



UNIVERSIDADE FEDERAL DO CEARÁ
CAMPUS QUIXADÁ
CURSO DE GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

RONILDO OLIVEIRA DA SILVA

**ANÁLISE DE DESEMPENHO DA *GOOGLE CLOUD VISION API* EM LEITURA DE
TEXTOS PROVENIENTES DE IMAGENS NATURAIS**

QUIXADÁ
2019

RONILDO OLIVEIRA DA SILVA

ANÁLISE DE DESEMPENHO DA *GOOGLE CLOUD VISION API* EM LEITURA DE
TEXTOS PROVENIENTES DE IMAGENS NATURAIS

Trabalho de Conclusão de Curso apresentado ao
Curso de Graduação em Ciência da Computação
do *Campus* Quixadá da Universidade Federal
do Ceará, como requisito parcial à obtenção do
grau de bacharel em Ciência da Computação.

Orientador: Prof. Dr. Regis Pires Magalhães

QUIXADÁ

2019

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Biblioteca Universitária
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

- S583a Silva, Ronildo Oliveira da.
Análise de desempenho da Google Cloud Vision API em leitura de textos provenientes de imagens naturais / Ronildo Oliveira da Silva. – 2019.
53 f. : il. color.
- Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Campus de Quixadá, Curso de Ciência da Computação, Quixadá, 2019.
Orientação: Prof. Dr. Regis Pires Magalhães.
1. Google Cloud Platform. 2. Visão Computacional. 3. Análise de Dados. I. Título.

CDD 004

RONILDO OLIVEIRA DA SILVA

ANÁLISE DE DESEMPENHO DA *GOOGLE CLOUD VISION API* EM LEITURA DE
TEXTOS PROVENIENTES DE IMAGENS NATURAIS

Trabalho de Conclusão de Curso apresentado ao
Curso de Graduação em Ciência da Computação
do *Campus* Quixadá da Universidade Federal
do Ceará, como requisito parcial à obtenção do
grau de bacharel em Ciência da Computação.

Aprovada em: ____ / ____ / ____

BANCA EXAMINADORA

Prof. Dr. Regis Pires Magalhães (Orientador)
Universidade Federal do Ceará (UFC)

Prof. Ma. Lívia Almada Cruz Rafael
Universidade Federal do Ceará (UFC)

Prof. Dr. Marcio Espíndola Freire Maia
Universidade Federal do Ceará (UFC)

Este trabalho é dedicado ao meu pai, minha referência de honestidade, responsabilidade e respeito.

AGRADECIMENTOS

Ao meu Pai, Seu Joaquim que de uma forma diferente se tornou muito mais meu amigo que alguém do mesmo sangue, que me ensinou a ser honesto, justo e responsável. Agradeço pela oportunidade que me deu aos seis anos: “_Ronildo, essa roça aqui é sua, se você quiser trabalhar, nós vamos trabalhar, mas eu prefiro que você estude...”

À minha mãe, Dona Francisca que sempre estava à disposição, pela preocupação (ainda chora quando eu pego o ônibus pra viajar) com os filhos, pelo peito cheio de orgulho que constantemente diz que tem.

Ao meu irmão Ricardo, que contribuiu e continua contribuindo para com a educação e formação de estudantes, inclusive a minha. Pelas lições de como não desistir e de que nunca é tarde pra fazer algo de útil.

À “Dona da minha cabeça que vem como um carnaval...”, Gleiciania.

Aos meus amigos: Leonardo, Marcelo, Marcello, Nathália, Arthur, Darliene, Amanda, Yuri, Michel, ‘Jon’, Dieinison, Paulo, Débora, ‘Colegiado’ e ao Dias, que sempre traz uma palavra amiga e sorriso no rosto...

Aos professores do IFCE - *Campus* Cedro: Emmanuel Diego, Róger Moura e em especial, Derig Almeida, que não só me ensinaram a desenvolver código, mas que me deram lições de honestidade, responsabilidade, profissionalismo e acreditaram no meu potencial.

A Deodato Aquino, que em uma conversa durante um evento, conheceu, reconheceu minhas raízes e me disse: “Você tem a capacidade de formar pessoas, eu me disponho a te ajudar no curso que você quiser... Hoje eu estou te ajudando, mas amanhã você vai ajudar alguém e continuar essa corrente do bem.”

Aos professores da UFC - *Campus* Quixadá: David Sena, Fábio Dias, Márcio Maia e Paulo Rego, meus orientadores enquanto bolsista. Foi muito bom trabalhar com vocês, jamais esquecerei a confiança depositada.

Ao Professor Joel Ramiro, um dos motivadores a continuar e acreditar que podemos somar bons valores como profissionais do ensino.

Ao Professor Regis Pires, não apenas pela ótima orientação, mas que desde o início acreditou na proposta deste trabalho, que me incentivou nos momentos mais complexos que houveram durante a produção deste projeto. Agradeço muito por aquele “Vamos em frente!”... Conseguimos.

“Hoje só acredito no pulsar das minhas veias.”

(Graco Sílvio Braz)

RESUMO

Grandes companhias do mercado de tecnologia da informação vem adotando técnicas voltadas à Inteligência Artificial, Visão Computacional para resolver problemas diários como conhecimento facial para desbloqueio de celulares, detecção de automóveis, pessoas, tradução de placas e avisos, etc. Tudo isso, em tempo real ou muito próximo dessa realidade. Sabendo que os modelos computacionais estão evoluindo rapidamente e ficando cada vez mais robustos e presentes no mundo real, que vem a curiosidade de analisar essas características, como robustez, rapidez, confiança e segurança dessas ferramentas. Este trabalho tem por objetivo analisar a acurácia da ferramenta *Google Cloud Vision* para detecção e reconhecimento de textos em imagens naturais e encontrar quais os fatores que existem no ambiente em que vivemos que podem reduzir a flexibilidade de uma ferramenta produzida por uma gigante do mercado.

Palavras-chave: *Google Cloud Vision*. Visão Computacional. Inteligência Artificial. Análise de Dados. Aprendizado de Máquina.

ABSTRACT

Great companies in the information technology market have been adopting techniques aimed to Artificial Intelligence and Computer Vision to solve daily problems such as facial recognition to unlock cell phones, detection of automobiles, people, translation of signs and warnings, etc. All of this, in real time or very close to this reality. Knowing that computer models are rapidly evolving and becoming increasingly robust and present in the real world, appears the curiosity to analyze these characteristics, such as robustness, speed, confidence and security of these tools. This work aims to analyze the accuracy of the Google Cloud Vision tool for detection and recognition of texts in natural images and find out what factors exist in the environment that we live and that can reduce the flexibility of a tool produced by a market giant.

Keywords: Google Cloud Vision. Computer Vision. Artificial Intelligence. Data Analysis. Machine Learning.

LISTA DE FIGURAS

Figura 1	– O custo total de edição de ‘SNOWY’ em ‘SUNNY’ é igual a 3. Uma inserção da letra ‘U’, substituição da letra ‘O’ pela letra ‘N’ e remoção da letra ‘W’.	18
Figura 2	– Exemplo das informações apresentadas numa imagem do <i>dataset Common Objects in Context Text (COCO-Text)</i> .	22
Figura 3	– Estrutura da anotação de uma imagem do <i>dataset COCO-Text</i> .	22
Figura 4	– Imagem 1008258.jpg do conjunto de dados da Tarefa 2.	23
Figura 5	– Exemplo de linhas de resultado da Tarefa 2 no formato configurado para submissão.	23
Figura 6	– Exemplo do <i>endpoint</i> (ponto de extremidade) compatível com o método de solicitação HTTP.	24
Figura 7	– Estrutura da solicitação de detecção de texto da <i>Google Cloud Vision Application Programming Interface (API)</i> .	25
Figura 8	– Estrutura da resposta de detecção de texto da <i>Google Cloud Vision API</i> .	25
Figura 9	– Descrição gráfica do <i>bounding box</i> e as anotações detectadas pela <i>Google Vision API</i> .	26
Figura 10	– Capturas de tela dos rótulos retornados pela <i>Google Cloud Vision API</i> imagens originais, ruidosas e restauradas. A imagem original (à esquerda) é pertencente ao <i>ImageNet</i> . Nenhuma das etiquetas retornadas para a imagem ruidosa (ao centro) está relacionada aos rótulos da imagem original, enquanto os rótulos da imagem restaurada (à direita) são basicamente os mesmos da imagem original.	28
Figura 11	– Estrutura da <i>Dense Chained Attention Network</i> .	29
Figura 12	– Exemplo de imagens que dificultam a detecção e o reconhecimento de textos.	30
Figura 13	– Diagrama dos passos metodológicos.	31
Figura 14	– Esquema de ambiente de execução dos experimentos.	33
Figura 15	– Exemplo contido no arquivo de saída para submissão na Tarefa 2 do IC-DAR2017 <i>Robust Reading Challenge on COCO-Text</i> , onde o valor à esquerda da vírgula é o número identificador da imagem e o valor à direita da vírgula, é a transcrição do texto. Os demais resultados estão apresentados no Anexo A.	35
Figura 16	– Gráfico das melhores colocações na competição sob a análise do <i>Correctly Recognize Words (CRW)</i> .	36

Figura 17 – Gráfico das melhores colocações na competição sob a análise do <i>Total Edit Distance</i> (TED)	37
Figura 18 – Observando as imagens e seus respectivos valores a baixo, é possível deduzir que o valor da estimativa é proporcional ao nível de ruído da imagem.	38
Figura 19 – A imagem da esquerda possui como <i>ground truth</i> panasonic e a API detecta panasonic. A imagem central possui como <i>ground truth</i> www.TheRoyalolive.com e a API detecta olive.com. Por fim, na imagem da direita o <i>ground truth</i> é Shore e a API não detecta conteúdo. Adicionalmente, há uma relação onde a quantidade de ruído é inversamente proporcional a quantidade de texto detectado.	39
Figura 20 – Novo esquema de ambiente de execução dos experimentos para análise com remoção de ruídos.	40
Figura 21 – Assinatura do método de remoção de ruídos.	41
Figura 22 – Gráfico das melhores colocações na competição sob a análise do CRW após remoção de ruídos.	43
Figura 23 – Gráfico das melhores colocações na competição sob a análise do TED após remoção de ruídos.	43

LISTA DE QUADROS

Quadro 1 – Comparativos entre os trabalhos relacionados e o trabalho proposto.	30
Quadro 2 – Resultado da Tarefa 2 na competição ICDAR 2017.	36
Quadro 3 – Veja que a relação entre baixa ruidosidade e resultado na detecção do texto pela <i>Google Cloud Vision API</i> permanece e solidifica a análise de Hosseini <i>et al.</i> (2017).	39
Quadro 4 – Quantidade de respostas dadas pela API que possuem algum texto.	42
Quadro 5 – Resultado da Tarefa 2 na competição ICDAR 2017 após remoção de ruídos das imagens.	42

LISTA DE ABREVIATURAS E SIGLAS

API	<i>Application Programming Interface</i>
BLSTM	<i>Bidirectional Long Short-Term Memory</i>
CNN	<i>Convolutional Neural Network</i>
COCO	<i>Common Objects in Context</i>
COCO-Text	<i>Common Objects in Context Text</i>
CRW	<i>Correctly Recognize Words</i>
CTC	<i>Connectionist Temporal Classification</i>
CVDF	<i>Common Visual Data Foundation</i>
GCP	<i>Google Cloud Platform</i>
HTTP	<i>Hypertext Transfer Protocol</i>
IAPR	<i>International Association for Pattern Recognition</i>
ICDAR	<i>International Conference on Document Analysis and Recognition</i>
IoT	<i>Internet of Things</i>
OCR	<i>Optical Character Recognition</i>
RCNN	<i>Recurrent Convolutional Neural Network</i>
REST	<i>Representational State Transfer</i>
RN	Redes Neurais
RNA	Rede Natural Artificial
ROF	<i>Rudin, Osher and Fatemi</i>
SQL	<i>Structured Query Language</i>
TED	<i>Total Edit Distance</i>
UTF-8	<i>8-bit Unicode Transformation Format</i>

SUMÁRIO

1	INTRODUÇÃO	15
2	FUNDAMENTAÇÃO TEÓRICA	17
2.1	Aprendizado de Máquina	17
2.2	<i>Convolutional Neural Networks (Convolutional Neural Network (CNN))</i>	17
2.3	Distância de Edição	17
2.4	Palavras Reconhecidas Corretamente	18
2.5	Variação Total de Remoção de Ruído	18
2.6	<i>Microsoft Common Objects in Context (MS COCO)</i>	19
2.7	ICDAR2017 Robust Reading Challenge on COCO-Text	20
2.7.1	<i>Tarefas da Competição</i>	21
2.7.1.1	<i>Tarefa 2 : Reconhecimento de texto em contexto (Task 2: Cropped Word Recognition)</i>	22
2.8	<i>Google Cloud Vision API</i>	23
2.8.1	<i>Amostras de detecção de texto</i>	24
2.8.1.1	<i>Protocolo da API</i>	24
3	TRABALHOS RELACIONADOS	27
3.1	<i>Google's Cloud Vision API is Not Robust to Noise</i>	27
3.2	<i>Dense Chained Attention Network for Scene Text Recognition</i>	28
3.3	<i>What is wrong with scene text recognition model comparisons? dataset and model analysis</i>	29
3.4	Comparativos Entre Trabalhos	30
4	METODOLOGIA	31
4.1	Coleta de Dados	31
4.2	Pré-processamento das Imagens	31
4.3	Organização dos Dados Recebidos	32
4.4	Análise das Imagens e Análise dos Dados Recebidos	32
4.5	Submissão dos resultados	32
5	EXPERIMENTOS E RESULTADOS	33
5.1	Coleta do <i>dataset</i> da Tarefa 2	33
5.2	Processamento das imagens	34

5.3	Organização dos dados recebidos	34
5.4	Submissão dos Resultados	35
5.5	Análise da Estimativa de Presença de Ruídos nas Imagens do <i>dataset</i> . .	38
5.6	Remoção de Ruídos	40
5.7	Análise dos Resultados Pós Remoção de Ruídos	42
6	CONCLUSÃO	45
6.1	Dificuldades Encontradas	45
6.2	Trabalhos Futuros	45
	REFERÊNCIAS	46
	APÊNDICES	48
	APÊNDICE A – Submissão de Resultados Sem Segmentação dos Dados .	48
	APÊNDICE B – Submissão de Resultados Com Segmentação dos Dados .	51

1 INTRODUÇÃO

Os problemas de reconhecimento de textos acompanham pesquisadores da área de visão computacional há anos. Em Mori *et al.* (1992), escrito há décadas, já tínhamos uma série de análises a respeito de técnicas voltadas à *Optical Character Recognition* (OCR), evoluções, dificuldades em criar uma solução que efetivamente faça um reconhecimento eficiente de textos.

Nos últimos anos, essas técnicas vem sendo aprimoradas com foco e grande atenção em imagens naturais, ou comumente chamadas de imagens do mundo real. Pode-se definir imagens do mundo real como as que imagens que são capturadas de radares para reconhecimento de placas de carro, do conteúdo de placas de trânsito, fachadas de lojas e placares de um jogo de futebol, por exemplo.

Mesmo que haja uma similaridade entre reconhecimento óptico de caracteres e reconhecimento de textos em cenas, essa última é uma abordagem muito desafiadora devido aos vários fatores que tornam as imagens, o objeto de análise, pouco padronizadas, como nível de detalhe, orientação da imagem, desfoque, iluminação, etc. O que é altamente contrastante com o ambiente utilizando técnicas de OCR, que em geral, são aplicadas a documentos claros, limpos e capturados por ferramentas de escaneamento de páginas Weinman *et al.* (2009).

Com a crescente procura e pesquisa a respeito desse tema, notou-se o surgimento de conferências voltadas para a solução e incremento de resultados sobre esses problemas, uma delas, a *International Conference on Document Analysis and Recognition* (ICDAR) apoiada pela *International Association for Pattern Recognition* (IAPR), também conhecida como a décima quarta IAPR, uma importantíssima conferência bienal que premia pesquisadores, entusiastas e cientistas em análise de documentos de todo o mundo. A última edição da ICDAR foi realizada em 2017, na cidade de Kyoto, Japão (ICDAR, 2017). A conferência além de reunir grandes nomes da comunidade de pesquisa em reconhecimento de textos, também propõe desafios para inflar e motivar a pesquisa nessa área.

O uso de técnicas de reconhecimento de textos em imagens naturais não se restringem a apenas entusiastas, mas também a gigantes do mercado como a API de Visão Computacional da *Microsoft Cognitive Services*¹, *Amazon Rekognition*² e *Google Cloud Vision API*³. Essa última, alvo desse trabalho de análise.

¹ <<https://azure.microsoft.com/pt-br/services/cognitive-services>>

² <<https://aws.amazon.com/pt/rekognition>>

³ <<https://cloud.google.com/vision>>

O objetivo deste trabalho é analisar o comportamento da *Google Cloud Vision API* na sua ferramenta de detecção de textos utilizando imagens naturais e avaliando sua acurácia (o quociente entre a quantidade de palavras extraídas e reconhecidas corretamente de uma imagem e o total de imagens analisadas) por meio dos resultados das competições do ICDAR.

Este trabalho está organizado em Capítulos. O Capítulo 2 diz respeito à fundamentação teórica, com um conjunto de estudos que são base e impulsionam a realização desse trabalho. O Capítulo 3, são descritos os trabalhos que serviram de embasamento teórico para fundamentar o este trabalho. O Capítulo 4 aborda os métodos de coleta de dados e procedimentos metodológicos. Por fim, no Capítulo 5, tem-se a conclusão deste trabalho, seus resultados que foram obtidos pela execução dos procedimentos metodológicos.

2 FUNDAMENTAÇÃO TEÓRICA

Nesse Capítulo, serão listados os principais conceitos pertencentes ao trabalho e que auxiliarão na compreensão da abordagem tratada.

2.1 Aprendizado de Máquina

Aprendizado de Máquina é uma área pertencente ao campo de estudo de Inteligência Artificial que, por meio de algoritmos, é capaz de fazer com que um sistema computacional aprenda algo. Arthur Samuel, o pioneiro do Aprendizado de Máquina, a descreve como: “O campo de estudo que dá aos computadores a capacidade de aprender sem serem explicitamente programados” (WIEDERHOLD; MCCARTHY, 1992).

Uma definição mais formal é que o Aprendizado de Máquina é: “Um programa de computador aprende a partir de uma experiência E com respeito a alguma classe de tarefas T medida por uma performance P , se essa performance na tarefa T medida por P , melhora com a experiência E .” (MITCHELL, 1997)

2.2 Convolutional Neural Networks (CNN)

Em Krizhevsky *et al.* (2012), tem-se o questionamento a respeito da detecção e reconhecimento de objetos e que é inevitável não se deparar com uma quantidade enorme de dados para se avaliar, treinar, manipular e testar. Automaticamente, para que haja um aprendizado eficiente sobre uma quantidade grande de objetos, precisa-se de um modelo robusto que tenha a capacidade de atender a essa coleção de dados. No entanto, a imensa complexidade dessa tarefa de reconhecimento de objetos implica num problema que não pode ser especificado nem mesmo por um conjunto de dados gigantesco. É necessário um modelo que também possua lotes de “conhecimento prévio” para compensar todos os dados que não se tem. É partindo dessa necessidade que foi criada as CNNs, que constituem uma dessas classes de modelos. As CNNs podem, em sua capacidade, fazer inferências com maior precisão e facilidade de treinamento.

2.3 Distância de Edição

Distância de edição ou Distância de *Levenshtein* é uma métrica criada em 1965 pelo matemático russo *Vladimir Iossifowitsch Levenshtein* (WIKIPÉDIA, 2019). O algoritmo

em questão resolve o problema de calcular quantas operações de edição são necessárias para transformar uma palavra em outra. A fim de se transformar uma palavra $x[1..m]$ numa $y[1..n]$, precisa-se aplicar zero ou várias operações de transformações. Dada as palavras x e y , o objetivo é a partir da aplicação dessas transformações, x seja idêntica a y e tenha o custo mínimo de operações (CORMEN *et al.*, 2009). A Figura 1 traz um exemplo de como funciona o algoritmo de cálculo da distância de edição proposto por *Levenshtein*.

Figura 1 – O custo total de edição de ‘SNOWY’ em ‘SUNNY’ é igual a 3. Uma inserção da letra ‘U’, substituição da letra ‘O’ pela letra ‘N’ e remoção da letra ‘W’.

1	S	_	N	O	W	Y
2	S	U	N	N	_	Y

Fonte: Dasgupta *et al.* (2009)

2.4 Palavras Reconhecidas Corretamente

Em Cremers *et al.* (2015), uma palavra é dita reconhecida corretamente (CRW) se todos os caracteres são reconhecidos corretamente, usando comparação sensível à maiúsculas. Neste trabalho, esse conceito é fundamental para o entendimento dos resultados na ICDAR2017 *Robust Reading Challenge on COCO-Text*.

2.5 Variação Total de Remoção de Ruído

“Os modelos de variação total alcançaram um grande sucesso no processamento de imagens. Eles têm sido usados em muitas aplicações como restauração de imagem, quebra de imagem e *inpainting* (processo de reconstrução de partes perdidas ou deterioradas de imagens e vídeos)¹. A restauração de imagens, especialmente para a remoção ruídos, forma um etapa preliminar significativa em muitas tarefas de visão de máquina como detecção e reconhecimento de objetos. O modelo típico de restauração de imagens foi introduzido pela primeira vez por Leonid I. Rudin, Stanley Osher e Emad Fatemi. O modelo criado por eles, conhecido como Rudin, Osher and Fatemi (ROF) ((RUDIN *et al.*, 1992)) pode preservar descontinuidades nítidas ou bordas em uma imagem enquanto remove o ruído” (YU *et al.*, 2009).

¹ <<https://en.wikipedia.org/wiki/Inpainting>>

2.6 Microsoft Common Objects in Context (MS COCO)

Para maior entendimento deste trabalho, é importante entender a organização de um conjunto de dados bastante utilizado por pesquisadores que se desafiam a compreender e resolver problemas que envolvem visão computacional, análise de dados, etc. Em (LIN *et al.*, 2014), temos a descrição do *Microsoft COCO*, um conjunto de dados de larga escala para detecção, segmentação e rotulagem finalizado em meados de 2014 e patrocinado por gigantes do mercado, como: a *Common Visual Data Foundation (CVDF)* ², *Mighty Ai* ³, *Facebook* ⁴ e claro, a *Microsoft* ⁵. Esse *dataset* possui uma quantidade grande de classes, características e anotações que são mencionadas na listagem a baixo:

- O *dataset* possui **segmentação de objetos**, ou seja, todos os objetos que se apresentam nas imagens do conjunto de dados são localizados pelo seu contorno ou fronteira, como curvas, linhas e polígonos;
- **Reconhecimento em contexto**, que pode ser definido como a detecção de um objeto não isolado numa imagem;
- **Segmentação com *superpixels*** ⁶. Um *superpixel* é um agrupamento de *pixels* semelhantes usados na discretização de imagens e busca viabilizar a manipulação de imagens de alta resolução;
- Mais de 330 mil imagens, com mais de 200 mil rótulos;
- 1.5 milhão de instâncias de objeto;
- 80 categorias de objetos;
- 91 categorias de coisas (*stuff categories*);
- 5 anotações *captions* por imagem;
- Pontos chave (*keypoints*) de imagens de 250.000 pessoas. Neste contexto, um *keypoint* pode ser entendido como um ponto de articulação do corpo humano, como o ombro, joelho, cotovelo, etc.

² <www.cvd.foundation.org>

³ <<https://mighty.ai>>

⁴ <<https://www.facebook.com>>

⁵ <<https://www.microsoft.com>>

⁶ <https://cran.r-project.org/web/packages/OpenImageR/vignettes/Image_segmentation_superpixels_clustering.html>

2.7 ICDAR2017 Robust Reading Challenge on COCO-Text

Uma das competições que fizeram parte do evento ICDAR 2017, foi a *Robust Reading Challenge on COCO-Text*, um desafio que tem como ambiente e material, imagens naturais de textos (textos em placas, camisetas, veículos, casas, etc). Este desafio aborda a necessidade de quantificar e acompanhar o progresso no domínio da compreensão do texto em ambientes irrestritos focado em textos da língua Inglesa, descrito em Gomez *et al.* (2017). O desafio é organizado dentro do contexto da Competição de Leitura Robusta (*Robust Reading Competition*). “Pode-se entender Leitura Robusta como a área de pesquisa que lida com a interpretação da comunicação escrita em contextos irrestritos. Tipicamente, a leitura robusta está vinculada à detecção e reconhecimento de informações textuais em imagens naturais e a técnicas e metodologias desenvolvidas especificamente para problemas que não sejam documentos em papel digitalizados e incluem imagens e vídeos nascidos digitalmente, também conhecido como imagens sintéticas”. (ICDAR2019, 2019) Essa competição está organizada da seguinte forma:

- localização de palavras;
- reconhecimento de palavras (alvo deste trabalho);
- localização e reconhecimento de palavras (reconhecimento *end-to-end*).

Este desafio fornece um conjunto de dados para as práticas de pesquisa para a solução do desafio, o COCO-Text ⁷, um *dataset* derivado do conjunto de dados em larga escala *Common Objects in Context* (COCO) que envolve apenas dados úteis para detecção e reconhecimento de textos, que reúne as seguintes características (COCO, 2019):

- localização dos textos (*bounding boxes*);
- transcrições dos textos legíveis;
- presença de múltiplas instâncias de texto por imagem;
- 63.686 imagens;
- 145.859 textos;
- categorização do texto como: manuscrito e impresso;
- categorização do texto como: legível e ilegível;
- categorização do texto como: palavra no idioma inglês e não pertencente ao idioma inglês.

A competição funciona pela submissão de resultados que os competidores produzem sobre os dados de testes e não os modelos, programas ou códigos propriamente ditos.

⁷ <<https://vision.cornell.edu/se3/coco-text-2>>

2.7.1 Tarefas da Competição

O *Robust Reading Challenge on COCO-Text* organizado em 2017, foi dividido em três tarefas onde os participantes poderiam escolher quais e quantas participariam. A enumeração seguinte descreve como as três tarefas foram organizadas.

- Localização de textos (*Text localization*): Tinha como objetivo obter a área estimada que o texto ocupa numa determinada imagem em termos de *bounding boxes* que correspondem as palavras encontradas;
- Reconhecimento de palavras (*Cropped word recognition*): Detectar e prover o conteúdo em termos de texto (transcrição) dentro dos limites de localizados numa imagem. Essa tarefa é o alvo da análise neste trabalho.
- Reconhecimento ponto-a-ponto (*End-to-end text*): Localizar e reconhecer e prover a transcrição dos textos numa imagem. Basicamente, a combinação das duas tarefas anteriores num único passo.

Na Figura 2, pode-se perceber os retângulos em verde (no centro do monitor do computador, e alguns sobre o *mousepad*) que representam a localização ou limites (*bounding-box*) da palavra detectada na imagem. Os retângulos em laranja (do lado esquerdo, na parte inferior, e do lado direito da imagem), tem as informações a respeito do conteúdo detectado que são conhecidas como ‘anotações’.

A anotação sobre cada um desses *bounding-boxes* possui a seguinte estrutura, como mostra a Figura 3:

As linhas 1 e 4 representam o identificador do *bounding box*, que tem como limites os pontos 150 e 66 que formam a coordenada do superior esquerdo do *bounding box*. Os valores 123.2 e 37.0 representam respectivamente a largura e a altura do *bounding box*. Na linha 3, o atributo “*class*” pode assumir os seguintes valores: “*machine printed*”, “*handwritten*” ou “*others*”, neste caso, o texto identificado é proveniente de uma máquina.

Já na linha 5, tem-se o id da imagem fonte, neste caso a imagem `COCO_train2014_000000286120.jpg` é a utilizada neste exemplo. A linha 6 traz informações sobre o idioma em que o texto foi escrito. As opções de idioma do texto pode ser: “*english*”, “*not english*” ou “*na*” quando se apresenta em outro idioma. A linha 7 define o conteúdo de acordo com sua legibilidade e por fim, na linha 8, o conteúdo de fato, ou transcrição.

Figura 2 – Exemplo das informações apresentadas numa imagem do *dataset* COCO-Text.



Fonte: ICDAR2017 (2017).

Figura 3 – Estrutura da anotação de uma imagem do *dataset* COCO-Text

```

1  "1027347":{
2  "bbox": [150,66,123.2,37.0],
3  "class": "machine printed",
4  "id": 1027347,
5  "image_id": 286120,
6  "language": "english",
7  "legibility": "ligible",
8  "utf8_string": "flickr"
9  }

```

Fonte: ICDAR2017 (2017).

2.7.1.1 Tarefa 2 : Reconhecimento de texto em contexto (Task 2: Cropped Word Recognition)

A Tarefa 2, tem como objetivo reconhecer o conteúdo de imagens de texto. Cada uma das imagens possui um texto como seu conteúdo e dois *pixels* de borda que contornam o conteúdo textual. Como resultado do processamento da imagem, apenas palavras da língua inglesa com mais de três caracteres são consideradas nesta tarefa independente se contém símbolos ou caracteres especiais.

Um exemplo de imagem contida no conjunto de dados da Tarefa 2 é apresentada na Figura 4.

Os resultados do reconhecimento de todas as imagens de teste deverão ser salvas

Figura 4 – Imagem 1008258 .jpg do conjunto de dados da Tarefa 2.



Fonte: ICDAR2017 (2017).

em um único arquivo de texto com extensão .txt na codificação *8-bit Unicode Transformation Format* (UTF-8) com a seguinte estrutura apresentada na Figura 5.

Figura 5 – Exemplo de linhas de resultado da Tarefa 2 no formato configurado para submissão.

```
1 1034977,http://www.thewebsite.com
2 1034977,Cake s
3 1199294,1993
4 1199294,dog
```

Fonte: ICDAR2017 (2017).

O primeiro parâmetro, é o identificador (id) da imagem, o segundo é o conteúdo da textual da imagem. A métrica de avaliação dos resultados é dada pela distância de edição entre a palavra predita e a real.

2.8 Google Cloud Vision API

Para que os conceitos da *Google Cloud Vision* sejam entendidos, é interessante saber sobre a *Google Cloud Platform* (GCP), um aglomerado de sistemas computacionais independente do seu nível de virtualização. A adicionalmente, a GCP possui um conjunto de serviços e ferramentas voltadas para uma série de áreas no ramo da computação, como: *Internet of Things* (IoT), *Structured Query Language* (SQL), armazenamento em nuvem, *Big Data*, aprendizado de máquina voltado à tradução, recomendação e visão computacional, por exemplo.

Essa última, nomeada de *Google Cloud Vision* oferece modelos baseados em Redes Neurais (RN) pré-treinadas por meio de uma API, ferramenta utilizada neste trabalho ou a capacita o usuário criar modelos personalizados usando o *AutoML Vision*. Assim, tem-se a

flexibilidade de acordo com as necessidades do usuário, dependendo do seu caso de uso. A *Google Cloud Vision* é usada principalmente para classificação de imagens de acordo com o seu conteúdo, de objetos, análise facial e textos.

“A API do Cloud Vision encapsula modelos avançados de *machine learning* em uma API *Representational State Transfer* (REST) fácil de usar, o que permite aos desenvolvedores entender o conteúdo de imagens. Essa API rapidamente classifica as imagens em milhares de categorias (por exemplo: ‘veleiro’), detecta objetos e rostos individuais e extrai palavras impressas contidas nas imagens” (GOOGLE, 2019b).

2.8.1 Amostras de detecção de texto

“O reconhecimento óptico de caracteres é feito com a detecção de texto. Com esse recurso, é possível detectar o texto em uma imagem e extraí-lo de maneira que seja compatível com uma ampla variedade de idiomas. Também é possível fazer a identificação automática de idioma.” (GOOGLE, 2019a)

2.8.1.1 Protocolo da API

Como mencionado, a *Google Cloud Vision* provê um serviço REST com funcionalidades bem definidas em seus verbos. Para utilizar as funcionalidades da API, é necessário possuir uma API KEY, um identificador de usuário permitido para uso da plataforma. Para executar a "Detecção de texto", basta fazer uma solicitação POST via *Hypertext Transfer Protocol* (HTTP) descrito na Figura 6 fornecendo a API KEY como *token OAuth*⁸ de acesso seguido do o corpo da solicitação apropriado que seguem explicados no texto.

Figura 6 – Exemplo do *endpoint* (ponto de extremidade) compatível com o método de solicitação HTTP.

```
1 POST https://vision.googleapis.com/v1/images:annotate?key=YOUR_API_KEY
```

Fonte: Google (2019a)

Na Figura 7, temos um exemplo da estrutura de como é organizada a requisição da detecção de texto feita pela *Google Cloud Vision API* que tem como parâmetro, a imagem. A assinatura da requisição está descrita na Figura 7.

⁸ <<https://oauth.net>>

Figura 7 – Estrutura da solicitação de detecção de texto da *Google Cloud Vision API*.

```

1  {
2  "requests": [
3  {
4  "image": {
5  "content": "1008258.jpg"},
6  "features": [{"type": "TEXT_DETECTION"}]
7  }
8  ]}

```

Fonte: Google (2019a)

Note que na linha 5 na Figura 7 o atributo `content` recebe o endereço da imagem armazenada no mesmo diretório do código de requisição que deve ser codificada na base 64⁹.

Após o processamento feito pela *Google Cloud Vision API*, é dada como resposta, uma série de informações a respeito da imagem enviada, uma das informações mais relevantes para esse trabalho é a transcrição do texto e a localização do conteúdo textual na imagem como mostra a Figura 8. O trecho da resposta da *Google Cloud Vision API* apresenta o conteúdo transcrito da imagem na linha 5, quanto os vértices que compõe o *bounding box* (limite) em que o texto na imagem se apresenta e valorados nas linhas 8, 9, 10 e 11. O atributo `text` (todo o conteúdo textual da imagem), atributo filho de `fullTextAnnotation`, também possui o conteúdo transcrito da imagem, que pode ser encontrado na linha 17 da Figura 8.

Figura 8 – Estrutura da resposta de detecção de texto da *Google Cloud Vision API*.

```

1  {
2  "textAnnotations": [
3  {
4  "locale": "en",
5  "description": "STOP\n",
6  "boundingPoly": {
7  "vertices": [
8  { "x": 4, "y": 5},
9  { "x": 86, "y": 9},
10 { "x": 84, "y": 50},
11 { "x": 2, "y": 46}
12 ] } },
13 ...
14 "fullTextAnnotation": {[
15 ...
16 ],
17 "text": "STOP\n"
18 }
19 ]}

```

Fonte: Google (2019a)

Pode-se reunir num único conceito, as informações das Figuras 8 e 4 na Figura 9. É possível enxergar o *bounding box* (discretizado numa cor mais clara) que delimita a palavra

⁹ <pt.wikipedia.org/wiki/Base64>

STOP e perceber o comportamento dela em relação ao conteúdo que nem sempre está inserido totalmente na fronteira fornecida.

Figura 9 – Descrição gráfica do *bounding box* e as anotações detectadas pela *Google Vision API*.



Fonte: Elaborada pelo Autor.

3 TRABALHOS RELACIONADOS

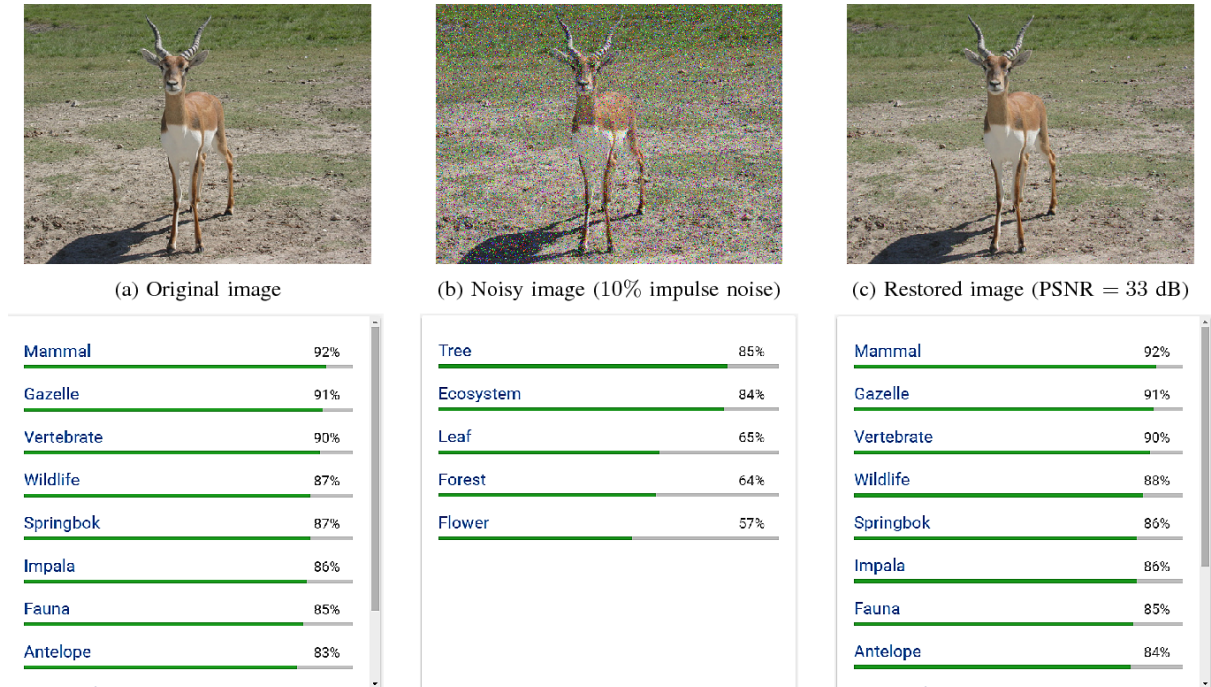
Neste Capítulo, seguem descrições breves dos trabalhos que fornecem embasamento teórico ao trabalho corrente, fazendo uma relação entre eles e como as técnicas se portam.

3.1 *Google's Cloud Vision API is Not Robust to Noise*

No trabalho de Hosseini *et al.* (2017), foi feita a análise de modelos de Aprendizagem de Máquina em ambientes com ruídos ou perturbações. Imagens proveniente de sensores, sofrem de influências como desfoque, ruídos e outras deformações. Isso implica que o design de modelos de visão computacional seja robusto, por exemplo, num ambiente como diagnóstico médico e condução autônoma de automóveis.

Além destas informações, recentes pesquisas apontam vulnerabilidades de modelos de Aprendizado de Máquina em ambientes reais ou receptores de várias influências externas. Adicionalmente, segurança nas avaliações de sistemas de Aprendizado de Máquina é uma área emergente de estudo e esses modelos vêm sendo desafiados por vários pesquisadores, escritores e entusiastas da área como modelos de detecção de voz, faces e texto em vídeo. Esse trabalho, avalia a acurácia da *Google Cloud Vision API* para entradas com ruídos, onde existe a possibilidade do ser humano perceber que no conteúdo da imagem existe um animal enquanto a API produz um resultado diferente (árvore) do previsto como é ilustrado na Figura 10.

Figura 10 – Capturas de tela dos rótulos retornados pela *Google Cloud Vision API* imagens originais, ruidosas e restauradas. A imagem original (à esquerda) é pertencente ao *ImageNet*. Nenhuma das etiquetas retornadas para a imagem ruidosa (ao centro) está relacionada aos rótulos da imagem original, enquanto os rótulos da imagem restaurada (à direita) são basicamente os mesmos da imagem original.



Fonte: Hosseini *et al.* (2017).

3.2 Dense Chained Attention Network for Scene Text Recognition

Já no trabalho de Gao *et al.* (2018), é possível entender que o reconhecimento de textos em imagens naturais, continua sendo um desafio atual e que métodos de reconhecimento desses textos quando se usam técnicas *offline* utilizando *Recurrent Convolutional Neural Network* (RCNN), possuem bons resultados mesmo enfrentando problemas como foco da imagem no texto propriamente dito.

Vários conjuntos de dados públicos são usados para a avaliação, incluindo o *Street View Text*¹, *IIIT5K*², *ICDAR 2003* e *ICDAR 2013*. O conjunto de treino desse trabalho foi treinado sob um conjunto puramente composto por dados sintéticos, sem quaisquer manipulações desses dados. O conjunto também obedece a restrição de possuir apenas palavras formadas por mais de três caracteres alfanuméricos.

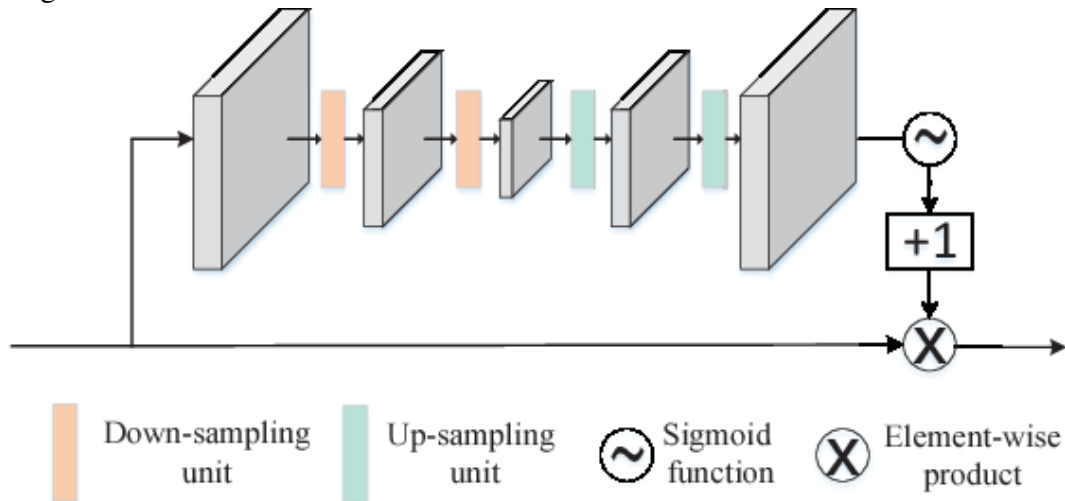
O modelo proposto trata-se de uma Rede Natural Artificial (RNA). Nesse trabalho,

¹ <http://www.iapr-tc11.org/mediawiki/index.php/The_Street_View_Text_Dataset>

² <<http://cvit.iiit.ac.in/projects/SceneTextUnderstanding/IIIT5K.html>>

chamada de *Dense Chained Attention Network*, tem sua estrutura ilustrada na Figura 11.

Figura 11 – Estrutura da *Dense Chained Attention Network*.



Fonte: Gao *et al.* (2018).

Os módulos de atenção empilhada extraem uma representação robusta de recursos para toda a imagem. Em seguida, os mapas de recursos são convertidos em uma representação sequencial. Especificamente, o mapa de características tridimensionais é cortado em mapas 2D ao longo de sua largura e, em seguida, cada mapa é aplainado em um vetor. Desta forma, cada elemento na sequência de recursos corresponde à uma região local da palavra na imagem e pode ser visto como a representação característica da região. Então a *Bidirectional Long Short-Term Memory* (BLSTM) aprende a informação do contexto e modela as dependências sequenciais dentro da sequência de recursos. Em seguida, a sequência gerada é transformada em sequência de distribuições de probabilidade. Finalmente, a *Connectionist Temporal Classification* (CTC) converte as distribuições de probabilidade no texto apresentado na imagem.

3.3 *What is wrong with scene text recognition model comparisons? dataset and model analysis*

Em Baek *et al.* (2019), também foram obtidos significantes resultados tanto no desenvolvimento de um modelo que faça o reconhecimento efetivo de textos em imagens do mundo real quanto nos bons resultados em competições de leitura robusta de textos em imagens naturais de nível internacional. Além disso, esse trabalho faz descrição de uma série de causas como: imagens sintéticas, com ruído, desfoque, regularidade e orientação como mostra a Figura 12, fatores que reduzem o valor da acurácia dos modelos de aprendizado de máquina baseado em

CNN que se propõem a resolver essa questão.

Figura 12 – Exemplo de imagens que dificultam a detecção e o reconhecimento de textos.



Fonte: Baek *et al.* (2019)

a) Fontes complexas; b) Textos verticais; c) Caracteres especiais; d) Bloqueio; e) Baixa resolução; f) Ruído no rótulo.

3.4 Comparativos Entre Trabalhos

No Quadro 1 é possível identificar as diferenças e semelhanças entre os trabalhos relacionados e o proposto. É visível dentre os trabalhos relacionados uma preocupação tanto em resolver problemas com detecção seja de objetos ou textos numa imagem, quanto tratar as inevitáveis alterações qualitativas das imagens trabalhadas. Que algumas alternativas para tentar reduzir a quantidade de interferências no mundo real é criar uma rede neural robusta, utilizar de imagens sintéticas (que geralmente possuem pouco ou nenhum ruído) ou até mesmo procurar entender como modelos de detecção funcionam quando se trabalha com imagens do mundo real.

Quadro 1 – Comparativos entre os trabalhos relacionados e o trabalho proposto.

	<i>Trabalho Proposto</i>	<i>Hosseini et al. (2017)</i>	<i>Gao et al. (2018)</i>	<i>Baek et al. (2019)</i>
Dataset	COCO-Text	ImageNet	SVT, IIIT5K, ICDAR ¹	SVT, IIIT5K, ICDAR ¹
Modelo de RN utilizado	GCV ² API (Text detection)	GCV ² API (Label detection)	CRNN	CNN
Aplicado à Textos	Sim	Não	Sim	Sim
Tipo de Dados	Não sintéticos	Não sintéticos	Sintéticos	Ambos
Segmentação de dados	Ambos	Sim	Não	Não

Fonte: Elaborado pelo Autor.

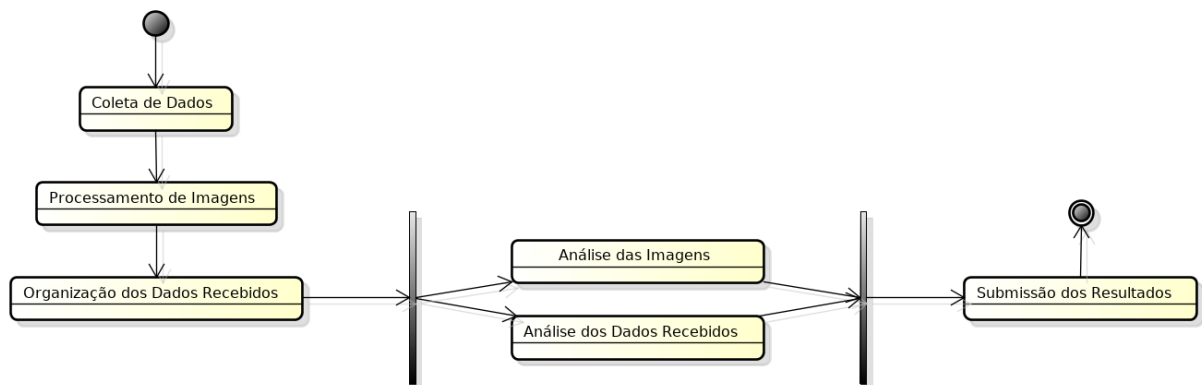
¹ IIIT5K-Words (IIIT), Street View Text (SVT), ICDAR2003 (IC03), ICDAR2013 (IC13), ICDAR2015 (IC15), SVT Perspective (SP), CUTE80 (CT);

² Google Cloud Vision.

4 METODOLOGIA

As Seções seguintes descrevem as metodologias, manipulação de dados, técnicas, procedimentos e estratégias de reconhecimento de textos manuscritos. Na Figura 13 é esboçado o diagrama que representam os passos a serem seguidos durante o processo de metodologia deste trabalho.

Figura 13 – Diagrama dos passos metodológicos.



Fonte: Elaborada pelo Autor.

4.1 Coleta de Dados

A etapa de coleta de dados se resume a adquirir um *dataset* que contenha dados correspondentes ao conjunto de dados para submissão na Tarefa 2 do desafio *ICDAR2017 Robust Reading Challenge on COCO-Text*. Esses dados estão disponíveis para *download* e podem ser acessados na página ¹ dessa competição, mediante a autenticação do usuário pela página ² do participante.

4.2 Pré-processamento das Imagens

A etapa de pré-processamento diz respeito a organização dos dados descarregados descritos na seção anterior e ao consumo desses dados que formam um conjunto de imagens utilizando a *Google Cloud Vision API* para detecção de textos em cada uma dessas imagens. Para isso, é proposto neste trabalho que uma aplicação seja implementada para que exista o vínculo entre as imagens que fazem parte desse conjunto criado para a participação na Tarefa 2 e a API

¹ <<https://rrc.cvc.uab.es/?ch=5&com=downloads>>

² <<https://rrc.cvc.uab.es/?com=contestant>>

da *Google*. Dessa forma, uma solução é a modelagem de um consumidor de serviços que vai enviar para a API, cada uma das imagens e um manipulador de respostas (descrita na Figura 8) da API.

4.3 Organização dos Dados Recebidos

Tendo o acesso a todas as repostas vindas da API, é indispensável que se analise as informações contidas nelas e que se extraia informações úteis, como o conteúdo do atributo `description` (descrito na Figura 8) de cada resposta do processamento a fim de organizá-las para a etapa de submissão de acordo com o formato exigido (Figura 5) pela organização.

4.4 Análise das Imagens e Análise dos Dados Recebidos

Após a organização dos resultados, avaliar, descobrir e interpretar relações entre entrada e saída no processamento de detecção de textos para que haja a possibilidade de enriquecer a solução.

4.5 Submissão dos resultados

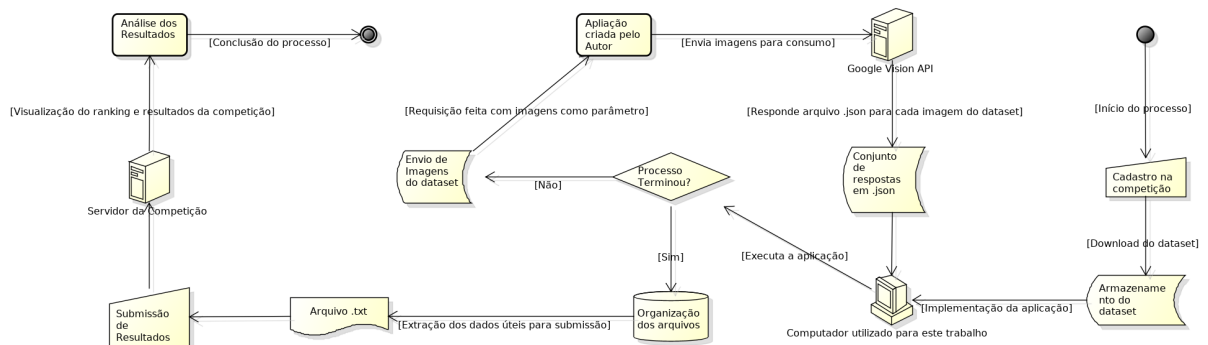
Por último, submeter os resultados recebidos, analisar as respostas dadas pelo sistema de avaliação da competição, avaliar as pontuações e colocações na competição de acordo com as métricas descritas nas Seções 2.3 e 2.4. Se necessário, realizar testes e experimentos que melhorem os resultados.

5 EXPERIMENTOS E RESULTADOS

Este Capítulo objetiva descrever os procedimentos e conjunto de técnicas, métodos adotados e os resultados obtidos em cada procedimento.

A Figura 14 ilustra o ambiente de execução dos experimentos adotados neste trabalho em forma de fluxograma.

Figura 14 – Esquema de ambiente de execução dos experimentos.



Fonte: Elaborada pelo Autor.

5.1 Coleta do *dataset* da Tarefa 2

O processo de coleta de dados, corresponde à primeira etapa da execução deste trabalho. Os dados foram obtidos por meio de *download* e são provenientes dos conjuntos de dados de teste na página ¹ da competição *ICDAR2017 Robust Reading Challenge on COCO-Text*. O *dataset* em questão, *Cropped words dataset* exclusivo para o trabalho da Tarefa 2, possui as seguintes características:

- Apenas as palavras legíveis e em inglês do *dataset* COCO-Text são utilizadas;
- Apenas palavras com mais de 3 caracteres compõe o *dataset*;
- Todas as imagens possuem 2 *pixels* de borda para os limites do texto na imagem;
- O formato esperado é um arquivo `.txt` com uma linha por palavra detectada em cada imagem, separada por `,` no formato: `nome_do_arquivo, transcrição`;
- As transcrições são no formado UTF-8;
- 9.837 imagens de teste. Imagens essas, que farão parte do conjunto de submissão para a competição.

Partindo do estado de início do processo ilustrado na Figura 14, após a realização

¹ <<https://rrc.cvc.uab.es/?ch=5&com=downloads>>

do *download* do conjunto de teste, o *dataset* é então armazenado no computador usado para a execução dos processos descritos na Seção 4.

5.2 Processamento das imagens

Todas as 9.837 imagens do conjunto de testes do *dataset* concedido para participação na Tarefa 2, foram parâmetros de processamento na Aplicação ² elaborada pelo Autor deste trabalho. A Aplicação consiste num *notebook jupyter*³ implementada na linguagem de programação *Python*⁴ que faz um percurso sobre todas as imagens do conjunto de testes para a submissão.

Para cada iteração sobre esses arquivos, uma requisição é feita enviando a imagem para que a API a processe. Todas as imagens foram enviadas para processamento utilizando a *Google Cloud Vision API*. A API em questão, dá uma resposta no formato *.json*, resposta essa que é recebida pela aplicação criada. Para cada uma das 9.837 imagens processadas, 9.837 arquivos no formato *.json* foram criados contendo a resposta da API. Esse procedimento é executado até o momento em que todas as imagens forem processadas e todas as respostas recebidas.

O resultado de cada processamento é armazenado num arquivo *.json* correspondente a cada imagem processada. Ao término do processo, uma nova iteração é feita sobre os arquivos *.json*, assim, extraíndo o conteúdo de cada uma dessas respostas, organizando-as num único arquivo *.txt* que será enviado à competição.

5.3 Organização dos dados recebidos

Com o término do consumo da API para todas as imagens, automaticamente, todas as respostas em formato *.json*, tem seus conteúdos avaliados e extraídos. Na Seção 2.8.1.1 viu-se que a imagem enviada para o consumo da API possui um nome, então, esse nome do arquivo formará a primeira parte da linha do arquivo de submissão, o que será feito para todas as 9.837 imagens do conjunto da Tarefa 2. Todas as repostas *.json* que possuem o atributo *textAnnotations* foram manipuladas de forma que o seu atributo filho *description* fizesse parte da segunda parte da linha que compõe o resultado para a submissão na Tarefa 2 do

² <<https://github.com/RonildoOliveira/Analise-Desempenho-GoogleCloud-Vision-API>>

³ <<https://jupyter.org/>>

⁴ <www.python.org>

ICDAR2017 *Robust Reading Challenge on COCO-Text*, caso contrário, essa segunda parte ficará vazia como mostra a Figura 15.

Figura 15 – Exemplo contido no arquivo de saída para submissão na Tarefa 2 do ICDAR2017 *Robust Reading Challenge on COCO-Text*, onde o valor à esquerda da vírgula é o número identificador da imagem e o valor à direita da vírgula, é a transcrição do texto. Os demais resultados estão apresentados no Anexo A.

```

1 1178021,grips
2 1074683,
3 1002777,PARKING
4 1028865,Reduced
5 1143729,
6 1232246,
7 1226971,CAFE
8 1160721,www.lhelova0ive.com

```

Fonte: Elaborada pelo Autor

5.4 Submissão dos Resultados

Após o processamento de todas as 9.837 imagens, captura e organização dos atributos extraídos, foi feita a submissão deles na página ⁵ de avaliação de resultados da Tarefa 2 na competição ICDAR 2017.

Como resultado, no Quadro 2, a solução utilizando a *Google Vision API* está em 10º lugar das melhores soluções em palavras reconhecidas corretamente (CRW (*case insensitive*)), exibido graficamente na Figura 16. O resultado que trata sobre a Distância Total de Edição (TED (*case insensitive*)) insensível a maiúsculas, que é justamente a soma de todas as distâncias de edição de cada transcrição predita em relação à transcrição esperada (*ground truth*). A solução ⁶ apresentada neste trabalho é a 13ª melhor no Quadro 2, como pode-se observar na Figura 17. Parte do documento que contém a listagem de palavras reconhecidas e seus respectivos identificadores, são apresentados Apêndice A.

⁵ <<https://rrc.cvc.uab.es/?ch=5&com=evaluation&task=2>>

⁶ <https://raw.githubusercontent.com/RonildoOliveira/Analise-Desempenho-GoogleCloud-Vision-API/master/resultados-apendice/37_submit_ch_5_task_2_file_1-GoogleVisionAPI.txt>

Quadro 2 – Resultado da Tarefa 2 na competição ICDAR 2017.

Date	Method	TED (C.S) ¹	CRW (C.S)	TED (C.I) ²	CRW (C.I)
2017-07-01	HIK_OCR	3,661.58	41.72%	899.10	76.11%
2017-06-30	Tencent-DPPR Team & USTB-PRIR	4,022.12	36.91%	1,233.46	70.83%
2019-02-25	CLOVA-AI	3,594.48	47.35%	1,583.77	69.27%
2018-12-19	SAR	4,002.36	41.27%	1,528.74	66.85%
2019-03-20	ustc_pr316	4,111.81	40.00%	1,615.44	65.35%
2017-06-30	HKU-VisionLab	3,921.94	40.17%	1,903.37	59.29%
2017-06-30	BRTRS-Recognition	4,895.96	28.18%	2,282.49	59.25%
2018-12-23	Airtel OCR - Readotron	4,892.62	31.64%	2,296.50	51.01%
2017-06-29	CCFLAB	4,743.28	26.52%	2,982.66	42.66%
2019-05-31	GoogleVisionAPI	6,164.12	24.83%	4,705.33	37.51%
2017-10-06	CRNN - Sravya	5,704.54	24.26%	3,532.96	36.98%
2017-06-30	3CNN_2BiLSTM_CTC	6,405.61	12.19%	4,395.42	30.17%
2017-06-30	Enhancing Text Recognition ³ .	7,231.87	17.88%	5,555.89	29.69%
2017-06-28	LSTM based text recognition	6,594.01	10.11%	4,638.83	26.25%

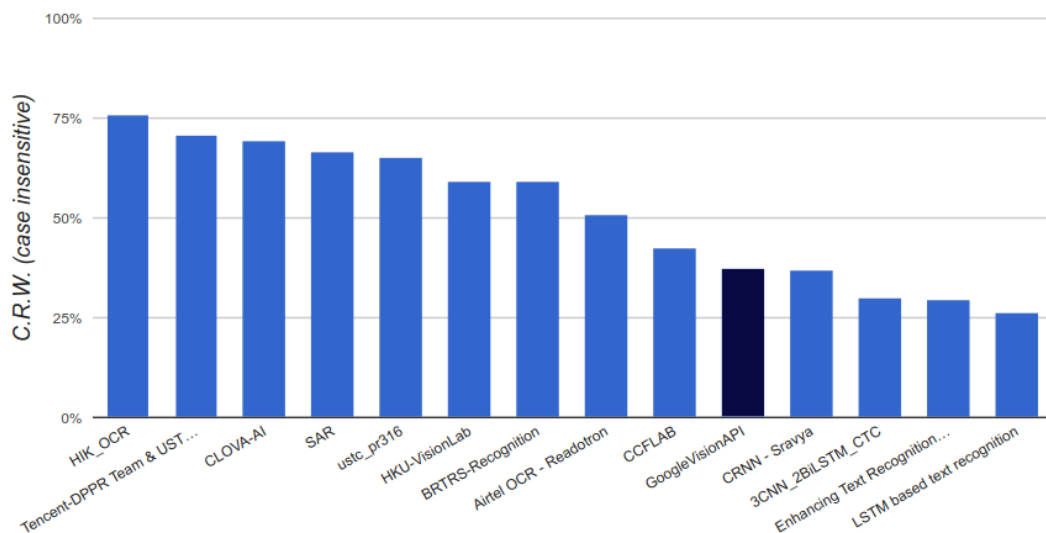
¹ C.S. (Case Sensitive);

² C.I. (Case Insensitive);

³ Enhancing Text Recognition Accuracy by Adding External Language Model).

Na Figura 16, tem-se um gráfico que traz algumas informações importantes do Quadro 2, quando se diz respeito a quantidade de palavras reconhecidas corretamente. Se compararmos esses dois conjuntos de dados, pode-se concluir que 37,51% das palavras do conjunto de dados, aproximadamente, 3690 palavras com mesmo valor do *ground truth* insensível à caracteres maiúsculos.

Figura 16 – Gráfico das melhores colocações na competição sob a análise do CRW

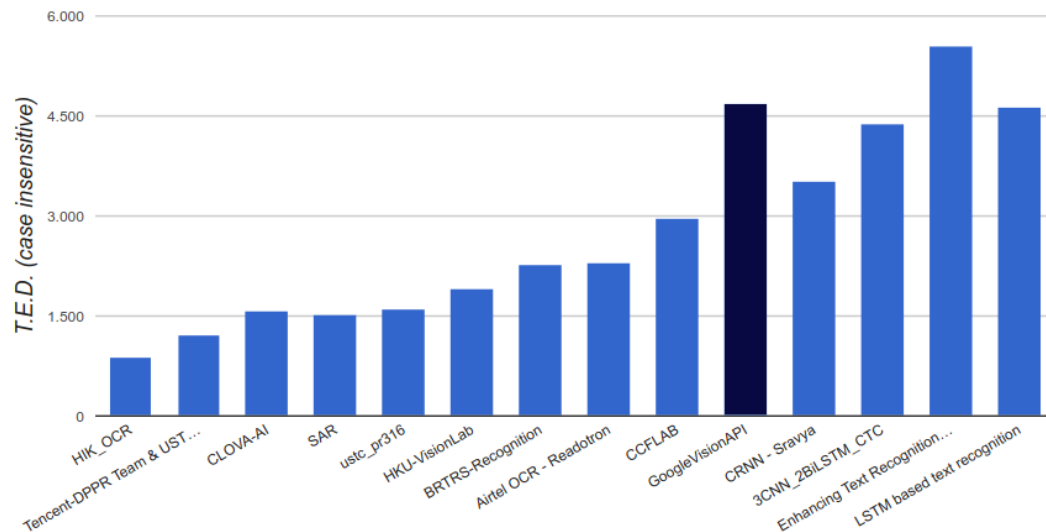


Fonte: ICDAR2017 (2019) (adaptada pelo Autor).

Similar à abordagem anterior, a Figura 17, traz uma organização entre a relação do total de distâncias de edição de todas as palavras do conjunto de dados. Também se apoiando no Quadro 2, a soma total de todas as médias de distâncias de edição é de aproximadamente

4.706, isto é, foram necessárias pouco mais de 4.700 operações (um pouco mais que 5 vezes mais operações que a melhor solução até o dia 31 de maio de 2019) para que todas as palavras preditas se tornassem idênticas as palavras que compõe o *ground truth*, mais uma vez, insensível à caracteres maiúsculos.

Figura 17 – Gráfico das melhores colocações na competição sob a análise do TED



Fonte: ICDAR2017 (2019) (adaptada pelo Autor).

5.5 Análise da Estimativa de Presença de Ruídos nas Imagens do *dataset*

As seções anteriores levantaram algumas discussões sobre o desempenho da *Google Cloud Vision API*, como o fato dos resultados na competição serem pouco satisfatórios. Em meio esses questionamentos, vem a necessidade de realizar experimentos que aborde a manipulação das imagens utilizadas no processo de submissão da solução para a competição e verificar os efeitos e resultados produzidos após essa segmentação. Para isso, utilizou-se da ferramenta gratuita, *scikit-image*⁷, uma coleção de algoritmos para processamento de imagens. Dentre esse algoritmos, do módulo *restoration*, tem-se a função nomeada como *estimate_sigma* que tem por funcionalidade estimar o desvio padrão de ruído numa imagem. Essa funcionalidade foi utilizada e seus resultados aplicados a duas imagens com níveis de ruído bastante distintos e ilustrada na Figura 18, que também traz seus respectivos valores para estimativa de ruídos.

Figura 18 – Observando as imagens e seus respectivos valores a baixo, é possível deduzir que o valor da estimativa é proporcional ao nível de ruído da imagem.



Ruidosidade: 19.593452664985904

Fonte: BoofCV (2019)



Ruidosidade: 0.7164722300562286

A partir dessa ferramenta, foi feita uma análise sob todas as imagens utilizadas na submissão do resultado para a Tarefa 2, tomando como dado, os seus respectivos valores de estimativa de nível de ruído. Interessa também saber qual a média de nível de ruído entre as respostas da API que possuem o atributo *description* e as que não possuem o atributo *description*. Essa análise levanta a questão de que possa haver uma relação entre um nível alto de ruído e uma baixa quantidade de caracteres reconhecidos pela API. O Quadro 3 mostra

⁷ <<https://scikit-image.org>>

essa relação a seguir.

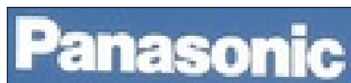
Quadro 3 – Veja que a relação entre baixa ruidosidade e resultado na detecção do texto pela *Google Cloud Vision API* permanece e solidifica a análise de Hosseini *et al.* (2017).

Texto detectado pela API?	Estimativa de ruidosidade média	Quantidade de imagens
Sim	8.189566086828773	5384
Não	8.850842266665568	4453

Fonte: Elaborada Pelo Autor.

É interessante observar que a média da estimativa de ruídos nas imagens que possuem conteúdo detectado pela API é menor que a média de estimativa das imagens em que texto algum foi detectado como é descrito no Quadro 3. Uma outra verificação é de visualização do que é predito pela *Google Cloud Vision API*. A Figura 19 mostra alguns resultados e suas respectivas comparações com o resultado previsto (*ground truth*) pela competição.

Figura 19 – A imagem da esquerda possui como *ground truth* panasonic e a API detecta panasonic. A imagem central possui como *ground truth* www.TheRoyalolive.com e a API detecta olive.com. Por fim, na imagem da direita o *ground truth* é Shore e a API não detecta conteúdo. Adicionalmente, há uma relação onde a quantidade de ruído é inversamente proporcional a quantidade de texto detectado.



Ground truth: panasonic
API: panasonic
Noise: 3.858348936568232



Ground truth: www.TheRoyalolive.com
API: olive.com
Noise: 10.193325244947049



Ground truth: Shore
API:
Noise: 11.192814854166935

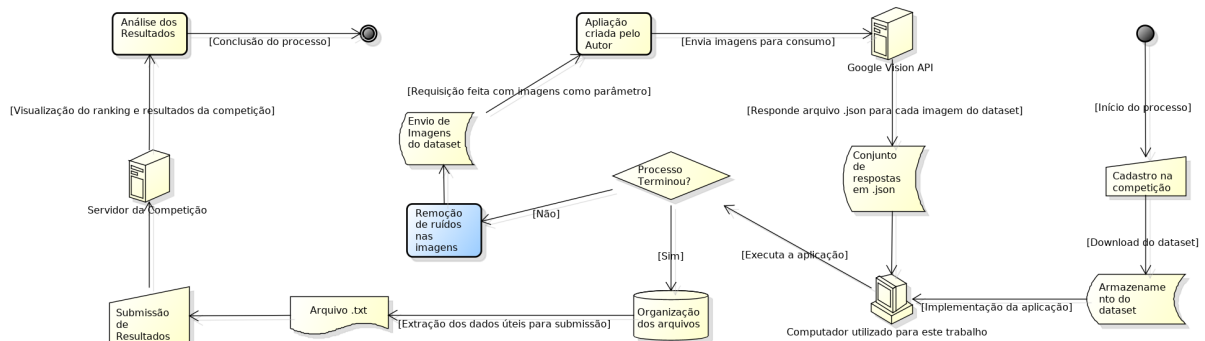
Fonte: Elaborado Pelo Autor.

Partindo dessa análise, torna-se obrigatória a verificação da quantidade de texto detectado quando temos um conjunto de dados que foi segmentado por um filtro de redução de ruídos e efetivamente, concluir a análise da acurácia da *Google Cloud Vision API*.

5.6 Remoção de Ruídos

Nesta etapa foi feita uma redução de ruídos nas imagens que não puderam ter seus textos transcritos pela API, que são exatas 4453 imagens como foi mostrado no Quadro 3. Após a aplicação da redução de ruídos em cada uma dessas imagens, um novo e positivo resultado foi gerado, 2688 das 4453 imagens, tiveram textos detectados e reconhecidos pela API. Daí vem a motivação tal como em Dasari *et al.* (2018) de que ao se aplicar um filtro de remoção de ruídos em imagens com características similares descritas na Figura 21, é possível aumentar de uma maneira simples a capacidade de se receber um resultado com conteúdo útil seja qual for a técnica voltada ao aprendizado de máquina. Assim, a execução dos experimentos recebe mais uma etapa que diz respeito à remoção do ruído nas imagens do conjunto de testes da Tarefa 2 como mostra a Figura 20 e uma nova etapa na Aplicação é adicionada. Todas as imagens passam por uma redução de ruído descrita na seção 2.5 e seguem novamente o percurso restante idêntico ao processo anterior.

Figura 20 – Novo esquema de ambiente de execução dos experimentos para análise com remoção de ruídos.



Fonte: Elaborada pelo Autor.

Na Figura 21 tem-se a descrição da função de restauração da biblioteca do *scikit-image*. Nela estão descritos os seguintes atributos:

Figura 21 – Assinatura do método de remoção de ruídos.

```
1  skimage.restoration.denoise_tv_chambolle(  
2  image,  
3  weight=0.5,  
4  eps=0.0002,  
5  n_iter_max=200,  
6  multichannel=False)
```

Fonte: Elaborada pelo Autor

- Na linha 2, *image* é o diretório onde a imagem a ter o filtro aplicado está armazenada;
- Na linha 3, *weight* é a largura da remoção do ruído que é diretamente proporcional ao nível de remoção de ruído;
- Na linha 4, *eps* é representa a diferença relativa do valor da função de custo que determina o critério de parada;
- Na linha 5, *n_iter_max* é o número de iterações usado para a otimização;
- Na linha 6, *multichannel*, aplica a remoção de ruído separadamente para cada canal. Se verdadeiro, deve ser usado para imagens coloridas, caso contrário a remoção de ruídos também é aplicada a todos os canais.

Já no Quadro 4, pode-se perceber que com a remoção de ruídos nas imagens, é possível ter um acréscimo na quantidade de textos detectados pela API, que implica numa menor distância total de edição.

Quadro 4 – Quantidade de respostas dadas pela API que possuem algum texto.

Texto Detectado pela API ?	Sem Segmentação (Número de imagens)	Com Remoção de Ruídos (Número de imagens)
Sim	5384 (54.73%)	8072 (82.06 %)
Não	4453 (45.27%)	1765 (17.94%)

Fonte: Elaborado pelo Autor.

5.7 Análise dos Resultados Pós Remoção de Ruídos

Esta etapa é responsável por descrever os resultados obtidos após a remoção de ruídos das imagens antes da consumição pela API da *Google*. Como resultado (organizados no Apêndice B), no Quadro 5, a solução utilizando a *Google Vision API* avança para a 10^a posição do *ranking*. Nesta análise, ocupa uma posição superior (10^a) quando se refere ao número de palavras reconhecidas corretamente (CRW (*case insensitive*)) como mostra a Figura 22. Já o resultado que trata sobre a Distância Total de Edição (TED (*case insensitive*)) aparece na 12^a melhor posição na Tarefa 2, assim, ilustrado na Figura 23.

Quadro 5 – Resultado da Tarefa 2 na competição ICDAR 2017 após remoção de ruídos das imagens.

Date	Method	TED (C.S) ¹	CRW (C.S)	TED (C.I) ²	CRW (C.I)
2019-06-14	SenseTime-CKD	4,068.9679	41.43%	845.1011	76.87%
2017-07-01	HIK_OCR	3,661.5785	41.72%	899.1009	76.11%
2017-06-30	Tencent-DPPR Team & USTB-PRIR	4,022.1224	36.91%	1,233.4609	70.83%
2019-02-25	CLOVA-AI	3,594.4842	47.35%	1,583.7724	69.27%
2018-12-19	SAR	4,002.3563	41.27%	1,528.7396	66.85%
2019-03-20	ustc_pr316	4,111.8119	40.00%	1,615.4420	65.35%
2017-06-30	HKU-VisionLab	3,921.9388	40.17%	1,903.3725	59.29%
2017-06-30	BRTRS-Recognition	4,895.9593	28.18%	2,282.4888	59.25%
2018-12-23	Airtel OCR - Readotron	4,892.6199	31.64%	2,296.4995	51.01%
2019-06-23	<i>GoogleVisionAPI</i>	<i>5,693.3547</i>	<i>27.61%</i>	<i>3,569.3361</i>	<i>43.01%</i>
2017-06-29	CCFLAB	4,743.2752	26.52%	2,982.6609	42.66%
2017-10-06	CRNN - Sravya	5,704.5379	24.26%	3,532.9616	36.98%
2017-06-30	3CNN_2BiLSTM_CTC	6,405.6129	12.19%	4,395.4174	30.17%
2017-06-30	Enhancing Text Recognition ³	7,231.8718	17.88%	5,555.8922	29.69%
2017-06-28	LSTM based text recognition	6,594.0069	10.11%	4,638.8345	26.25%

Fonte: Elaborado pelo Autor.

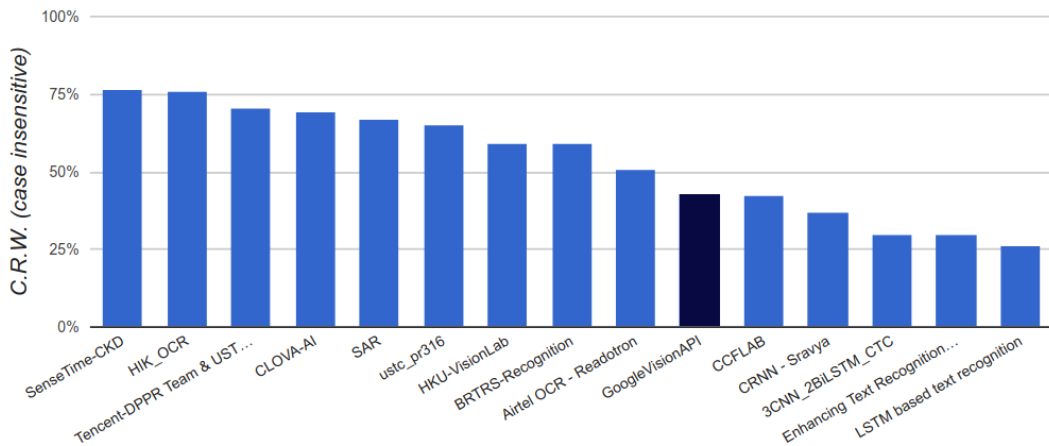
¹ C.S. (*Case Sensitive*);

² C.I. (*Case Insensitive*);

³ *Enhancing Text Recognition Accuracy by Adding External Language Model*).

A Figura 22, trata da nova quantidade de palavras reconhecidas corretamente. Se compararmos esses dois conjuntos de dados, pode-se concluir que 43.01% das palavras do conjunto de dados, aproximadamente, 4231 palavras com mesmo valor do *ground truth* insensível a caracteres maiúsculos.

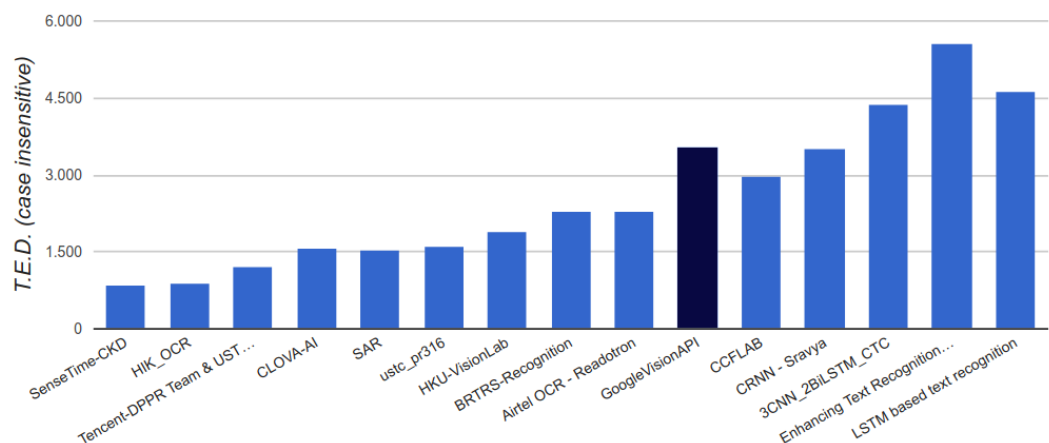
Figura 22 – Gráfico das melhores colocações na competição sob a análise do CRW após remoção de ruídos.



Fonte: ICDAR2017 (2019) (adaptada pelo Autor).

Na Figura 23, tem-se que a soma total de todas as médias de distâncias de edição é de aproximadamente 3.570, um número bem distante quando se compara com o melhor resultado (aproximadamente 846 operações) no *ICDAR2017 Robust Reading Challenge on COCO-Text*, isto é, foram necessárias pouco mais de 3.500 operações para que todas as palavras previstas se tornassem idênticas as palavras que compõe o *ground truth*, mais uma vez, insensível à caracteres maiúsculos.

Figura 23 – Gráfico das melhores colocações na competição sob a análise do TED após remoção de ruídos.



Fonte: ICDAR2017 (2019) (adaptada pelo Autor).

Após a remoção de ruídos, pode-se perceber tanto uma redução significativa na distância total de edição insensível à maiúsculas quanto um acréscimo na quantidade de palavras reconhecidas corretamente.

6 CONCLUSÃO

Foi apresentado neste trabalho uma análise da acurácia da *Google Cloud Vision API* para detecção de textos em imagens naturais e que ainda é conveniente o tratamento ou pré-processamento dessas imagens antes da extração de localização e reconhecimento, seja de objetos, pessoas, veículos ou textos. Pode-se também concluir que o estudo de aprendizado de máquina e profundo ainda tem espaço para crescer e perder as limitações que o mundo real possui.

6.1 Dificuldades Encontradas

A partir dos resultados encontrados no Capítulo 5, pode-se concluir que a estrutura da *Google Cloud Vision API* foi projetada para ter um tempo de resposta muito curto, o que inviabiliza, muitas vezes uma melhor extração de informações textuais de uma imagem, que muitas vezes leva um tempo maior, devido a complexidade do problema. Um outro ponto que pode ser destacado é que a RN *Google Cloud Vision API* não foi ajustada especificamente (ao contrário das demais técnicas que fizeram parte do *ranking* presente nos Quadros 2 e 5) para o conjunto de dados utilizado neste trabalho.

6.2 Trabalhos Futuros

Uma continuação interessante para este trabalho é continuar a análise dessas ferramentas que o mercado de tecnologia da informação oferece, como a *Amazon Rekognition* e *Microsoft Cognitive Services* que também tem soluções similares no ramo de Visão Computacional e Inteligência Artificial.

REFERÊNCIAS

- BAEK, J.; KIM, G.; LEE, J.; PARK, S.; HAN, D.; YUN, S.; OH, S. J.; LEE, H. What is wrong with scene text recognition model comparisons? dataset and model analysis. **arXiv preprint arXiv:1904.01906**, 2019.
- BOOFCV. **Example Wavelet Noise Removal - BoofCV**. 2019. Disponível em: https://boofcv.org/index.php?title=Example_Wavelet_Noise_Removal. Acesso em: 04 de maio de 2019.
- COCO. **COCO-Text Dataset for Text Detection and Recognition | SE(3) Computer Vision Group at Cornell Tech**. 2019. Disponível em: <https://vision.cornell.edu/se3/coco-text-2>. Acesso em: 08 de maio de 2019.
- CORMEN, T. H.; LEISERSON, C. E.; RIVEST, R. L.; STEIN, C. **Introduction to algorithms**. [S.l.]: MIT press, 2009.
- CREMERS, D.; REID, I.; SAITO, H.; YANG, M.-H. **Computer Vision—ACCV 2014: 12th Asian Conference on Computer Vision, Singapore, Singapore, November 1-5, 2014, Revised Selected Papers**. [S.l.]: Springer, 2015. v. 9007.
- DASARI, S. *et al.* **Character recognition of handwritten alphabets from Telugu language using Machine Learning**. Tese (Doutorado) — California State University, Northridge, 2018.
- DASGUPTA, S.; PAPADIMITRIOU, C.; VAZIRANI, U. **Algoritmos**. [S.l.]: AMGH Editora, 2009.
- GAO, Y.; CHEN, Y.; WANG, J.; TANG, M.; LU, H. Dense chained attention network for scene text recognition. In: IEEE. **2018 25th IEEE International Conference on Image Processing (ICIP)**. [S.l.], 2018. p. 679–683.
- GOMEZ, R.; SHI, B.; GOMEZ, L.; NUMANN, L.; VEIT, A.; MATAS, J.; BELONGIE, S.; KARATZAS, D. Icdar2017 robust reading challenge on coco-text. In: IEEE. **2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)**. [S.l.], 2017. v. 1, p. 1435–1443.
- GOOGLE, I. **Amostras de detecção de texto - Documentação da API Cloud Vision - Google Cloud**. 2019. Disponível em: <https://cloud.google.com/vision/docs/detecting-text?hl=pt-br>. Acesso em: 10 de maio de 2019.
- GOOGLE, I. **API Vision - Análise do conteúdo de imagens - API Cloud Vision - Google Cloud**. 2019. Disponível em: <https://cloud.google.com/vision/?hl=pt-br>. Acesso em: 10 de maio de 2019.
- HOSSEINI, H.; XIAO, B.; POOVENDRAN, R. Google’s cloud vision api is not robust to noise. In: IEEE. **2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)**. [S.l.], 2017. p. 101–105.
- ICDAR. **ICDAR2017 Home**. 2017. Disponível em: <http://u-pat.org/ICDAR2017/index.php>. Acesso em: 08 de maio de 2019.
- ICDAR2017. **Tasks - ICDAR2017 Robust Reading Challenge on COCO-Text - ICDAR 2019 RobustReading Competition**. 2017. Disponível em: <http://rrc.cvc.uab.es/?ch=5&com=tasks>. Acesso em: 08 de maio de 2019.

ICDAR2017. **Results - ICDAR2017 Robust Reading Challenge on COCO-Text - ICDAR 2019 RobustReading Competition**. 2019. Disponível em: <https://rrc.cvc.uab.es/?ch=5&com=evaluation&task=2>. Acesso em: 04 de maio de 2019.

ICDAR2019. **Introduction - ICDAR 2019 RobustReading Competition**. 2019. Disponível em: <https://rrc.cvc.uab.es>. Acesso em: 03 de julho de 2019.

KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. Imagenet classification with deep convolutional neural networks. In: **Advances in neural information processing systems**. [S.l.: s.n.], 2012. p. 1097–1105.

LIN, T.-Y.; MAIRE, M.; BELONGIE, S.; HAYS, J.; PERONA, P.; RAMANAN, D.; DOLLÁR, P.; ZITNICK, C. L. Microsoft coco: Common objects in context. In: SPRINGER. **European conference on computer vision**. [S.l.], 2014. p. 740–755.

MITCHELL, T. M. Machine learning. 1st edition. **IBM Journal of Research and Development**, McGraw-Hill, Inc., 1997.

MORI, S.; SUEN, C. Y.; YAMAMOTO, K. Historical review of ocr research and development. **Proceedings of the IEEE**, IEEE, v. 80, n. 7, p. 1029–1058, 1992.

RUDIN, L. I.; OSHER, S.; FATEMI, E. Nonlinear total variation based noise removal algorithms. **Physica D: nonlinear phenomena**, Elsevier, v. 60, n. 1-4, p. 259–268, 1992.

WEINMAN, J. J.; LEARNED-MILLER, E.; HANSON, A. R. Scene text recognition using similarity and a lexicon with sparse belief propagation. **IEEE Transactions on pattern analysis and machine intelligence**, IEEE, v. 31, n. 10, p. 1733–1746, 2009.

WIEDERHOLD, G.; MCCARTHY, J. Arthur samuel: Pioneer in machine learning. **IBM Journal of Research and Development**, IBM, v. 36, n. 3, p. 329–331, 1992.

WIKIPÉDIA. **Vladimir Levenshtein – Wikipédia, a enciclopédia livre**. 2019. Disponível em: https://pt.wikipedia.org/wiki/Vladimir_Levenshtein. Acesso em: 27 de maio de 2019.

YU, G.; QI, L.; DAI, Y. On nonmonotone chambolle gradient projection algorithms for total variation image restoration. **Journal of Mathematical Imaging and Vision**, Springer, v. 35, n. 2, p. 143–154, 2009.

APÊNDICE A – SUBMISSÃO DE RESULTADOS SEM SEGMENTAÇÃO DOS DADOS

Neste apêndice, serão apresentados uma breve lista dos resultados obtidos durante a execução dos processos descritos na Seção 4, Seção 5 e em especial, a Figura 14 que descreve graficamente todo o processo deste trabalho. A versão completa possui mais de 9 mil linhas e está disponível em: <https://raw.githubusercontent.com/RonildoOliveira/Analise-Desempenho-GoogleCloud-Vision-API/master/resultados-apidice/37_submit_ch_5_task_2_file_1-GoogleVisionAPI.txt>

```
1 1030340,land
2 1157660,level
3 1213077,home
4 1223888,speclalsts
5 1207119,
6 1030858,2013
7 1161655,garte
8 1104762,b.a
9 1215525,
10 1188823,cuts
11 1231099,park
12 1185646,paper
13 1020780,thowed
14 1070923,
15 1083592,
16 1109826,chadds
17 1217440,emporaf
18 1236737,arg
19 1172778,3d
20 1172778,4008841
21 1172778,00024224
22 1217857,2.90
23 1061374,
24 1125977,
25 1216543,
26 1024865,williams
27 1217992,
28 1097342,
29 1026516,utilize
30 1033015,powexrhouse
31 1123038,plants
32 1073350,
33 1104909,aerismaud
34 1025537,renavias
35 1136359,
36 1215465,
```

```
37 1003593,police
38 1239833,
39 1238263,
40 1226785,
41 1160471,
42 1110136,
43 1017234,
44 1088509,38
45 1210866,
46 1041321,start
47 1034976,closed
48 1012533,erideworld
49 1211504,ground
50 1118509,
51 1024256,xing
52 1236377,
53 1014390,huaraches
54 1028215,gear
55 1031672,statefarm
56 1233515,162010
57 1039450,ees
58 1028839,ransparent
59 1220266,
60 1133291,
61 1222118,
62 1008236,ross
63 1143059,
64 1232529,varela
65 1094003,wuarnet
66 1029532,long
67 1125857,
68 1157484,
69 1016045,7300
70 1144654,
71 1095458,stop
72 1224760,2770
73 1213736,stoutlager
74 1007323,vehicles
75 1125527,
76 1107631,
77 1223390,plntaeipu
78 1223390,normasgenerales
79 1223390,9
80 1150851,650.-
81 1218269,architecture:
82 1233493,mand
83 1146892,
```

```
84 1225526,mary
85 1088459,
86 1010651,connexodon
87 1043599,
88 1178021,
89 1074683,
90 1002777,parking
91 1028865,reduced
92 1143729,
93 1232246,
94 1226971,cafe
95 1160721,www.theroyoloive.com
96 1160721,olive.com
97 1237621,oferta
98 1026564,10/28/2011
99 1043141,grapese
100 1233567,real
```

APÊNDICE B – SUBMISSÃO DE RESULTADOS COM SEGMENTAÇÃO DOS DADOS

Neste apêndice, serão apresentados uma breve lista com a transcrição de 100 imagens que forma parte dos resultados obtidos durante a execução dos processos descritos na Seção 4, Seção 5 e Figura 20 deste trabalho. A versão completa possui mais de 9 mil linhas e pode ser encontrado do *link* seguinte: <https://raw.githubusercontent.com/RonildoOliveira/Analise-Desempenho-GoogleCloud-Vision-API/master/resultados-apidice/43_submit_ch_5_task_2_file_1-GoogleVisionAPI.txt>

```
1 1030340,land
2 1157660,level
3 1213077,heme
4 1223888,spcialsts
5 1207119,
6 1030858,2013
7 1161655,garte
8 1104762,
9 1215525,pure
10 1188823,cuts
11 1231099,park
12 1185646,paper
13 1020780,haredwod
14 1070923,auarblewww.nb
15 1083592,crane
16 1109826,hadd
17 1217440,temporary,
18 1236737,hanget
19 1172778,4006641zt
20 1217857,2.90
21 1061374,
22 1125977,ishb
23 1216543,fihorne
24 1024865,williams
25 1217992,
26 1097342,touro
27 1026516,tutilze
28 1033015,powerhoust
29 1123038,iants
30 1073350,toto
31 1104909,hertsmaud
32 1025537,kehavias
33 1136359,ge
34 1215465,
35 1003593,police
36 1239833,your
```

37 1238263,
38 1226785,
39 1160471,moreton
40 1110136,
41 1017234,dery
42 1088509,w
43 1210866,show
44 1041321,start
45 1034976,closed
46 1012533,eerideworld
47 1211504,punoi
48 1118509,hyun
49 1024256,xing
50 1236377,parefrightsprotecta
51 1014390,huaraches
52 1028215,gear
53 1031672,statefarm
54 1233515,4
55 1039450,bees
56 1028839,transparent
57 1220266,newa
58 1133291,
59 1222118,2etix
60 1008236,rossi
61 1143059,
62 1232529,varela
63 1094003,muarnet2car
64 1029532,long
65 1125857,tours
66 1157484,locatin
67 1016045,7300
68 1144654,mob
69 1095458,stop
70 1224760,2770
71 1213736,stoutlager
72 1007323,vehicles
73 1125527,
74 1107631,people
75 1223390,plannormasgenerales
76 1150851,6s0
77 1218269,architecture:
78 1233493,hand
79 1146892,roo
80 1225526,mary
81 1088459,totall
82 1010651,cannexdon
83 1043599,kempa

```
84 1178021,grips
85 1074683,
86 1002777,parkingeni
87 1028865,reduced
88 1143729,
89 1232246,
90 1226971,cafe
91 1160721,www.lhelovaoive.com
92 1237621,oferta
93 1026564,10/28/2011
94 1043141,grapesed
95 1233567,real
96 1215303,
97 1214255,al
98 1005162,03:43
99 1223241,wark
100 1230760,wrarovar
```