

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ**

**MESSIAS ANTONIO DA ROSA JUNIOR**

**USO DA ARQUITETURA YOLO PARA SUPORTAR O  
RECONHECIMENTO DE TEXTOS BRAILLE**

**DOIS VIZINHOS**

**2022**

**MESSIAS ANTONIO DA ROSA JUNIOR**

**USO DA ARQUITETURA YOLO PARA SUPORTAR O  
RECONHECIMENTO DE TEXTOS BRAILLE**

**Using YOLO framework to support the recognition of Braille texts**

Trabalho de Conclusão de Curso de Graduação apresentado como requisito para obtenção do título de Bacharel em Engenharia de Software do Curso de Bacharelado em Engenharia de Software da Universidade Tecnológica Federal do Paraná.

Orientador: Prof. Dr. Marlon Marcon

**DOIS VIZINHOS**

**2022**



[4.0 Internacional](https://creativecommons.org/licenses/by/4.0/)

Esta licença permite compartilhamento, remixe, adaptação e criação a partir do trabalho, mesmo para fins comerciais, desde que sejam atribuídos créditos ao(s) autor(es). Conteúdos elaborados por terceiros, citados e referenciados nesta obra não são cobertos pela licença.

**MESSIAS ANTONIO DA ROSA JUNIOR**

**USO DA ARQUITETURA YOLO PARA SUPORTAR O  
RECONHECIMENTO DE TEXTOS BRAILLE**

Trabalho de Conclusão de Curso de Graduação  
apresentado como requisito para obtenção do  
título de Bacharel em Engenharia de Software  
do Curso de Bacharelado em Engenharia de  
Software da Universidade Tecnológica Federal  
do Paraná.

Data de aprovação: 07/dezembro/2022

---

Marlon Marcon  
doutorado  
Universidade Tecnológica Federal do Paraná

---

André Roberto Ortoncelli  
doutorado  
Universidade Tecnológica Federal do Paraná

---

Franciele Beal  
doutorado  
Universidade Tecnológica Federal do Paraná

**DOIS VIZINHOS  
2022**

## **AGRADECIMENTOS**

Agradeço aos meus pais, Juliana Correa Bernardo e Messias Antonio da Rosa, por sempre me apoiar e me incentivar nos estudos durante todos esses anos. As minhas irmãs Karen, Manuella, Julia e Maria Fernanda, e minha tia Julia, por também me incentivarem e depositarem suas confianças em mim.

Ao meu orientador, Prof. Dr. Marlon Marcon, pela confiança, paciência, todo o conhecimento repassado e auxílio prestados que foram essenciais para o desenvolvimento deste trabalho. Também agradeço à banca, Profa. Dra. Franciele Beal e Prof. Dr. André Roberto Ortoncelli, por todas as contribuições e sugestões para melhorar o trabalho.

Agradeço à UTFPR e todos os professores, que me acompanharam em todos esses anos e foram fundamentais para a minha evolução intelectual e pessoal.

Aos meus amigos, que me ajudaram a superar momentos difíceis e me acompanharam durante toda esta jornada, tanto no Brasil quanto no Japão, compartilhando bons momentos e experiências.

## RESUMO

Braille é um sistema de leitura e escrita utilizado principalmente pelas pessoas com deficiência visual, apesar de ser um sistema conhecido mundialmente, poucas pessoas que são videntes conseguem ler uma cela Braille. Acreditamos que o uso de visão computacional possa solucionar esse problema para pessoas videntes, eliminando uma das barreiras existentes entre pessoas videntes e cegas. Para o desenvolvimento desse trabalho, realizamos um mapeamento sistemático com o objetivo de identificar quais metodologias estavam sendo utilizadas pela comunidade acadêmica para a realização de detecção de celas Braille e onde esses estudos estão sendo publicados, além de realizar treinamentos de redes neurais utilizando o framework YOLO (You Only Look Once) com três bases de dados diferentes, DSBI, Donnie Braille e Braille Character Dataset, obtendo uma precisão média de 19%, 100% e 85% respectivamente.

**Palavras-chave:** obr; carcateres braile; reconhecimento tico de imagens em braille; deteção de pontos; documentos em braille.

## ABSTRACT

Braille is a reading and writing system used mainly by people with visual impairments, despite being a system known worldwide, few people who are sighted can read a Braille cell. We believe that the use of computer vision can solve this problem for sighted people, eliminating one of the existing barriers between sighted and blind people. For the development of this work, we carried out a systematic mapping in order to identify which methodologies were being used by the academic community to carry out detection of Braille cells and where these studies are being published, in addition to training neural networks using the YOLO (You Only Look Once) framework with three different databases, DSBI, Donnie Braille and Braille Character Dataset, obtaining an average accuracy of 19%, 100% and 85% respectively.

**Keywords:** obr; braille characters; optical braille image recognition; dot detection; braille documents.

## LISTA DE FIGURAS

<b>Figura 1</b>	<b>– Representação das celas Braille na língua portuguesa. . . . .</b>	<b>9</b>
<b>Figura 2</b>	<b>– Representação letras com diacríticos em Braille na língua portuguesa. .</b>	<b>10</b>
<b>Figura 3</b>	<b>– Telas da aplicação desenvolvida. . . . .</b>	<b>14</b>
<b>Figura 4</b>	<b>– Arquitetura da CNN BraUNET. . . . .</b>	<b>15</b>
<b>Figura 5</b>	<b>– Representação da base de dados do DSBI. . . . .</b>	<b>16</b>
<b>Figura 6</b>	<b>– Representação da base de dados do DSBI com linhas auxiliares. . . . .</b>	<b>16</b>
<b>Figura 7</b>	<b>– Representação da base de dados do DSBI com celas reconhecidas. . . . .</b>	<b>16</b>
<b>Figura 8</b>	<b>– Representação da base de dados Donnie Braille. . . . .</b>	<b>17</b>
<b>Figura 9</b>	<b>– Representação da base de dados Braille Character Dataset. . . . .</b>	<b>17</b>
<b>Figura 10</b>	<b>– Representação das tecnologias mais utilizadas em trabalhos para reco- nhecimento de celas Braille. . . . .</b>	<b>21</b>
<b>Figura 11</b>	<b>– Representação das formas de aquisição de imagens mais utilizadas em trabalhos para reconhecimento de celas Braille. . . . .</b>	<b>21</b>
<b>Figura 12</b>	<b>– Representação dos trabalhos que fazem o reconhecimento em arquivos <i>Double-sided</i> . . . . .</b>	<b>22</b>
<b>Figura 13</b>	<b>– Representação dos trabalhos que fazem processo de segmentação e re- conhecimento separadamente . . . . .</b>	<b>22</b>
<b>Figura 14</b>	<b>– Representação dos países . . . . .</b>	<b>23</b>
<b>Figura 15</b>	<b>– Representação do ano em que foram publicados os artigos selecionados.</b>	<b>23</b>
<b>Figura 16</b>	<b>– Representação da sequencia utilizada nas celas Braille . . . . .</b>	<b>25</b>
<b>Figura 17</b>	<b>– Representação das anotações providas na base de dados . . . . .</b>	<b>26</b>
<b>Figura 18</b>	<b>– Representação das anotações convertidas para o formato requerido . .</b>	<b>27</b>
<b>Figura 19</b>	<b>– Visualização de como são as anotações para o treinamento . . . . .</b>	<b>27</b>
<b>Figura 20</b>	<b>– Representação das anotações em PASCAL VOC XML . . . . .</b>	<b>28</b>
<b>Figura 21</b>	<b>– Representação da interseção sobre união . . . . .</b>	<b>33</b>
<b>Figura 22</b>	<b>– Representação da linha do tempo do treinamento utilizando a base DSBI</b>	<b>35</b>
<b>Figura 23</b>	<b>– Representação da linha do tempo do treinamento utilizando a base Donnie Braille . . . . .</b>	<b>36</b>
<b>Figura 24</b>	<b>– Representação da linha do tempo do treinamento utilizando a base Braille Character Dataset . . . . .</b>	<b>37</b>

## LISTA DE TABELAS

<b>Tabela 1 – Tabela de palavras-chave utilizadas para realização do protocolo . . . .</b>	<b>19</b>
<b>Tabela 2 – Representação da quantidade de artigos encontrados no protocolo . . . .</b>	<b>19</b>



## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>9</b>
<b>1.1</b>	<b>Objetivo Geral</b>	<b>10</b>
<b>1.2</b>	<b>Objetivos Específicos</b>	<b>11</b>
<b>1.3</b>	<b>Motivação</b>	<b>11</b>
<b>1.4</b>	<b>Organização do documento</b>	<b>12</b>
<b>2</b>	<b>ESTADO DA ARTE</b>	<b>13</b>
<b>2.1</b>	<b>Base de Dados</b>	<b>14</b>
2.1.1	DSBI	15
2.1.2	Donnie Braille	16
2.1.3	Braille Character Dataset	17
<b>3</b>	<b>MAPEAMENTO SISTEMÁTICO</b>	<b>18</b>
<b>3.1</b>	<b>Crerios de exclusão</b>	<b>18</b>
<b>3.2</b>	<b>Questões de pesquisa</b>	<b>18</b>
<b>3.3</b>	<b>Respondendo as questões de pesquisa</b>	<b>20</b>
3.3.1	QP1: Qual é a tecnologia utilizada?	20
3.3.2	QP2: Qual é a forma de aquisição das imagens?	20
3.3.3	QP3: Realiza a detecção em <i>Double-sided</i> ?	20
3.3.4	QP4: Executa processos de segmentação e reconhecimento separados?	21
3.3.5	QP5: Quando e onde têm sido publicados estes trabalhos?	23
3.3.6	Conclusão	24
<b>4</b>	<b>METODOLOGIA</b>	<b>25</b>
<b>4.1</b>	<b>Detalhamento das atividades desenvolvidas</b>	<b>25</b>
4.1.1	Seleção das bases de dados para testes e treinamento utilizando YOLO	25
4.1.2	Realização do mapeamento sistemático acerca das soluções para reconhecimento de textos Braille	29
4.1.3	Anotação das bases de dados de treinamento	29
4.1.4	Treinamento da YOLO utilizando as bases de dados	29
<b>4.2</b>	<b>Tecnologias</b>	<b>29</b>
4.2.1	Python	30
4.2.2	YOLO	30

<b>5</b>	<b>DESENVOLVIMENTO</b>	<b>31</b>
<b>5.1</b>	<b>DSBI</b>	<b>31</b>
<b>5.2</b>	<b>Donnie Braille</b>	<b>32</b>
<b>5.3</b>	<b>Braille Character Dataset</b>	<b>32</b>
<b>6</b>	<b>RESULTADOS</b>	<b>33</b>
<b>6.1</b>	<b>Base de dados: DSBI</b>	<b>33</b>
<b>6.2</b>	<b>Base de dados: Donnie Braille</b>	<b>34</b>
<b>6.3</b>	<b>Base de dados: Braille Character Dataset</b>	<b>35</b>
<b>7</b>	<b>CONCLUSÕES</b>	<b>38</b>
	<b>REFERÊNCIAS</b>	<b>39</b>

## 1 INTRODUÇÃO

A deficiência visual ocorre quando se tem perda ou redução da capacidade visual em ambos os olhos sem a possibilidade de corrigir o problema com cirurgia, lentes de contato ou óculos (JUSTIÇA, s.d). Segundo (OTTAIANO *et al.*, 2019), existem três níveis de classificação para a deficiência visual, sendo eles:

- Deficiência visual moderada;
- Deficiência visual grave;
- Cegueira.

De acordo com o censo do IBGE (IBGE, 2010), no Brasil, existem 29 milhões de pessoas com deficiência visual em alguma dificuldade, de 6 milhões de pessoas com grande dificuldade visual e aproximadamente 500 mil pessoas não conseguem enxergar de modo algum. Não se sabe ao certo quantas pessoas no mundo possuem alguma deficiência visual, pois com as pesquisas atuais, não há dados suficientes para se chegar a uma conclusão certa, estima-se que se tenha 36 milhões de pessoas cegas e 216 milhões de pessoas com visão subnormal no mundo (OTTAIANO *et al.*, 2019).

O Braille é um sistema que possibilita a leitura e escrita de deficientes visuais criado por Louis Braille em 1829, sendo testado e aprimorado até a sua versão final em 1837. (JIMÉNEZ *et al.*, 2009).

O método criado por Braille consiste em códigos escritos em relevo, que são lidos através do tato. Atualmente o modelo é o mais utilizado no mundo, sendo reconhecido na UNESCO - Organização das Nações Unidas para a Educação, a Ciência e a Cultura como uma “língua vital de comunicação” (JIMÉNEZ *et al.*, 2009), tendo somente algumas variações de acordo com cada país. O sistema é formado por duas colunas de três pontos, possibilitando até 64 combinações, que no fim, formam letras, números e símbolos (BRASIL, 2018). As Figuras 1 e 2 apresentam a codificação Braille associada a cada letra ou símbolo, para a Língua Portuguesa.

**Figura 1 – Representação das celas Braille na língua portuguesa.**

a	b	c	d	e	f	g	h	i	j	k	l	m
⠁	⠃	⠉	⠑	⠅	⠋	⠗	⠈	⠊	⠽	⠏	⠇	⠍
n	o	p	q	r	s	t	u	v	w	x	y	z
⠞	⠕	⠎	⠖	⠗	⠘	⠙	⠛	⠜	⠝	⠞	⠟	⠡

Fonte: (BRASIL, 2018).

**Figura 2 – Representação letras com diacríticos em Braille na língua portuguesa.**

<b>Vogais</b>	a	⠁	e	⠑	i	⠃	o	⠕	u	⠥
<b>Acento agudo</b>	á	⠁⠗	é	⠑⠗	í	⠃⠗	ó	⠕⠗	ú	⠥⠗
<b>Acento grave</b>	à	⠁⠗	–	–	–	–	–	–	–	–
<b>Acento circunflexo</b>	â	⠁⠗	ê	⠑⠗	–	ô	⠕⠗	–	–	–
<b>Til</b>	ã	⠁⠗	–	–	–	õ	⠕⠗	–	–	–

<b>Consoante</b>	c	⠉
<b>Cedilha</b>	ç	⠉⠗

**Fonte: (BRASIL, 2018).**

O desenvolvimento de tecnologias que favoreçam a compreensão do Braille é importante para melhorar a qualidade de vida de quem as precisa, apesar de haver muitas pessoas que utilizam o sistema Braille, ele não é bem conhecido por pessoas que não necessitam utilizar o sistema para se comunicar. Um modo de transcrição dessas informações poderia auxiliar pessoas videntes na aprendizagem do sistema ou na troca de informações com pessoas que fazem o uso do sistema Braille para a comunicação.

A Visão Computacional, que é composta de ferramentas para processar imagens e extrair informações delas, oferece uma grande variedade de aplicações. É usada para análise de imagens médicas, detecção de objetos, reconhecimento facial, detecção de movimento, análise de vídeo, modelagem 3D, rastreamento de veículos e muitos outros sistemas especialistas. Um sistema especialista em reconhecimento de textos Braille poderia auxiliar professores, médicos, atendentes, terapeutas, entre outros profissionais, a entender e decodificar a escrita Braille de maneira rápida e precisa. O sistema poderia ser usado para traduzir textos escritos em Braille para outros formatos de texto, como textos digitais, para melhorar a acessibilidade e a inclusão de pessoas com deficiência visual.

No contexto atual, aprendizagem profunda tem demonstrado capacidade de superar métodos tradicionais para reconhecimento de objetos em imagens<sup>1</sup>. Diversas arquiteturas de Redes Neurais Convolucionais (do inglês *Convolutional Neural Networks*, CNNs) conseguem solucionar problemas de detecção de objetos e realizar um bom trabalho quando se trata de reconhecimento de imagem, como no caso do YOLO (REDMON *et al.*, 2016).

## 1.1 Objetivo Geral

Esta proposta tem como objetivo geral, **desenvolver uma solução, baseada em aprendizagem profunda através do framework YOLO para detecção de celas Braille.**

<sup>1</sup> <https://paperswithcode.com/sota/image-classification-on-imagenet>

## 1.2 Objetivos Específicos

Para realização do objetivo geral, alguns objetivos específicos são vislumbrados:

- Selecionar bases de dados para testes e treinamento utilizando YOLO;
- Realizar um mapeamento sistemático para identificar como estão sendo feitos os trabalhos focados na detecção e transcrição de celas Braille;
- Conversão ou criação das anotações das caixas delimitadoras de cada cela Braille;
- Treinar e testar a rede neural utilizando as bases de dados selecionadas.

## 1.3 Motivação

Para uma pessoa sem deficiência visual, muitas vezes fica complicado compreender os desafios envolvidos e a importância de se realizar o reconhecimento de textos Braille. Visando exemplificar as possibilidades de utilização de sistemas de visão computacional aplicados para este fim, são apresentados os seguintes cenários:

**Cenário 1:** É muito importante que toda criança seja estimulada a explorar os seus sentidos para que se tenha um melhor desenvolvimento no futuro, e, por conta do pouco ou até mesmo ausência de estímulo dos familiares e grupo social por conta da privação de um dos sentidos, pode causar problemas no desenvolvimento da criança (BRASIL, 2000). A ferramenta proposta poderá auxiliar os familiares a incentivar a pessoa com deficiência visual através da transcrição dos textos Braille para textos digitais.

Para realizar a leitura e escrita, deficientes visuais utilizam máquinas de escrever e computadores adaptados ao Braille, estes, geram folhas que possibilitam a leitura de deficientes visuais, porém, pessoas que não necessitam de fazer o uso do sistema em seu dia a dia, possuem dificuldade em compreender o mesmo. O sistema quebra esta barreira através da tradução da anotação braille para texto, possibilitando que pessoas videntes que não possuem conhecimento no sistema possam compreender o que está escrito e auxiliar o deficiente visual.

**Cenário 2:** Um aluno com deficiência visual utiliza uma máquina de escrever em Braille para o desenvolvimento de seus trabalhos acadêmicos, seu colega de classe, que não compreende o sistema Braille e deseja realizar o trabalho acadêmico com seu colega de classe, poderá utilizar a ferramenta para a compreensão do que está sendo desenvolvido pelo seu colega.

**Cenário 3:** Para a realização de provas e exames, como o ENEM, o aluno possui o direito de receber todo o material em Braille, se preferir. Também é direito do aluno utilizar uma máquina de escrever em Braille para a realização da prova. Sendo assim, o avaliador, necessitaria de uma ferramenta capaz de transcrever o texto feito pelo aluno caso o mesmo não possuía conhecimentos sobre o sistema Braille.

**Cenário 4:** Um médico prescreve um medicamento para um paciente deficiente visual, para a comodidade do paciente, é lhe passado duas receitas com o mesmo conteúdo, sendo uma em braille e outra em texto tradicional. Ao levar as receitas para a farmácia para a busca dos medicamentos, o paciente leva apenas a receita prescrita em Braille, onde ninguém no local possui conhecimento sobre o sistema Braille, sendo assim, o sistema proposto poderia auxiliar os farmacêuticos a entregar os medicamentos certos ao cliente. Ainda no contexto de medicamentos, caso o deficiente visual necessite de auxílio para tomar os medicamentos e só possuir as instruções em Braille, o sistema poderia ser utilizado para a consulta dessas informações.

**Cenário 5:** O sistema pode ser utilizado no atendimento ao público, onde há a necessidade de um deficiente visual repassar informações para um vidente, que não possui conhecimento sobre o sistema Braille.

Para promover a inclusão social de pessoas cegas, é importante levar em consideração as necessidades e as barreiras que essas pessoas enfrentam. Isso pode incluir a implementação de medidas de acessibilidade, como a utilização de braile, áudio descrição e outros recursos para tornar os ambientes mais acessíveis. Além disso, é importante promover a sensibilização e o entendimento da comunidade sobre as necessidades das pessoas cegas, bem como trabalhar para eliminar atitudes e comportamentos discriminatórios. Criar formas de melhorar a comunicação e interação entre videntes e pessoas com deficiência visual promovem a inclusão e colaboram para uma sociedade mais justa e igualitária.

#### **1.4 Organização do documento**

Nessa seção é apresentada a organização da monografia, ela está organizada em nove seções: Na Seção 2 são apresentados conceitos relacionados ao processo padrão de aprendizado de máquina utilizado para o reconhecimento de celas Braille, bem como a exemplificação e demonstração de propostas similares que são utilizadas como referência para a realização da monografia. Além disso é apresentada as base de dados selecionadas para a realização dos treinamentos e testes a serem realizados; A Seção 3 apresenta o protocolo utilizado para a realização de mapeamento sistemático, bem como a apresentação das respostas encontradas; A Seção 4 detalha as atividades que foram desenvolvidas na monografia, tecnologias que foram utilizadas para a realização dos objetivos, e ações realizadas para a conclusão do projeto; A seção 5 apresenta como foram realizadas as atividades para o desenvolvimento da monografia. A Seção 6 trata dos resultados obtidos com o desenvolvimento deste projeto; Por fim, são apresentadas as conclusões obtidas e recomendações para trabalhos futuros.

## 2 ESTADO DA ARTE

Um processo padrão para reconhecimento de textos em Braille por meio de visão computacional consiste nas seguintes etapas, segundo (FARINELLA; LEONARDI; STANCO, 2015):

- Selecionar parte de interesse na imagem para focar somente no texto em Braille;
- Binarização e reconhecimento dos pontos da imagem;
- Preenchimento dos pontos que foram parcialmente reconhecidos;
- Segmentação da imagem em linhas e colunas;
- Por fim, a transcrição das celas Braille é realizada.

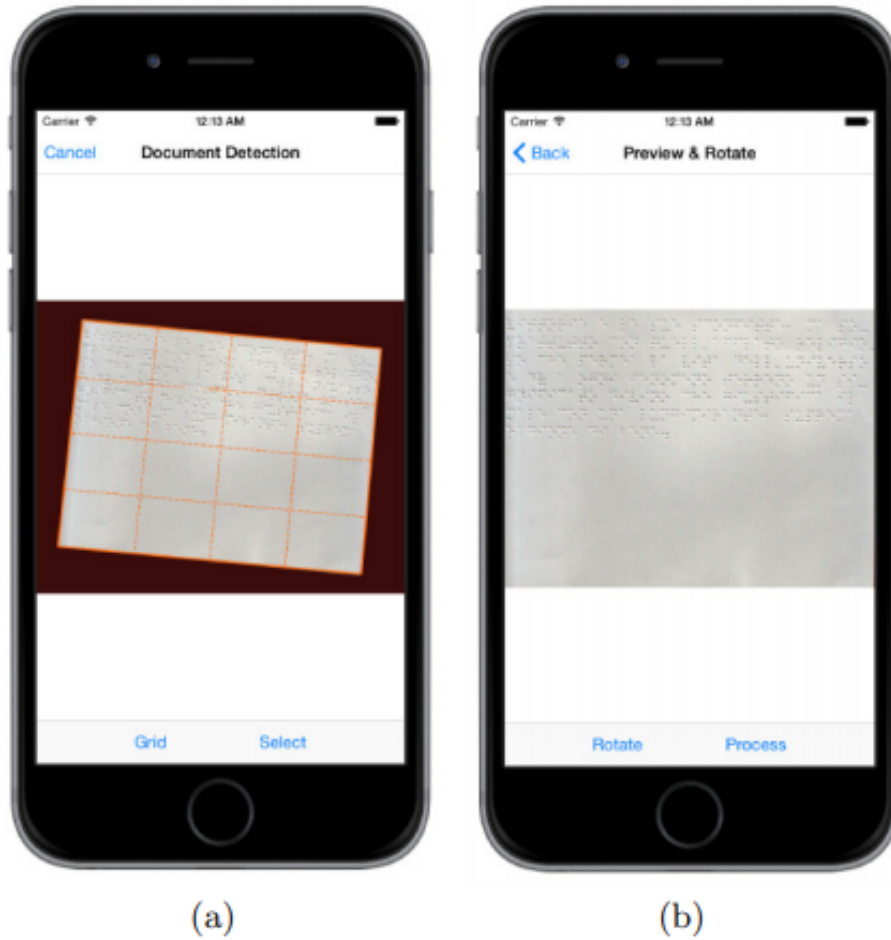
(FARINELLA; LEONARDI; STANCO, 2015) propuseram um aplicativo para smartphone capaz de traduzir textos em Braille para texto clássico, com eficácia de 91%. Algumas telas deste aplicativo são exibidas na Figura 3, demonstrando as etapas de seleção da área de interesse (b) a partir de uma foto capturada pelo dispositivo (a).

Apesar da eficácia obtida por (FARINELLA; LEONARDI; STANCO, 2015) o processo utilizado para reconhecimento é suscetível a erros de reconhecimento devido a natureza das imagens, que podem apresentar variações de iluminação, contraste, escala etc. Para contornar tais problemas, soluções baseadas em Aprendizagem Profunda (do inglês, Deep Learning) têm se mostrado mais robustas a este tipo de problema (LI *et al.*, 2020; OVODOV, 2021)

Existem diversas ferramentas que conseguem alcançar o *state-of-the-art* em reconhecimento de objetos genérico em imagens, dentre elas estão o YOLO (REDMON *et al.*, 2016), um sistema que utiliza apenas uma rede neural para reconhecer a imagem toda, conseguindo também fazer o reconhecimento em tempo real. Assim como o YOLO, o Single Shot MultiBox Detector (SSD) (LIU *et al.*, 2016) também usa apenas uma rede neural para o reconhecimento, sua vantagem é que não é necessário se preocupar com o redimensionamento da imagem ao ser inserida, melhorando a sua performance por ser um processamento a menos. Mask R-CNN (HE *et al.*, 2017) é uma extensão do Faster R-CNN (REN *et al.*, 2015), o objetivo do trabalho foi adicionar mais uma camada de reconhecimento de máscaras em paralelo com a camada de reconhecimento das caixas delimitadoras, com a utilização de segmentação semântica, a rede obteve excelentes resultados, conseguindo superar o desempenho de todos os modelos de entrada única.

Algumas soluções propõem a utilização de *deep learning* no reconhecimento de textos Braille, como o BraUNET (LI *et al.*, 2020). Está arquitetura é baseada na U-NET (RONNEBERGER; FISCHER; BROX, 2015), porém focada na detecção e identificação de caracteres Braille, e para tal, sua metodologia aplica a segmentação semântica em textos digitalizados. A

**Figura 3 – Telas da aplicação desenvolvida.**



**Fonte: Farinella, Leonardi e Stanco (2015).**

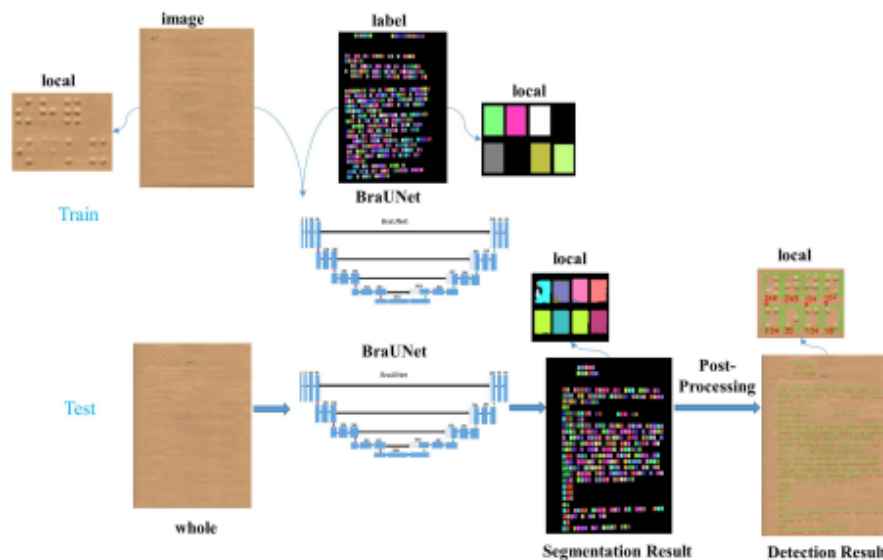
Figura 4 apresenta a arquitetura da rede, que tem como entrada uma imagem de uma página de texto digitalizado e retorna os caracteres segmentados e sua respectiva classe (código Braille).

## 2.1 Base de Dados

Como mostrado na Seção 3, reconhecimento de textos Braille usando aprendizagem profunda não é uma área muito explorada pela comunidade acadêmica, por este motivo, ainda não existem muitas bases de dados completas que possam ser utilizadas para o treinamento da inteligência artificial, a seguir, é demonstrado as base de dados escolhidas para o desenvolvimento da monografia:



**Figura 4 – Arquitetura da CNN BraUNET.**



**Fonte: Li et al. (2020).**

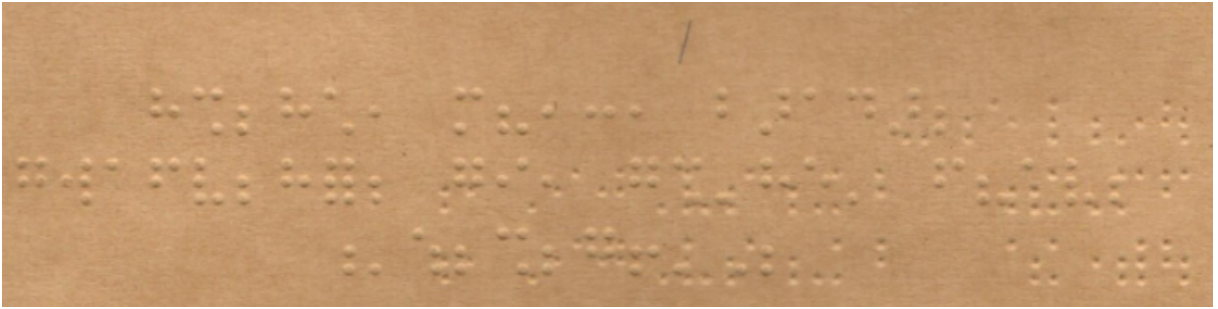
### 2.1.1 DSBI

DSBI (LI et al., 2018), consiste em 144 imagens coloridas com textos em Braille Double-sided, ou seja, textos que possuem dupla face (escrita em ambos os versos da folha) de seis livros diferentes, com imagens com resolução média de 1700x2300 pixels. A escolha da base de dados se deu em conta da mesma estar sendo utilizada em dois outros projetos que são referência para comparações com o projeto a ser desenvolvido, sendo chamados de BraUNet(LI et al., 2020) e Angelina(OVODOV, 2021), que são trabalhos que realizam a detecção de e transcrição de celas Braille. Sendo a base de dados mais completa encontrada em relação as combinações possíveis, a base possui dados para todas as 64 combinações possíveis.

A Figura 5 representa a primeira página dos textos pertencente ao grupo “Ordinary Printed Document”. É importante ressaltar que apesar de alguns livros estarem escritos em Braille para o mandarim, ainda sim esses seriam úteis para o treinamento em qualquer idioma, tendo em vista que as celas Braille são as mesmas em qualquer idioma apenas mudando a sua ordem e em alguns casos o que cada uma representa. Porém a tradução pode ser facilmente realizada através da codificação de cada cela para um determinado idioma.

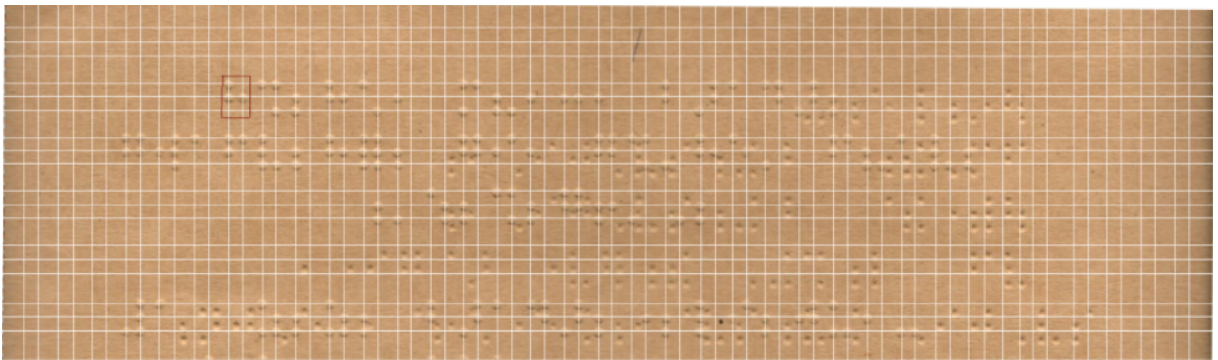
A base de dados disponibiliza em suas anotações, a posição em pixels das linhas verticais e horizontais. A Figura 6 representa como as linhas são distribuídas, as linhas brancas foram adicionadas para melhor visualização. Podemos notar que existem pontos de intersecção das linhas em que pode se visualizar os caracteres Braille,.

**Figura 5 – Representação da base de dados do DSBI.**



**Fonte: Autoria própria.**

**Figura 6 – Representação da base de dados do DSBI com linhas auxiliares.**



**Fonte: Autoria própria.**

A seguir, na Figura 7, é demonstrado como foi feito a leitura das celas Braille e quais valores foram inseridos para o treinamento rede neural. As linhas vermelhas representam os dados a serem reconhecidos.

**Figura 7 – Representação da base de dados do DSBI com celas reconhecidas.**



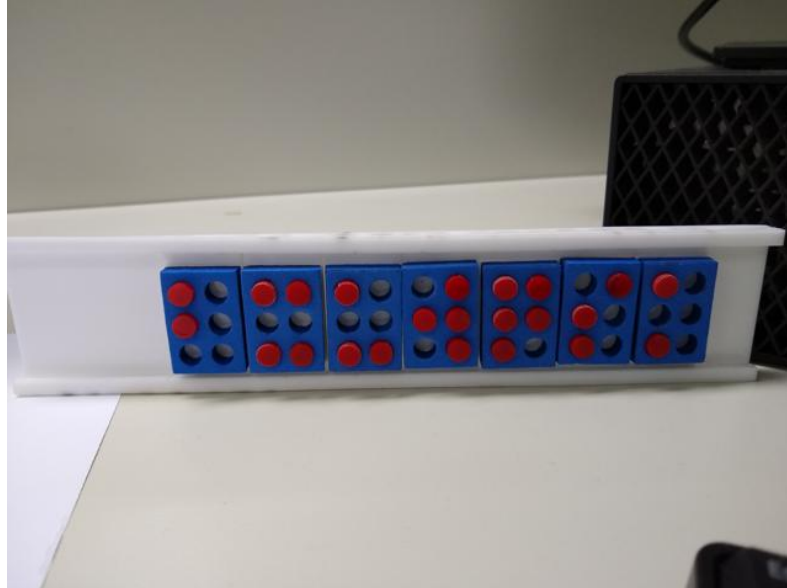
**Fonte: Autoria própria.**

### 2.1.2 Donnie Braille

A segunda base de dados utilizada é a Donnie Braille (GONÇALVES *et al.*, 2020), a base consiste em 2818 imagens coloridas com uma resolução de 640 x 480 pixels em um

ambiente real, sendo uma base menos completa, possui dados apenas para 26 combinações, sendo essas combinações referentes a todas as letras do alfabeto, de A à Z.

**Figura 8 – Representação da base de dados Donnie Braille.**



**Fonte: Gonçalves *et al.* (2020).**

### 2.1.3 Braille Character Dataset

Braille Character Dataset (SHANKS0465, 2019) é uma base de dados que consiste em 1560 imagens preto e branco de tamanho 28 x 28 pixels, sendo cada imagem referente aos 26 caracteres do alfabeto latino. Para cada letra, são disponibilizadas 60 variações, com diferentes posições, brilhos e rotações, resultando então, nas 1560 possibilidades.

A escolha dessa base de dados, se deu por conta de ser uma base com características diferentes das bases apresentadas anteriormente, sendo assim, possível fazer uma análise com três opções diferentes.

**Figura 9 – Representação da base de dados Braille Character Dataset.**



**Fonte: SHANKS0465 (2019).**

### 3 MAPEAMENTO SISTEMÁTICO

Com o objetivo de identificar as tecnologias de visão computacional utilizadas para o reconhecimento de caracteres Braille, além da forma de obtenção dos textos a serem convertidos, foi realizado um mapeamento sistemático aplicado da seguinte forma: (i) definição das questões de pesquisa (QPs), (ii) busca por estudos preliminares de relevância, (iii) identificação das palavras-chaves utilizadas com mais frequência (iv) seleção dos estudos primários, (v) revisão da string de busca criada (vi) categorização dos artigos e (vii) mapeamento e extração das informações que respondem as questões previamente estabelecidas.

#### 3.1 Critérios de exclusão

- Estudos que não utilizam Visão Computacional para reconhecimento de textos em Braille;
- Estudos que não estejam na língua inglesa;
- Estudos que foram publicados antes de 2013;
- Relatórios técnicos, documentos que estão disponíveis na forma de resumos ou apresentações e também revisões de literatura secundárias (ou seja, revisões e mapeamentos sistemáticos da literatura) foram excluídos.

#### 3.2 Questões de pesquisa

- QP1: Qual é tecnologia utilizada?
- QP2: Qual é a forma de aquisição das imagens?
- QP3: Realiza a detecção em *Double-sided*?
- QP4: Executa processos de segmentação e reconhecimento separados?
- QP5: Quando e onde têm sido publicados estes trabalhos?

A partir da análise de estudos previamente selecionados, as palavras-chave da pesquisa foram identificadas. Os termos identificados foram concatenados para gerar uma *string* de busca que foi aplicada às bases indexadoras de publicações científicas. A estrutura das strings de busca segue o princípio de conectar termos sinônimos com o conectivo lógico **OR**, e com o conectivo **AND** para ligar grupos diferentes de termos. As palavras-chave identificadas juntamente com seus sinônimos são representadas na Tabela 1;

**Tabela 1 – Tabela de palavras-chave utilizadas para realização do protocolo**

	Palavra chave	Sinônimos
1	Braille	-
2	Optical Braille Recognition	OBR, braille characters, Dot Detection, Optical Braille image recognition, Optical Braille Images, Braille documents
3	Computer Vision	Object recognition, object tracking, object learning, object interpretation, image recognition

**Fonte: Autoria própria.**

Inicialmente foram utilizadas todas as palavras-chave e sinônimos da tabela1 para a geração da string de busca, porém, por limitação da base do *Science Direct*, houve a necessidade de reduzir a quantidade de palavras-chave utilizadas na busca, os resultados obtidos em ambas strings geradas e os resultados obtidos ao se pesquisar apenas a palavra “Braille”, são representados na Tabela 2. A primeira coluna denominada Braille apresenta os resultados retornados a partir da busca de somente este termo nas bases, a coluna “Braille OR OBR” apresenta os resultado das linhas 1 e 2 da Tabela 1 combinados. Por fim, as colunas “String completa” e “String de 9 termos” apresentam a combinação das linhas 1, 2 e 3 da tabela1. A diferença entre elas é a quantidade de termos utilizado na busca, que devido à limitação no número para o *Science Direct* foi necessária. É possível notar que a String final foi reconfigurada de modo a retornar o mesmo número de resultados.

**Tabela 2 – Representação da quantidade de artigos encontrados no protocolo**

Plataforma	Apenas Braille	Braille OR OBR	String completa	String de 9 termos
IEEE	570	1,775	58	58
Scopus	10,964	19,677	1,023	1023
Science Direct	4,903	381,084	Impossível	358
ACM	1,623	496,528	235	235
Total	18,060	899,064	1,598	1,674

**Fonte: Autoria própria.**

Os resultados obtidos demonstrados na tabela2, buscando apenas a palavra “Braille” e apenas o “OR” referente ao Braille, serviu para se ter uma noção do teto de resultados que poderiam ser obtidos, podemos perceber que algumas bases de dados não possuem um acervo vasto relacionado ao tema. Vale destacar que, exceto para a base de dados da *Science Direct*, onde não se foi possível fazer a busca com a *string* completa, os resultados obtidos com a string de 9 termos e a string completa são os mesmos, ou seja, não houve perda de resultados ao fazer a redução de palavras-chave.

A seguir são apresentadas as *strings* de busca que foram utilizadas no mapeamento sistemático. A primeira consiste na string originalmente vislumbrada e a segunda a *string* readequada para compatibilizar a busca em todas as bases de maneira uniforme.

**String Original:**

(“Braille” OR “optical braille recognition” OR “obr” OR “braille characters” OR “optical braille image recognition” OR “dot detection” OR “optical braille images” OR “braille documents”) AND (“computer vision” OR “object recognition” OR “image recognition” OR “object tracking” OR “object learning” OR “object interpretation”)

**String modificada:**

(“Braille” OR “obr” OR “dot detection”) AND (“computer vision” OR “object recognition” OR “image recognition” OR “object tracking” OR “object learning” OR “object interpretation”)

As informações coletadas, juntamente com os gráficos gerados são disponibilizados através de um arquivo no Google Drive<sup>1</sup>.

### 3.3 Respondendo as questões de pesquisa

#### 3.3.1 QP1: Qual é a tecnologia utilizada?

A primeira questão de pesquisa (QP1) refere-se a qual tecnologia vem sendo utilizada para o treinamento das ferramentas para reconhecimento de texto Braille, a Figura 10 mostra que 26,2% dos projetos selecionados utilizam *Deep Learning* para o treinamento, seguido por *Image Segmentation* com 19%. Já as outras tecnologias aparecem com menos de 10%. Um fato notado é que a maioria dos projetos que utilizam *Image segmentation* foram publicados entre os anos 2013 e 2015, enquanto os que utilizam *Deep Learning* foram publicados entre 2019 e 2022.

#### 3.3.2 QP2: Qual é a forma de aquisição das imagens?

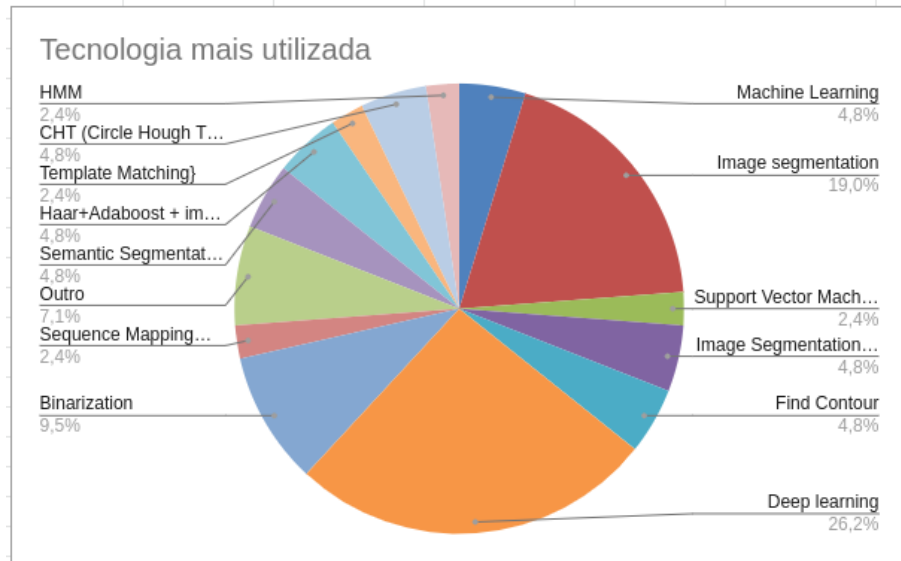
A segunda questão de pesquisa (QP2) refere-se ao método de aquisição das imagens utilizadas para o treinamento, como visto na Figura 11, 64,3% dos projetos utilizam scanner como método de aquisição, podemos notar uma grande dominância deste método, pois é o método onde se é possível eliminar a maior parte dos ruídos e se obter uma boa resolução.

#### 3.3.3 QP3: Realiza a detecção em *Double-sided*?

A terceira questão de pesquisa (QP3) refere-se a quantidade de artigos que realizam a detecção em páginas *Double-sided*, ou seja, páginas que possuem impressões Braille em suas duas faces. Por se tratar de uma grafia tátil, é necessário se criar um relevo na superfície que será utilizada para a leitura, sendo assim, o verso de onde foi criado o relevo fica com “ruídos”.

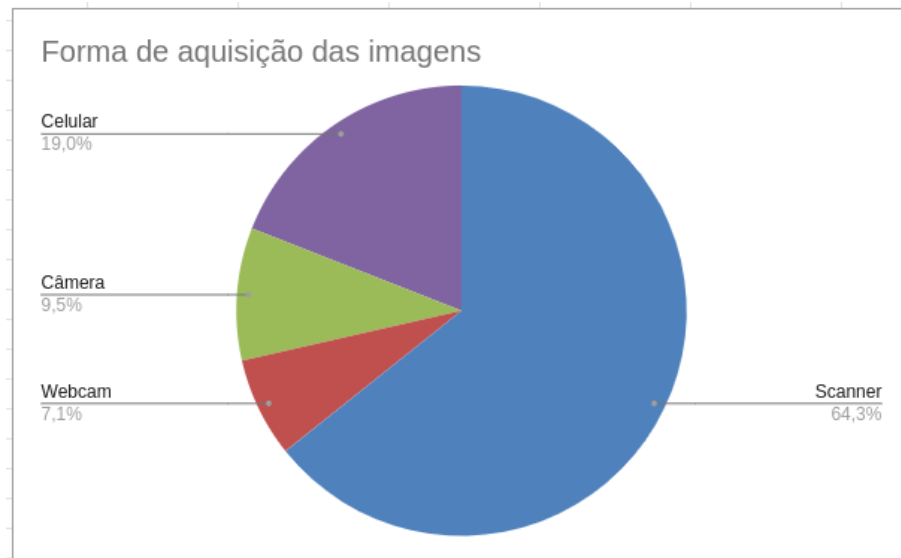
<sup>1</sup> <https://docs.google.com/spreadsheets/d/1gz0Fu9GBcUBtNMFtmva05pyKo7U40T2V/edit?usp=sharing&oid=116566424443329145346&rtpof=true&sd=true>

**Figura 10 – Representação das tecnologias mais utilizadas em trabalhos para reconhecimento de celas Braille.**



Fonte: Autoria própria.

**Figura 11 – Representação das formas de aquisição de imagens mais utilizadas em trabalhos para reconhecimento de celas Braille.**



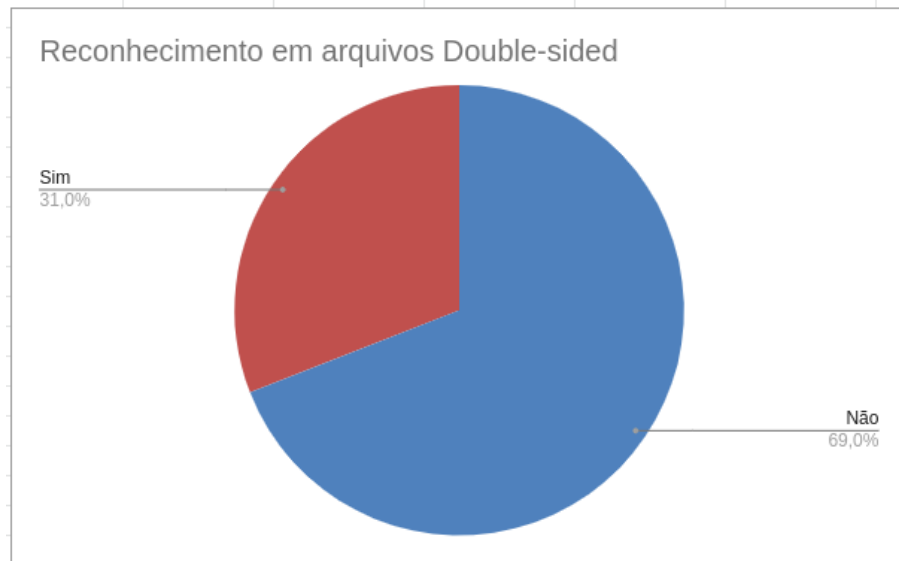
Fonte: Autoria própria.

Por conta disso, utilizar base de dados que possuem celas nas duas faces da superfície, tem um desafio a mais durante a detecção, pois precisariam lidar com esses ruídos gerados pelas celas do verso e não às detectar como sendo parte dos textos.

#### 3.3.4 QP4: Executa processos de segmentação e reconhecimento separados?

Por meio da quarta questão de pesquisa (QP4) buscamos verificar se os trabalhos estão executando processo de segmentação previamente ao de reconhecimento. Processo de segmen-

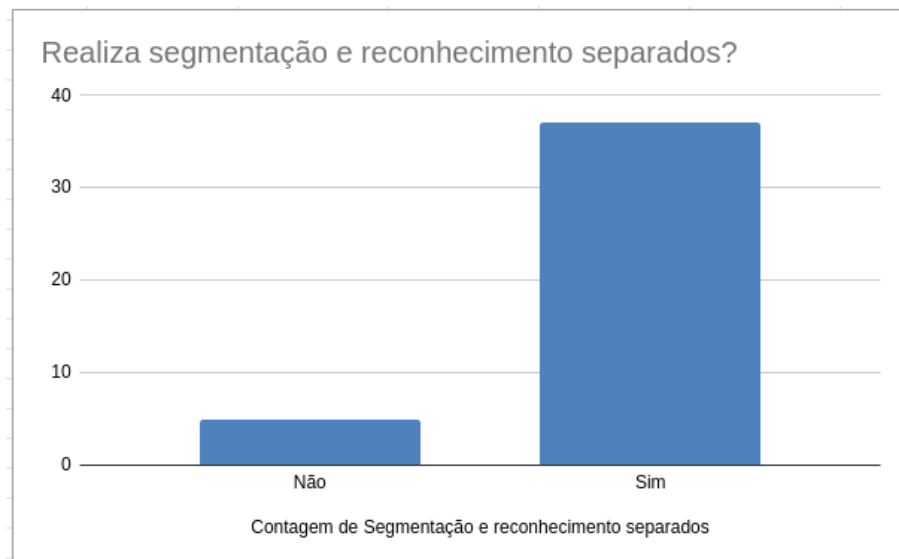
**Figura 12 – Representação dos trabalhos que fazem o reconhecimento em arquivos *Double-sided***



**Fonte: Autoria própria.**

tação consiste em identificar os objetos/regiões de interesse da imagem, e pode ser feito através de técnicas de processamento digital de imagens. De forma geral, é uma etapa importante em muitas aplicações de reconhecimento de objetos por meio de imagens digitais, e permite identificar objetos com maior precisão. Por meio da Figura 13 podemos perceber que a maioria do trabalho reconhece essa importância e vem utilizando essa metodologia como forma de aumentar o seu desempenho.

**Figura 13 – Representação dos trabalhos que fazem processo de segmentação e reconhecimento separadamente**



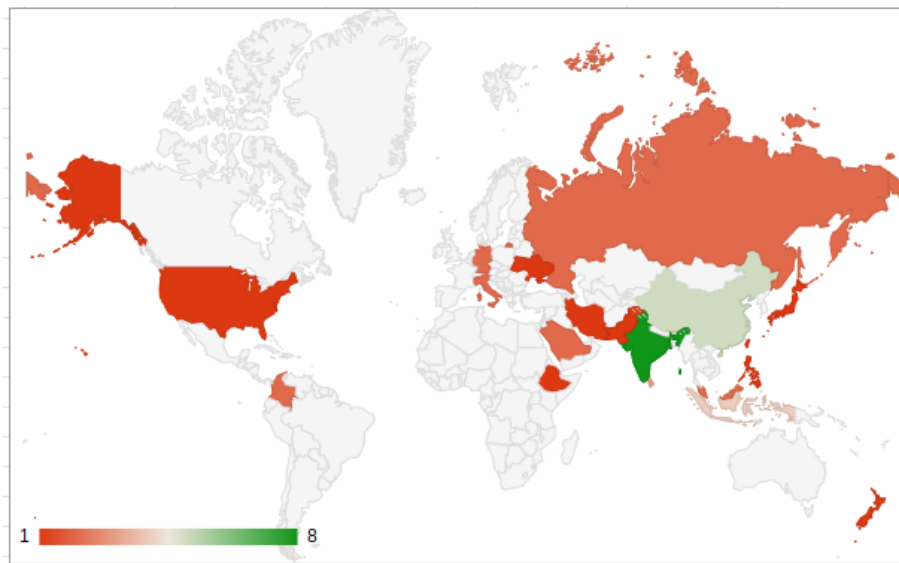
**Fonte: Autoria própria.**



### 3.3.5 QP5: Quando e onde têm sido publicados estes trabalhos?

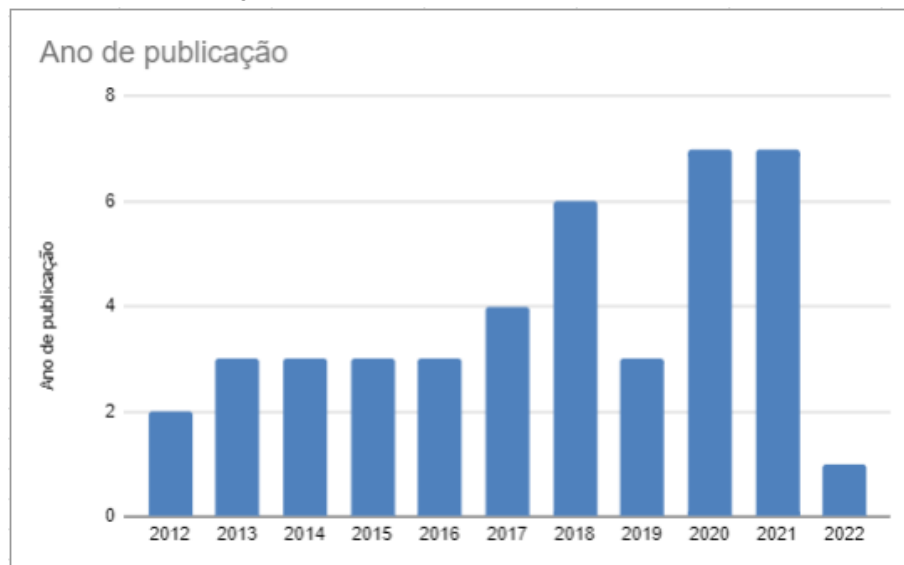
Já na quinta questão de pesquisa (QP5) buscou-se identificar quais países estão realizando pesquisas referente ao tema proposto. Foram identificados 20 países sendo a maioria deles pertencentes ao oriente. Os três países com mais publicações são Índia, China e Indonésia, com 8, 5 e 4 publicações respectivamente. A Figura 14 apresenta a disposição geográfica das publicações identificadas no mapeamento realizado.

**Figura 14 – Representação dos países**



Fonte: Autoria própria.

**Figura 15 – Representação do ano em que foram publicados os artigos seleccionados.**



Fonte: Autoria própria.

Até o momento da realização do mapeamento, não foram identificadas publicações provenientes do Brasil focadas na detecção e transcrição de textos Braille, o que ressalta ainda mais

a importância de pesquisas nessa área. Na literatura internacional, a temática de transcrição de textos Braille é explorada de forma mais ampla, sendo que a maioria desses trabalhos é de cunho acadêmico e ocorreu no período de 2017 a 2021, como percebido na Figura 15.

### 3.3.6 Conclusão

Com o mapeamento, podemos notar que Braille não é um conteúdo muito pesquisado na comunidade acadêmica, a quantidade de estudos é ainda menor quando se é relacionado a visão computacional.

Em suma, o mapeamento dos estudos existentes na área de visão computacional e Braille mostra que ainda há muito o que pesquisar nesta área. Esses estudos podem ajudar a melhorar a comunicação entre pessoas cegas e não-cegas, bem como ajudar as pessoas videntes a entender o texto em Braille.

## 4 METODOLOGIA

A seguir são apresentados aspectos relacionados à metodologia para o desenvolvimento da presente proposta. Na Seção 4.1, são detalhados os objetivos específicos apresentados.

### 4.1 Detalhamento das atividades desenvolvidas

#### 4.1.1 Seleção das bases de dados para testes e treinamento utilizando YOLO

Para que se tenha um comparativo, é necessário que se utilize base de dados conhecidas e utilizadas no meio acadêmico, para isso, nessa atividade selecionamos as bases de dados que possam nos fornecer um comparativo para avaliar a performance de nossa aplicação. Para a execução desse trabalho, foram selecionadas três bases de dados. Como demonstrado na Seção 2.1, utilizamos três bases com características diferentes para a execução desse trabalho.

- **DSBI:** Sendo a base de dados mais completa encontrada, os autores, além das imagens a serem utilizadas, disponibilizam anotações referentes às classes das celas Braille e as posições das caixas delimitadoras em cada uma das imagens fornecidas. O formato das anotações é dado da seguinte forma: As primeiras duas linhas representam a posição das linhas verticais e horizontais em relação a imagem. Cada linha subsequente é um conjunto de oito números, sendo os dois primeiros referente a posição das linhas verticais e horizontais e os próximos seis números são relacionados às celas Braille em si, sendo um para representar um ponto preenchido e zero para representar um ponto vazio.

**Figura 16 – Representação da sequencia utilizada nas celas Braille**



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

As anotações fornecidas pelos autores é diferente dos padrões requeridos para serem utilizados no framework YOLOv4, para que fosse possível fazer o treinamento utilizando a mesma base de dados, foi necessário realizar uma adaptação das anotações para o seguinte formato:

– `<classe> <x-central> <y-central> <largura> <altura>`

Onde “classe” representa qual objeto aquela linha está relacionada, “x-central” a posição no eixo horizontal da imagem, “y-central” a posição no eixo vertical, “largura” a largura total da caixa delimitadora e “altura” a altura total da caixa delimitadora. Para realizar a adaptação, foi desenvolvido um algoritmo utilizando a tecnologia Python, onde ele recebe como parâmetro as anotações providas por (LI *et al.*, 2018) e converte para o formato apresentado acima.

A Figura 17 representa como as anotações são fornecidas na base de dados DSBI, as duas primeiras linhas representam posições das linhas e colunas na imagem. Enquanto os números subsequentes representam as anotações das celas. Sendo que os dois primeiros números referem-se a qual índice de linha e coluna eles correspondem e os números subsequentes são relacionados aos pontos a sequência de pontos das celas, utilizando 1 para um ponto escrito e 0 para um ponto vazio, uma representação de como é a sequência utilizada pode ser vista na imagem 16.

**Figura 17 – Representação das anotações providas na base de dados**

2	69	89	116	136	164	184	211	231	258	278	306	326	353	373	400	420	448	471	500	521	552	573	606	626	658	679	709	731	
3	57	77	97	135	155	175	213	233	253	291	313	337	375	395	415	464	486	511	551	572	596	637	659	683	721	741	761	810	
4	4	9	1	1	0	1	1	0																					
5	4	10	0	1	1	0	1	0																					
6	4	11	1	0	0	1	1	0																					
7	4	12	0	0	1	1	1	1																					
8	4	14	0	1	0	1	0	0																					
9	4	15	1	0	0	0	0	0																					
10	4	16	0	0	1	1	1	0																					
11	4	17	0	1	1	0	0	0																					
12	4	19	1	1	1	1	0	1																					

**Fonte: Autoria própria.**

A Figura 18 é uma representação de como é a saída do algoritmo desenvolvido e também das anotações que são utilizadas para o treinamento utilizando a arquitetura YOLO. Após a adaptação das anotações, foi realizada uma verificação para certificar de que as anotações foram convertidas de maneira correta, validando se as posições das caixas delimitadoras estavam na posição esperada.

A Figura 19 demonstra como estão as caixas delimitadoras nas imagens utilizadas para o treinamento, podendo se perceber que estão na posição esperada e também demonstra uma variação de saturação realizada pela arquitetura para que seja obtido um melhor resultado.

Com a validação das anotações, foi possível iniciar os treinamentos utilizando a placa de vídeo GeForce RTX 3070, sendo realizadas diversas tentativas utilizando métodos de entradas diferentes para o treinamento. Iniciamos os treinamentos utilizando 30% do tamanho original da imagem, aumentando gradativamente em outros treinamentos

**Figura 18 – Representação das anotações convertidas para o formato requerido**

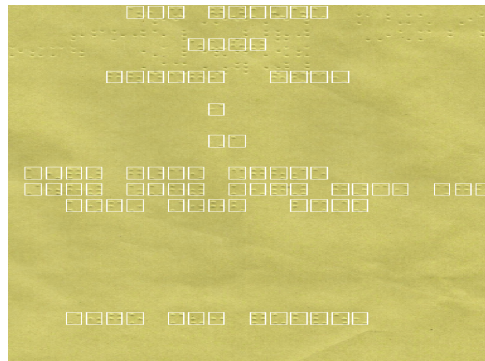
```

1 27 0.175294 0.059025 0.023529 0.025663
2 56 0.205294 0.059025 0.023529 0.025663
3 3 0.236471 0.059025 0.023529 0.025663
4 55 0.297059 0.059025 0.023529 0.025663
5 36 0.327647 0.059025 0.023529 0.025663
6 53 0.358235 0.059025 0.023529 0.025663
7 18 0.388824 0.059025 0.023529 0.025663
8 13 0.450000 0.059025 0.023529 0.025663
9 39 0.480588 0.059025 0.023529 0.025663
10 50 0.511176 0.059025 0.023529 0.025663
11 1 0.541765 0.059025 0.023529 0.025663
12 1 0.572353 0.059025 0.023529 0.025663
13 1 0.602941 0.059025 0.023529 0.025663
14 1 0.664118 0.059025 0.023529 0.025663
15 1 0.694706 0.059025 0.023529 0.025663
16 1 0.725882 0.059025 0.023529 0.025663
17 1 0.755882 0.059025 0.023529 0.025663
18 1 0.787059 0.059025 0.023529 0.025663
19 1 0.817059 0.059025 0.023529 0.025663

```

**Fonte: Autoria própria.**

**Figura 19 – Visualização de como são as anotações para o treinamento**



**Fonte: Autoria própria.**

até chegarmos ao limite de 45%, onde a placa de vídeo RTX 3070 começou a apresentar problemas por falta de memória RAM.

Para solucionar tal problema, fomos em busca de outros meios de realizar o treinamento dessa base. Através do Google Cloud Computing<sup>1</sup>, foi possível implementar o ambiente YOLOv4 e alugar uma placa de vídeo para processamento em nuvem, sendo ela uma GeForce Tesla T4, uma placa de vídeo que possui 16GB de memória, sendo o dobro disponível na RTX 3070, porém, possui um poder de processamento menor. Por ter mais memória, foi possível utilizar imagens com resoluções maiores nas bases de dados que possuem imagens grandes no aprendizado de máquina, a fim de se obter um melhor resultado na identificação e treinamento da rede neural.

<sup>1</sup> <https://cloud.google.com/compute/>

- **DONNIE BRAILLE:** As anotações disponibilizadas para a base de dados Donnie Braille estão no formato PASCAL VOC XML, uma representação pode ser visto na figura 20, um formato que se tornou muito conhecido na comunidade acadêmica de visão computacional por serem utilizados nos desafios criados pela (PASCAL, 2014) onde eles ofereciam as bases de dados, juntamente com as anotações em formato PASCAL VOC XML.

**Figura 20 – Representação das anotações em PASCAL VOC XML**

```

<annotation>
  <folder>definite_lego_images</folder>
  <filename>20180822_142204.jpg</filename>
  <path>/home/gabrielwh/dev/braille-scripts/definite_lego_images/20180822_142204.jpg</path>
  <source>
    <database>Unknown</database>
  </source>
  <size>
    <width>640</width>
    <height>480</height>
    <depth>3</depth>
  </size>
  <segmented>0</segmented>
  <object>
    <name>e</name>
    <pose>Unspecified</pose>
    <truncated>0</truncated>
    <difficult>0</difficult>
    <bndbox>
      <xmin>194</xmin>
      <ymin>319</ymin>
      <xmax>239</xmax>
      <ymin>370</ymin>
    </bndbox>
  </object>
</annotation>

```

**Fonte: Autoria própria.**

Para que fosse possível fazer o treinamento utilizando YOLOv4, também foi necessário realizar a conversão das anotações providas para o formato apresentado na Seção ?? através de um outro algoritmo feito em Python e fazer a validação se as anotações foram convertidas de maneira correta. Apesar das imagens já estarem em um tamanho adequado para o treinamento (640 x 480 pixels), a fim de otimizar o tempo de processamento, as imagens de entrada foram enviadas para treinamento com uma escala de 65% (416 x 320 pixels) de seu tamanho original.

- **BRAILLE CHARACTER DATASET:** Diferentemente das outras bases de dados utilizadas, esta base de dados oferece apenas as imagens, sem quaisquer anotações referente as caixas delimitadoras, para contornar tal problema, foi utilizado uma ferramenta

chamada YoloLabel<sup>2</sup> para criar as anotações de cada uma das imagens disponibilizadas. Além das anotações para as caixas delimitadoras, foi necessário separar e criar um arquivo para definir quais seriam as imagens utilizadas para treinamento e validação do treinamento, por se tratar de uma base de dados onde todas as imagens possuem um tamanho de 28 x 28 pixels, o tamanho original da imagem foi utilizado para treinamento, não havendo necessidade de mudar a sua escala para melhorar o tempo de treinamento.

#### 4.1.2 Realização do mapeamento sistemático acerca das soluções para reconhecimento de textos Braille

Um mapeamento sistemático é importante para identificar como estão sendo feitos os trabalhos focados na detecção e transcrição de celas Braille. O mapeamento pode ajudar os pesquisadores a entender quais técnicas foram usadas, quais foram os resultados obtidos e quais são os próximos passos. Além disso, o mapeamento também pode ajudar a identificar oportunidades de melhoria, como a criação de algoritmos mais eficazes e a melhoria dos sistemas de detecção. Realizamos um mapeamento sistemático com base em 1674 artigos encontrados. Os resultados obtidos são apresentados na Seção 3;

#### 4.1.3 Anotação das bases de dados de treinamento

Existem diversas arquiteturas que são utilizadas para a visão computacional, onde cada arquitetura possui um modelo de anotação diferente para as caixas delimitadoras. Sendo assim, é necessário adaptar as anotações para o formato aceito pela arquitetura YOLO ou então criá-las caso não sejam fornecidas, para que seja possível realizar o treinamento. Um exemplo de como esses dados foram obtidos está na Seção 2.1;

#### 4.1.4 Treinamento da YOLO utilizando as bases de dados

Nessa etapa, foram realizados os treinamentos das redes neurais utilizando as bases de dados selecionadas. Foram ajustados os parâmetros das redes para obtenção de melhores resultados no treinamento onde foram calculadas métricas como precisão e acurácia para avaliação dos resultados.

## 4.2 Tecnologias

Essa Seção apresentará as tecnologias e frameworks que foram utilizados.

<sup>2</sup> [https://github.com/developer0hye/Yolo\\_Label](https://github.com/developer0hye/Yolo_Label)

#### 4.2.1 Python

Python é uma linguagem de programação open-source de alto nível, simples e de fácil aprendizado. Python é famoso pela facilidade com que permite aos usuários importar as muitas bibliotecas disponíveis e muitas delas são adaptadas para programas de Machine Learning e Inteligência Artificial (IBM, 2019).

A linguagem foi utilizada no contexto deste trabalho principalmente para realizar a conversão das anotações providas pelas bases de dados caso necessário.

#### 4.2.2 YOLO

You Only Look Once(YOLO) é um sistema de detecção de objetos em tempo real de última geração que usa uma única rede neural para a imagem completa. Sua técnica é baseada em um problema de regressão para separar espacialmente as caixas delimitadoras e associar as probabilidades das classes, uma vez que todo o pipeline de detecção é uma única rede, pode ser otimizado ponta a ponta diretamente no desempenho da detecção(REDMON *et al.*, 2016).

O algoritmo YOLO é conhecido por ser capaz de detectar vários objetos em uma imagem de uma só vez, com uma taxa de detecção rápida e alta precisão. Ele funciona dividindo a imagem em uma grade de células e, em seguida, usando uma rede neural para prever a probabilidade de um determinado objeto estar presente em cada célula. Ele também é capaz de identificar a localização precisa dos objetos na imagem, permitindo que ele possa ser usado para desenhar caixas de detecção em torno dos objetos detectados.



## 5 DESENVOLVIMENTO

Esta seção foi dividida em três partes, sendo cada uma delas relacionada às bases de dados apresentadas na seção 2.1.

### 5.1 DSBI

Para realizar a conversão das anotações, foi desenvolvido um algoritmo em Python para ler os arquivos .txt das anotações, converter seu conteúdo para o formato desejado e salvar o resultado em outro arquivo também em .txt. O algoritmo foi desenvolvido de forma que os resultados possam ser adaptados para outros formatos de anotação, como XML, por exemplo. Além disso, foi utilizada a biblioteca Pandas para ler e manipular os dados das anotações.

O algoritmo lê cada linha do arquivo e separa os campos em variáveis, como o nome da entidade ou o tipo de anotação. Depois, o programa verifica qual é o formato de saída desejado e converte as informações para esse formato, salvando o resultado em um arquivo .txt.

Apesar da GeForce RTX 3070 ser uma placa de vídeo atual e com alto poder de processamento e memória, utilizar esta base de dados requer que se utilize uma escala da imagem muito pequena, ou, utilize uma subdivisão muito alta. Iniciou-se o treinamento com uma escala de 30% do tamanho original da imagem, e, ao perceber que a baixa resolução estava impactando no treinamento ao ponto da rede neural não reconhecer nenhuma cédula após várias iterações, foi-se aumentando gradativamente a proporção das imagens até chegar em 45% do tamanho total, sendo que o tempo de processamento estimado era de quase três meses e meio.

Por ainda não se ter obtido resultados satisfatórios, e, subdividir ainda mais a base de dados aumentaria muito o tempo de processamento, tornando a execução inviável, alugamos uma placa de vídeo para computação em nuvem através do Google Cloud, onde foi possível aumentar a escala das imagens para 55% do tamanho original, e, obtendo o mesmo resultado anterior. Aumentar a escala para 60%, onde foi possível notar os primeiros resultados positivos, teve uma previsão de tempo de processamento de 3 meses e meio e uma previsão de custos de \$650,00, tendo em vista que utilizou-se o processamento em nuvem, e se trata de um serviço pago, a continuação do treinamento foi inviável.

Para ser possível prosseguir com o treinamento, o projeto foi submetido no Edital Nº 29/2022 DIRGRAD/DIRGE-DV APOIO À EXECUÇÃO DE TCC, sendo este contemplado com o montante de R\$ 1000,00. Infelizmente, devido a todos os processos burocráticos existentes, o dinheiro foi repassado pela instituição somente alguns dias antes da entrega desse trabalho, o que impossibilitaria com que se fosse realizado um treinamento com o mínimo de iterações recomendadas. Mesmo assim, optou-se por iniciar o treinamento para que se obtivesse um resultado prévio.

## 5.2 Donnie Braille

Para esta base de dados, foi necessário realizar um ajuste no código de conversão utilizado para a base de dados DSBI, pois os arquivos de entrada estavam em outro formato, porém, como os arquivos de saída seriam os mesmos, o código pode ser reaproveitado.

Como esta base de dados utiliza uma resolução de 640 x 480 pixels totais, não é necessário utilizar uma placa de vídeo com muita memória RAM durante o treinamento, sendo assim, o treinamento foi realizado utilizando a placa de vídeo GeForce RTX 3070, que possui um poder maior de processamento, levando cerca de 82 horas para o treinamento total.

## 5.3 Braille Character Dataset

Diferentemente das outras bases, para a Braille Character Dataset foi necessário criar todas as anotações, sendo assim, foi utilizado YoloLabel para criar a anotação de uma cela Braille. A partir dessa anotação, como todas as imagens possuíam o mesmo tamanho e as imagens consistem apenas das celas Braille em si, a anotação foi replicada para todas as outras imagens existentes, possuindo como variável apenas a classe a qual a cela se referia.

Dentre todas as bases de dados utilizadas, além de ser a menor na proporção das imagens, também é a que teve o menor tempo de treinamento total, levando cerca de 40 horas para finalizar o treinamento.

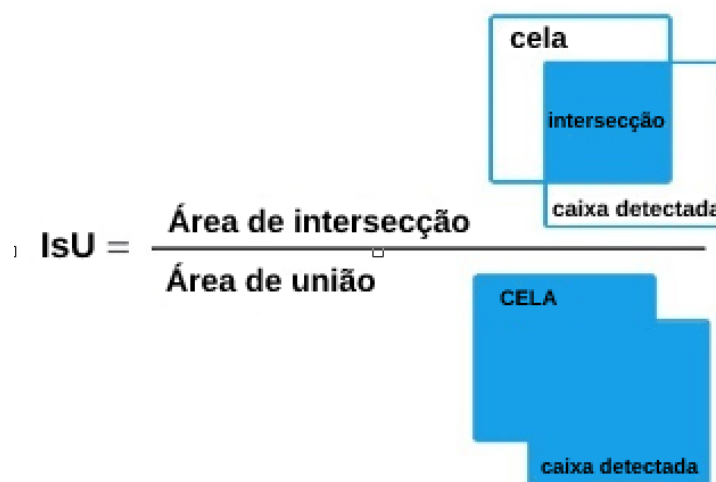
## 6 RESULTADOS

Para calcular a precisão de uma rede neural, YOLO utiliza um método chamado MAP (Precisão Média, do inglês *Mean Average Precision*), esse método de avaliação foi criado pela (PASCAL, 2014).

Seu método consiste em comparar a caixa delimitadora com a caixa detectada e retornar uma pontuação. Quanto maior a pontuação, mais preciso é o modelo em suas detecções, uma representação visual dessa comparação pode ser vista na figura 21. Isso é importante porque, apesar da detecção de objetos ter uma tendência a obter muitos falsos positivos (predição incorreta de um objeto) e uma taxa de detecção aceitável, um classificador pobre pode facilmente errar a classe de uma cela, ou seja, uma cela que representa a letra A pode ser classificado como um cela B ou qualquer outro valor.

Além da predição correta do objeto detectado, MAP leva em consideração quão próximo a caixa delimitadora gerada pela rede neural está da caixa delimitadora nas anotações. A Figura 21 representa como funciona a função de interseção sobre união, quanto mais próximo a caixa detectada estar da cela e maior for o ponto de interseção entre elas, menor vai ser a taxa de erro. A parte de resultados foi dividida em três tópicos, cada um deles relacionado a uma contribuição obtida durante o desenvolvimento da monografia.

**Figura 21 – Representação da interseção sobre união**



**Fonte: Autoria própria.**

### 6.1 Base de dados: DSBI

Devido às dificuldades encontradas e apresentadas na Seção 5, podemos perceber que YOLO-v4 não lida muito bem com imagens com grandes resoluções. Utilizar essa base de dados em questão, que possui imagens com resolução média de 1700 x 2300 pixels, requer um poder computacional muito grande ou então um tempo muito grande para realizar o treinamento.

É possível subdividir os lotes de imagens enviados para o processamento na placa de vídeo, assim, cada imagem ocuparia menos espaço na memória, porém, à medida que aumentam as subdivisões, o tempo necessário para o processamento aumenta proporcionalmente.

Até a data limite de entrega desse projeto, está sendo executado o treinamento da rede neural em nuvem, através do Google Cloud Computing. Foi alcançado um MAP de 19%, o que pode ser considerado um resultado ruim, pois, em média, a cada cinco palavras detectadas, apenas uma estaria correta. Porém, o resultado é promissor, tendo em vista que não houve tempo hábil para que nem 10% das iterações recomendadas fossem realizadas e se trata de um treinamento preliminar.

Isso indica que, caso fossem realizadas todas as iterações recomendadas, o resultado poderia ser ainda mais satisfatório. Além disso, o resultado obtido foi muito melhor que o esperado, pois apenas uma pequena parte do tempo de desenvolvimento foi gasta na realização das iterações. Logo, é possível concluir que a realização de iterações foi um bom investimento e que, com o tempo adequado, pode ser ainda mais eficaz. A Figura 22 mostra a linha do tempo do treinamento utilizando essa base. Como demonstra o gráfico, o treinamento ainda estava nas etapas iniciais e possuía um tempo estimado de treinamento de 3809 horas, ou seja, cerca de 158 dias restantes para o treinamento.

O alto tempo estimado se dá por conta da complexidade e tamanho das imagens utilizadas para o treinamento. Fazendo uma analogia simples, ensinar para uma criança o que é uma maçã é uma tarefa simples, porém, é mais complexo ensinar como diferenciar 64 tipos de maçãs diferentes. Para essa base de dados, além das imagens possuírem um tamanho médio de 1700 x 2300 pixels, também possui todas as 64 combinações possíveis para as celas Braille, justificando o tempo elevado necessário para o treinamento.

Um repositório contendo o arquivo de configuração e todas as anotações necessárias para realizar o treinamento utilizando a arquitetura YOLO criadas, é disponibilizado através de um repositório no Github<sup>1</sup>

## 6.2 Base de dados: Donnie Braille

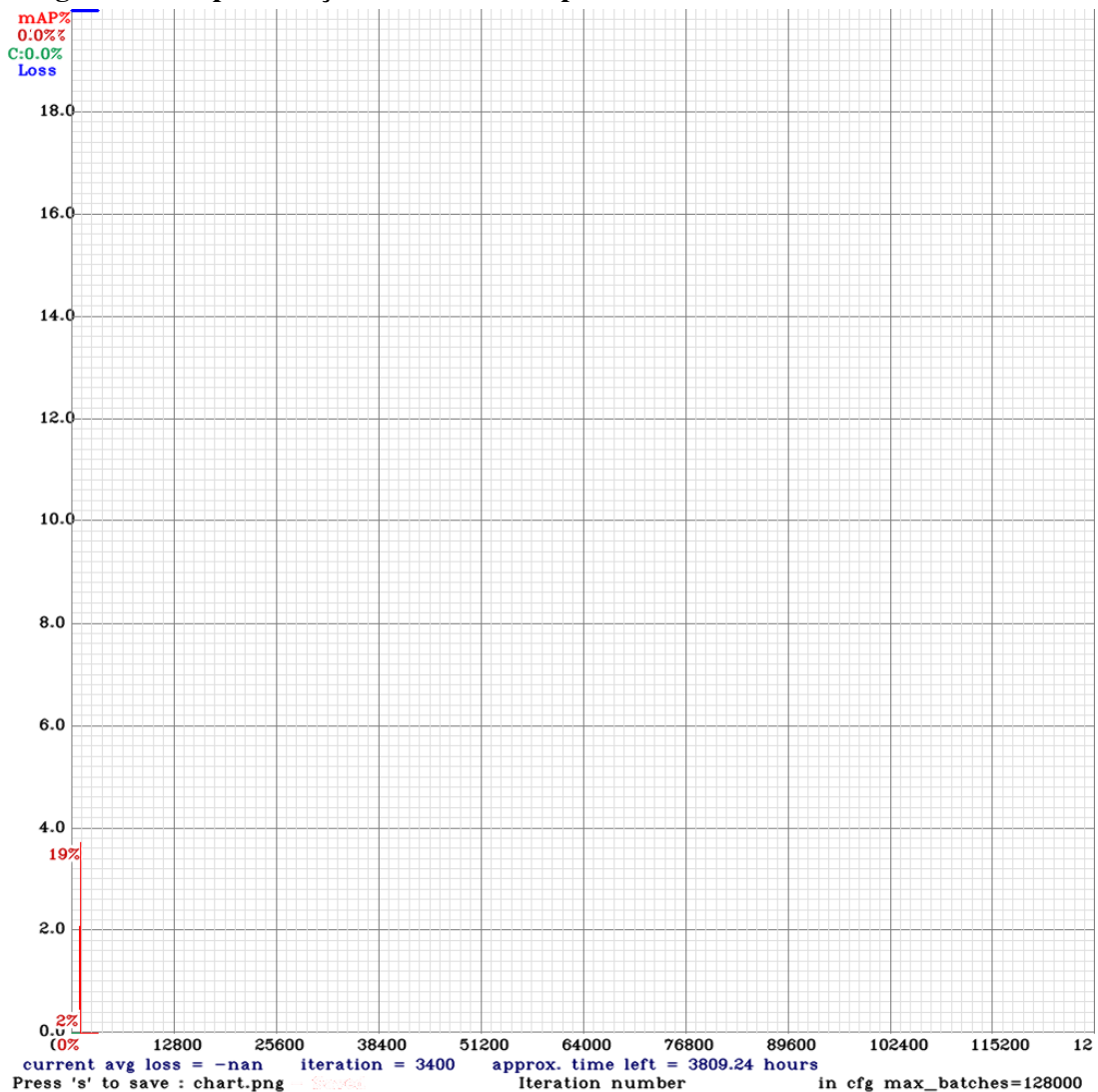
Ao utilizar a base de dados Donnie Braille, obtemos os melhores resultados possíveis, tendo um MAP de 100%. Como obtemos uma média de precisão de 100%, isso significa que além das predições estarem todas corretas, a posição das caixas de predição estavam muito próximas ou exatamente como esperado.

Na Figura 23, o eixo X representa a quantidade de iterações realizadas, o eixo Y representa a função de erro, que leva em alguns fatores, como erro de classificação e interseção sobre a união.

---

<sup>1</sup> <https://github.com/MessiasJunior/DSBI>

**Figura 22 – Representação da linha do tempo do treinamento utilizando a base DSBI**



Fonte: Autoria própria.

Um repositório contendo o arquivo de configuração e todas as anotações necessárias para realizar o treinamento utilizando a arquitetura YOLO criadas, é disponibilizado através de um repositório no Github<sup>2</sup>.

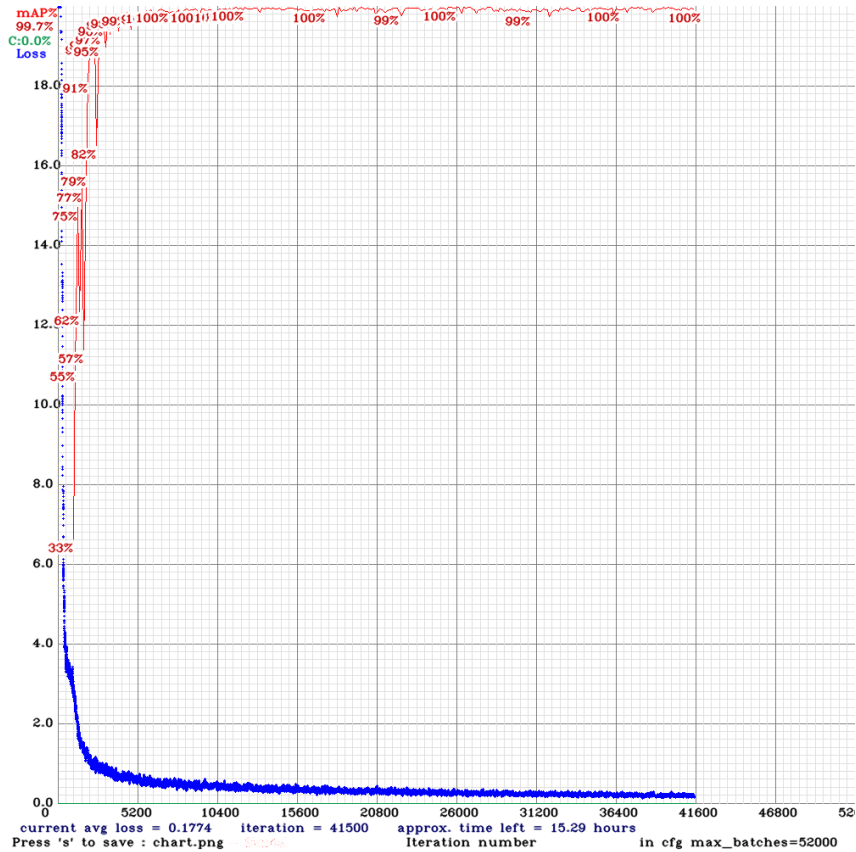
### 6.3 Base de dados: Braille Character Dataset

Diferentemente dos resultados obtidos com a base Donnie Braille, onde tivemos uma grande evolução e depois uma constância nos resultados, ao utilizar Braille Character Dataset tivemos um pico de 85% no MAP e podemos perceber uma grande inconsistência durante o treinamento.

Quando se trata de visão computacional, uma taxa de erro de 15% é considerada alta. No contexto em questão, uma cela detectada incorretamente pode dificultar ou até impossibilitar a

<sup>2</sup> <https://github.com/MessiasJunior/donniebraille>

**Figura 23 – Representação da linha do tempo do treinamento utilizando a base Donnie Braille**



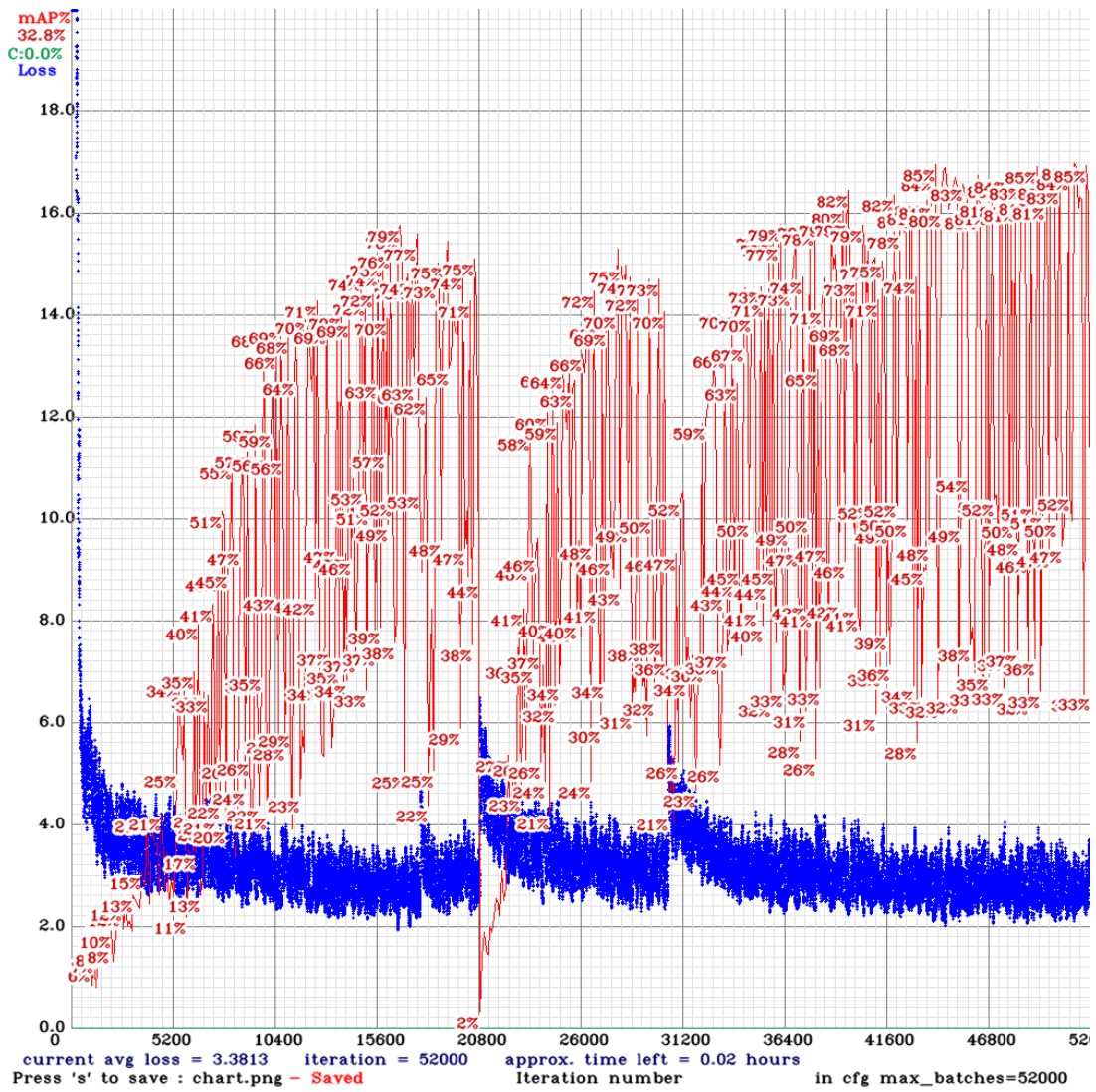
**Fonte: Autoria própria.**

transcrição correta das celas Braille, mesmo utilizando software para correções de ortografia. Uma alternativa para correção é a implementação em conjunto com a proposta de (ORTON-CELLI; MARCON; BEAL, 2021), que propõe correção de erros de transcrição de celas Braille.

Um repositório contendo o arquivo de configuração e todas as anotações necessárias para realizar o treinamento utilizando a arquitetura YOLO criadas, é disponibilizado através de um repositório no Github<sup>3</sup>

<sup>3</sup> <https://github.com/MessiasJunior/braillecharacterdata>

Figura 24 – Representação da linha do tempo do treinamento utilizando a base Braille Character Dataset



Fonte: Autoria própria.

## 7 CONCLUSÕES

Quando o trabalho foi iniciado, o objetivo era desenvolver um aplicativo capaz de traduzir células Braille para texto, no entanto, com todas as dificuldades e tempo gasto tentando obter um resultado válido utilizando a base de dados DSBI de (LI *et al.*, 2018), foi necessário alterar o escopo e validar se a arquitetura YOLO-v4 seria uma boa alternativa para se utilizar para o reconhecimento de textos Braille.

Por meio do mapeamento sistemático realizado, podemos concluir que Braille ainda é um tema pouco estudado na visão computacional, conseqüentemente, torna-se mais difícil encontrar outros estudos e base de dados para serem utilizados e comparados, justamente um dos problemas encontrados durante o desenvolvimento do trabalho.

Reconhecimento óptico Braille é um tema que merece atenção, por se tratar de um tema que propõe a inclusão de pessoas com deficiência visual. Os avanços tecnológicos possibilitam maior aproximação e interação com pessoas videntes através da conversão das células Braille para texto.

Apesar de não se ter obtido um resultado satisfatório em relação a detecção de células Braille utilizando a base de dados DSBI, o desenvolvimento foi importante para entendermos as necessidades e limitações do YOLO-v4 ao se utilizar imagens com grandes resoluções.

YOLO-v4 se provou um framework muito eficiente para reconhecimento de células Braille, principalmente pelo seu desempenho utilizando a base de dados “Donnie Braille”, onde obteve uma precisão média de 100%.

Sendo assim, podemos concluir que YOLO-v4 é uma ferramenta poderosa para a visão computacional, podendo ter resultados excelentes em um curto espaço de tempo, dependendo da base de dados a ser utilizada.



## REFERÊNCIAS

- BRASIL, M. d. E. **Deficiência Visual**. 2000. Disponível em: <http://portal.mec.gov.br/seed/arquivos/pdf/deficienciavisual.pdf>. Acesso em: 01 de outubro de 2021.
- BRASIL, M. d. E. **Grafia Braille para a Língua Portuguesa**. 2018. Disponível em: <http://portal.mec.gov.br/docman/dezembro-2018-pdf/104041-anexo-grafia-braille-para-lingua-portguesa/file>. Acesso em: 01 de outubro de 2021.
- FARINELLA, G. M.; LEONARDI, P.; STANCO, F. A mobile application for braille to black conversion. *In: SPRINGER. International Conference on Advanced Concepts for Intelligent Vision Systems*. [S.l.], 2015. p. 741–751.
- GONÇALVES, D. *et al.* Braille character detection using deep neural networks for an educational robot for visually impaired people. *In: SBC. Anais do XVI Workshop de Visão Computacional*. [S.l.], 2020. p. 123–128.
- HE, K. *et al.* Mask r-cnn. *In: Proceedings of the IEEE international conference on computer vision*. [S.l.: s.n.], 2017. p. 2961–2969.
- IBGE. **CENSO DEMOGRÁFICO 2010**. 2010. Disponível em: <https://www.ibge.gov.br/estatisticas/sociais/populacao/9662-censo-demografico-2010.html?edicao=9749&t=destaques>. Acesso em: 25 de novembro de 2021.
- IBM. **Why machine learning is primarily written in Python**. 2019. Disponível em: <https://developer.ibm.com/blogs/why-machine-learning-is-primarily-written-in-python/>. Acesso em: 25 de novembro de 2021.
- JIMÉNEZ, J. *et al.* Biography of louis braille and invention of the braille alphabet. **Survey of ophthalmology**, Elsevier, v. 54, n. 1, p. 142–149, 2009.
- JUSTIÇA, F. e. T. BRASIL. Secretaria da **Deficiência Visual**. s.d. Disponível em: <https://www.justica.pr.gov.br/Pagina/Deficiencia-Visual>. Acesso em: 25 de novembro de 2021.
- LI, R. *et al.* Dsbi: double-sided braille image dataset and algorithm evaluation for braille dots detection. *In: Proceedings of the 2018 the 2nd International Conference on Video and Image Processing*. [S.l.: s.n.], 2018. p. 65–69.
- LI, R. *et al.* Optical braille recognition based on semantic segmentation network with auxiliary learning strategy. *In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*. [S.l.: s.n.], 2020. p. 554–555.
- LIU, W. *et al.* Ssd: Single shot multibox detector. *In: SPRINGER. European conference on computer vision*. [S.l.], 2016. p. 21–37.
- ORTONCELLI, A. R.; MARCON, M.; BEAL, F. An automated approach to mitigate transcription errors in braille texts for the portuguese language. **arXiv preprint arXiv:2103.03935**, 2021.
- OTTAIANO, J. A. A. *et al.* **As Condições de Saúde Ocular no Brasil**. 2019. Disponível em: [https://www.cbo.com.br/novo/publicacoes/condicoes\\_saude\\_ocular\\_brasil2019.pdf](https://www.cbo.com.br/novo/publicacoes/condicoes_saude_ocular_brasil2019.pdf). Acesso em: 25 de novembro de 2021.

- OVODOV, I. G. Optical braille recognition using object detection neural network. *In: Proceedings of the IEEE/CVF International Conference on Computer Vision*. [S.l.: s.n.], 2021. p. 1741–1748.
- PASCAL. **PASCAL Visual Object Classes**. 2014. Disponível em: <http://host.robots.ox.ac.uk/pascal/VOC/>. Acesso em: 15 de abril de 2022.
- REDMON, J. *et al.* You only look once: Unified, real-time object detection. *In: Proceedings of the IEEE conference on computer vision and pattern recognition*. [S.l.: s.n.], 2016. p. 779–788.
- REN, S. *et al.* Faster r-cnn: Towards real-time object detection with region proposal networks. **Advances in neural information processing systems**, v. 28, p. 91–99, 2015.
- RONNEBERGER, O.; FISCHER, P.; BROX, T. U-net: Convolutional networks for biomedical image segmentation. *In: SPRINGER. International Conference on Medical image computing and computer-assisted intervention*. [S.l.], 2015. p. 234–241.
- SHANKS0465. **Braille Character Dataset**. 2019. Disponível em: <https://www.kaggle.com/datasets/shanks0465/braille-character-dataset>. Acesso em: 15 de abril de 2022.