOLOGIA OCR E SUAS DISPONIBILIDADES: UMA ANÁLISE
COMPARATIVA ENTRE FUNÇÕES DE RECONHECIMENTO ÓPTICO
DE CARACTERES**

**LINS/SP
2º SEMESTRE/2022**

CENTRO ESTADUAL DE EDUCAÇÃO TECNOLÓGICA PAULA SOUZA
FACULDADE DE TECNOLOGIA DE LINS PROF. ANTÔNIO SEABRA
CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE
SISTEMAS

FELIPE MORAIS CUNHA
GUILHERME HENRIQUE CURIEL CAETANO

TECNOLOGIA OCR E SUAS DISPONIBILIDADES: UMA ANÁLISE
COMPARATIVA ENTRE FUNÇÕES DE RECONHECIMENTO ÓPTICO
DE CARACTERES

Trabalho de Conclusão de Curso (TCC) apresentado à Faculdade de Tecnologia de Lins para a obtenção do título de Tecnólogo (a) em Análise e Desenvolvimento de Sistemas.

Orientador: Prof. Dr. Alexandre Ponce de Oliveira

Morais Cunha, Felipe

M827t Tecnologia OCR e suas disponibilidades: uma análise comparativa entre funções de Reconhecimento Óptico de Caracteres / Felipe Moraes Cunha, Guilherme Henrique Curiel Caetano. — Lins, 2022.

66f.

Trabalho de Conclusão de Curso (Tecnologia em Análise e Desenvolvimento de Sistemas) — Faculdade de Tecnologia de Lins Professor Antonio Seabra: Lins, 2022.

Orientador(a): Dr. Alexandre Ponce de Oliveira

1. OCR. 2. API. 3. Computação. I. Curiel Caetano, Guilherme Henrique. II. Ponce de Oliveira, Alexandre. III. Faculdade de Tecnologia de Lins Professor Antonio Seabra. IV. Título.

CDD 004.21

**FELIPE MORAIS CUNHA
GUILHERME HENRIQUE CURIEL CAETANO**

**TECNOLOGIA OCR E SUAS DISPONIBILIDADES: UMA ANÁLISE COMPARATIVA
ENTRE FUNÇÕES DE RECONHECIMENTO ÓPTICO DE CARACTERES**

Trabalho de Conclusão de Curso apresentado à Faculdade de Tecnologia de Lins "Prof. Antônio Seabra", como parte dos requisitos necessários para a obtenção do título de Tecnólogo em ADS sob orientação do Prof. Dr. Alexandre Ponce de Oliveira.

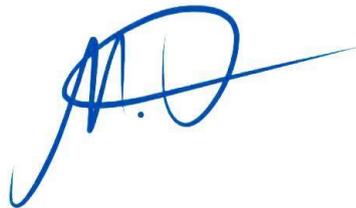
Data de aprovação: 06 de dezembro de 2022



Orientador: Prof. Dr. Alexandre Ponce de Oliveira



Prof. Me. Júlio Fernando Lieira



Prof. Me. Felipe Maciel Rodrigues

Dedico à minha família, que me forneceu apoio no decorrer do curso. E aos meus colegas de sala que me auxiliaram nos momentos de dúvida, clareando a minha mente.

Felipe Moraes Cunha

Dedico esta mais nova conquista à minha família, que esteve sempre ao meu lado fornecendo apoio, conforto e motivação.

Guilherme Henrique Curiel Caetano

AGRADECIMENTOS

O primeiro ano de curso foi o mais complicado, logo após o início das aulas o mundo sofreu com uma pandemia. E, assim como muitos, também fui afetado, mas não em questão de saúde, e sim, financeira. Quando ingressei na Fatec, estava desempregado, e obtive diversas ajudas, e por isso eu sou grato. Minha mãe, meu irmão, e acima de tudo, agradeço à minha prima, Priscila Moraes, que mesmo de longe me auxiliou neste momento tão complicado.

Agradeço imensamente meu parceiro de trabalho, Guilherme Henrique Curiel Caetano, que desde o início do curso me ajudou nos estudos e esteve sempre ao meu lado em trabalhos da faculdade, tornando-se também um dos meus melhores amigos, que levarei para a vida.

Agradeço também os meus colegas de classe, que mesmo passando a maior parte do curso à distância, sempre estiveram perto, e nos unimos cada vez mais.

Felipe Morais Cunha

Esta etapa da minha vida tem sido de extrema importância. Sempre fui inspirado pela tecnologia e pela ciência, e hoje tenho enfim a chance de concluir um curso superior nesta área. Só posso agradecer aos meus professores e colaboradores da Fatec Lins, que direta ou indiretamente contribuíram para que eu concluísse essa jornada. Todos os docentes que passaram pela minha vida acadêmica se empenharam para compartilhar seu conhecimento a todos os alunos da melhor forma possível, não somente com conhecimentos técnicos e acadêmicos, mas com conselhos gerais sobre o comportamento da vida e do mercado de trabalho.

Agradeço ao meu parceiro de trabalho e grande amigo, Felipe Morais Cunha, que igualmente a mim, empenhou-se para o desenvolvimento e conclusão deste Trabalho de Graduação. Assim como nunca deixou de estar ao meu lado nos maiores momentos que passei nesta instituição. Concluo o curso sabendo que não fiz um amigo somente na vida acadêmica, mas também um amigo para a vida toda.

Agradeço, também, ao Prof. Dr. Alexandre Ponce de Oliveira, por ter aceitado ser o nosso orientador, a qual disponibilizou seu tempo e conhecimento para a orientação no decorrer do desenvolvimento.

Não poderia deixar de agradecer a minha família, que teve paciência e foi um ombro durante esses anos, em especial, durante a pandemia da COVID-19, em que passei a maior parte da vida acadêmica com meu Notebook em meu quarto, me esforçando para que mesmo à distância e com o cenário que se encontrava não só o país, mas o mundo, eu pudesse adquirir conhecimento e realizar as atividades, e foi nesse cenário que minha família foi meu grande suporte, meu ombro amigo. Sempre me incentivando a dar o melhor de mim.

Por fim, só tenho a agradecer todos os meus colegas de classe, que por três anos confiaram a mim para ser o representante de sala, e que durante todos esses anos, mesmo com praticamente dois deles em Ensino à distância, formamos laços estreitos e saio sabendo que fiz amizades sólidas e duradouras.

Guilherme Henrique Curiel Caetano

RESUMO

Nos primórdios da era computacional, na década de 1950, iniciou-se o que hoje chamamos de *Optical Character Recognizer* (OCR) que, por meio de pesquisas da automação e dados realizados pela antiga Agência de Segurança das Forças Armadas (AFSA), aspirava o estudo da leitura de textos de forma automatizada. Posteriormente, em 1974, fora desenvolvido pelo inventor Ray Kurzweil, o primeiro produto que realizava o reconhecimento de fontes impressas, que inicialmente, seria utilizada para auxiliar pessoas com deficiência visual a ouvir textos que a máquina reconhecia e vocalizava. A tecnologia OCR evoluiu desde então, não somente sendo uma ferramenta de acessibilidade, como também empresarial, residencial e de segurança. Tendo em vista o aumento da utilização do reconhecimento óptico de caracteres, foi realizado, neste trabalho um estudo de diversos ambientes de *Application Programming Interface* (API) as quais disponibilizam a tecnologia para a decodificação de imagens, a fim de averiguar, analisar e comparar o grau de acuracidade dessas plataformas. Atualmente, algumas empresas disponibilizam API's com tal tecnologia, a citar: Microsoft, Amazon, Google, Clarifai, SentiSight e Api4AI. Expondo as ferramentas a determinados tipos de imagens, foi obtido, por meio do estudo, como cada qual se comporta com três categorias de figuras em dois formatos de obtenção de imagem – fotografia e digitalização. Possibilitando, com os dados obtidos, identificar semelhanças, diferenças e melhorias.

Palavras-chave: OCR. API. Computação.

ABSTRACT

In the early days of the computational era, in the 1950s, what we now call Optical Character Recognizer (OCR) began, which, through automation research and data carried out by the former Armed Forces Security Agency (AFSA), aspired to study of automated text reading. Later, in 1974, inventor Ray Kurzweil had developed the first product that recognized printed fonts, which would initially be used to help people with visual impairments to hear texts that the machine recognized and vocalized. OCR technology has since evolved into not only an accessibility tool, but also an enterprise, home, and security tool. Bearing in mind the increased use of optical character recognition, this work carried out a study of several Application Programming Interface (API) environments which provide the technology for decoding images, in order to ascertain, analyze and compare the degree accuracy of these platforms. Currently, some companies provide APIs with such technology, namely: Microsoft, Amazon, Google, Clarifai, SentiSight and Api4AI. By exposing the tools to certain types of images, it was obtained, through the study, how each one behaves with three categories of figures in two image acquisition formats – photography and digitization. Making it possible, with the data obtained, to identify similarities, differences and improvements.

Keywords: Web. OCR. API. Computing.

LISTA DE ILUSTRAÇÕES

Figura 2.1 - De-skew	21
Figura 2.2 - Binarization	21
Figura 2.3 - Zoning	22
Figura 2.4 - Normalização	22
Figura 2.5 - Detecção de palavras e linhas de texto	23
Figura 2.6 - Reconhecimento do guião	23
Figura 2.7 – Pós-processamento	24
Figura 2.8 – Funcionamento de uma API	25
Figura 2.9 – Home do Eden AI	27
Figura 2.10 – Página principal do VS Code	29
Figura 3.1 – Aba OCR no Eden	33
Figura 3.2 – Página do OCR.	33
Figura 3.3 – Página de teste do OCR	34
Figura 3.4 – imagem de exemplo	34
Figura 3.5 – Upload da imagem para análise	35
Figura 3.6 – Resultados Google Vision e SentiSight	35
Figura 3.7 – reconhecimento do guião no Google	38
Figura 3.8 – reconhecimento do guião no Clarifai	38
Figura 3.9 – Cálculo da precisão do Google	41
Figura 3.10 – Cálculo da precisão do Clarifai	41

Figura 4.1 – Foto base do currículo – imagem 1	44
Figura 4.2 – Response Microsoft Azure e Google – imagem 1	45
Figura 4.3 – Response API4AI e AWS – imagem 1	45
Figura 4.4 – Response SentiSight e Clarifai – imagem 1	46
Figura 4.5 – Fotografia base do documento – imagem 2	47
Figura 4.6 – Response MS Azure e API4AI – imagem 2	47
Figura 4.7 – Response Google e AWS – imagem 2	48
Figura 4.8 – Response SentiSight e Clarifai – imagem 2	48
Figura 4.9 – Imagem base do documento em inglês – imagem 3	49
Figura 4.10 – Response MS Azure e API4AI – imagem 3	49
Figura 4.11 – Response Google e AWS – imagem 3	50
Figura 4.12 – Response SentiSight e Clarifai – imagem 3	50
Figura 4.13 – Response Microsoft Azure e API4AI – imagem 4	51
Figura 4.14 – Response Google e AWS – imagem 4	52
Figura 4.15 – Response SentiSight e Clarifai – imagem 4	52
Figura 4.16 – Response da Microsoft e API4AI – imagem 5	53
Figura 4.17 – Response do Google e AWS – imagem 5	53
Figura 4.18 – Response SentiSight e Clarifai – imagem 5	54
Figura 4.19 – Response Microsoft e API4AI – imagem 6	54
Figura 4.20 – Response Google e AWS – imagem 6	55
Figura 4.21 – Response SentiSight e Clarifai – imagem 6	55

Figura 4.22 – Estrutura do precisaoOCR.py	56
Figura 4.23 – Precisão API4AI na imagem 1	56
Figura 4.24 – reconhecimento do guião no Google	60
Figura 4.25 – reconhecimento do guião na API4AI	61

LISTA DE GRÁFICOS

Gráfico 3.1 – Análise final no Excel	42
Gráfico 4.1 – Precisão das bibliotecas OCR em cada imagem	58
Gráfico 4.2 – Total pela média dos valores obtidos	59

LISTA DE QUADROS

Quadro 3.1 – Response da API do Google	36
Quadro 3.2 – Response da API Clarifai	37
Quadro 3.3 – Estrutura do PrecisaOCR.py	39

LISTA DE TABELAS

Tabela 3.1 - Detalhes PrecisaOCR.py	39
Tabela 4.1 - Especificações da imagem – foto currículo	44
Tabela 4.2 - Especificações da imagem - foto prestação de serviços.....	46
Tabela 4.3 - Especificações da imagem - documento em inglês.....	49
Tabela 4.4 - Especificações da imagem – digitalização do currículo	51
Tabela 4.5 - Especificações da imagem - digitalização prestação de serviços	53
Tabela 4.6 - Especificações da imagem – digitalização do doc em inglês	54

LISTA DE ABREVIATURAS E SIGLAS

AFSA - Air Force Security Agency

AI - Artificial Intelligence

API - Application Programming Interface

CSS - Cascading Style Sheet

Docs - Documents

GB - Giga Bytes

GHz - GigaHertz

HTML - Linguagem de Marcação de HiperTexto

IDE - ambiente de desenvolvimento integrado

JSON - JavaScript Object Notation

Libs - Libraries

NSA - National Security Agency

OCR - Optical Character Recognizer

PVM - Python Virtual Machine

RAM - Random Access Memory

SDK - Software Development Kit

SO - Sistema Operacional

SQL - Standard Query Language

VS - Visual Studio

WWW - World Wide Web

LISTA DE SÍMBOLOS

\$ - Dólar

% - Porcentagem

* - Multiplicação

/ - Barra

@ - Arroba

[] - Colchetes

{ } - Chave

+ - Adição

<> - Tags

SUMÁRIO

1 INTRODUÇÃO	16
2 CONCEITOS BÁSICOS	19
2.1 Definição dos conceitos	19
2.1.1 Optical Character Recognizer	19
2.1.2 Application Programming Interface	24
2.2 Objetos analisados	25
2.2.1 Eden AI	26
2.2.1.1 APIs do Eden AI	27
2.2.2 Visual studio code	28
2.2.3 Python	30
3 METODOLOGIA	31
3.1 Definição dos critérios e métodos de análise	31
3.2 Definições do ambiente	32
3.3 Planejamento e execução	32
4 LEVANTAMENTO DE DADOS E ANÁLISE DE RESULTADOS	43
4.1 Teste no Eden AI	43
4.2 Cálculos da precisão no precisaoocr.py	55
4.3 Totalizando os resultados no Excel e API mais precisa	57
5 CONSIDERAÇÕES FINAIS	62
REFERÊNCIAS BIBLIOGRÁFICAS	64

1 INTRODUÇÃO

O primeiro computador eletrônico foi desenvolvido entre os anos de 1943 e 1946, levando o nome de *Electronic Numerical Integrator and Computer* (ENIAC). Neste período, as pesquisas computacionais, em especial nos Estados Unidos, eram realizadas para fins militares, e com o ENIAC não foi diferente, devido ao fato de que foi o exército estadunidense que encomendou a criação do computador, montado pela Universidade da Pensilvânia, com o objetivo de calcular trajetórias balísticas de mísseis. No decorrer da história, os computadores tornaram-se cada vez mais presentes em empresas, laboratório e, por fim, nos lares.

Atualmente os computadores fazem parte de diversos períodos da rotina do ser humano, desde trabalhos, estudos e entretenimento. Nas empresas, as máquinas auxiliam no gerenciamento de negócios, organização de projetos e controle de acesso. Computadores são essenciais também dentro de governos, na parte de segurança, disponibilização de informações à população e cadastro de beneficiários de programas sociais.

A Fundação Getúlio Vargas (2020) afirma, em uma pesquisa realizada, que “o volume de venda anual de computadores em 2019 foi de 12 milhões (-3%), essa é a mesma quantidade de 2016 e 2017, metade do pico de 2013”, somente no Brasil. O aumento no número de pessoas que se encontram em *Home Office* também contribuiu no aumento da quantidade de computadores presentes nas casas.

Uma vez que a tecnologia e sua utilização se tornaram cada vez mais presentes no dia a dia, atividades que até então eram simples – como escrever uma redação ou ler um livro – também foram se integrando à tecnologia, o mundo passou a escrever com editores de texto e ler por meio de *eBooks*. É nesse contexto que está inserido o *Optical Character Recognizer* (OCR), em tradução literal, “Reconhecimento Óptico de Caracteres”.

O OCR possui duas origens diferentes, como explica Folter (2022), no ano de 1974, Ray Kurzweil - pioneiro no estudo dessa tecnologia - desenvolveu um produto que processava e reconhecia qualquer fonte impressa. O objetivo, inicialmente, era a utilização desse produto para a acessibilidade de pessoas com deficiência visual, tendo em vista que a máquina desenvolvida analisava a fonte e vocalizava o que estava escrito. Por outro lado, também é considerado como início da tecnologia de

reconhecimento óptico a década de 1950, tempo que a computação ainda era voltada para a área militar, como explica

Folter (2022) o OCR começou no "início da automação de dados em um processo que começou com a antiga Agência de Segurança de Forças Armadas (AFSA), atual agência de Segurança Nacional (NSA), nos Estados Unidos".

Tinha por objetivo o envio de informações de documentos sem a necessidade da parte humana, ou seja, é realizada uma análise da informação pelo computador do que estava escrito e enviar os dados a outros lugares. Atualmente, o conceito da funcionalidade da ferramenta não foge dos seus objetivos primordiais.

Basicamente, o OCR funciona como um escrivão, supondo que um escriba esteja reescrevendo um documento é necessário a leitura, extração da informação e escrita. Ou partindo para a área da tecnologia, em um cenário onde um funcionário de uma empresa necessita cadastrar um novo colaborador no sistema, é realizado a leitura das informações na ficha de cadastro e as digitar no sistema da empresa, segundo a Microsoft (2022) o Reconhecimento Óptico tem dentre seus diversos benefícios a redução dos processos manuais de entrada de dados.

O OCR, em poucas palavras, realiza a mesma ação. Entretanto, como qualquer ferramenta de tecnologia, faz o procedimento com mais acuracidade, visto que, qualquer trabalho humano está sujeito a erro, bem como a digitalização de textos para torná-los editáveis com ampla utilização, como explica Sophia (2021). Klippa (2022) suscita que "A tecnologia melhora a qualidade de um texto digitalizado ou de uma imagem e segue vários passos para extrair dados que foram capturados. A diferença é que o trabalho manual leva mais tempo e é mais propenso a erros humanos". O site demonstra também que há quatro passos que toda interface que disponibiliza a tecnologia deve seguir: pré-processar a imagem, segmentá-la, reconhecer os caracteres e realizar o pós-processamento. Por interface, entenda-se que o OCR não é uma tecnologia monopolizada por um único fornecedor, visto que há diversas empresas que disponibilizam a ferramenta por meio de *libraries* (libs) – "bibliotecas" em tradução literal – de APIs.

Uma *Application Programming Interface* (API), em português: Interface de Programação de Aplicativos; são, de acordo com o site Hashtag (2021), conectores de sistemas, softwares e aplicativos, possibilitando a entrega de uma experiência familiar na utilização de aplicações. É a permissão de que o usuário de um aplicativo consulte, faça alterações e armazene dados de inúmeros sistemas sem a necessidade

do acesso direto, suscinta Costa (2022). Esses softwares podem ter diversas utilizações, como uma lanchonete que precisa emitir boleto e utiliza uma API de um determinado banco para tal, note que, a lanchonete não precisa contratar uma empresa que desenvolva um sistema completo de geração de boletos.

O presente trabalho tem por objetivo a análise de ambientes que disponibilizam bibliotecas de APIs de OCR. Dentre elas estão: Microsoft, Amazon, Google, Clarifai,

SentiSight e Api4AI. Todas as bibliotecas estão dispostas no site Eden AI – que organiza, centraliza e disponibiliza as APIs para teste da função OCR, ainda que cada possua seu próprio ambiente.

Salienta-se que o objetivo da pesquisa realizada é analisar e comparar o grau de acerto dos caracteres analisados pelas APIs, disponibilizando em todas diversas categorias de imagens para análise.

Tal feito é importante para auxiliar futuros desenvolvedores que desejem aplicar as tecnologias utilizadas em seus sistemas, aplicando a de melhor desempenho.

O presente trabalho está dividido em cinco capítulos. No primeiro é introduzido o trabalho e o contexto. No segundo são descritos os conceitos de OCR, os objetos que foram analisados, os ambientes e métricas utilizados. No terceiro capítulo é apresentada a metodologia adotada para a realização dos testes. O quarto é exclusivo para detalhar o levantamento dos dados e o resultado dos testes aplicados e o quinto capítulo trata das considerações finais e possíveis melhorias para as ferramentas analisadas.

2 CONCEITOS BÁSICOS

A tecnologia está em constante evolução e, junto a ela, toda e qualquer ferramenta está sujeita a manutenção. Deve-se ao fato de que, algo desenvolvido hoje, amanhã já estará desatualizado e novas tecnologias cada vez mais sofisticadas substituem outras já obsoletas ou atrasadas.

O site Klippa (2022) conceitua que com o OCR, não é diferente, ainda que o conceito seja o mesmo desde sua concepção, empresas que se dispõem a trabalhar com a tecnologia possuem campos de pesquisa que sempre tem o objetivo de refinar a precisão do Reconhecimento Óptico, o que antes era um sistema que reconhecia apenas fontes impressas, atualmente possuem sistemas que conseguem ler placas de automóveis em alta velocidade. É neste contexto de análise que existe o Eden AI, uma ferramenta centralizadora de algumas APIs de OCR, disponibilizando um centro de pesquisa da mesma tecnologia aplicada e desenvolvida por diferentes instituições.

2.1 Definição de conceitos

Neste tópico é apresentado as características que englobam o Reconhecimento Óptico de Caracteres, sua história, evolução e utilização. Tal qual, é descrito, igualmente, para as Interfaces de Programação de Aplicativos.

2.1.1 Optical Character Recognizer

Os anos da década de 1950 marcaram o início oficial da tecnologia OCR, por meio de pesquisas realizadas pela antiga Agência de Segurança de Forças Armadas (AFSA) dos Estados Unidos. Eles tinham em mente, na época, uma ideia de enviar documentos e informações de um local a outro sem a necessidade do trabalho manual – ou humano – por meio da leitura dessas informações em um computador. Folter (2022) explica que “a ideia era fazer com que informações de documentos pudessem chegar a outros locais sem ter que passar por mãos humanas, ou seja, um computador poderia analisar o que estaria ali e, automaticamente, entender e passar os dados para outros lugares”.

Enquanto os militares estadunidenses pensaram nessa ideia e em seu desenvolvimento, ressalta Folter (2022), foi um estudioso, pioneiro na área, Ray

Kurzweil – nascido em 12 de fevereiro de 1948 em Nova Iorque – no ano de 1974 inventou a primeira máquina que realizava o reconhecimento de qualquer fonte impressa. A ideia de Kurzweil, inicialmente, era que o produto desenvolvido auxiliasse na acessibilidade de pessoas com deficiência visual, pois seu sistema reconheceria as fontes, analisaria e vocalizaria o que estava escrito.

Na década de 80, o pioneiro na tecnologia OCR vendeu a empresa para a Xerox – empresa sinônimo de fotocópia – que ao saberem da inovação de Kurzweil, demonstravam imenso interesse, após a venda da empresa, uma revolução teve início e o OCR é o que tornou possível a existência de fotocopiadoras, digitalizadores e muitas outras ferramentas que utilizam a tecnologia, demonstra Folter (2022), numa clara demonstração que Kurzweil é um personagem importante na história da digitalização.

O funcionamento da tecnologia é simples, conforme Folter (2022) deixa claro, é uma tecnologia cuja é operada por meio de um software, que tem por objetivo a extração do conteúdo de uma imagem, entendendo-a, processando-a e por fim, transformando-a em um texto escrito. O site Klippa (2022) explica que o funcionamento do OCR é como a capacidade humana de ler um texto, porém, diferentemente do trabalho visual do ser humano, o Reconhecimento Óptico além de ser mais rápido na leitura, ainda melhora a qualidade do texto digitalizado e está sujeito a menos erros humanos.

A tecnologia possui quatro passos, de acordo com o Klippa (2022), a citar: pré-processamento da imagem, segmentação, reconhecimento dos caracteres e o pós-processamento.

1. Pré-processamento: melhoria da qualidade da imagem disponibilizada para o reconhecimento, para que a saída dos dados seja mais precisa, é necessária uma imagem clara e de boa qualidade. Nessa primeira etapa é onde o motor do OCR localiza automaticamente erros e os corrige, incluindo as seguintes tecnologias:

- **De-skew:** processo onde é trabalhado o ângulo da fotografia.

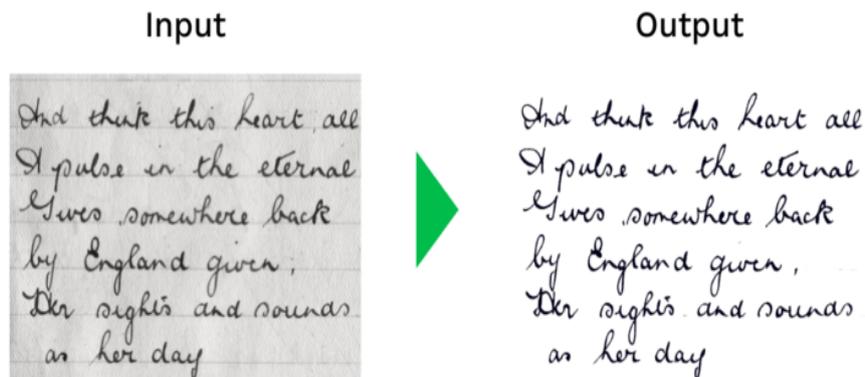
Figura 2.1 – De-skew



Fonte: Klippa, 2022.

- **Binarization:** a imagem ou documento que é digitalizado é convertido para preto e branco, permitindo a separação precisa do texto do fundo da imagem.

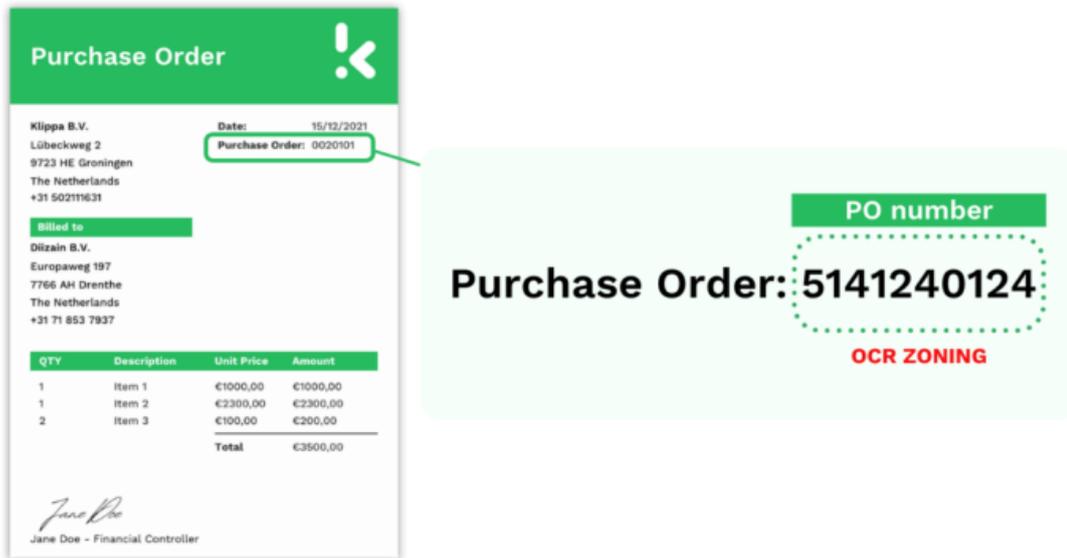
Figura 2.2 – Binarization



Fonte: Klippa, 2022.

- **Zoning:** é feita uma análise do layout da imagem, identificando colunas, linhas, blocos, legendas, parágrafos e tabelas.

Figura 2.3 – Zoning



Fonte: Klippa, 2022.

- **Normalização:** é reduzido o ruído da imagem por meio de ajustes no valor da intensidade dos pixels.

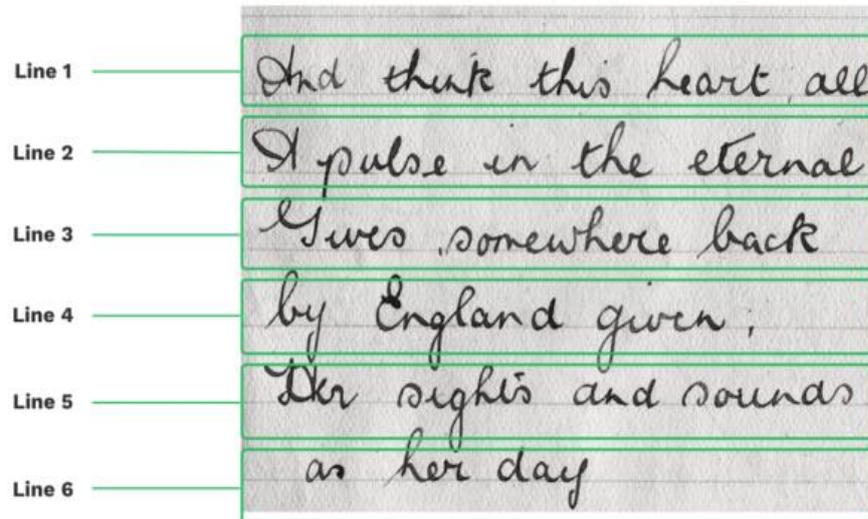
Figura 2.4 – Normalização



Fonte: Klippa, 2022.

2. **Segmentação:** realiza o reconhecimento de uma linha de texto de cada vez. Possui os seguintes processos:
 - **Deteção de palavras e linhas de texto:** identifica as linhas e as palavras pertencentes a elas.

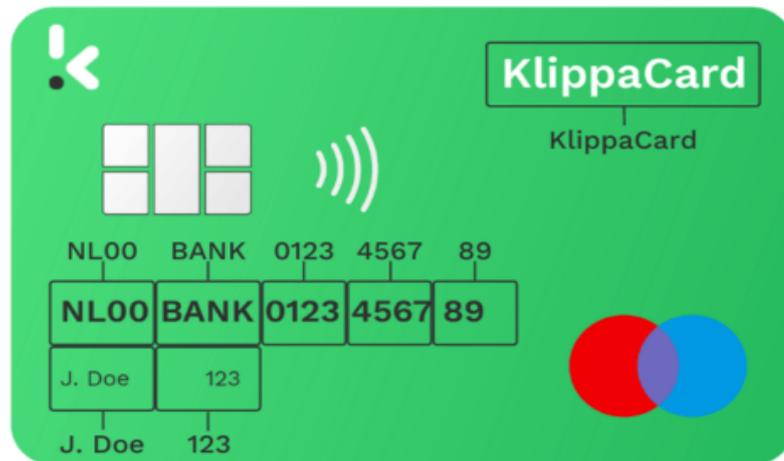
Figura 2.5 – Detecção de palavras e linhas de texto



Fonte: Klippa, 2022.

- **Reconhecimento do guião:** baseia-se em documentos, páginas, linhas, parágrafos, palavras e caracteres.

Figura 2.6 – Reconhecimento do guião

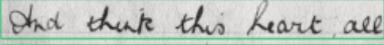
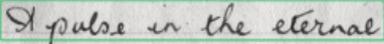
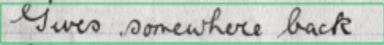
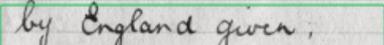
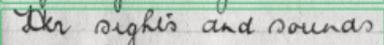
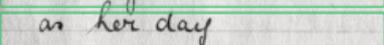


Fonte: Klippa, 2022.

- 3. Reconhecimento do carácter:** o terceiro passo divide a imagem em partes, secções ou zonas, permitindo o reconhecimento dos caracteres. Tendo os seguintes passos:

- **Correspondência de matrizes:** é feita uma comparação pixel por pixel para reconhecer os caracteres de uma imagem, comparando-as com uma biblioteca de matrizes de caracteres.
 - **Reconhecimento de características:** reconhece tamanho, forma, linhas e estruturas.
4. **Pós-processamento:** a quarta e última etapa é responsável pela utilização de técnicas e algoritmos para melhorias de precisão da extração de dados, a fim de obter um bom resultado. Também realizando correções, dependendo da tecnologia disponibilizada.

Figura 2.7 – Pós-processamento

	And thik this heart all	And think this heart all
	A pulse in the elernal	A pulse in the eternal
	Yives somewhere back	Gives somewhere back
	by England given,	by England given,
	Her sighis and sounds	Her sights and sounds
	on her day	on her day

Fonte: Klippa, 2022.

2.1.2 Application Programming Interface

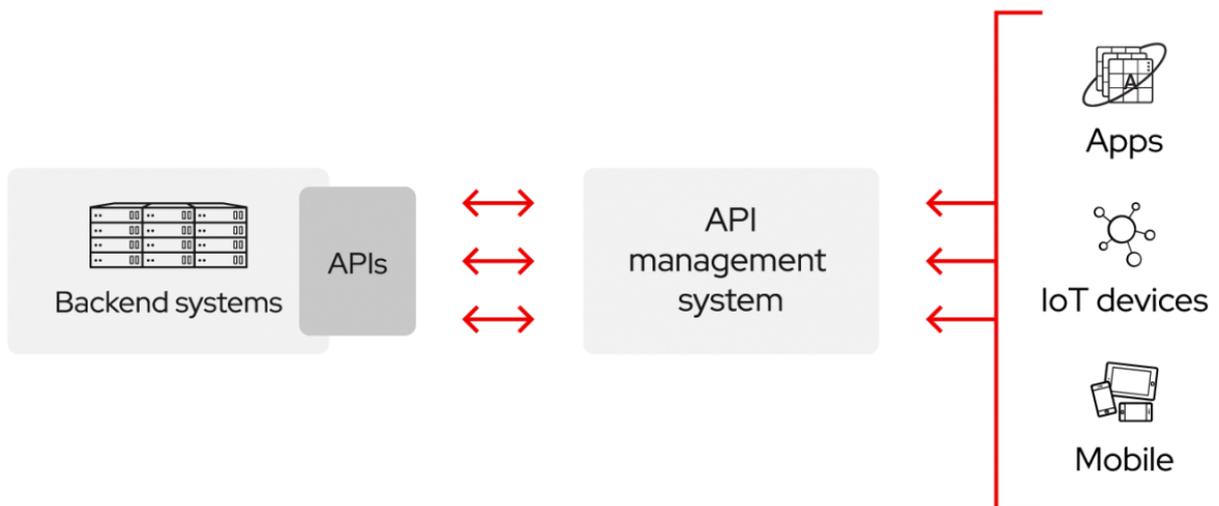
As *Application Programming Interface* (API), como argumenta o site Redhat (2022), estão presentes no mundo tecnológico desde os primeiros dias da computação, remontando tempos em que era impensável a utilização de computadores para uso pessoal. No passado, as APIs eram utilizadas como bibliotecas para Sistemas Operacionais, porém, após os anos 2000, estavam tornando-se cada vez mais presentes e importantes em integrações remotas de dados.

Essa tecnologia é uma solução cuja permite a comunicação com outros produtos sem a necessidade do conhecimento de sua implementação, processo que simplifica o desenvolvimento de sistemas, sendo econômico e diminuindo o tempo de

trabalho. São uma espécie de contrato, com acordo entre as partes interessadas e existem para auxiliar empresas e suas equipes de Tecnologia da Informação. Pois, com um mundo cada vez mais veloz tecnologicamente, as necessidades das empresas tendem a mudar rapidamente. “As APIs simplificam a forma como os desenvolvedores integram novos componentes de aplicações a uma arquitetura preexistente”. (Redhat, 2022)

É, neste contexto, que as Interfaces são aplicadas, como explica o site Redhat (2017) que as APIs “são uma maneira simplificada de conectar a própria infraestrutura por meio do desenvolvimento de aplicações nativas em nuvem”. Possibilitam também, o compartilhamento de dados com clientes e outros usuários externos, agregando bastante valor de negócios devido a sua simplicidade.

Figura 2.8 – Funcionamento de uma API



Fonte: Redhat, 2017.

Neste estudo, foram utilizadas APIs de algumas empresas que possuem a tecnologia OCR, em um ambiente centralizador que disponibiliza essas bibliotecas para serem testadas. Como explicado, tal feito diminui o tempo que uma determinada empresa levaria pesquisando formas de implementar a tecnologia, podendo comparar qual das bibliotecas é mais precisa no reconhecimento de textos.

2.2 Objetos analisados

As principais ferramentas utilizadas na pesquisa são: Eden AI, Visual Studio Code, Linguagem Python e o Microsoft Excel.

O Eden AI¹ é um *website* com a proposta de centralizar e disponibilizar determinadas tecnologias de Inteligência Artificial, dentre as quais estão presentes ferramentas de OCR. Neste trabalho o site foi utilizado para testar as bibliotecas de API's disponíveis em seu ambiente, a citar: Google, Microsoft, Amazon, Clarifai, Sentsight e API4AI.

O visual Studio Code, ou VS Code, é um editor de código aberto desenvolvido pela Microsoft, é disponibilizado para o Windows, Mac e o Linux, conhecido por ser uma ferramenta poderosa devido a sua loja de extensões que permite a instalação de diversas tecnologias para desenvolvimento de sistemas, tendo consigo também um imenso suporte a diversas Linguagens de Programação. Na presente pesquisa, o VS Code é utilizado para o desenvolvimento de um programa que calcula a precisão das bibliotecas de OCR.

O Python é uma Linguagem de Programação com o principal objetivo de ser de simples entendimento e fornecendo uma facilidade de aprendizado, podendo ser empregada em diversas tarefas. Atualmente, é a linguagem mais utilizada no mundo e no Brasil. No estudo, foi utilizada para o desenvolvimento do programa de cálculo de precisão das *libs*.

O MS Excel foi utilizado, neste estudo, apenas para realização do cálculo da média e de gráficos dos resultados obtidos.

2.2.1 Eden AI

A Redação (2022) expõe que o Eden AI é uma empresa startup francesa. Tem como missão a democratização de tecnologias de *Artificial Intelligence* (AI), tal feito é realizado por meio da construção de uma interface de programação que permite o acesso de ferramentas AI de diversos fornecedores. A função da companhia é unificar APIs dentro de seu ambiente para que os usuários possam experimentá-las, misturá-las e combiná-las.

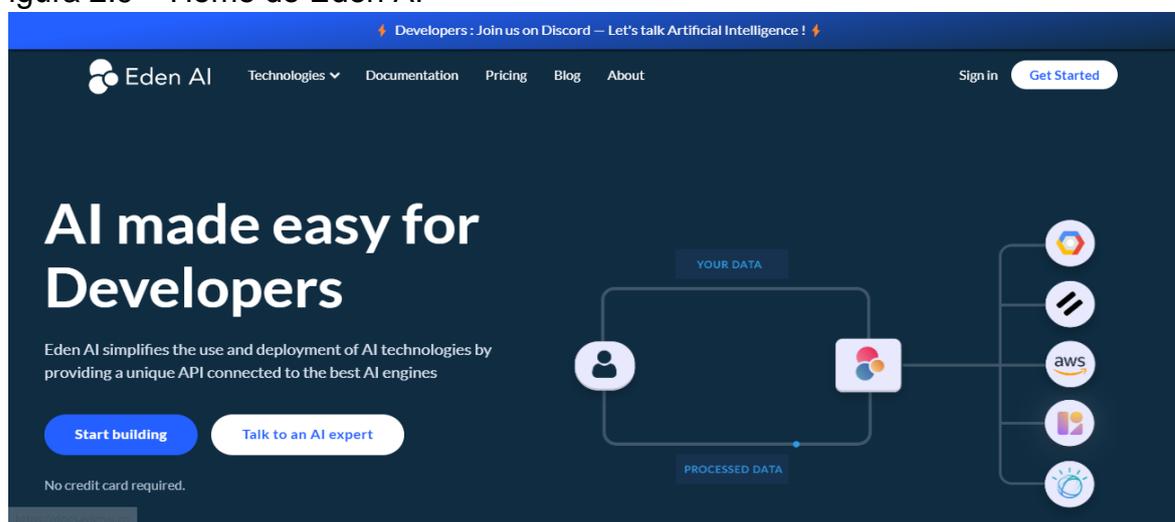
Diversos fornecedores - de pequeno à grande porte - construíram APIs em nuvem permitindo o processamento de dados na infraestrutura utilizando modelos de aprendizagem de máquinas, o OCR é um exemplo dessas bibliotecas. Contudo, cada

1 Eden AI: <https://www.edenai.co>

biblioteca possui sua característica individual, com cada qual sendo mais eficiente que as demais, diferenciando-se idiomas, meios de implementação e utilização, conforme esclarece a Redação (2022)

A utilização das tecnologias fornecidas pelo Eden não é gratuita, tendo preços diferentes para cada tecnologia, mas no começo, quando é criado um primeiro cadastro, é liberado \$10,00 (dez dólares) para serem utilizados e, os preços, são acessíveis, tendo em vista o objetivo da empresa de democratizar o acesso as tecnologias, evidencia a missão descrita no site do Eden AI (2022).

Figura 2.9 – Home do Eden AI



Fonte: Eden AI, 2022.

2.2.1.1 APIs do Eden AI

Dentre as diversas tecnologias fornecidas pelo Eden AI, as bibliotecas de APIs de OCR estão dentre elas. A empresa possui contrato com fornecedores diversificados, o que garante uma gama maior de experimentação dentro do ambiente.

Dentre as *libs* disponíveis, encontram-se: o Google Vision, pertencente a empresa multinacional de mesmo nome da qual é uma das fornecedoras de API em nuvem para utilização do Reconhecimento de Caracteres, além da biblioteca, a tecnologia já é empregada pela empresa em um aplicativo, Google Docs. O AWS da Amazon, outra multinacional, também fornece uma biblioteca em nuvem para teste, possuindo também seu ambiente OCR denominado Amazon Textract. O Microsoft Azure, da Microsoft, que possui seu próprio ambiente e uma API em nuvem para testes presente no Eden. Possuindo suporte para diversos idiomas. O SentiSight é um

sistema desenvolvido pela Neurotechnology UAB, com foco no reconhecimento de imagens, não somente caracteres, seu aplicativo, disponível para sistemas móveis, reconhece características de imagens, como por exemplo: se uma foto possuir um fungo, o sistema retorna as possibilidades de ser de fato um fungo ou uma planta. No Eden, é disponibilizado a sua biblioteca em nuvem de API de reconhecimento de caracteres. O Clarifai, uma empresa focada em Inteligência Artificial, em específico no reconhecimento de imagens e vídeos, com soluções via APIs, SDK móvel e local. Presente no site por meio da API em nuvem. A Api4AI possui a proposta de reconhecer textos em imagens, sendo uma das mais diferenciadas dada a sua forma de organizar o reconhecimento do guião das imagens.

A funcionalidade das aplicações no presente estudo são semelhantes, o usuário realiza um upload da imagem que deseja obter o resultado e cada biblioteca realiza a análise com base em suas próprias configurações. Dessa forma, a diferença está no response da solicitação em formato JSON, como cada tem configurações únicas vindas de seus fornecedores, todas se diferenciam.

2.2.2 Visual studio code

O Visual Studio Code é um editor de código aberto criado pela Microsoft. Possui diversas funcionalidades, a citar:

- Edição de código, suportando uma variedade de linguagens de programação;
- Terminal;
- Controle de versão.

O VS Code é considerado extremamente leve, sem onerar o ambiente do desenvolvedor e possui uma gama poderosa de funcionalidades, ainda que simples, seu diferencial é a loja de extensões, que permite o enriquecimento da ferramenta.

A loja permite que variados fornecedores possam disponibilizar tecnologias novas que vão desde edição visual do editor até ferramentas de desenvolvimento.

De acordo com Hanashiro (2021) o VS Code é tão poderoso que pode ser comparado com vários outras IDEs não gratuitas.

Está disponível para os principais Sistemas Operacionais do mercado, sendo eles: Windows, Mac e Linux. É desenvolvido com Electron, criada pelo GitHub, a ferramenta permite o desenvolvimento de *softwares Desktop* com HTML, CSS e JavasCript.

Com o VS Code é permitido:

- Desenvolvimento de sistemas para a Web;
- Criação de aplicativos móveis;
- Desenvolver sistemas locais para desktop;

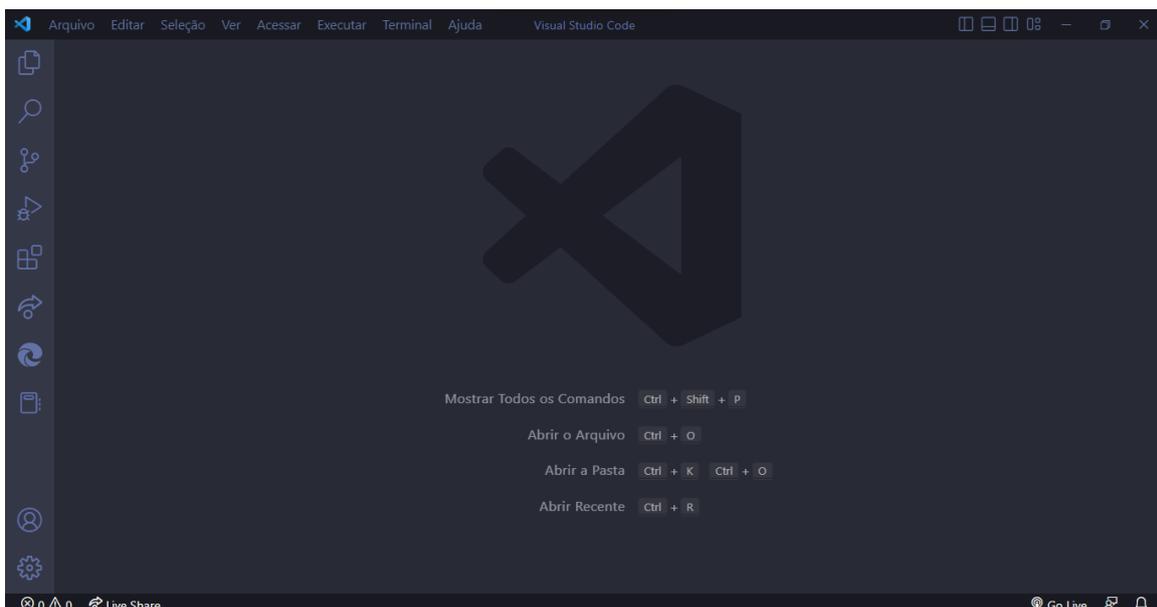
Possuí suporte para as principais linguagens de programação e de marcação do mercado, como o PHP, JavaScript, Java, HTML, CSS, TypeScript, Python, Ruby, C#, C++ e SQL.

Sua instalação é simples e seus pré-requisitos fáceis de serem cumpridos, não necessitando de uma máquina muito potente para realizar suas funções. É necessário para instalar o Visual Studio Code:

- Processador 1.6GHz e com 1GB de RAM;
- Microsoft.NET Framework para sistemas Windows;
- No sistema Linux é necessário ter GLIBCXX e BLIC.

Com os pré-requisitos de acordo, então é necessário baixar o instalador do VS Code no site oficial e seguir o passo-a-passo para sua instalação.

Figura 2.10 – Página principal do VS Code



Fonte: elaborada pelos autores, 2022.

2.2.3 Python

O Python é, segundo o site Hashtag (2022), uma Linguagem de Programação (LP) com o objetivo de fornecer simplicidade e fácil aprendizagem devido a sua versatilidade, sendo possível utilizar a Linguagem para uma variedade de tarefas que vão desde desenvolvimento de aplicativos móveis, websites, jogos, análise de dados e inteligência artificial.

A linguagem é bastante indicada aos iniciantes da área de programação, como explica o Hashtag (2022), sendo uma porta de entrada ao mundo da tecnologia e do desenvolvimento de sistemas. O Python é uma das LPs mais utilizadas no mundo e no Brasil, em um crescimento constante, estando disponível para Windows, Linux e macOS, sofrendo alterações ao variar o Sistema Operacional (SO).

As principais características do Python são:

- Multiplataforma: suporte em diversos SOs;
- Multiparadigma: abrange a diversos estilos de programação - orientada a objetos, programação funcional ou procedural;
- Linguagem interpretada: não possui a geração de arquivos executáveis. Há a necessidade de um interpretador que traduz o código fonte em linguagem de máquina, ou byte code, que envia ao ambiente *Python Virtual Machine* (PVM) responsável pela execução do programa.

O sistema é valorizado e está em constante crescimento devido a suas principais vantagens, tais quais: a gratuidade, portabilidade, multiplataformidade e extensibilidade e devido a sua simplicidade. Sua instalação no VSCode é feita pela loja de extensões do editor e sua extensão de arquivo é “.py”.

3 METODOLOGIA

Neste capítulo são descritos critérios e planejamento da análise da precisão do Reconhecimento de Caracteres, a fim de chegar a um resultado da melhor eficácia.

Com o auxílio do ambiente Eden é possível observar os resultados que serão mensurados pelo programa desenvolvido em Python e totalizados no Excel.

3.1 Definição de critérios e métodos de análise

O teste de Reconhecimento foi realizado utilizando o ambiente web do Eden AI com imagens de documentos de categoria empresarial e de escritório – para centralizar a análise em uma categoria – e com um programa desenvolvido em Python para medir a precisão das ferramentas.

As ferramentas utilizadas no trabalho são Eden AI, programa Python no VS Code e o Excel.

O primeiro passo é selecionar as imagens para teste no ambiente Eden. Após a seleção é feito o *upload* da imagem no site, com o idioma desejado – no presente estudo, português e inglês - para informar as bibliotecas. Após a análise de todas as imagens, o programa desenvolvido em Python irá comparar o texto original das imagens com o texto resultante da análise pelas *libs*.

Por fim, após a coleta de todas as comparações, o Excel irá centralizar os resultados por meio do cálculo da média após o cálculo da acuracidade, totalizando-os em um gráfico e definindo qual biblioteca obteve mais sucesso na aplicação do OCR. A pesquisa é realizada para definir a melhor API em nuvem para o reconhecimento óptico, a fim de apontar qual necessita de melhorias e qual é sugerido para a aplicação em sistemas a serem desenvolvidos tendo o estudo como base.

3.2 Definições de ambiente

Os testes foram realizados a partir de um Notebook Acer com processador Intel(R) Core(TM) i5-8265U CPU @ 1.60GHz 1.80GHz, com RAM de 8GB com o Windows 11 instalado. Nele está instalado o navegador Opera GX que foi utilizado para acessar o site Eden AI.

Também foi instalado, na máquina, o VS Code versão 1.73.1 para o desenvolvimento do programa de métricas em Python - esse instalado pela loja de extensões do Visual Studio. O Excel, utilizado para centralizar os resultados está na versão 2210 64 bits. O teste foi realizado com os dispositivos ligados à internet por rede Wi-Fi.

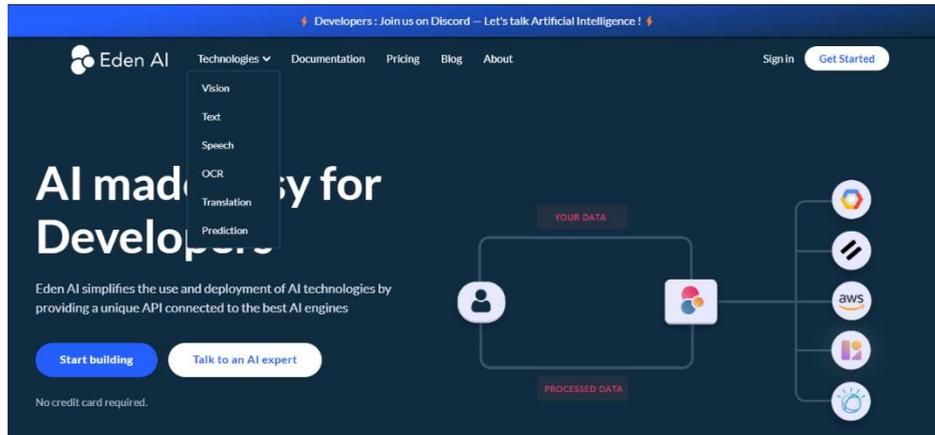
3.3 Planejamento e execução

O Notebook Acer está com o sistema Operacional e os *softwares* – VSCode e Excel – completamente atualizados em suas últimas versões. O navegador Opera GX também se encontra atualizado.

Os passos para o estudo estão definidos em: reconhecer as imagens no Eden AI, comparar texto base com o resultado no programa desenvolvido em Python e totalizar e centralizar resultados no Excel.

O site do Eden AI está em sua última versão, acessado por meio de seu endereço: <https://www.edenai.co>. Dentro do ambiente é necessário acessar a aba de tecnologias disponíveis e selecionar o OCR, como demonstrado na figura 3.1 a seguir:

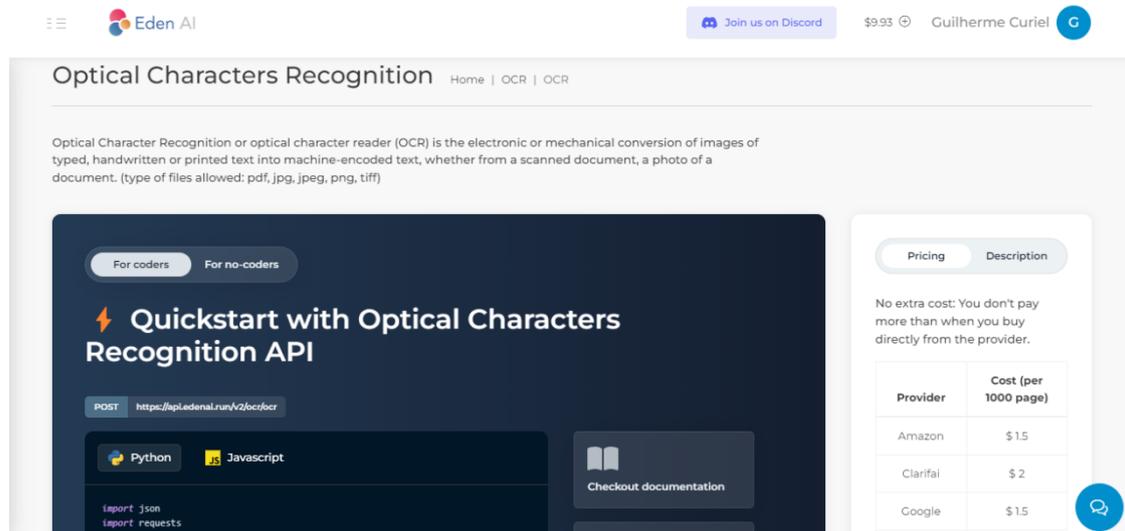
Figura 3.1 – Aba OCR no Eden



Fonte: Eden AI, 2022.

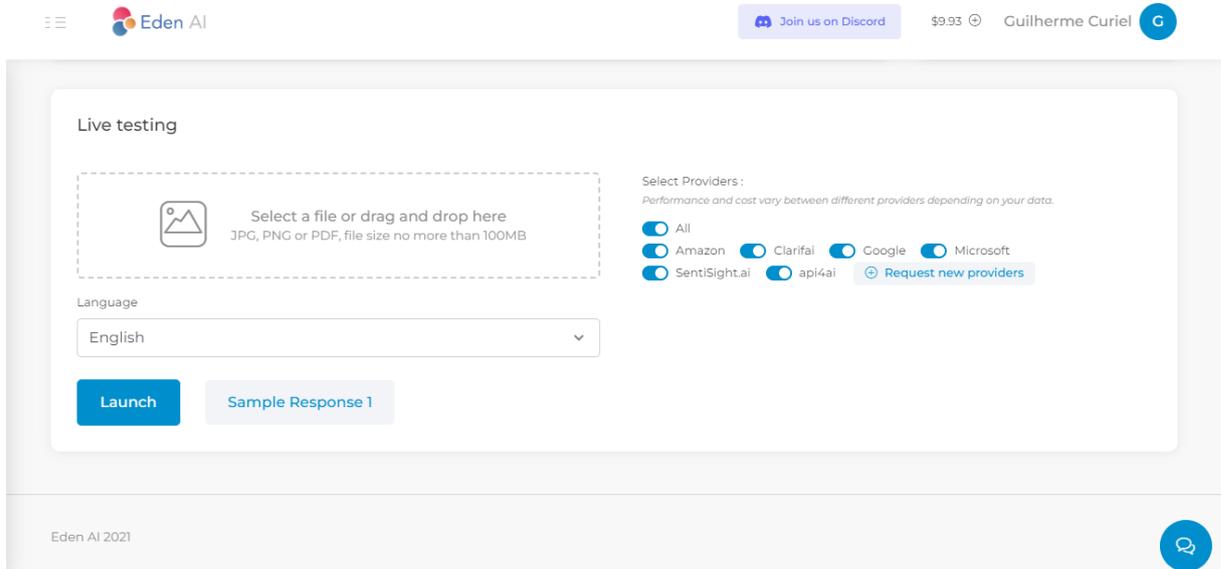
Uma vez na página de Reconhecimento Óptico, é apresentado brevemente uma explicação do que se trata o OCR. Também é demonstrado o preço que cada biblioteca possui e um rápido tutorial para ensinar como trabalhar com as *libs*. A figura 3.2 apresenta o início da página e a figura 3.3 a parte de análise.

Figura 3.2 – Página do OCR



Fonte: Eden AI, 2022.

Figura 3.3 – Página de teste do OCR



Fonte: Eden AI, 2022.

Para realização do teste, é necessário realizar o upload – neste capítulo de exemplo foi utilizado uma foto no formato JPEG contendo o texto: <Hello World/> - de uma foto para a análise das APIs, logo em seguida, é retornado um response contendo os resultados.

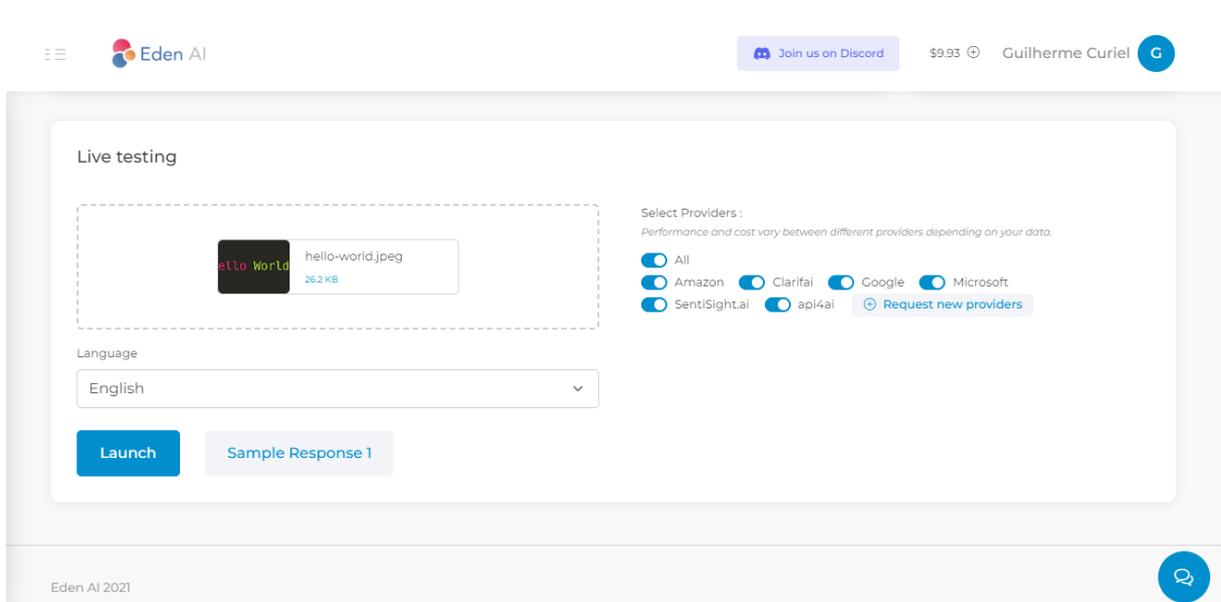
de cada biblioteca, como demonstra a figura 3.5 o upload da imagem para análise e a figura 3.6 o resultado retornado.

Figura 3.4 – imagem de exemplo



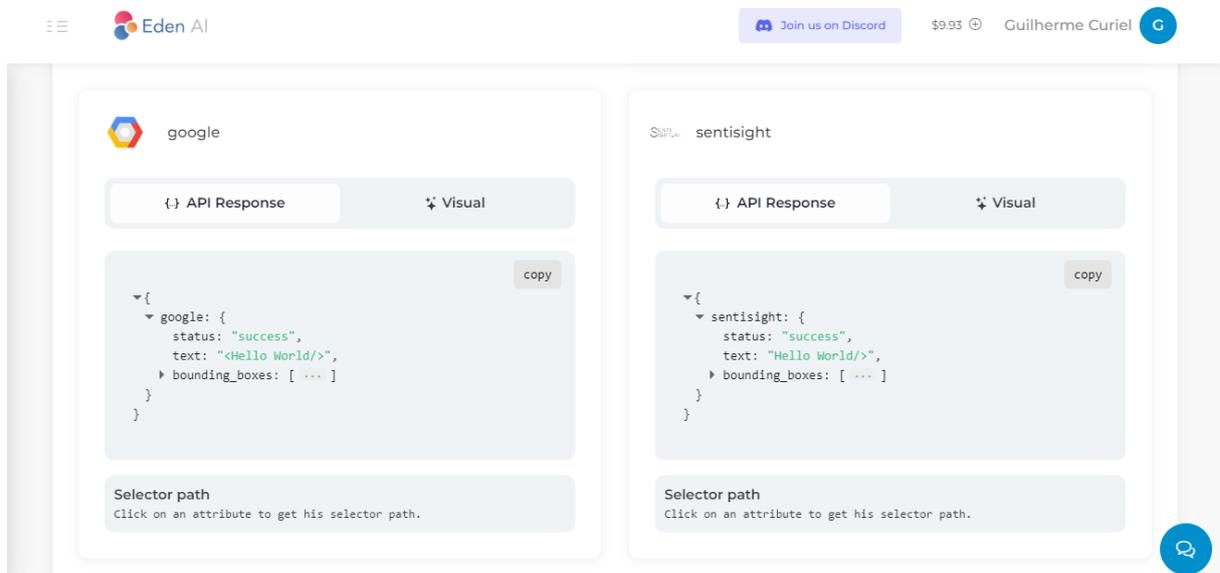
Fonte: martinsjackson, 2022.

Figura 3.5 – Upload da imagem para análise



Fonte: Eden AI, 2022.

Figura 3.6 – Resultados Google Vision e SentiSight



Fonte: Eden AI, 2022.

Observando ambos os testes preliminares ambas as bibliotecas retornaram com resultados divergentes, dado ao comportamento único de cada API ao analisar a imagem fornecida. Enquanto o Google trabalhou perfeitamente o texto presente na imagem o SentiSight obteve um retorno aquém do esperado.

Cada biblioteca retorna um extenso response após a análise do texto, retornando o *status* da chamada, *“success”* ou *“failed”*, o texto retornado e as

coordenadas de cada palavra. O quadro 3.1 demonstra o response do Google, enquanto o quadro 3.2 o response do Clarifai.

Quadro 3.1 – Response da API do Google

```
{
  "google": {
    "status": "success",
    "text": "<Hello World/>",
    "bounding_boxes": [
      {
        "text": "<",
        "left": 0.24625,
        "top": 0.38228941684665224,
        "width": 0.03125,
        "height": 0.17278617710583152
      },
      {
        "text": "Hello",
        "left": 0.28375,
        "top": 0.38228941684665224,
        "width": 0.174375,
        "height": 0.17926565874730022
      },
      {
        "text": "World",
        "left": 0.495625,
        "top": 0.38876889848812096,
        "width": 0.178125,
        "height": 0.18142548596112312
      },
      {
        "text": "/",
        "left": 0.68,
        "top": 0.3952483801295896,
        "width": 0.03375,
        "height": 0.17494600431965443
      }
    ]
  }
}
```

```

    },
    {
      "text": ">",
      "left": 0.7125,
      "top": 0.39740820734341253,
      "width": 0.035,
      "height": 0.17494600431965443
    }
  ]
}}

```

Fonte: Eden AI, 2022.

Quadro 3.2 – Response da API Clarifai

```

{
  "clarifai": {
    "status": "success",
    "text": "World/>Hello",
    "bounding_boxes": [{
      "text": "World/>",
      "left": 0.4893749952316284,
      "top": 0.35205182433128357,
      "width": 0.2681249976158142,
      "height": 0.2483801543712616
    },
    {
      "text": "Hello",
      "left": 0.27562499046325684,
      "top": 0.35853132605552673,
      "width": 0.19249999523162842,
      "height": 0.2332613170146942
    }
  ]
}}

```

Fonte: Eden AI, 2022.

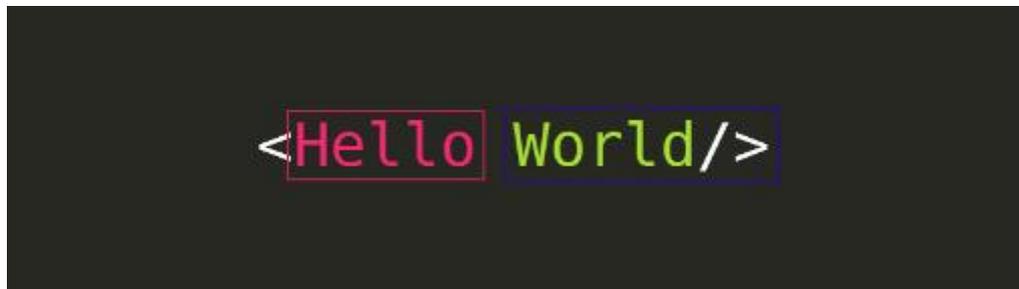
O Eden AI também retorna o resultado obtido por meio da divisão do texto em *boundaries* – ou reconhecimento do guião – ilustrando a performance da biblioteca. No trabalho o guião é utilizado como forma de desempate. As figuras 3.7 e 3.8 ilustram o retorno das imagens analisadas com o reconhecimento do guião.

Figura 3.7 – reconhecimento do guião no Google



Fonte: desenvolvida pelos autores, 2022.

Figura 3.8 – reconhecimento do guião no Clarifai



Fonte: desenvolvida pelos autores, 2022.

A segunda fase é analisar cada texto base e o texto do OCR em um programa desenvolvido em Python para cálculo. Foi utilizado como métrica a divisão do número total de caracteres do texto original dividido pela quantidade de caracteres remanescentes do texto do OCR. O quadro 3.3 traz a estrutura do sistema de precisão, enquanto a tabela 3.1 detalha o código.

Quadro 3.3 – Estrutura do PrecisaoOCR.py

```

from difflib import SequenceMatcher

main_text = "<Hello World/>" //texto original
result_ocr = "Hello World" //texto OCR

sm = SequenceMatcher(None, result_ocr, main_text)
true_positive_char_num = 0
for tag, i1, i2, j1, j2 in sm.get_opcodes():
    if tag== 'equal':
        true_positive_char_num += (j2 - j1)
    else:
        pass

print(f'precisão = {true_positive_char_num/len(main_text) * 100:.2f}%')

```

Fonte: elaborado pelos autores, 2022.

Tabela 3.1 – Detalhes PrecisaoOCR.py

Função	Descrição
<code>from difflib import SequenceMatcher</code>	Realiza a importação da classe “SequenceMatcher” que irá comparar cada caractere das variáveis definidas para realizar o cálculo dos caracteres divergentes entre o texto original e o texto do OCR.
<code>main_text = "<Hello World/>"</code>	Define a variável contendo o texto original para o cálculo final;
<code>result_ocr = "Hello World"</code>	Define a variável contendo o texto analisado pelo OCR.
<code>sm = SequenceMatcher(None, result_ocr, main_text)</code>	A variável sm aplica a classe “SequenceMatcher” onde irá verificar as similaridades entre as strings declaradas nas variáveis anteriores.
<code>true_positive_char_num = 0</code>	É declarado essa variável para que seja realizado o cálculo final da precisão. É definido o valor zero para garantir que a variável não tenha

	nenhum valor que possa interferir no resultado.
<pre>for tag, i1, i2, j1, j2 in sm.get_opcodes():</pre>	É iniciado um laço de repetição para incluir as informações coletadas na linha anterior, onde tag é a operação que posteriormente o programa irá realizar enquanto i1 e i2 são os valores coletados na variável do texto original e j1 e j2 coletados da variável do texto OCR. Todos são inseridos em get_opcodes() que irá verificar as informações presentes em cada variável.
<pre> if tag== 'equal':</pre>	A condicional if dá início ao cálculo dos caracteres, visto que, ao ser definido o valor equal para a tag, o processo é verificar a quantidade de caracteres iguais para dar continuidade na operação ou, caso não houver caracteres divergentes, continuar com o cálculo da precisão.
<pre> true_positive_char_num += (j2 - j1)</pre>	Essa operação irá definir a quantidade de caracteres divergentes está presente nas variáveis analisadas e será alocada na variável true_positive_char_num.
<pre> else: pass</pre>	Se não houver nenhum caractere divergente, a condicional irá seguir seu curso para dar fim ao cálculo.
<pre> print(f'precisão = {true_positive_char_num/len(main_text) * 100:.2f}%')</pre>	Por fim, é realizado o print do cálculo da precisão, a função "f" permite que sejam definidas as variáveis dentro de uma string para operações, nota-se que, para o cálculo final é dividido o número de caracteres presentes na variável true_positive_char_num com a quantidade de caracteres (feita pela função len que irá obter a quantidade) da variável main_text, e enfim multiplicando-os por 100 para obter o valor final da porcentagem, deixando duas casas após a vírgula (2f).

Fonte: elaborada pelos autores, 2022.

A seguir é demonstrado o resultado do código, coletando o texto original da imagem de teste, "<Hello World/>", com dois resultados das tratativas acima, da API

Google e Clarifai. Na figura 3.9 é apresentado o resultado da biblioteca do Google e na figura 3.10 o resultado da precisão do Clarifai.

Figura 3.9 – Cálculo da precisão do Google

```

PrecisaoOCR.py > ...
1  from difflib import SequenceMatcher
2
3  main_text = "<Hello World/>"
4  result_ocr = "<Hello World/>"
5
6  sm = SequenceMatcher(None, result_ocr, main_text)
7  true_positive_char_num = 0
8  for tag, i1, i2, j1, j2 in sm.get_opcodes():
9      if tag == 'equal':
10         true_positive_char_num += (j2 - j1)
11     else:
12         pass
13
14  print(f'precisão = {true_positive_char_num / len(main_text) * 100:.2f} %')

```

Python 3.11.0 (main, Oct 24 2022, 18:26:48) [MSC v.1933 64 bit (AMD64)]
Type 'copyright', 'credits' or 'license' for more information
IPython 8.6.0 -- An enhanced Interactive Python. Type '?' for help.

✓ from difflib import SequenceMatcher ...
... precisão = 100.00%

Fonte: elaborada pelos autores, 2022.

Figura 3.10 – Cálculo da precisão do Clarifai

```

PrecisaoOCR.py > ...
1  from difflib import SequenceMatcher
2
3  main_text = "<Hello World/>"
4  result_ocr = "World/>Hello"
5
6  sm = SequenceMatcher(None, result_ocr, main_text)
7  true_positive_char_num = 0
8  for tag, i1, i2, j1, j2 in sm.get_opcodes():
9      if tag == 'equal':
10         true_positive_char_num += (j2 - j1)
11     else:
12         pass
13
14  print(f'precisão = {true_positive_char_num / len(main_text) * 100:.2f} %')

```

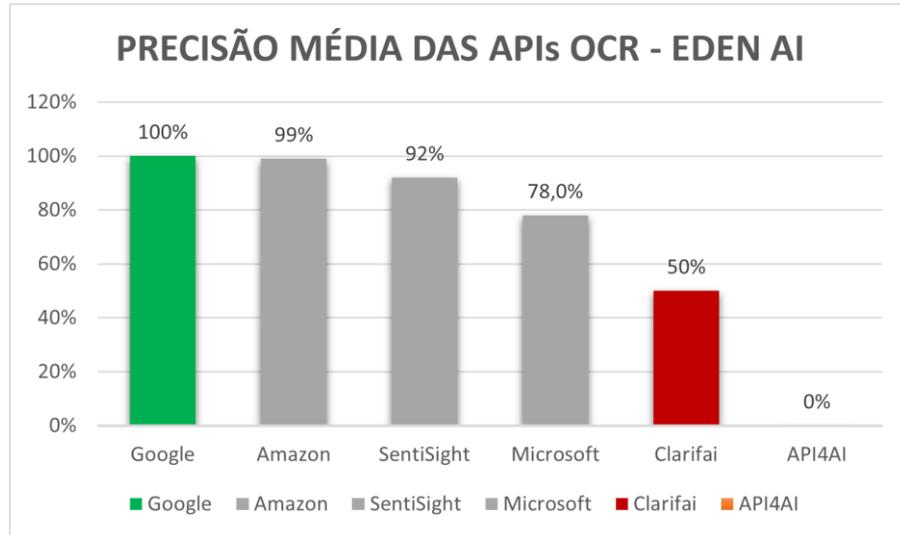
Python 3.11.0 (main, Oct 24 2022, 18:26:48) [MSC v.1933 64 bit (AMD64)]
Type 'copyright', 'credits' or 'license' for more information
IPython 8.6.0 -- An enhanced Interactive Python. Type '?' for help.

✓ from difflib import SequenceMatcher ...
... precisão = 50.00%

Fonte: elaborada pelos autores, 2022.

Por fim, a última etapa da análise é alocar todos os resultados em um documento Excel, para totalizar as análises, somá-las e por fim, obter a API mais eficaz. O gráfico 3.1 unifica os resultados obtidos utilizando a média aritmética dos resultados.

Gráfico 3.1 – Análise final no Excel



Fonte: elaborado pelos autores, 2022.

Vale ressaltar, que no instante da realização dos testes preliminares a API4AI estava fora de serviço, dessa forma, seus resultados não foram contabilizados, levando em consideração somente as APIs que retornaram a análise do texto.

4 LEVANTAMENTO DE DADOS E ANÁLISE DE RESULTADOS

Neste capítulo é demonstrado todo o processo da análise de fato, após documentar todo o processo que será realizado. Primeiramente, foram analisadas as imagens no ambiente do Eden AI, logo em seguida é colocado os textos das imagens originais e analisadas no `PrecisaoOCR.py` para calcular a efetividade da análise de cada API, após o cálculo é centralizado os resultados no Excel para montagem do gráfico da *lib* mais eficaz.

Deve-se observar que, em caso de empate, o parâmetro para desempatar a pontuação é verificar o reconhecimento do guião nos resultados em formato de imagem, trata-se de, se a API isolar cada caractere em um segmento, diferente é uma biblioteca que realiza um reconhecimento mais organizado, em contraste com as que isolam todo o texto em uma única *bounderie*.

Foram utilizadas no total, seis imagens para análise – três imagens iguais, porém três fotografias e três digitalizadas - totalizando trinta e seis testes.

4.1 Teste no Eden AI

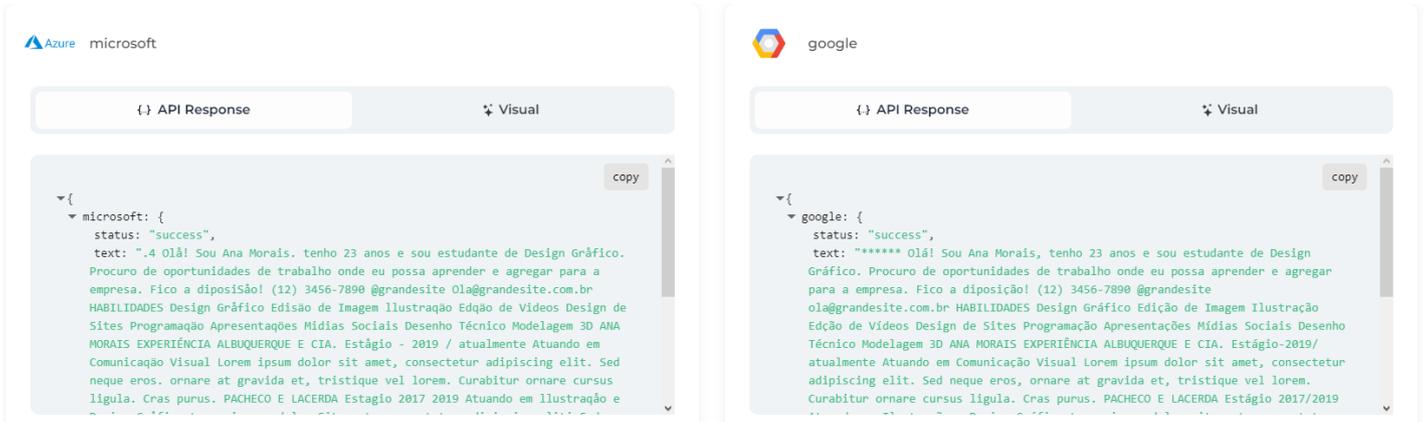
a) Imagem 1: Currículo

A primeira imagem utilizada é um currículo que possui uma foto 3x4 e linhas divisórias. A princípio, é uma imagem difícil para as bibliotecas analisarem devido aos elementos gráficos e de *layout*. A foto foi realizada por meio de um smartphone.

Após a realização da fotografia, foi feito o *upload* dela no Notebook Acer para que pudesse estar disponível na pasta de documentos, possibilitando seu *upload* no ambiente do Eden AI para análise. Nota-se que todas as imagens utilizadas nesta pesquisa são modelos retirados da internet, não expondo informações de nenhuma pessoa real.

- **Microsoft Azure e Google**

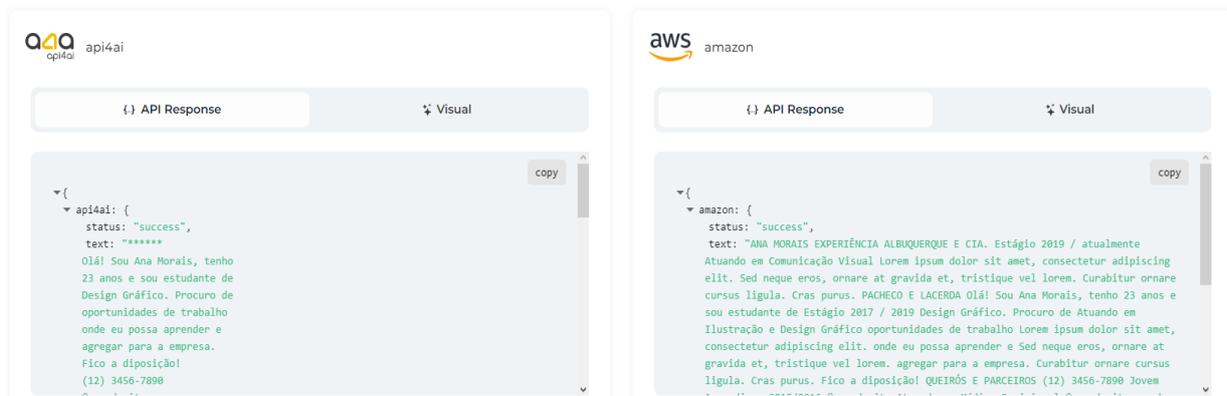
Figura 4.2 – Response Microsoft Azure e Google – imagem 1



Fonte: elaborada pelos autores, 2022.

- **API4AI e Amazon AWS**

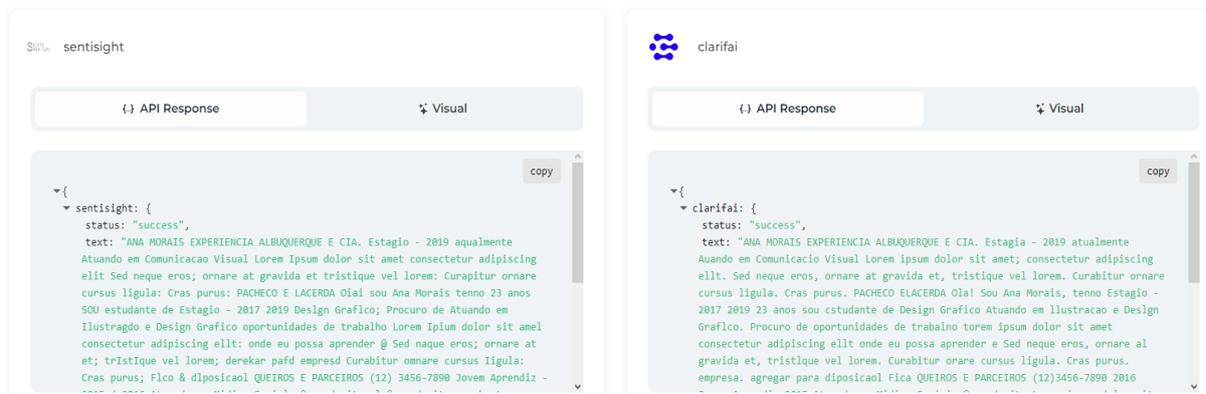
Figura 4.3 – Response API4AI e AWS – imagem 1



Fonte: elaborada pelos autores, 2022.

- **SentiSight e Clarifai**

Figura 4.4 – Response SentiSight e Clarifai – imagem 1



Fonte: elaborada pelos autores, 2022.

b) Imagem 2: Prestação de serviço

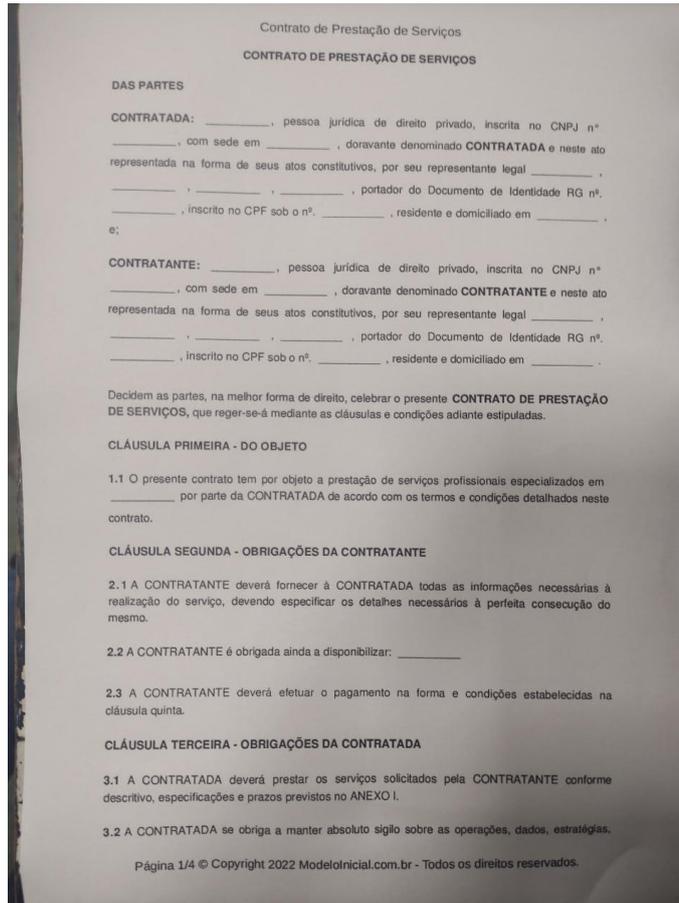
A segunda imagem analisada é um documento de prestação de serviços. Possui em sua formatação diversos números e linhas de preenchimento, porém, a princípio é um documento básico, visto conter somente elementos textuais. A foto foi realizada por meio de um smartphone.

Tabela 4.2 – Especificações da imagem – foto prestação de serviços

Fotografia – Prestação de serviço	
Tipo	Fotografia – câmera de smartphone
Idioma	Português
Resolução	961x1280

Fonte: elaborada pelos autores, 2022.

Figura 4.5 – Fotografia base do documento – imagem 2



Fonte: elaborada pelos autores, 2022.

- **Microsoft Azure e API4AI**

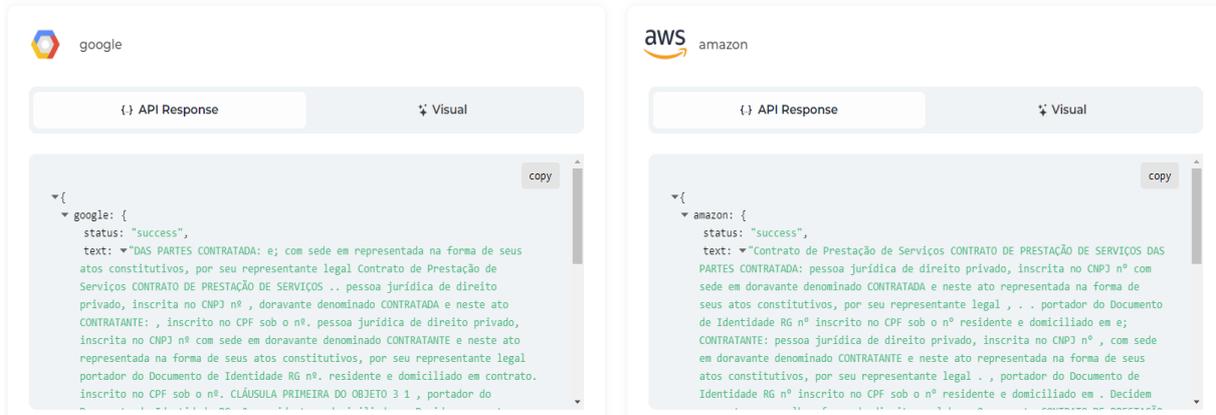
Figura 4.6 – Response MS Azure e API4AI – imagem 2



Fonte: elaborada pelos autores, 2022.

- **Google e AWS**

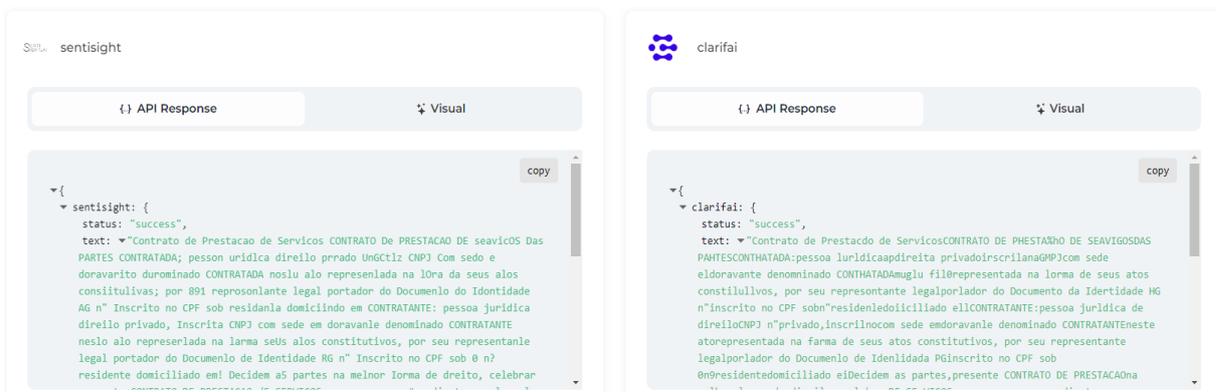
Figura 4.7 – Response Google e AWS – imagem 2



Fonte: elaborada pelos autores, 2022.

- **SentiSight e Clarifai**

Figura 4.8 – Response SentiSight e Clarifai – imagem 2



Fonte: elaborada pelos autores, 2022.

c) Imagem 3: Documento em inglês

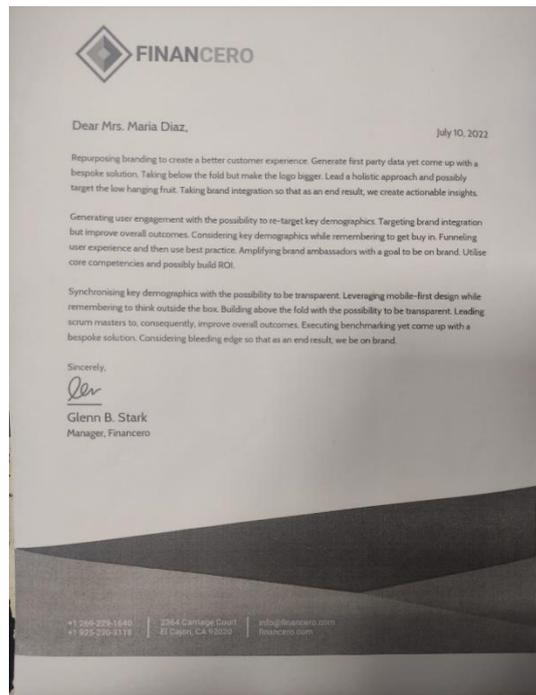
A terceira reconhecida é um documento na Língua Inglesa. Tem como formatação cabeçalho com logotipo, fontes com tamanhos diferentes, campo de assinatura preenchida e um rodapé. A foto foi realizada por meio de um *smartphone*.

Tabela 4.3 – Especificações da imagem - documento em inglês

Fotografia – Documento em inglês	
Tipo	Fotografia – câmera de smartphone
Idioma	Inglês
Resolução	1201x1600

Fonte: elaborada pelos autores, 2022.

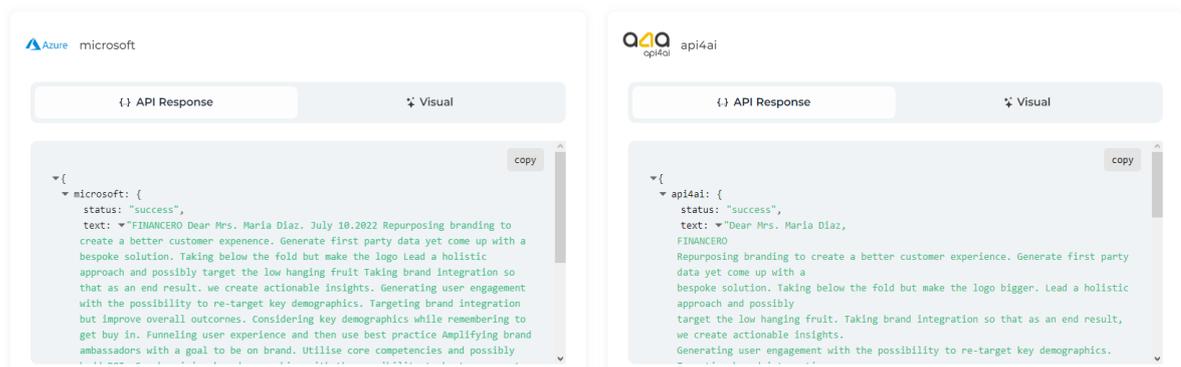
Figura 4.9 – Imagem base do documento em inglês – imagem 3



Fonte: elaborada pelos autores, 2022.

- **Microsoft Azure e API4AI**

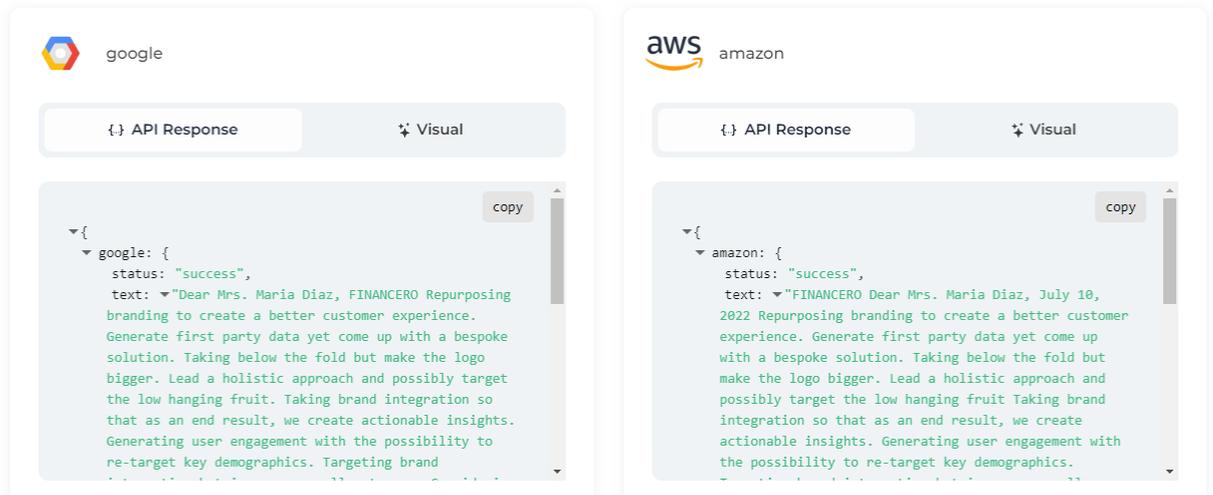
Figura 4.10 – Response MS Azure e API4AI – imagem 3



Fonte: elaborada pelos autores, 2022.

- **Google e AWS**

Figura 4.11 – Response Google e AWS – imagem 3



Fonte: elaborada pelos autores, 2022.

- **SentiSight e Clarifai**

Figura 4.12 – Response SentiSight e Clarifai – imagem 3



Fonte: elaborada pelos autores, 2022.

d) Imagem 4: Currículo digitalizado

A partir deste tópico as imagens são repetidas, tendo como diferença as resoluções, pois os documentos foram escaneados, aumentando sua resolução. O

objetivo é identificar se as bibliotecas sofrem impacto a partir de fontes diferentes de obtenção das imagens. A primeira é a digitalização do currículo.

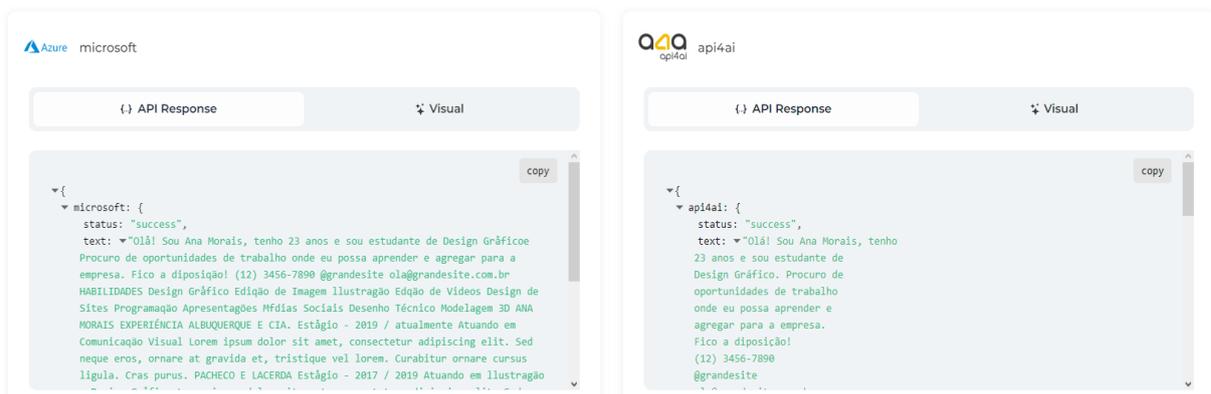
Tabela 4.4 – Especificações da imagem – digitalização do currículo

Scanner - Currículo	
Tipo	Scanner
Idioma	Português
Resolução	2528x3507

Fonte: elaborada pelos autores, 2022.

- **Microsoft Azure e API4AI**

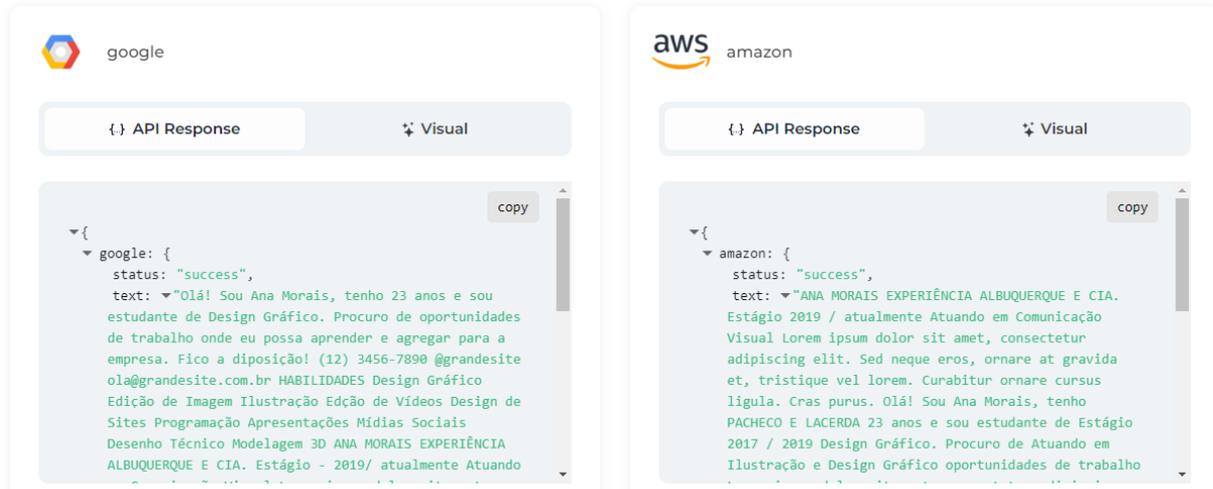
Figura 4.13 – Response Microsoft Azure e API4AI – imagem 4



Fonte: elaborada pelos autores, 2022.

- **Google e AWS**

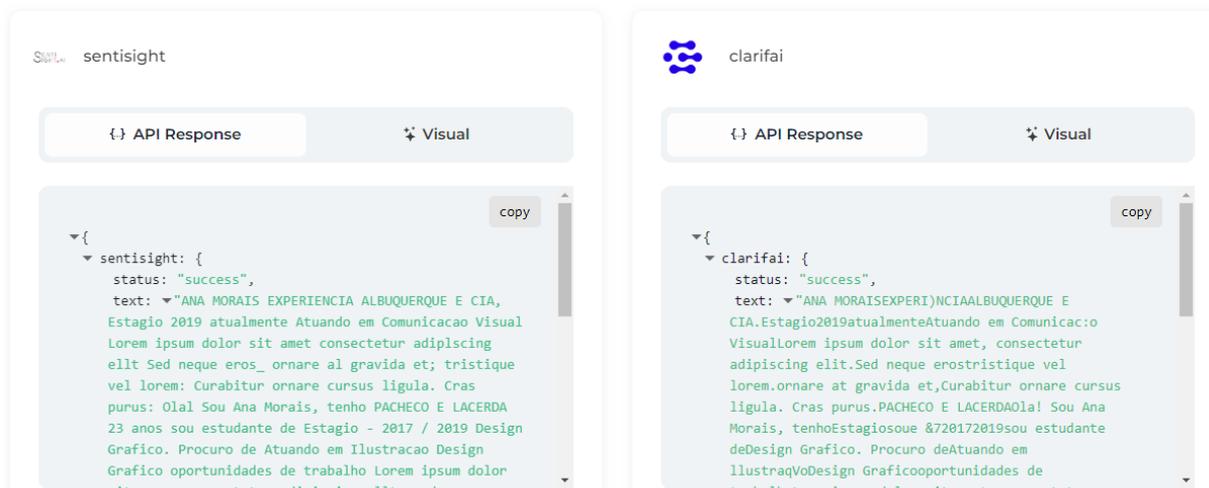
Figura 4.14 – Response Google e AWS – imagem 4



Fonte: elaborada pelos autores 2022.

- **SentiSight e Clarifai**

Figura 4.15 – Response SentiSight e Clarifai – imagem 4



Fonte: elaborada pelos autores, 2022.

e) Imagem 5: Prestação de serviço digitalizado

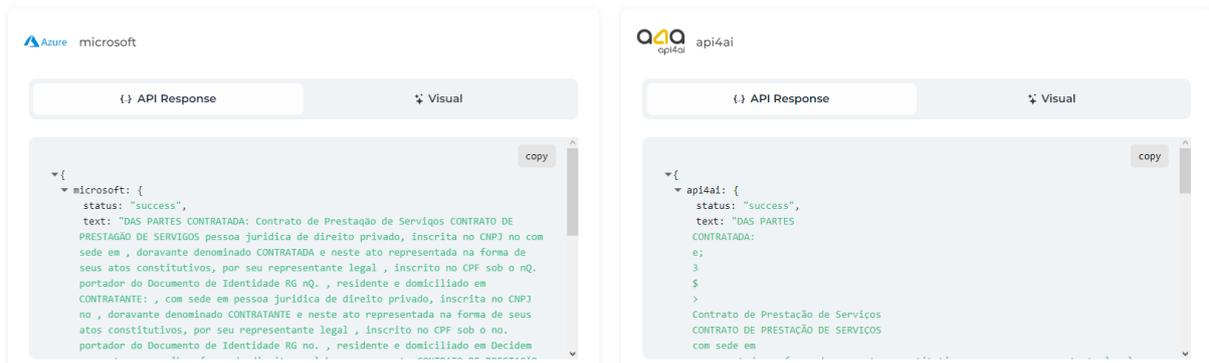
Tabela 4.5 – Especificações da imagem - digitalização prestação de serviços

Scanner – Prestação de serviços	
Tipo	Scanner
Idioma	Português
Resolução	2528x3507

Fonte: elaborada pelos autores, 2022.

- **Microsoft Azure e API4AI**

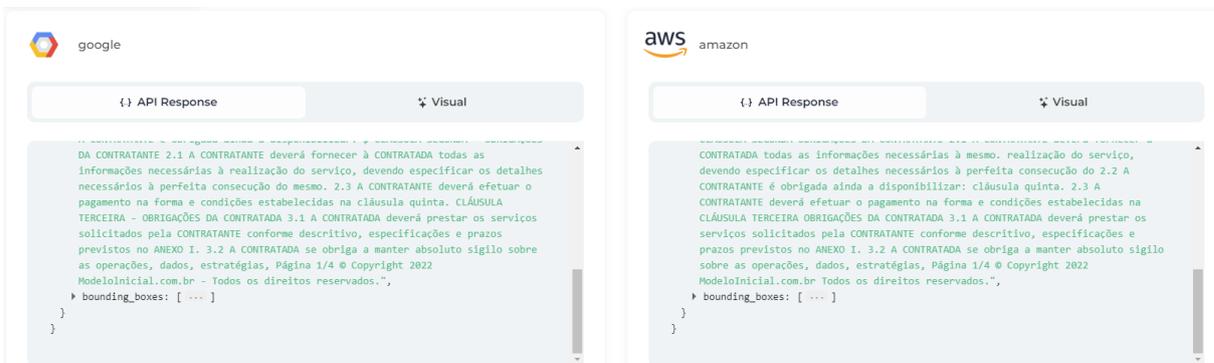
Figura 4.16 – Response da Microsoft e API4AI – imagem 5



Fonte: elaborada pelos autores, 2022.

- **Google e AWS**

Figura 4.17 – Response do Google e AWS – imagem 5



Fonte: elaborada pelos autores, 2022.

- **SentiSight e Clarifai**

Figura 4.18 – Response SentiSight e Clarifai – imagem 5



Fonte: elaborada pelos autores, 2022.

f) Imagem 6: Documento em inglês digitalizado

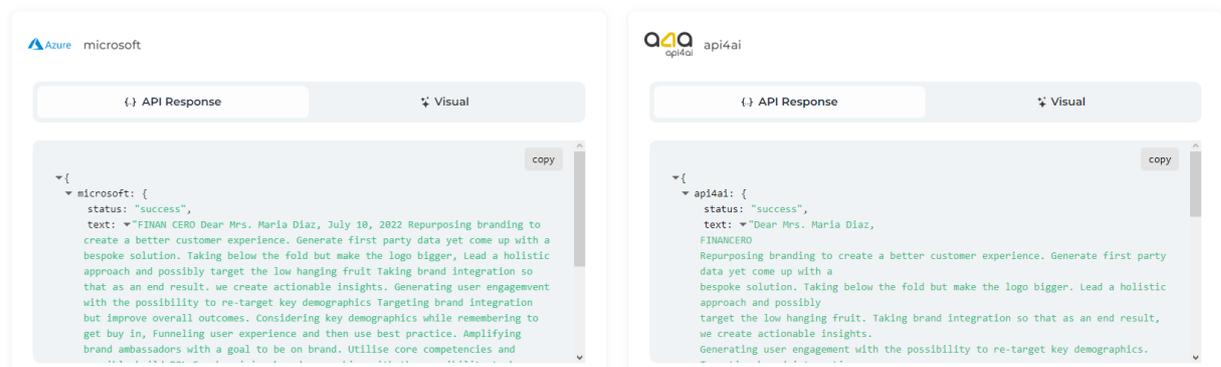
Tabela 4.6 – Especificações da imagem – digitalização do doc em inglês

Scanner – documento em inglês	
Tipo	Scanner
Idioma	Inglês
Resolução	2528x3507

Fonte: elaborada pelos autores, 2022.

- **Microsoft e API4AI**

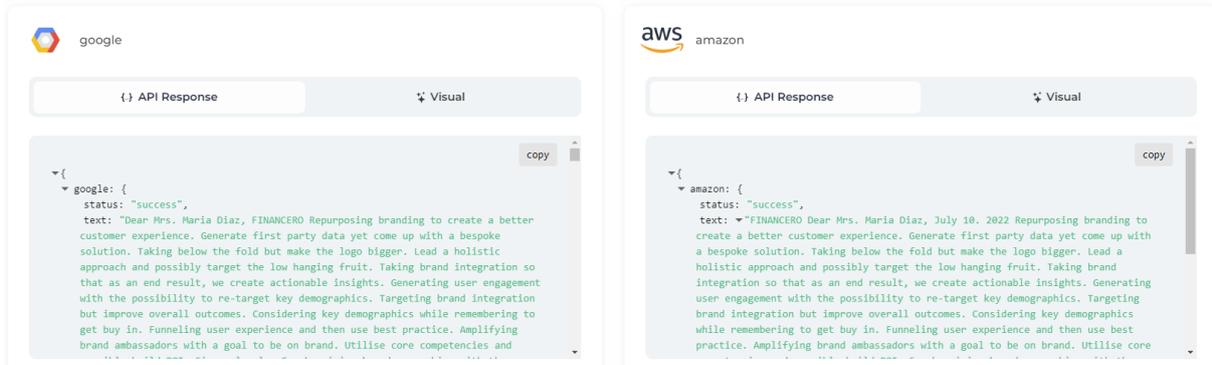
Figura 4.19 – Response Microsoft e API4AI – imagem 6



Fonte: elaborada pelos autores, 2022.

- **Google e AWS**

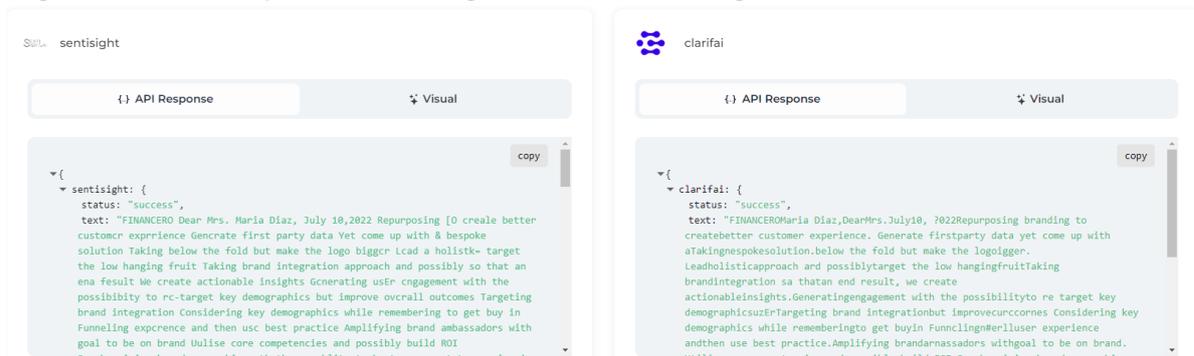
Figura 4.20 – Response Google e AWS – imagem 6



Fonte: elaborada pelos autores, 2022.

- **SentiSight e Clarifai**

Figura 4.21 – Response SentiSight e Clarifai – imagem 6



Fonte: elaborada pelos autores, 2022.

Após a coleta dos resultados é necessário calcular a precisão do reconhecimento óptico de caracteres. Vale ressaltar que, observando os responses, já é possível notar uma diferença entre uma biblioteca e outra, tendo algumas não tendo configuração para reconhecer o padrão utf-8 - textos com acentuações - unificação de palavras e acréscimos de caracteres inexistentes na imagem de entrada. Vale ressaltar, que a API4AI é a única que retorna os textos com o mesmo *layout* disposto nas imagens originais.

4.2 Cálculos da precisão no `precisaocr.py`

Neste tópico é apresentado e analisado os resultados da comparação entre o texto original dos seis documentos analisados e reconhecidos, realizando uma

comparação com o texto reconhecido pelas bibliotecas do Eden AI. O objetivo é coletar dados da precisão para centralizá-los no Excel para totalização dos resultados.

A figura 4.40 apresenta a estrutura do código que está sendo utilizado para a realização do cálculo de acuracidade – também pode ser observado no teste apresentado no capítulo 3, quadro 3.4 – bem como a figura 4.41 demonstra o cálculo da precisão da biblioteca API4AI. Logo em seguida, é descrito os resultados de todas as demais bibliotecas.

- **Imagem 1 – Fotografia currículo**

Figura 4.22 – Estrutura do precisaoOCR.py

```

PrecisaoOCR.py > ...
3  main_text = "texto base"
4  result_ocr = "texto ocr"
5
6  sm = SequenceMatcher(None, result_ocr, main_text)
7  true_positive_char_num = 0
8  for tag, i1, i2, j1, j2 in sm.get_opcodes():
9      if tag == 'equal':
10         true_positive_char_num += (j2 - j1)
11     else:
12         pass
13
14 print('Biblioteca')
15 print(f'precisão = {true_positive_char_num/len(main_text) * 100:.2f}%')
16

```

Fonte: elaborada pelos autores, 2022.

Figura 4.23 – Precisão API4AI na imagem 1

```

main_text = "Olá! Sou Ana Moraes, tenho 30 anos e moro em São Paulo."
result_ocr = "***** Olá! Sou Ana Moraes, tenho 30 anos e moro em São Paulo."

sm = SequenceMatcher(None, result_ocr, main_text)
true_positive_char_num = 0
for tag, i1, i2, j1, j2 in sm.get_opcodes():
    if tag == 'equal':
        true_positive_char_num += (j2 - j1)
    else:
        pass

print('API4AI')
print(f'precisão = {true_positive_char_num/len(main_text) * 100:.2f}%')

```

```

Python 3.11.0 (main, Oct 24 2022, 18:26:48) [MSC
v.1933 64 bit (AMD64)]
Type 'copyright', 'credits' or 'license' for more
information
IPython 8.6.0 -- An enhanced Interactive Python. Type
'?' for help.

✓ from difflib import SequenceMatcher ...

... API4AI
   precisão = 69.33%

```

Fonte: elaborada pelos autores, 2022.

Levando em consideração as figuras anteriores foram coletadas os seguintes resultados de acuracidade:

Na primeira imagem – fotografia do currículo – a API4AI obteve 69,33% de precisão, o Google com 69,23%, Microsoft com 45,11%, SentiSight com precisão de 20,17%, Amazon com 16,84% e a biblioteca do Clarifai com 15,70%.

Na segunda imagem – fotografia de prestação de serviços – considerando em ordem decrescente dos valores, a Amazon possui precisão de 89,03%, Google e API4AI estão empatados em 74,90%, Microsoft resultou em 71,16%, SentiSight fez 43,89% e o Clarifai possui acuracidade na imagem dois de 43,97%.

Seguindo para a terceira imagem – fotografia do documento em inglês – Amazon teve resultado de 98,20%, Google e API4AI obtiveram 92,16%, Microsoft em 90,36%, Clarifai com 46,94% e por último, SentiSight com valor de 37,70%.

Na quarta imagem – currículo digitalizado – em ordem decrescente: API4AI marcou 69,75%, Google obteve um resultado de 69,65%, Microsoft fez 45,74%, SentiSight obteve 20,89%, Amazon teve 15,90% e o Clarifai fez 11,85%.

Os resultados da quinta imagem – prestação de serviços digitalizada - tiveram a seguinte sequência, a Amazon obteve um resultado de 85,88%, o MS Azure 70,67%, a API4AI teve 60,40% de resultado, o Google com a acurácia de 60,24% e o Clarifai com 51,93%.

A sexta e última imagem – documento em inglês digitalizada - teve a Amazon 99,76%, a API4AI totalizou 91,85%, o Google com 91,54%, Microsoft obteve 58,93%, Clarifai com 42,63% e SentiSight obteve a menor pontuação, com 34,33%. Nota-se que, a Amazon foi a biblioteca mais próxima a obter 100% de precisão em uma imagem.

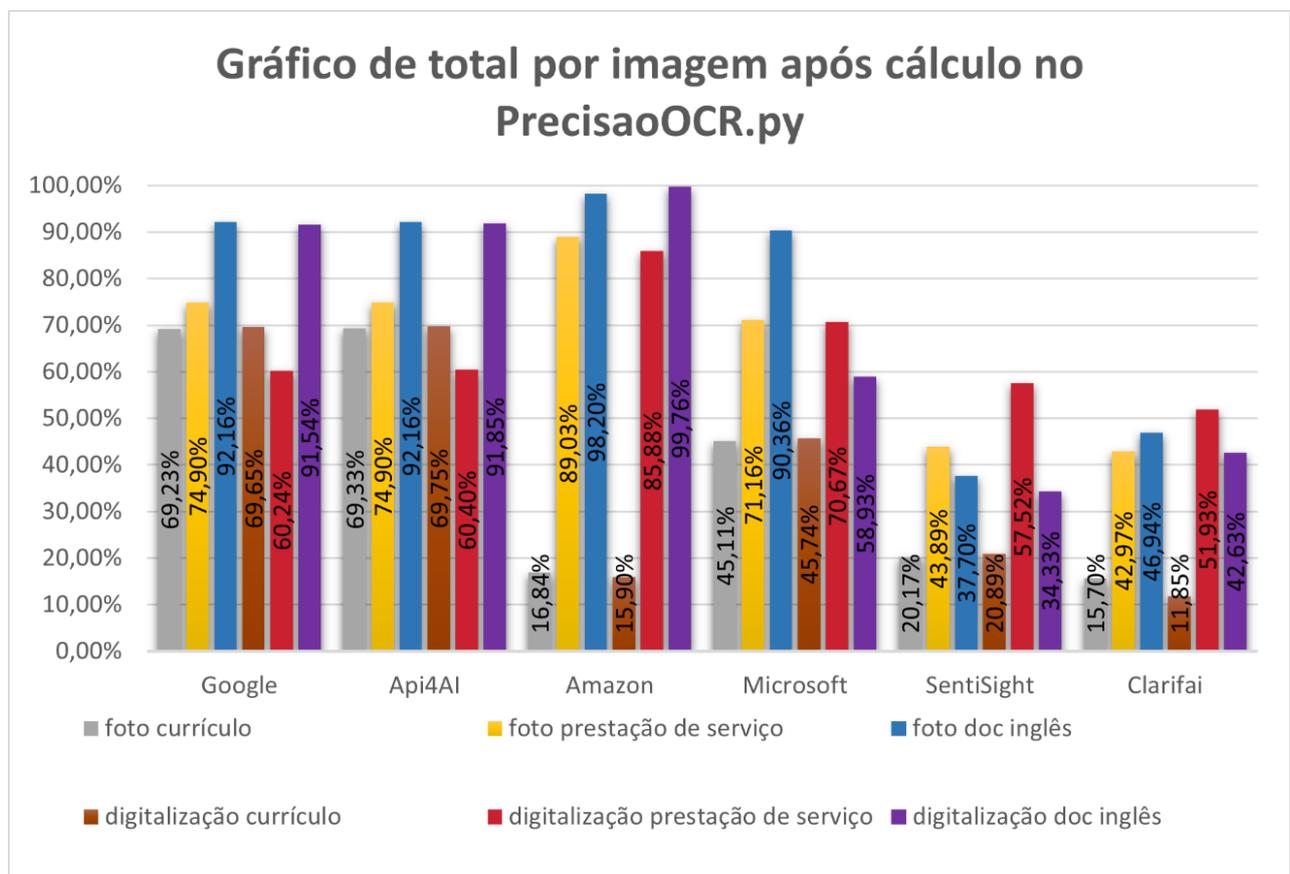
4.3 Totalizando os resultados no Excel e api mais precisa

Neste tópico é demonstrada as totalizações por meio de gráficos, onde se encontram ranqueadas as bibliotecas conforme sua precisão – no gráfico de resultados por imagem – e pela média dos resultados – no gráfico que define a melhor

biblioteca dos resultados médios. Nota-se que o cálculo foi efetuado com base na média aritmética dos valores, ou seja, é somado os valores de todas as seis imagens e logo em seguida é feita a divisão pela quantidade das imagens.

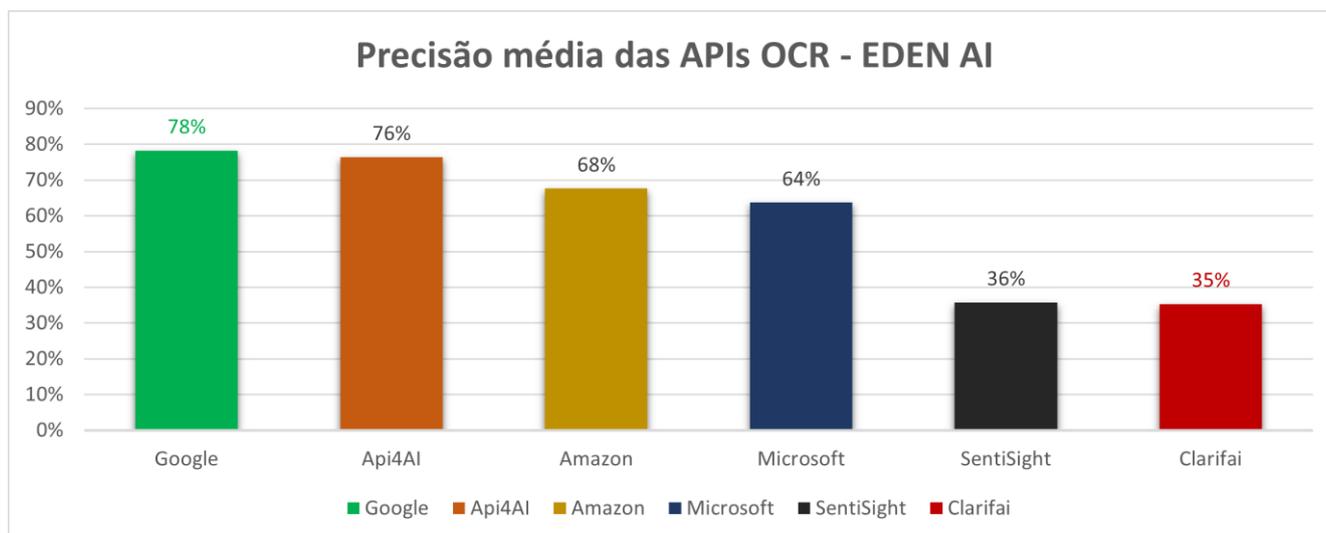
O gráfico 4.1 apresenta a precisão que cada biblioteca obteve após a análise das imagens e dos cálculos realizados no `precisaoOCR.py`. Enquanto o gráfico 4.2 demonstra os resultados totais pela média dos valores.

Gráfico 4.1 – Precisão das bibliotecas OCR em cada imagem



Fonte: elaborado pelos autores, 2022.

Gráfico 4.2 – Total pela média dos valores obtidos



Fonte: elaborado pelos autores, 2022.

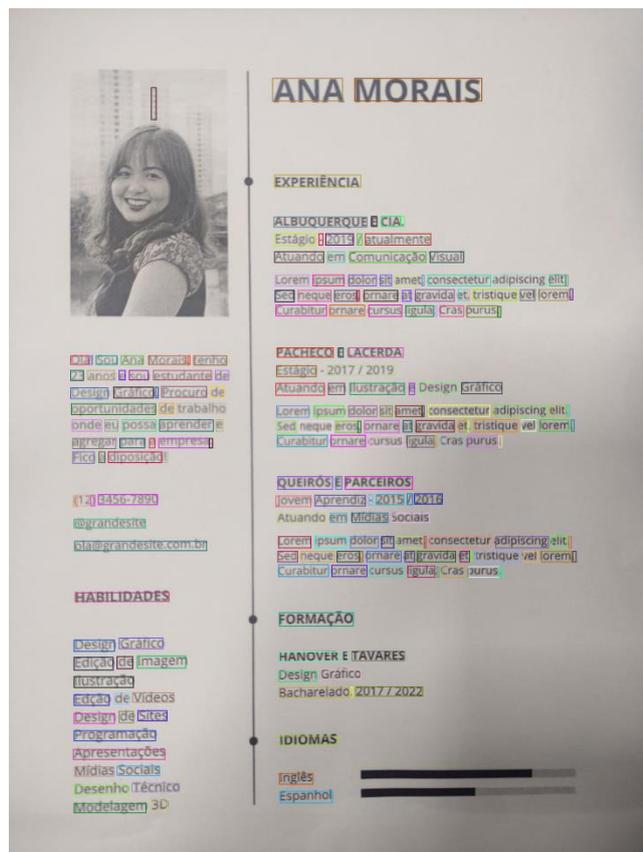
O estudo forneceu diversos resultados notáveis. Na realidade, houve um empate entre o Google e a API4AI, com uma pontuação de 76% de ambos no total, o critério de desempate realizado foi o reconhecimento do guião – figura 4.24 e 4.25 – que, enquanto o Google definia suas segmentações de acordo com o espaçamento dos caracteres, a API4AI definia apenas uma segmentação para o reconhecimento e, apesar de obter um resultado satisfatório, a regra define que a separação de segmentações siga o definido na figura 2.6, dessa forma, foi acrescentado 0,9 pontos – foi realizado uma análise de que, apesar de realizar as segmentações corretamente, o desempenho em algumas imagens analisadas pelo Google foi inferior ao da API4AI, dessa forma não foi adicionado um ponto a mais para o desempate e sim 0,9 – o que levou ao Google a pontuar 78%.

Também é necessário pontuar o grau de desempenho da lib da Amazon nas imagens três e seis, onde o idioma é inglês, a partir desses dados surge a hipótese de que a biblioteca possui uma maior familiaridade com a Língua Inglesa, ainda que reconheça o português – vale citar que, o desempenho da Amazon em ambas as imagens pode não estar vinculada ao idioma e sim o layout, a disposição do texto, a quantidade de linhas e imagens, sendo a língua apenas uma linha de raciocínio com base na comparação entre ambas as figuras – ainda nessa biblioteca, devido ao cálculo da média aritmética, o desempenho aquém nas imagens um e quatro ocasionaram em uma pontuação menor que o Google e a API4AI.

A Microsoft obteve resultados equilibrados, exceto pela imagem três, da qual obteve uma pontuação acima do restante das análises. Nota-se que, era esperado um melhor desempenho nas imagens digitalizadas por conta da resolução, no entanto, o resultado foi inferior, a hipótese levantada neste estudo é que, devido ao brilho da imagem digitalizada, a análise não foi tão bem-sucedida quanto as imagens fotografadas, que possuem um brilho menor.

O SentiSight e o Clarifai possuem pontuações semelhantes, havendo apenas um ponto de diferença entre ambos, as segmentações de ambos também estavam dispostas de forma confusa, não obtendo as coordenadas corretas, a conjectura é que a disposição das segmentações afeta a análise final da biblioteca, ainda que a API4AI também realize uma segmentação única para o documento todo, sua configuração não parece afetar o reconhecimento dos caracteres.

Figura 4.24 – reconhecimento do guião no Google



Fonte: elaborada pelos autores, 2022.

5 CONSIDERAÇÕES FINAIS

Com a análise realizada no ambiente Eden AI foi possível perceber as diferenças e semelhanças de todas as bibliotecas disponíveis no site. Mesmo que as opções disponibilizadas representam o mínimo da capacidade e das possibilidades do que a tecnologia OCR permite realizar, é uma visão micro do poder de aprendizagem que a ferramenta possui, e um ambiente que centraliza diversas bibliotecas para teste é de suma importância ao desenvolvedor que deseja aplicar o código em um eventual sistema.

Ainda as *libs* presentes no Eden AI possam melhorar de acordo com a manutenção realizada pelos fornecedores possam vir a ocorrer um dia seguinte após esse estudo, fato é que, cada uma se comporta de uma maneira única, mesmo havendo empates – no caso do Google e da API4AI – e tratando-se da API4AI é surpreendente a velocidade com que foi realizada sua manutenção, visto que, no início da fase de testes apresentava diversas falhas por conta do suporte de idioma, logo em seguida retornou a normalidade com capacidade de ficar no topo do pódio.

O Google segue, sendo uma empresa que investe e propõe-se a melhorar as tecnologias desenvolvidas, mantendo um padrão de acerto em todas as imagens. Enquanto a Amazon pode trabalhar para que sua ferramenta se torne ainda mais eficaz, sobretudo, em textos da Língua Portuguesa, tendo o tamanho do grau de acertos nas imagens de Língua inglesa. É perceptível, também, que a AWS tem certa dificuldade com textos editados em diferentes posicionamentos e com linhas e imagens.

A Microsoft também possui dificuldades com textos em português e com diversos elementos dentro dos textos, podendo ter melhorias no que diz respeito a essas parametrizações. E enfim, as bibliotecas do SentiSight e Clarifai podem ter manutenção tanto nas definições de segmentações quanto no reconhecimento, tendo em vista a dificuldade no reconhecimento de caracteres.

Todas as informações coletadas por meio das métricas definidas levam à conclusão de que a tecnologia OCR é diversificada e única em cada ambiente que é proposta, seus aplicativos que utilizam essas bibliotecas são mais trabalhados devido ao uso exclusivo da plataforma, mas é de suma importância a disponibilização em

nuvem das *libs* de APIs, tendo como contexto social tecnologias cada vez mais acessíveis.

Com a presente pesquisa buscou-se contribuir para a comunidade acadêmica com dados reais acerca da precisão das APIs analisadas, contribuindo para o conhecimento da tecnologia de Reconhecimento Óptico que, apesar de muito utilizada, não é muito conhecida, tanto por usuários comuns, quanto colaboradores da área de tecnologia.

Com a experiência adquirida durante o desenvolvimento deste estudo, sugere-se como trabalhos futuros, o desenvolvimento de novas bibliotecas de API de Reconhecimento Óptico de Caracteres – utilizando a base de análise desse trabalho, disponível no Google Drive² dos desenvolvedores do presente estudo - a fim de realizar o teste de acurácia no decorrer do desenvolvimento dessa nova API a para a verificação de sua acurácia, o código para o cálculo da precisão pode ser encontrado no quadro 3.3.

Também é sugerido o desenvolvimento de um aplicativo que reconheça placas de diversos idiomas para traduzir ao idioma selecionado pelo turista. O aplicativo pode utilizar uma das bibliotecas presentes no estudo, verificando qual a melhor para o tipo de desenvolvimento proposto.

Por fim, é possível melhorar a própria pesquisa realizada, aplicando o teste em mais bibliotecas ou sistemas de OCR, utilizando a mesma base de análise do trabalho, assim como aplicar um outro tipo de cálculo da média, a ponderada, onde é definido pesos para cada imagem analisada, tendo por objetivo não impactar no resultado.

² Google Drive com as imagens para teste:
<https://drive.google.com/drive/folders/1MAw-qPZbkCgrOUlbt4rmciYMWkrUYZ49?usp=sharing>

REFERÊNCIAS BIBLIOGRÁFICAS

AMAZON. Amazon Textract. Amazon, 2022. Disponível em: <<https://aws.amazon.com/pt/textract/#:~:text=O%20Amazon%20Textract%20é%20um,dados%20de%20formulários%20e%20tabelas.>>. Acesso em: 12 nov 2022.

COSTA, Janine. APIs: Tudo sobre o que é API, exemplos e importância! Zenvia, 2022. Disponível em: <l1nq.com/CJMzG>. Acesso em: 12 nov 2022.

Eden AI. Eden AI API Documentation (v1). Eden AI, 2022. Disponível em: <<https://api.edenai.run/v1/redoc/>>. Acesso em: 12 nov 2022.

Eden AI. OCR Technologies. Eden AI, 2022. Disponível em: <<https://www.edenai.co/technologies/ocr>>. Acesso em: 12 nov 2022.

FGV. Brasil tem 424 milhões de dispositivos digitais em uso, revela a 31ª Pesquisa Anual do FGVcia. FGV, 2020. Disponível em: <<https://portal.fgv.br/noticias/brasil-tem-424-milhoes-dispositivos-digitais-uso-revela-31a-pesquisa-anual-fgvcia>>. Acesso em: 19 fev 2022.

FOLTER, Rodrigo. O que é OCR? | Reconhecimento óptico de caracteres. Canaltech, 2022. Disponível em: <<https://canaltech.com.br/internet/o-que-e-ocr-reconhecimento-optico-de-caracteres/>>. Acesso em: 05 nov 2022.

HANASHIRO, Akira. VS Code - O que é e por que você deve usar? Treinaweb, 2021. Disponível em: <<https://www.treinaweb.com.br/blog/vs-code-o-que-e-e-por-que-voce-deve-usar>>. Acesso em: 13 nov 2022.

HASHTAG. O QUE É PYTHON E POR QUE APRENDER? GUIA PARA INICIANTES. Hashtag, 2022. Disponível em: <l1nq.com/CFUuw>. Acesso em: 13 nov 2022.

HASHTAG. O QUE É UMA API E COMO UTILIZÁ-LA? Hashtag, 2021. Disponível em: <<https://www.hashtagtreinamentos.com/o-que-e-uma-api-python#:~:text=Nada%20mais%20é%20do%20que,como%20por%20exemplo%20o%20Facebook.>>. Acesso em: 12 nov 2022.

KLIPPA. O que é OCR? O melhor guia de OCR 2022. Klippa, 2022. Disponível em: <<https://www.klippa.com/pt/blog/informacao/o-que-e-ocr/>>. Acesso em: 05 nov 2022.

MICROSOFT. O que é OCR (Reconhecimento Óptico de Caracteres). Microsoft, 2022. Disponível em: <<https://learn.microsoft.com/pt-br/azure/cognitive-services/computer-vision/overview-ocr>>. Acesso em: 12 nov 2022.

PYTHON. difflib — Helpers for computing deltas. Python, 2022. Disponível em: <<https://docs.python.org/3/library/difflib.html>>. Acesso em: 12 nov 2022.

REDAÇÃO. Eden AI unifica APIs de ML de vários fornecedores de nuvem. Teg6, 2022. Disponível em: <<https://teg6.com/40607/noticias/eden-ai-unifica-apis-de-ml-de-varios-fornecedores-de-nuvem/>>. Acesso em: 13 nov 2022.

REDHAT. O que é uma API? Redhat, 2017. Disponível em: <<https://www.redhat.com/pt-br/topics/api/what-are-application-programming-interfaces#história-das-apis>>. Acesso em: 12 nov 2022.

SentiSight. About. SentiSight, 2022. Disponível em: <<https://www.sentisight.ai/about/>>. Acesso em: 12 nov 2022.

SOPHIA, Ana. O que é OCR e como ele funciona. Wondershare, 2021. Disponível em: < <https://pdf.wondershare.com.br/pdf-knowledge/what-is-ocr.html> >. Acesso em: 19 fev 2022.