



UNIVERSIDADE FEDERAL DE SERGIPE
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA
DEPARTAMENTO DE COMPUTAÇÃO

Avaliação de técnicas para detecção de produtos e preços em folhetos digitais de supermercados

Trabalho de Conclusão de Curso

Mayara de Santana Barbosa



São Cristóvão – Sergipe

2022

UNIVERSIDADE FEDERAL DE SERGIPE
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA
DEPARTAMENTO DE COMPUTAÇÃO

Mayara de Santana Barbosa

**Avaliação de técnicas para detecção de produtos e preços em
folhetos digitais de supermercados**

Trabalho de Conclusão de Curso submetido ao Departamento de Computação da Universidade Federal de Sergipe como requisito parcial para a obtenção do título de Bacharel em Ciência da Computação.

Orientador(a): Leonardo Nogueira Matos
Coorientador(a): Thiago Dias Bispo

São Cristóvão – Sergipe

2022

Resumo

Os folhetos de supermercados são uma das mídias mais importantes para apresentar ofertas, todavia seu formato não estruturado dificulta a comparação de valores entre produtos de diferentes supermercados. Com o propósito de facilitar esse processo de comparar preços, surgiu o LudiiPrice, um aplicativo desenvolvido na UFS, sob coordenação dos orientadores desta monografia, que facilita o monitoramento de preços entre diversos estabelecimentos comerciais. Sua base de dados é construída através da digitalização colaborativa de notas fiscais e da mineração de texto de preços divulgados na Internet. O presente trabalho integra o LudiiPrice, com a finalidade de possibilitar também o uso de imagens de folhetos como fonte de dados para o aplicativo. Com o objetivo de localizar os preços e produtos nos folhetos comerciais, foram analisadas técnicas de *deep learning* para detecção de objetos em imagens, área que apresentou um grande crescimento nas últimas décadas. Para o treinamento do modelo de detecção foram rotuladas 900 imagens de folhetos de 20 supermercados distintos, as classes utilizadas foram Preço, Descrição, Imagem e Oferta. Utilizando a base rotulada, treinamos os modelos de detecção YOLOv5 e SSD e, através de uma análise comparativa, selecionamos o primeiro para a construção do detector. Em seguida, construímos algumas regras para melhorar o desempenho do nosso modelo. Com isso, o modelo proposto conseguiu atingir 79% de *average precision* para descrição, 93% para imagem, 93% para oferta e 86% para preço.

Palavras-chave: detecção de objetos. aprendizado profundo. folhetos.

Abstract

Supermarket flyers are one of the most important ways to communicate offers, however its unstructured format makes the value comparison between products from different stores arduous. With smoothing the price comparison process in mind, Ludiiprice emerged, an application developed at UFS, under this monography's advisors supervision, that eases the price monitoring between several comercial estabilishments. Its database is build through the collaborative digitalization of invoices and the text mining of prices published on the Internet. The present paper integrates LudiiPrice, with the finality of enabling flyer images as data sources for the application as well. In order to detect the prices and products on commercial flyers, we will analyze deep learning object detection technics on images, area that showed an significantly increase in the last decades. For the detection model training, 900 images of flyers from 20 different supermarkets were labeled, the classes used were Price, Description, Image and Offer. Using the labeled data, we trained the YOLOv5 and SSD detection models and, through from a comparative analysis, we selected the first for the construction of the detector. Then, we built some rules to improve the performance of our model. In this way, the model proposed had 79% of average precision for description, 93% for image, 93% for offer and 86% for price.

Keywords: object detection. deep learning. flyers.

Lista de ilustrações

Figura 1 – Funcionamento do YOLO	15
Figura 2 – Comparação das arquiteturas do YOLOv5	16
Figura 3 – Intersection Over Union	17
Figura 4 – Gráfico de interpolação	18
Figura 5 – Algoritmo de classificação de folhetos	21
Figura 6 – Algoritmo proposto para mesclar	22
Figura 7 – Etapas da abordagem	23
Figura 8 – Planos de recursos	23
Figura 9 – Etapas da nova abordagem	25
Figura 10 – Página do supermercado	27
Figura 11 – Estrutura da página do supermercado	28
Figura 12 – Página mensal de folhetos	28
Figura 13 – Página individual do folheto	29
Figura 14 – Exemplo de rotulação	30
Figura 15 – Interface do Dataturks	31
Figura 16 – Detecção de descrição	35
Figura 17 – Ofertas duplicadas	36
Figura 18 – Classes fora de ofertas	37
Figura 19 – Erro de categorias duplicadas	38
Figura 20 – Erro de preço	40
Figura 21 – Erro de descrição	40
Figura 22 – Erro de descrição em imagem	42
Figura 23 – Detecção de imagem	56

Lista de quadros

Quadro 1 – Outros supermercados selecionados	31
--	----

Lista de tabelas

Tabela 1 – Quantidade de artigos encontrados por base	20
Tabela 2 – Matriz de confusão	24
Tabela 3 – Métricas obtidas	24
Tabela 4 – Matriz de confusão na nova abordagem	25
Tabela 5 – Métricas obtidas na nova abordagem	25
Tabela 6 – Quantidade de imagens por supermercado selecionado	29
Tabela 7 – Métricas Yolo	34
Tabela 8 – Métricas SSD	34
Tabela 9 – Métricas Yolo pós regras	43
Tabela 10 – Métricas com o novo treino	44
Tabela 11 – Métricas com a nova base de treino com aplicação das regras	44
Tabela 12 – Métricas com a nova base de teste	44
Tabela 13 – Métricas com a nova base de teste com aplicação das regras	44
Tabela 14 – Métricas com novos supermercados	45
Tabela 15 – Métricas com novos supermercados após regras	45
Tabela 16 – Métricas com supermercados conhecidos	45
Tabela 17 – Métricas com supermercados desconhecidos	46
Tabela 18 – Métricas com a detecção em ofertas	55

Lista de abreviaturas e siglas

AP	<i>Average Precision</i>
API	<i>Application Programming Interface</i>
AWS	<i>Amazon Web Services</i>
CNN	<i>Convolutional Neural Network</i>
COCO	<i>Common Objects in Context</i>
CTPN	<i>Connectionist Text Proposal Network</i>
GPU	<i>Graphics Processing Unit</i>
HTML	<i>HyperText Markup Language</i>
IEEE	<i>Institute of Electrical and Electronic Engineers</i>
IOU	<i>Intersection Over Union</i>
MAP	<i>Mean Average Precision</i>
MLP	<i>Multi-Layer Perceptron</i>
OCR	<i>Optical Character Recognition</i>
PDF	<i>Portable Document Format</i>
QR	<i>Quick Response</i>
RPN	<i>Region Proposal Network</i>
SSD	<i>Single Shot Detector</i>
SVM	<i>Support Vector Machine</i>
TCC	Trabalho de Conclusão de Curso
UFS	Universidade Federal de Sergipe
URL	<i>Uniform Resource Locator</i>
YOLO	<i>You Only Look Once</i>

Sumário

1	Introdução	10
1.1	Objetivos	11
1.1.1	Objetivos específicos	11
1.2	Metodologia	11
1.3	Estrutura do documento	12
2	Fundamentação Teórica	13
2.1	<i>Web crawler</i>	13
2.2	Deteção de objetos	13
2.2.1	R-CNN, Fast R-CNN e Faster R-CNN	14
2.2.2	YOLO	14
2.2.3	SSD	16
2.2.4	Métricas para avaliação de modelos de detecção	16
2.2.4.1	<i>Intersection Over Union</i>	16
2.2.4.2	<i>Precision, recall</i> e <i>F1-score</i>	17
2.2.4.3	<i>Average precision</i>	18
3	Trabalhos Relacionados	20
3.1	Estratégia de busca	20
3.2	Recognition and Classifying Sales Flyers Using Semi-Supervised Learning	21
3.3	Content Extraction from Marketing Flyers	23
3.4	Using Convolutional Neural Networks for Content Extraction from Online Flyers	24
4	Construção da Base	26
4.1	<i>Web crawler</i>	26
4.2	Seleção das imagens	27
4.3	Rotulação	29
5	Desenvolvimento e Resultados	32
5.1	Ferramentas utilizadas	32
5.2	Avaliação dos modelos	33
5.2.1	YOLOv5	33
5.2.2	SSD	34
5.2.3	Comparação dos modelos	34
5.3	Regras para aperfeiçoamento do modelo	35
5.3.1	Regra 1 - Remover ofertas contidas em ofertas	35

5.3.2	Regra 2 - Remover descrição, preço e imagens não contidos em uma oferta	37
5.3.3	Regra 3 - Remover <i>bounding boxes</i> de mesma categoria com grande intersecção dentro de uma oferta	38
5.3.4	Regra 4 - Remover preço ou descrição quando existe grande interseção entre os dois e existe outro da mesma categoria na oferta	39
5.3.5	Regra 5 - Remover imagem ou descrição quando existe grande interseção entre as duas e existe outra da mesma categoria na oferta	41
5.3.6	Resultados após aplicação das regras	42
5.4	Nova fase de rotulação	43
5.4.1	Base de treinamento	43
5.4.2	Base de teste	44
5.4.3	Novos supermercados	44
5.4.4	Experimento com ofertas	45
5.5	Resultados	45
6	Considerações Finais	47
	Referências	48
	Apêndices	50
	APÊNDICE A Manual para Rotulação	51
	APÊNDICE B Experimento de Detecção em Ofertas	55

1

Introdução

A comparação de preços é uma das maneiras mais assertivas para apoiar a realização de compras mais inteligentes, “o consumidor deve pesquisar preços, estar preocupado em manter seu padrão de consumo, comparando preços e adequando o consumo à sua realidade”(GANDRA, 2015). Os folhetos de supermercados são uma das mídias mais importantes para apresentar promoções de varejo, seu impacto é resultado da motivação dos consumidores de realizarem compras com melhor valor, indica Luceri et al. (2014). Entretanto, as ofertas não estruturadas presentes nesses folhetos dificultam a comparação entre os produtos vendidos em lojas físicas ou online.

O LudiiPrice é um aplicativo para dispositivos Android desenvolvido originalmente por dois alunos de TCC ao longo do ano de 2020, orientados pelos mesmos orientadores deste. Este aplicativo vem continuamente sendo aperfeiçoado através de outros TCC que se seguiram desde então. A ideia aprimorada em cada um destes trabalhos é realizar um buscador de preços para produtos de supermercado. Na primeira versão do aplicativo, a coleta de dados dos preços praticados era feita pela leitura do código QR dos cupons fiscais. Sendo assim, o sistema funcionava como uma plataforma colaborativa que oferecia respostas às consultas mas requeria que os usuários em troca alimentassem a base de dados através da nota fiscal eletrônica. Este tipo de aplicativo exige muito engajamento da comunidade de usuários.

Para manter uma base de dados atualizada, capaz de capturar as flutuações de preços ao longo do tempo, dimensão temporal, e as discrepâncias em uma ampla região geográfica, seria necessário ter muitos usuários constantemente usando o aplicativo. Ainda, mais do que somente fazer consultas, os usuários precisariam, principalmente, fazer a leitura do código QR dos cupons fiscais. O presente trabalho integra o LudiiPrice com a finalidade de fornecer uma nova fonte de dados para a aplicação, imagens de folhetos de supermercados divulgadas na Internet e, viabilizar a comparação entre estabelecimentos que divulgam suas ofertas utilizando exclusivamente esse formato. Para esse propósito, serão analisadas técnicas de *deep learning*, tema muito discutido na

última década e com grandes avanços apresentados nas tarefas de localização de texto e objetos a partir de imagens e documentos digitais, como folhetos de venda (MOSQUERA; GENC, 2019).

A área de detecção de objetos apresentou um grande crescimento nas últimas décadas, seja com a classificação de objetos em uma simples imagem ou até mesmo em casos mais complexos, como a localização e classificação de vários objetos em uma mesma imagem. Entretanto, a detecção em folhetos de supermercados apresenta um grande desafio por reunir uma grande variedade de fontes, imagens, cores e tamanhos que precisam ser identificados, como explicam Mosquera e Genc (2019). Para vencer esse obstáculo, neste trabalho, avaliamos os modelos de última geração na área de detecção de objetos, como é o caso da família de modelos YOLO - You Only Look Once (REDMON et al., 2016) para que possamos identificar com precisão produtos e seus preços, facilitando a comparação de valores a partir de folhetos de supermercados.

1.1 Objetivos

O objetivo geral deste trabalho é avaliar técnicas de *deep learning* e desenvolver um modelo de detecção de produtos e preços em folhetos digitais, isto é, imagens dos encartes de papel veiculados pelos supermercados, com a finalidade de ampliar a base de dados do LudiiPrice, além de facilitar a comparação de preços entre supermercados que divulgam seus anúncios exclusivamente através desse meio.

1.1.1 Objetivos específicos

- Criar um *web crawler* para obter imagens de folhetos de supermercados divulgados na Internet;
- Construir uma base de dados de folhetos;
- Desenvolver um modelo de detecção de preços e produtos em folhetos;

1.2 Metodologia

Para a obtenção dos resultados, o presente trabalho emprega uma metodologia que consiste em 4 etapas, descritas a seguir:

1. *Web crawler*: Esta etapa consiste no desenvolvimento de um *web crawler*, responsável por extrair folhetos digitais de diversos supermercados para a construção da base de dados a ser utilizada no treinamento do modelo que será desenvolvido. Para isso, foram selecionados os *websites* que reuniam a maior quantidade e variedade de folhetos e uma análise foi feita sobre a estrutura do código HTML dos *websites* para a adequação do *web crawler*.

2. Base de Dados: Nesta fase, foram selecionadas, de forma aleatória, imagens dos folhetos dos supermercados que obtiveram mais resultados na etapa anterior, em que cada imagem correspondia a uma página de um folheto. Além disso, os anúncios contidos em cada imagem foram rotulados em: preço, descrição, imagem e oferta. Isso foi feito por meio da criação de *bounding boxes* em cada imagem, fazendo o uso de um software adequado para anotação e rotulagem de dados.
3. Detecção: Nesta etapa, foram explorados modelos de detecção de imagens do estado da arte. Realizamos o treinamento e comparação das métricas obtidas com a utilização de alguns modelos. A partir dos resultados, selecionamos o modelo para a construção do detector, realizamos experimentos e construímos algoritmos para o aperfeiçoamento dos resultados do modelo selecionado.

1.3 Estrutura do documento

Este documento está organizado nos seguintes capítulos:

- Capítulo 1 - Introdução: Seção atual, contendo motivação, objetivos e metodologia a ser utilizada;
- Capítulo 2 - Fundamentação Teórica: Contém a apresentação dos conceitos e tecnologias existentes para facilitar o entendimento do trabalho;
- Capítulo 3 - Trabalhos Relacionados: Corresponde a uma revisão da literatura sobre o tema estudado;
- Capítulo 4 - Construção da Base: Aborda o processo de obtenção, seleção e rotulação das imagens para construção da base de dados;
- Capítulo 5 - Desenvolvimento e Resultados: Discute as ferramentas utilizadas para treinamento e avaliação dos modelos, o processo de treinamento, estratégias utilizadas para o aprimoramento das métricas obtidas e a exposição dos resultados alcançados;
- Capítulo 6 - Considerações Finais: Apresenta os resultados conquistados, limitações e sugestões para trabalhos futuros.

2

Fundamentação Teórica

Com o objetivo de facilitar a compreensão do presente trabalho, este capítulo apresenta conceitos básicos que serão abordados no texto, além de modelos existentes para detecção de objetos e métricas para avaliação desses.

2.1 *Web crawler*

Um *web crawler* é um software que percorre a World Wide Web e baixa documentos de maneira metódica e automatizada. (KAUSAR; DHAKA; SINGH, 2013). Ele percorre várias páginas a partir de uma página web inicial, para isso, utiliza a estrutura HTML do *website*, realizando o download e indexação da página atual, para facilitar o resgate quando for solicitado. Neste momento os links presentes são salvos e uma verificação é realizada para confirmar se as páginas com as URLs salvas foram baixadas ou não, como apontam Kausar, Dhaka e Singh (2013). Este comportamento será proveitoso na obtenção das imagens para formação da base de dados, pois impedirá a duplicação de downloads de imagens já obtidas, além de que a possibilidade de navegar entre as páginas facilitará a extração automatizada.

2.2 Detecção de objetos

De acordo com VENTURIERI e Santos (1998) a classificação de imagens consiste no “estabelecimento de um processo de decisão no qual um grupo de *pixels* é definido como pertencente a uma determinada classe”. Esse processo torna-se consideravelmente complicado com o crescimento do número de classes buscadas e também quando vários objetos de diferentes categorias estão presentes na imagem (DRUZHKOVA; KUSTIKOVA, 2016).

O problema da detecção de objetos refere-se a identificação da classe dos objetos, contudo, encarrega-se também de determinar as coordenadas desses objetos na imagem. Como vários

fatores devem ser considerados, como variedade de formas, cores, iluminação, perspectiva, etc., a detecção de objetos é um grande desafio, como destacam [Druzhkov e Kustikova \(2016\)](#). São exemplos de detectores de objetos: o Faster R-CNN ([REN et al., 2015](#)), YOLO ([REDMON et al., 2016](#)) e SSD ([LIU et al., 2016](#)).

2.2.1 R-CNN, Fast R-CNN e Faster R-CNN

Uma das abordagens para a detecção de imagens baseia-se na proposta de regiões. Os modelos que empregam essa estratégia recebem uma imagem e retornam um conjunto de regiões que provavelmente conterão objetos ([HOSANG et al., 2015](#)). O R-CNN ([GIRSHICK et al., 2016](#)) é um dos modelos que utiliza essa técnica e faz uso do algoritmo de busca seletiva ([UIJLINGS et al., 2013](#)) para obter as possíveis coordenadas. Em seguida, cada proposta é redimensionada para uma imagem de tamanho fixo que alimenta um modelo *convolutional neural network* (CNN) para extrair as *features*. Finalmente, classificadores *support vector machine* (SVM) são usados para reconhecer as categorias dos objetos.

Embora a R-CNN proporcione bons resultados, o cálculo para obter as *features* leva a uma velocidade de detecção extremamente lenta ([ZOU et al., 2019](#)). Com o objetivo de melhorar o tempo de execução, surge o Fast R-CNN ([GIRSHICK, 2015](#)), permitindo treinar simultaneamente a rede de detecção e classificação, além de adicionar a etapa de regressão ao treinamento, resultando numa redução do tempo e aumento da precisão em comparação com o modelo anterior.

Apesar do Fast R-CNN apresentar melhorias significativas, o tempo gasto na proposta de regiões é o gargalo desse sistema de detecção ([REN et al., 2015](#)). A principal contribuição do Faster-RCNN é a substituição do algoritmo de busca seletiva pela *region proposal network* (RPN), que permite propostas de região quase sem custo ([ZOU et al., 2019](#)). Além disso, [Ren et al. \(2015\)](#) integraram todas as etapas de aprendizado, tornando a detecção ainda mais rápida e precisa.

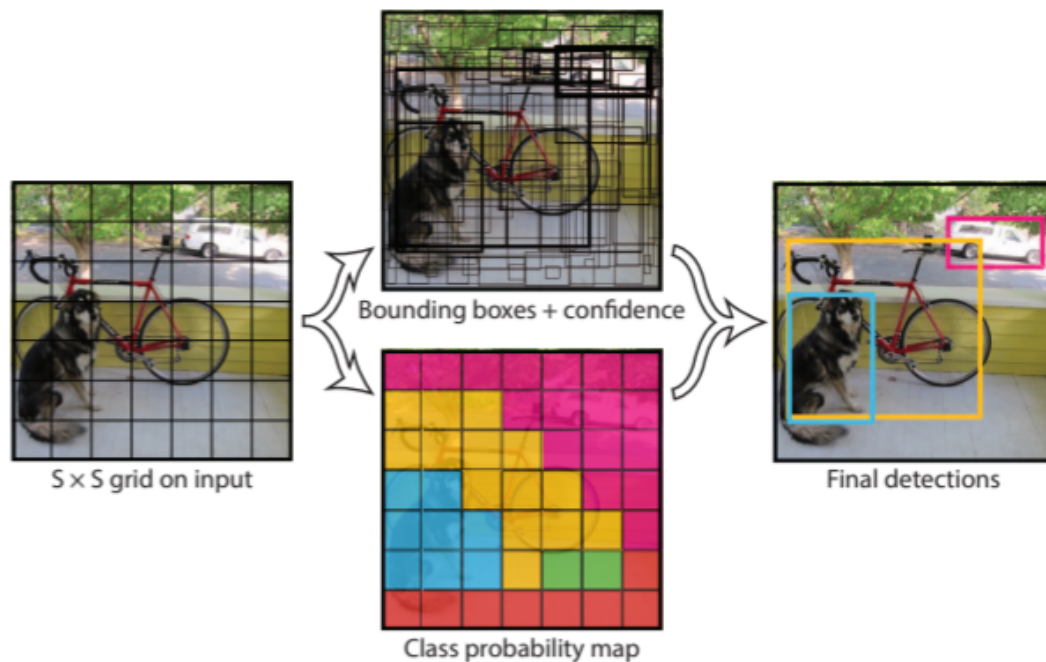
2.2.2 YOLO

O YOLO ([REDMON et al., 2016](#)): You Only Look Once é um detector que utiliza a abordagem de um estágio. Nessa técnica, uma única rede neural é responsável por detectar regiões e as categorias dos objetos simultaneamente. Por essa razão, o YOLO conseguiu manter acurácia e alcançar uma velocidade superior às técnicas anteriores.

O funcionamento do YOLO se inicia na divisão da imagem em uma grade de tamanho $S \times S$, em seguida, cada uma das células da grade prevê um *bounding box*, composto das coordenadas (x, y) representando o centro da caixa, além da sua altura e largura. Além disso, a célula é responsável por prever uma pontuação de confiança, que indica o grau de certeza do *bounding box* previsto conter um objeto.

Paralelamente, uma previsão de classe também é feita. Para cada classe possível é fornecida uma probabilidade. A combinação do grau de confiança e da classe prevista resulta em uma pontuação final, responsável por determinar a probabilidade do *bounding box* conter um objeto específico. As etapas descritas anteriormente são representadas na figura [Figura 1](#).

Figura 1 – Funcionamento do YOLO

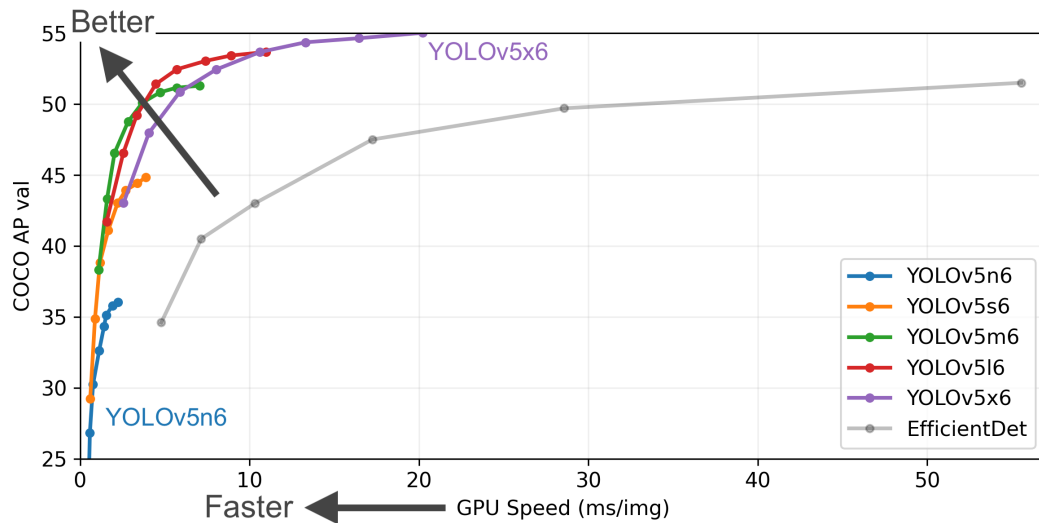


Fonte: ([REDMON et al., 2016](#))

Diferentemente da utilização das propostas de regiões ou de janelas deslizantes, que consiste em percorrer a imagem com um retângulo e aplicar uma função classificadora a cada uma dessas sub-imagens ([LAMPERT; BLASCHKO; HOFMANN, 2008](#)), YOLO enxerga a imagem inteira durante o estágio de treinamento e teste, o que permite a análise do contexto, reduzindo erros de fundo, por exemplo, como acontece com o Fast R-CNN ([REDMON et al., 2016](#)).

Ao longo dos anos, novas versões do YOLO foram criadas com o objetivo de aprimorar o modelo original. Atualmente, o YOLOv7 é a versão mais recente. A YOLOv5 ([ULTRALYTICS, 2020](#)) foi a primeira implementação desenvolvida utilizando PyTorch. Ela possui cinco arquiteturas, a nano (YOLOv5n), *small* (YOLOv5s), *medium* (YOLOv5m), *large* (YOLOv5l) e *xlarge* (YOLOv5x) e a principal diferença entre elas consiste no número de módulos de extração de *features* ([HORVAT; GLEDEC, .](#)). A [Figura 2](#) exibe um gráfico comparativo das cinco arquiteturas utilizando o mAP obtido com o dataset COCO val2017.

Figura 2 – Comparação das arquiteturas do YOLOv5



Fonte: (ULTRALYTICS, 2020)

2.2.3 SSD

O Single Shot Detector (SSD) (LIU et al., 2016) é um detector de objetos que também utiliza um único estágio, resultando em economia de tempo de processamento sem abrir mão de uma alta precisão na detecção. Para atingir isso, o SSD elimina as etapas de geração de propostas de regiões e agrega todas as fases de processamento em uma única rede, gerando previsões em diferentes proporções e escalas por localização das *features*.

Esse modelo é formado por uma rede de classificação que é responsável por extrair as *features* e uma ou mais camadas convolucionais que resultam nos *bounding boxes* e classes dos objetos detectados na imagem. Seu funcionamento é baseado nas redes convolucionais que produzem *bounding boxes* de vários tamanhos e um *score* de confiança sobre a presença de cada classe de objeto nos *boxes*, além de uma etapa de supressão para realizar ajustes nos *bounding boxes* para melhor corresponder à forma dos objetos. Por combinar previsões de vários *feature maps* com diferentes resoluções, o modelo é capaz de lidar com objetos de vários tamanhos. (LIU et al., 2016).

2.2.4 Métricas para avaliação de modelos de detecção

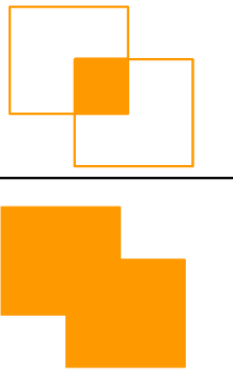
Nesta seção apresentaremos algumas métricas utilizadas para avaliação de modelos de detecção de objetos, além de conceitos necessários para a compreensão dessas.

2.2.4.1 Intersection Over Union

A *intersection over union* (IOU) é a métrica de avaliação mais popular usada nos *benchmarks* de detecção de objetos (REZATOFIGHI et al., 2019). Ela mede a interseção da área

entre o *bounding box* previsto e o esperado, dividido pela área de união de ambos. Quanto mais próximo de um for a IOU, mais fiel a detecção foi à rotulação. A representação gráfica desse cálculo pode ser observado na [Figura 3](#).

Figura 3 – Intersection Over Union

$$IOU = \frac{\text{area of overlap}}{\text{area of union}} = \frac{\text{Diagrama de IOU}}{\text{Diagrama de Área de União}}$$


Fonte: Adaptado de ([PADILLA; NETTO; SILVA, 2020](#))

Esse conceito é importante para definir se uma previsão é correta ou não. Para isso, é definido um limiar, caso a IOU obtida ultrapasse esse valor, a detecção é considerada correta, caso contrário é tida como incorreta.

2.2.4.2 *Precision, recall e F1-score*

Para entender o cálculo da *precision* e do *recall*, é necessário conhecer alguns conceitos:

- *True positive* (TP): Uma detecção correta, ou seja, IOU obtido foi maior que o limiar definido.
- *False positive* (FP): Uma detecção de objeto não existente ou detecção incorreta de objeto existente, IOU abaixo do limiar.
- *False negative* (FN): Detecção ausente de um objeto existente.

Precision (precisão) é a proporção de detecções corretas dentre todas as previstas pelo modelo, é uma métrica que calcula o quanto o modelo é capaz de detectar corretamente os objetos existentes. Seu cálculo é realizado da seguinte forma:

$$P = \frac{TP}{TP + FP} \quad (2.1)$$

O *recall* (cobertura) é a proporção de detecções corretas dentre todas as ocorrências reais, ele mede a capacidade do modelo de obter o máximo das detecções esperadas. A fórmula para calcular essa métrica é a seguinte:

$$R = \frac{TP}{TP + FN} \quad (2.2)$$

Outra métrica é a *F1-score*, média harmônica entre o *recall* e a *precision*. Seu objetivo é combinar as duas métricas anteriores em uma só, facilitando a avaliação do modelo. A fórmula do seu cálculo é a seguinte:

$$F1\text{-score} = 2 * \frac{P * R}{P + R} \quad (2.3)$$

2.2.4.3 Average precision

Outra métrica utilizada para avaliar modelos de detecção de objetos é a *average precision* (AP). Ela é obtida calculando-se a área debaixo da curva *precision-recall*, curva que mostra o contrapeso entre os valores entre *precision* e *recall* para diferentes limites. Assim, a AP resume a curva em único valor, representando a média de todas as precisões. Uma forma de calcular a AP é utilizando a técnica de interpolação de 11 pontos, onde se calcula a *precision* em 11 níveis de *recall* igualmente espaçados e, em seguida, é feita a média dos valores encontrados. Esse cálculo pode ser expresso como:

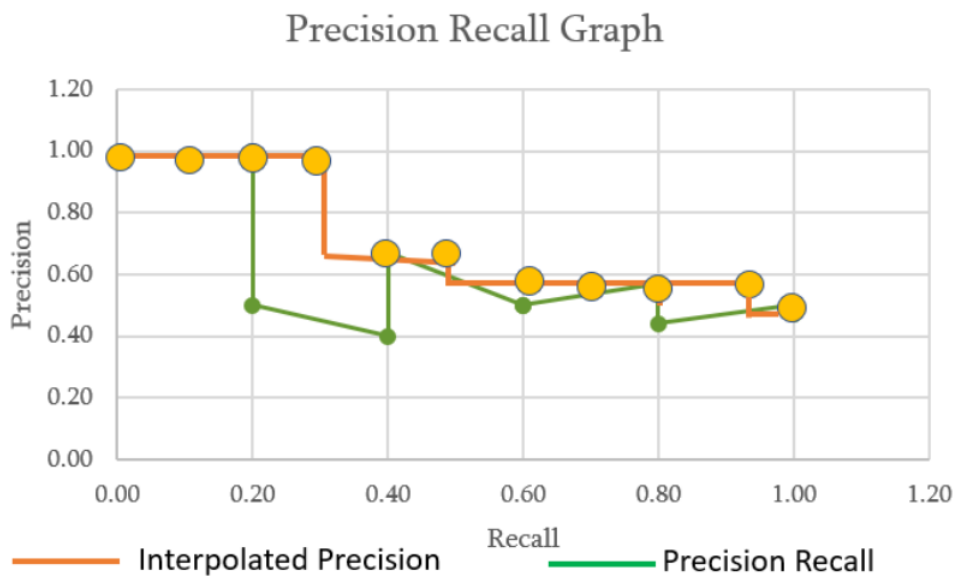
$$AP = \frac{1}{11} \sum_{r \in \{0,0.1,...,1\}} \rho_{interp(r)} \quad (2.4)$$

Onde

$$\rho_{interp(r)} = \max_{\tilde{r}: \tilde{r} \geq r} \rho(\tilde{r}) \quad (2.5)$$

E $\rho_{interp(r)}$ é a *precision* máxima para o *recall* r . A Figura 4 demonstra o gráfico da *precision* interpolada, bem como a curva *precision-recall*.

Figura 4 – Gráfico de interpolação



Fonte: (KHANDELWAL, 2020)

A *mean average precision* (mAP) é a média da AP de todas as classes detectadas, definida pela equação a seguir.

$$mAP = \frac{1}{N} \sum_{i=1}^N AP_i \quad (2.6)$$

Onde AP_i é o valor da AP para a i -ésima classe e N é o total de classes.

3

Trabalhos Relacionados

Este capítulo apresenta uma revisão da literatura a respeito de algoritmos de detecção de produtos e preços, aplicada em folhetos de vendas. A Seção 3.1 explica a estratégia de busca utilizada e as Seções 3.2, 3.3 e 3.4 apresentam os artigos científicos encontrados.

3.1 Estratégia de busca

Para obter trabalhos relacionados relevantes, a *string* de busca que obteve resultados mais promissores ao tema pesquisado foi "(extraction OR recognition OR detection) AND (supermarket OR marketing OR sales) AND flyers".

As pesquisas foram realizadas em março de 2022 nas seguintes bases: Scopus¹, Web of Science² e IEEE Xplore Digital Library³. A quantidade de resultados encontrados em cada base está exibida na Tabela 1.

Tabela 1 – Quantidade de artigos encontrados por base

Base	Quantidade de artigos
Scopus	10
Web of Science	9
IEEE	1

Após análise inicial, foi constatado que apenas cinco artigos encontrados no Scopus eram relacionados ao tema de pesquisa, três da Web of Science foram relevantes, porém já estavam incluídos nos resultados do Scopus, assim como o único artigo encontrado no IEEE, totalizando assim cinco trabalhos relevantes.

¹ <https://www.scopus.com>

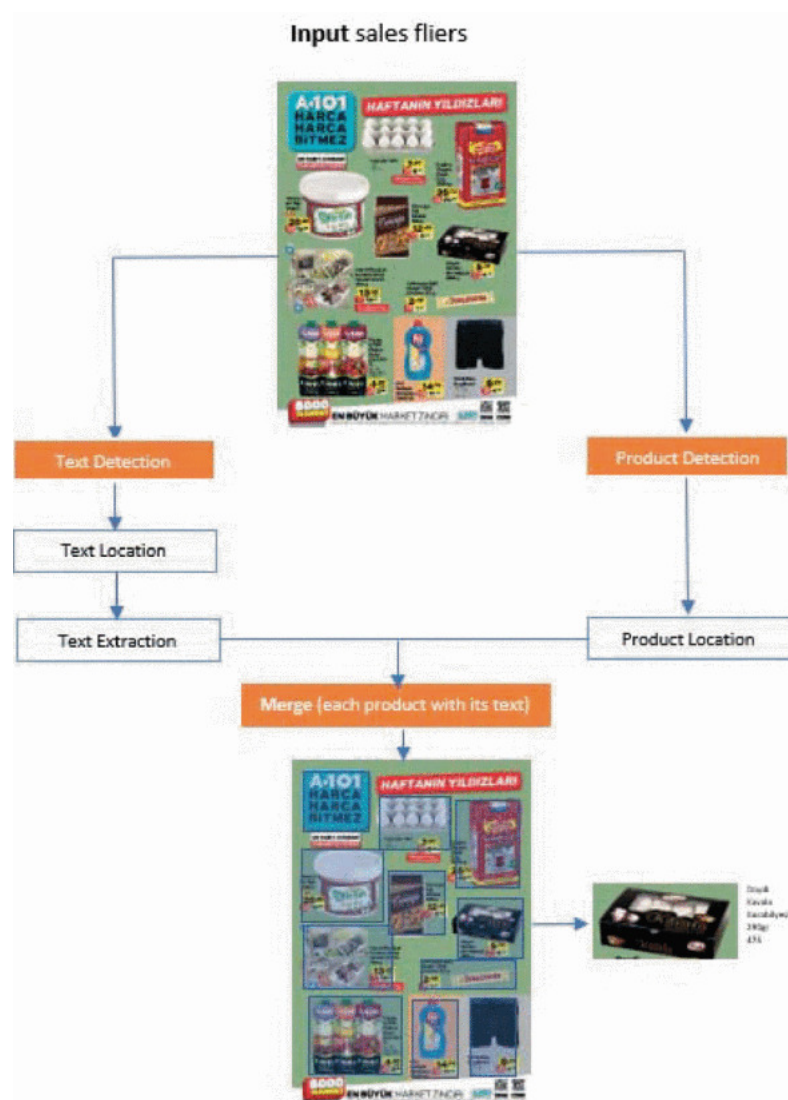
² <https://www.webofscience.com>

³ <https://ieeexplore.ieee.org>

3.2 Recognition and Classifying Sales Flyers Using Semi-Supervised Learning

Mosquera e Genc (2019) apresentam um algoritmo para detecção e classificação de produtos e informações, como descrição e preço, contidas em folhetos de diversificados estilos e pertencentes a diferentes domínios, tal como setor eletrônico, de alimentos, etc. Na Figura 5 estão ilustradas as 3 etapas referentes ao algoritmo: detecção dos produtos, detecção dos textos e a mesclagem dos resultados dos passos anteriores.

Figura 5 – Algoritmo de classificação de folhetos



Fonte: (MOSQUERA; GENC, 2019)

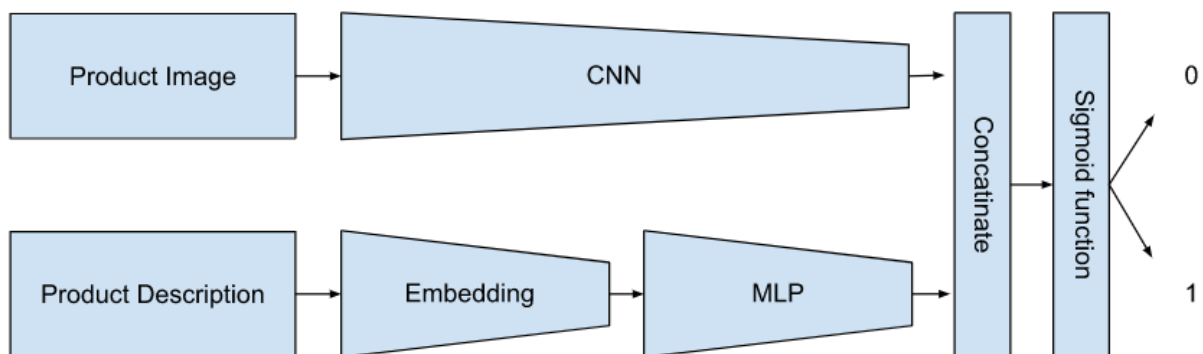
Segundo Mosquera e Genc (2019), as duas primeiras fases são tarefas de detecção e segmentação, o desafio principal nesse momento são os diferentes padrões de texto e o fundo altamente desordenado. Para superar esses obstáculos, os autores utilizaram na detecção de objetos a rede neural Faster-R-CNN e a rede neural ResNet como *feature map*. Já na detecção

e extração de texto, foram comparados os efeitos utilizando uma *connectionist text proposal network* (CTPN) e a OCR Microsoft API, sendo que esta obteve resultados mais precisos.

Mosquera e Genc (2019) aplicaram três diferentes algoritmos para mesclar os objetos detectados com as informações extraídas. Todos eles começam com uma etapa em comum de mesclagem dos *bounding boxes* de palavras mais próximas, formando um novo *bounding box* que representa a descrição de cada produto. O primeiro algoritmo busca por distância de cada um desses *bounding boxes* de informações com cada um dos produtos detectados, com o objetivo de formar um novo *bounding box* que une a descrição ao produto. Esse algoritmo dispõe de um bom desempenho quando existe um espaço considerável entre os produtos, porém quando isso não acontece, o algoritmo resulta em diversas agregações incorretas. A fim de melhorar os resultados nessas situações, o segundo algoritmo extrai o texto presente nos produtos e o utiliza em comparação com as descrições já extraídas, o que resultou num aumento de desempenho do algoritmo.

Entretanto, quando não é possível obter o texto no produto, o problema anterior permanece. Por isso, Mosquera e Genc (2019) propõem um novo algoritmo, combinação de *multi-layer perceptron* (MLP) e *convolutional neural network* (CNN). Enquanto a CNN opera sobre as imagens, os textos passam por uma aplicação do MLP, o resultado dessas duas partes é mesclado através de uma função sigmoide, que retorna a probabilidade do produto corresponder a descrição recebida.

Figura 6 – Algoritmo proposto para mesclar



Fonte: Adaptado de (MOSQUERA; GENC, 2019)

O terceiro método de mesclagem proposto por Mosquera e Genc (2019) obteve um excelente desempenho na classificação dos folhetos. Apesar disso, a abordagem possui algumas limitações nos casos em que os folhetos possuem produtos com mesmo nome ou descrição e tamanhos diferentes.

3.3 Content Extraction from Marketing Flyers

Calefati et al. (2016) propõem um algoritmo de extração de conteúdo em folhetos de venda, que utiliza recursos visuais presentes nos folhetos como suporte às classificações, pois acreditam na importância que os recursos têm na distinção das entidades de interesse presentes em um documento não estruturado como um encarte.

A abordagem possui quatro etapas (Figura 7): extração de textos e recursos visuais, utilização dos recursos para classificação dos textos, mesclagem de textos vizinhos em parágrafos e, por fim, mesclagem de título, descrição e preço para identificação das ofertas.

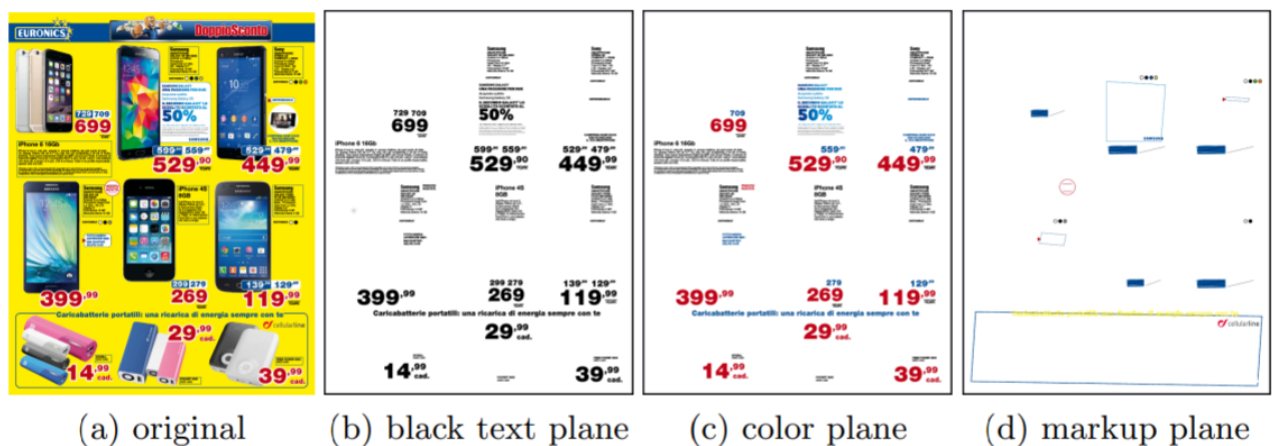
Figura 7 – Etapas da abordagem



Fonte: (CALEFATI et al., 2016)

O primeiro passo é a divisão dos folhetos em planos de recursos, como pode ser observado na Figura 8. O *black text plane* é utilizado para extração dos textos, o *color plane* é obtido removendo todos os elementos que não são texto e mantendo posição, cor e opacidade. Já no *markup plane* é possível obter todos os elementos geométricos, marcações e sublinhados, por exemplo, recursos muito utilizados para destacar informações.

Figura 8 – Planos de recursos



Fonte: (CALEFATI et al., 2016)

A partir da etapa anterior, os autores definiram diversas *features*, como cor, tamanho da fonte, orientação, etc., que são utilizadas na etapa de classificação. Esta etapa é feita utilizando o *random forest classifier* e os *tokens* são classificados em Preço, Título, Descrição ou Outro. Em seguida, eles são agregados através de um algoritmo de *clustering*, que considera a classe e posição do *token* e resulta na criação dos *bounding boxes* representando o título, descrição ou preço.

A última etapa é a agregação dos *tokens* para que sejam formadas as tuplas (Preço, Título, Descrição), a abordagem de [Calefati et al. \(2016\)](#) procura levar em consideração não só a menor distância entre os 3 elementos buscados, mas também a minimização da sobreposição entre os *bounding boxes* das ofertas finais. Na [Tabela 2](#) é exibida a matriz de confusão para o *random forest classifier* e na [Tabela 3](#) as métricas obtidas na classificação e agregação.

Tabela 2 – Matriz de confusão

	Title	Descr.	Price	Other
Title	95.39%	5.57%	3.70%	3.27%
Descr.	2.70%	91.51%	2.94%	3.51%
Price	0.19%	0.67%	87.31%	2.04%
Other	1.72%	2.26%	6.05%	91.18%

Fonte: ([CALEFATI et al., 2016](#))

Tabela 3 – Métricas obtidas

	Precision	Recall	F-measure
Title	0.740	0.655	0.695
Description.	0.789	0.837%	0.812
Price	0.815	0.916	0.862
Aggr. offers	0.487	0.547	0.515

Fonte: ([CALEFATI et al., 2016](#))

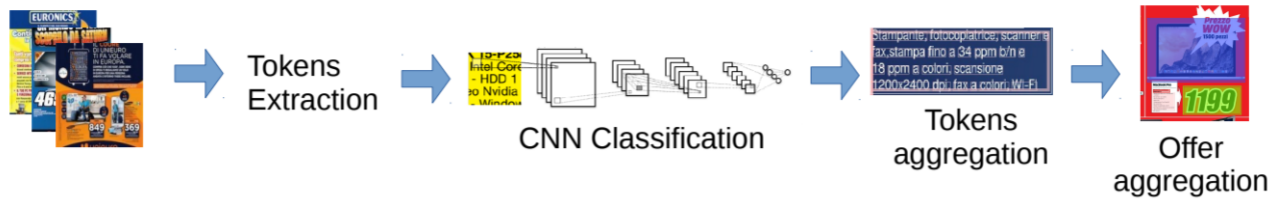
3.4 Using Convolutional Neural Networks for Content Extraction from Online Flyers

A abordagem de [Gallo, Zamberletti e Noce \(2015\)](#) consiste na extração de informações de folhetos levando em consideração recursos visuais. Mas, diferentemente de [Calefati et al. \(2016\)](#), a nova análise substitui a etapa de extração de recursos do PDF para a utilização de CNN na classificação das imagens.

A primeira etapa consiste na obtenção dos *tokens*. Para isso, utilizou-se PDFTextStripper⁴ para extrair todos os textos e outras informações relevantes, como as coordenadas de cada palavra. Logo depois, foi realizada a classificação dos tokens nas entidades Título, Descrição, Preço e

⁴ <https://pdfbox.apache.org/docs/2.0.7/javadocs/org/apache/pdfbox/text/PDFTextStripper.html>

Figura 9 – Etapas da nova abordagem



Fonte: (GALLO; ZAMBERLETTI; NOCE, 2015)

Outro, utilizando CNN com a arquitetura AlexNet. Quando todos os *tokens* são classificados, eles são agregados utilizando o algoritmo de *clustering* apresentado no artigo de [Calefati et al. \(2016\)](#), resultando também nos *bounding boxes* que representam o título, descrição e preço.

Na agregação dos *tokens* em tuplas (Preço, Título, Descrição) são consideradas a menor distância entre os três elementos buscados e a menor sobreposição entre os *bounding boxes* das ofertas resultantes. As métricas apresentadas a seguir ([Tabela 5](#)) utilizaram o indicador intersection over union (IOU) entre os *bounding boxes* detectados e formados na rotulação. Foi considerado que IOU maior que 0.5 indicava que a detecção estava correta. Na [Tabela 4](#) se encontra a matriz de confusão para a classificação utilizando CNN e na [Tabela 5](#) os resultados da agregação, onde pode ser observado um aumento de desempenho em relação a abordagem anterior de [Calefati et al. \(2016\)](#).

Tabela 4 – Matriz de confusão na nova abordagem

	Title	Descr.	Price	Other
Title	85.31%	44.26%	1.87%	6.89%
Descr.	0.88%	37.81%	0.94%	1.81%
Price	0.11%	1.93%	73.54%	4.07%
Other	13.70%	16.00%	23.65%	87.24%

Fonte: (GALLO; ZAMBERLETTI; NOCE, 2015)

Tabela 5 – Métricas obtidas na nova abordagem

	Precision	Recall	F-measure
Title	0.417	0.838	0.557
Description.	0.495	0.807	0.614
Price	0.799	0.443	0.570
Aggr. offers	0.925	0.541	0.683

Fonte: (GALLO; ZAMBERLETTI; NOCE, 2015)

4

Construção da Base

Este capítulo apresenta o processo de obtenção, seleção e rotulação das imagens para formação da base de dados. A Seção 4.1 descreve a estratégia para aquisição das imagens utilizando um *web crawler*, a Seção 4.2 detalha os critérios para escolha das imagens e, por fim, a Seção 4.3 expõe o procedimento de rotulação.

4.1 *Web crawler*

A primeira etapa realizada foi a designação do *website* onde seriam efetuadas as extrações das imagens. O site selecionado foi o Seus Folhetos¹, por centralizar folhetos de mercados de todo o Brasil e, além disso, por possuir um acervo que possibilita a visualização dos encartes de todo o período em que foram registrados.

Para a implementação do *web crawler*, a ferramenta escolhida foi o Scrapy², *framework open source* desenvolvido na linguagem de programação Python para extração de dados estruturados em *websites*. Em conjunto com o Scrapy, foi utilizado o Selenium³, ferramenta que permite simular interações em navegadores, ação que foi necessária em algumas páginas do site definido.

A Figura 10 exibe a primeira página acessada para cada supermercado. A estrutura (Figura 11) simples do acervo permite a navegação entre os meses que possuem folhetos publicados. A partir da visualização mensal (Figura 12), cada folheto é acessado individualmente. Para que isso seja possível, o Selenium atua clicando no botão de expandir, fazendo com que todos os encartes sejam listados.

¹ <https://seusfolhetos.com.br/>

² <https://scrapy.org/>

³ <https://www.selenium.dev>

⁴ Disponível em <<https://seusfolhetos.com.br/arquivo/gbarbosa>>. Acesso em 5 maio 2022.

Figura 10 – Página do supermercado



Fonte: Arquivo do Gbarbosa no SeusFolhetos⁴

Finalmente, na visualização individual (Figura 13), o download da imagem é realizado e ela é armazenada no diretório do supermercado em questão e, enquanto há mais páginas do mesmo folheto, o *web crawler* encaminha-se para a seguinte.

4.2 Seleção das imagens

Como filtragem inicial dos 600 supermercados presentes no acervo e tentativa de priorizar os vendedores com mais encartes publicados, foram selecionados apenas aqueles que possuíam folhetos em todos os meses de 2021 e nos três primeiros meses de 2022, totalizando 56 supermercados. O *web crawler* foi então executado, realizando o download de todos os folhetos disponíveis entre 2021 e 2022 para cada um desses estabelecimentos.

Após a execução do algoritmo, foi constatada uma grande variedade de modelos de folhetos, como por exemplo, imagens com ofertas únicas, apresentação de campanhas ou revistas do estabelecimento que não continham anúncios de produtos. Por esse motivo, foi necessário realizar uma limpeza manualmente para eliminar esses exemplares que não contribuiriam para o modelo, já que não possuíam ofertas. Outras amostras descartadas foram imagens de folhetos mais complexos, com produtos sobrepostos, por exemplo, onde seria necessário um modelo de aprendizado mais complexo para conseguir identificar os produtos e associá-los aos preços.

⁵ Disponível em <<https://seusfolhetos.com.br/arquivo/gbarbosa/03-2022>>. Acesso em 5 maio 2022.

⁶ Disponível em <<https://seusfolhetos.com.br/n/gbarbosa-ofertas/folheto-350077-0>>. Acesso em 5 maio 2022.

Figura 11 – Estrutura da página do supermercado

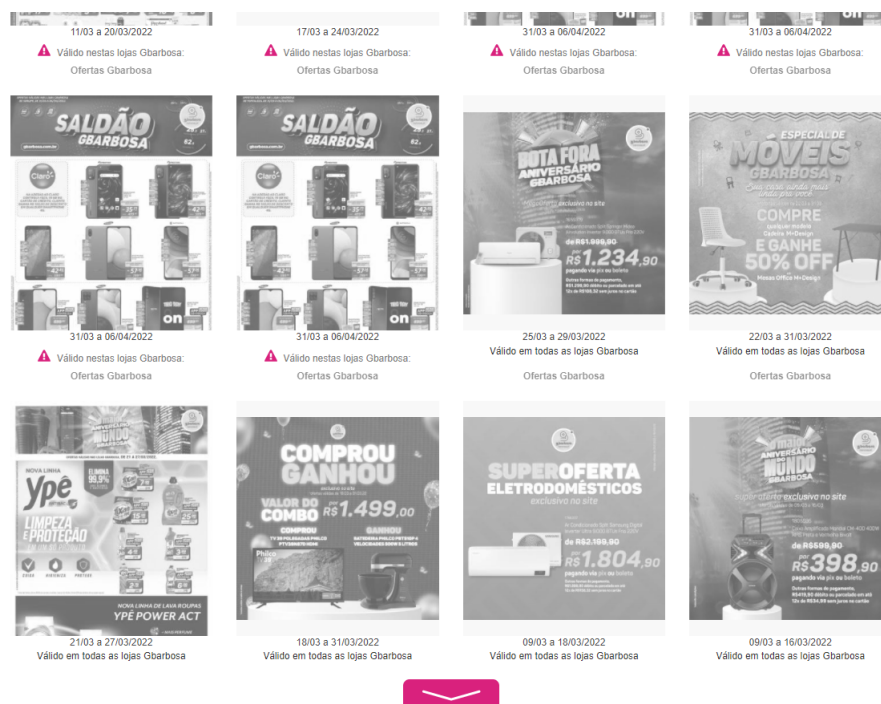
```

<div class="store-archive-date">
  <div class="year">2022</div>
  <a href="/arquivo/gbarbosa/05-2022" rel="nofollow">05-2022</a>
  <br>
  <a href="/arquivo/gbarbosa/04-2022" rel="nofollow">04-2022</a>
  <br>
  <a href="/arquivo/gbarbosa/03-2022" rel="nofollow">03-2022</a>
  <br>
  <a href="/arquivo/gbarbosa/02-2022" rel="nofollow">02-2022</a>
  <br>
  <a href="/arquivo/gbarbosa/01-2022" rel="nofollow">01-2022</a>
  <br>
  <div class="year">2021</div>
  <a href="/arquivo/gbarbosa/12-2021" rel="nofollow">12-2021</a>
  <br>
  <a href="/arquivo/gbarbosa/11-2021" rel="nofollow">11-2021</a>
  <br>
  <a href="/arquivo/gbarbosa/10-2021" rel="nofollow">10-2021</a>
  <br>
  <a href="/arquivo/gbarbosa/09-2021" rel="nofollow">09-2021</a>
  <br>
  <a href="/arquivo/gbarbosa/08-2021" rel="nofollow">08-2021</a>
  <br>
  <a href="/arquivo/gbarbosa/07-2021" rel="nofollow">07-2021</a>
  <br>
  <a href="/arquivo/gbarbosa/06-2021" rel="nofollow">06-2021</a>
  <br>
  <a href="/arquivo/gbarbosa/05-2021" rel="nofollow">05-2021</a>
  <br>
  <a href="/arquivo/gbarbosa/04-2021" rel="nofollow">04-2021</a>
  <br>
  <a href="/arquivo/gbarbosa/03-2021" rel="nofollow">03-2021</a>
  <br>
  <a href="/arquivo/gbarbosa/02-2021" rel="nofollow">02-2021</a>

```

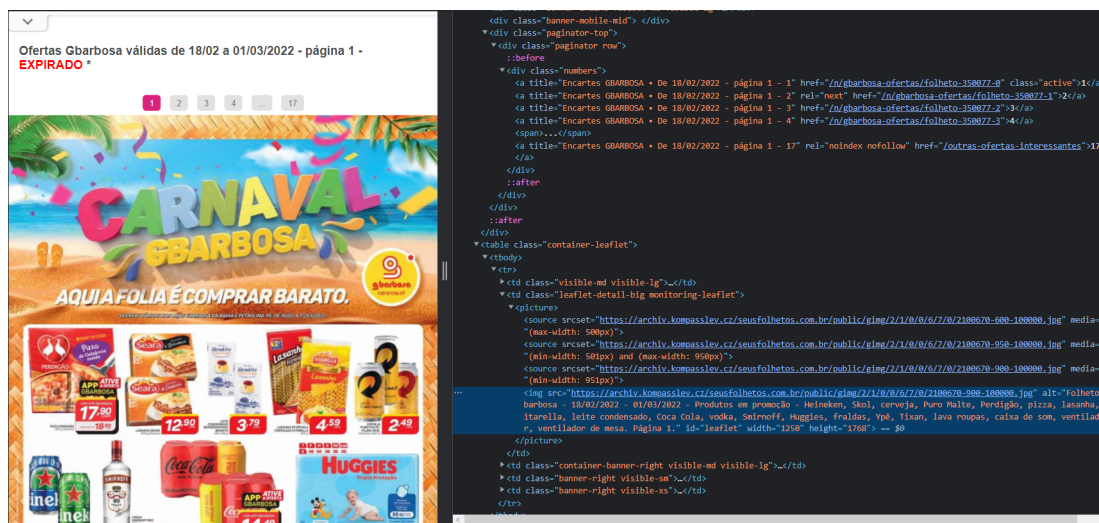
Fonte: Arquivo do Gbarbosa no SeusFolhetos⁴

Figura 12 – Página mensal de folhetos



Fonte: Arquivo do Gbarbosa no SeusFolhetos⁵

Figura 13 – Página individual do folheto



Fonte: Arquivo do Gbarbosa no SeusFolhetos⁶

Em seguida, a contagem de imagens remanescentes definiu os 10 supermercados que seriam utilizados para rotulação e criação da base de dados. Na Tabela 6 estão listados os mercados selecionados e a quantidade de imagens obtidas de cada um. Por fim, com a finalidade de totalizar 2500 imagens para rotulação, foi construído um algoritmo para selecionar aleatoriamente 250 imagens de cada supermercado. A seguir os folhetos foram armazenados no Google Drive.

Tabela 6 – Quantidade de imagens por supermercado selecionado

Supermercado	Quantidade de imagens
Carrefour	2536
Atacadão	2202
Big	1406
Assaí Atacadista	919
Nacional	883
Bompreço	824
Gbarbosa	735
Cooper	734
Econômico Atacadão	626
Supermercados Imperatriz	625

4.3 Rotulação

Na etapa de rotulação, a ferramenta escolhida foi o Daturks⁷, software *open source* que permite a rotulagem de texto, imagem ou vídeo. A instalação foi baseada em Docker⁸. Para hospedá-lo, foi criada uma instância do AWS EC2⁹.

⁷ <https://docs.daturks.com>

⁸ <https://hub.docker.com/r/klimentij/daturks>

⁹ <http://3.92.192.23>

As imagens foram carregadas através de um arquivo de texto simples contendo os links do Drive para todos os arquivos de imagens, cada arquivo corresponde a uma página de um folheto (Figura 15). O processo de rotulação foi baseado na identificações dos seguintes alvos: descrição, preço, imagem e oferta. A descrição consiste em informações do produto, pode conter o nome do produto, quantidade, formato, unidade (15(a)), o preço se refere ao valor (15(b)), a imagem é a figura que representa o produto (15(c)), já a oferta é o *bounding box* do anúncio completo, envolvendo imagem, preço e descrição (15(d)). Foram desconsideradas ofertas complexas, como por exemplo, anúncio com foto ou descrição compartilhada para mais de um produto.

Figura 14 – Exemplo de rotulação



Fonte: Elaborada pelo autor

O trabalho de rotulação é bastante exaustivo e lento porque requer uma intervenção manual minuciosa. A fim de viabilizar a rotulação da base em um tempo compatível com o próprio desenvolvimento do TCC, esta tarefa foi distribuída a uma equipe de colaboradores formada principalmente por alunos, alguns sem vinculação com o projeto e orientadores do trabalho, além de colaboradores externos, sem vínculo com a UFS. Para que estas pessoas pudessem contribuir com a rotulação dos alvos foi elaborado um manual (Apêndice A) com orientações detalhadas sobre o processo.

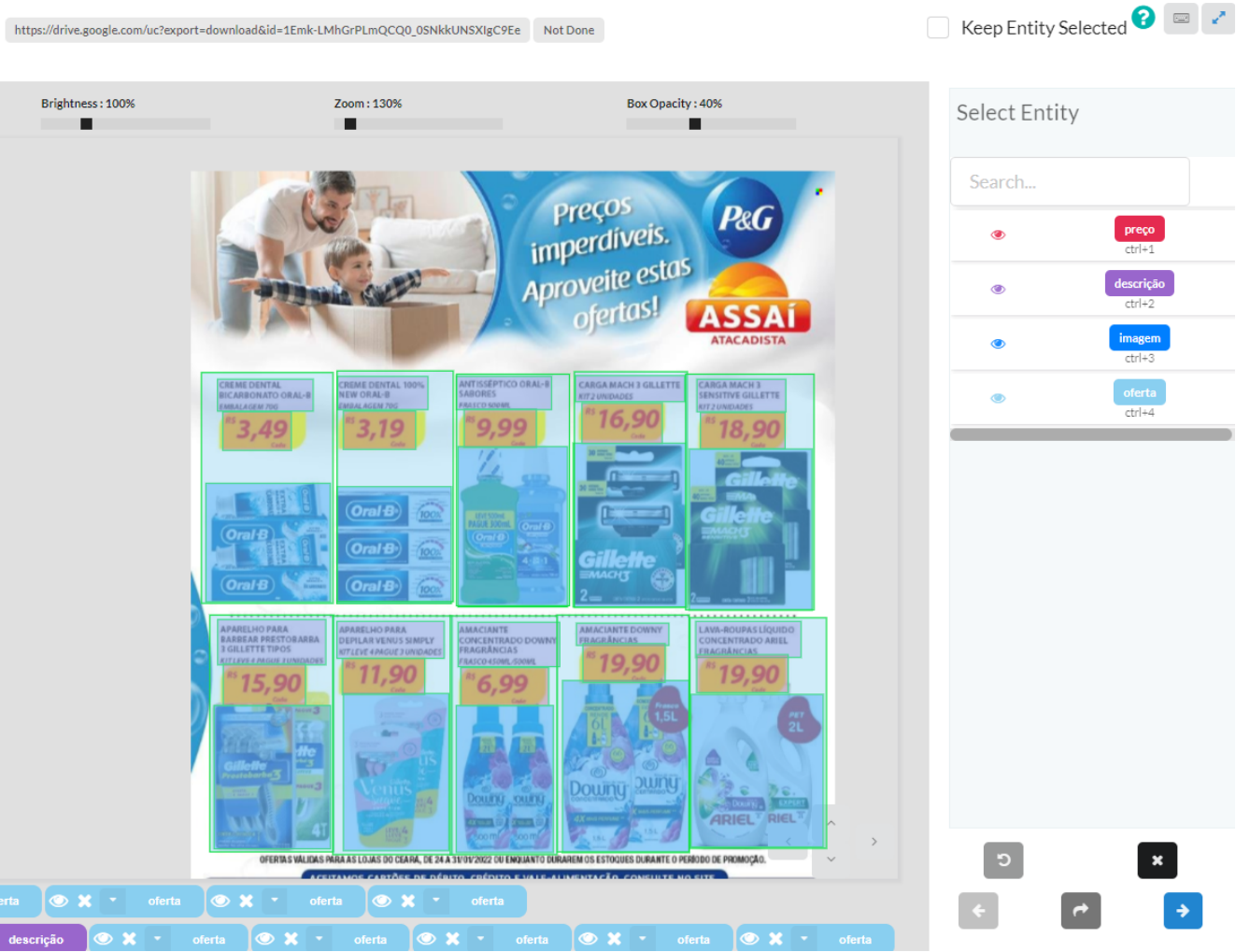
Como o processo de rotulação é muito custoso, foi realizada uma fase inicial de rotulação com 500 folhetos, 20% da base total obtida. Essa parcela foi necessária para que a avaliação dos modelos pudesse ser iniciada. No decorrer dos experimentos mais 400 imagens foram rotuladas para a realização de testes, sendo 300 folhetos da base inicial e 100 folhetos de outros supermercados listados na Quadro 1, totalizando 900 folhetos. Da base de 900 folhetos, foram rotuladas 13841 ofertas, 13900 imagens, 14175 descrições e 15829 preços.

¹⁰ Disponível em <<http://3.92.192.23/projects/mayarabarbosa2206/Folhetos>>. Acesso em 5 maio 2022.

Quadro 1 – Outros supermercados selecionados

Apoio Mineiro	Atacadão Atakarejo
Bretas	Comercial Esperança
TodoDia	Maxxi Atacado
Makro	CasaGrande Supermercados
Mercadão Atacadista	Redeconomia

Figura 15 – Interface do Datururks



Fonte: Página do projeto no Datururks⁷

5

Desenvolvimento e Resultados

Este capítulo aborda todo o processo de desenvolvimento do detector e os resultados alcançados. A Seção 5.1 informa quais ferramentas foram utilizadas. O processo de avaliação entre dois modelos e a seleção de um deles é relatada na Seção 5.2. A Seção 5.3 descreve algoritmos que foram construídos para reduzir erros do detector e a Seção 5.4 mostra outras estratégias utilizadas para aprimoramento do modelo. Por fim, a Seção 5.5 apresenta um sumário dos resultados obtidos.

5.1 Ferramentas utilizadas

A linguagem de programação Python foi a escolhida para a construção e avaliação dos modelos, visto que é uma das mais utilizadas em Inteligência Artificial, por dispor de diversas bibliotecas que auxiliam todas as etapas da construção e uso de um modelo de aprendizado de máquina, desde a manipulação dos dados até a apresentação dos resultados. Além disso, ela permite a utilização do Jupyter, um ambiente de desenvolvimento que facilita o processo de codificação e depuração do código.

O ambiente de desenvolvimento utilizado foi o Google Colaboratory¹, que permite a execução de código Python pelo navegador e é especialmente adequado para aprendizado de máquina, análise de dados e educação (GOOGLE, 2021). O Google Colab permite a execução do Jupyter na nuvem, não requer nenhuma configuração e permite a integração com o Google Drive e acesso sem custo financeiro a recursos de computação, como GPUs (*Graphics Processing Units*), tornando o processo de execução muito mais eficiente.

As principais bibliotecas Python utilizadas foram: Pandas, para a importação e manipulação dos dados; Scikit-learn e suas funções auxiliares para separação dos dados entre treino e teste; Numpy, para facilitar a execução de cálculos na obtenção das métricas; PyTorch, que

¹ <https://colab.research.google.com/drive/1u2BtLGmQl82T6ZkazpGOyu6CDyWqz9bl>

auxiliou na execução dos modelos e cálculo das métricas e OpenCV, que foi utilizada para a plotagem e visualização das detecções.

5.2 Avaliação dos modelos

Para selecionar qual método de detecção seria utilizado para a construção do modelo, foram feitas análises comparativas entre dois detectores, o YOLOv5 e o SSD. Para o treinamento dos dois modelos foram utilizadas as primeiras 500 imagens rotuladas, sendo 450 folhetos para treino e 50 folhetos de teste. Das imagens de teste esperava-se obter 710 descrições, 659 imagens, 672 ofertas e 716 preços.

5.2.1 YOLOv5

A implementação do YOLOv5 selecionada foi a do Ultralytics ([ULTRALYTICS, 2020](#)), desenvolvida por Glenn Jocher, utilizando o framework PyTorch. Dentre as arquiteturas disponíveis, foi escolhida a YOLOv5x, que apesar de ter o maior tempo de execução também apresenta os melhores resultados para o tamanho de imagem 640*640, tamanho padrão que foi mantido para o treinamento. Além da escolha da arquitetura e criação dos arquivos de configuração do modelo e dos dados de entrada, também foi feita a conversão do json obtido no Daturks para o formato de labels aceito pelo YOLOv5, onde cada linha representa um *bounding box* e é formada pelo id da classe, o centro de x, centro de y, largura e altura da imagem.

O treinamento utilizou *transfer learning*, ou seja, o modelo foi treinado com a utilização dos pesos de uma YOLOv5x pré-treinada no dataset COCO, disponíveis no repositório da Ultralytics ([ULTRALYTICS, 2020](#)). Para a aferição das métricas após o treino, foram utilizados 50 folhetos de teste. Para fazer o cálculo se fez necessário medir a correspondência entre a área detectada e a área anteriormente rotulada, com o objetivo de definir se a detecção foi válida ou não. Em vista disso, foi empregada a medida (IOU) entre detecções e rotulações e o limiar utilizado para conferir se a detecção foi correta foi de 0.5, essa escolha foi feita baseando-se nos trabalhos relacionados, que também utilizaram esse limite.

As métricas são apresentadas na tabela [Tabela 7](#). A mAP obtida foi 94,2%. O modelo apresentou melhores resultados para as classes imagem e oferta, obtendo resultados de *precision* e *recall* acima de 90%. Para os casos de descrição e preço, o *recall* também apresentou resultados acima de 90%, porém a *precision* não atingiu esse valor, sendo de 84,85% para descrições e 83,87% para preços. Ou seja, classificou de forma incorreta como preço e descrição em quantidade maior que as classes anteriores.

Tabela 7 – Métricas Yolo

	Precision	Recall	F-score	AP
descrição	84,85%	93,10%	88,78%	90,34%
imagem	94,29%	97,35%	95,80%	96,55%
oferta	95,09%	97,92%	96,48%	96,55%
preço	83,87%	94,41%	88,31%	93,37%

5.2.2 SSD

Para o treinamento com o Single Shot MultiBox Detector, foi adotado o repositório de Sagar Vinodababu ([VINODABABU, 2018](#)), devido a implementação também aplicar o PyTorch no seu desenvolvimento. Foi necessária a conversão das rotulações para o Pascal VOC XML, formato de entrada para a execução do SSD, no qual cada imagem possui um XML correspondente contendo as anotações das classes dos objetos. Os 50 folhetos de teste utilizados no cálculo das métricas do YOLO também foram utilizados para o cálculo das métricas do SSD.

Para a classificação das detecções em corretas também foi considerado o IOU maior que 0.5. As métricas *precision*, *recall*, *F-Score* e *AP* são apresentadas na [Tabela 8](#). A mAP obtida com o SSD foi de 67,28%. As categorias descrição e preço alcançaram resultados abaixo do esperado, tanto a *precision*, mostrando que o modelo classificou incorretamente uma parte considerável dos objetos, quanto o *recall*, revelando uma parcela de cerca de 40% de objetos não detectados. Já as categorias imagem e oferta obtiveram melhores coberturas, porém com *precision* muito abaixo das obtidas com o YOLO.

Tabela 8 – Métricas SSD

	Precision	Recall	F-score	AP
descrição	56,91%	58,59%	57,74%	41,89%
imagem	62,28%	95,58%	75,42%	88,81%
oferta	49,25%	97,47%	65,44%	91,87%
preço	61,92%	63,13%	62,52%	46,57%

5.2.3 Comparação dos modelos

Com base nas métricas obtidas, o YOLOv5 demonstrou melhores resultados na avaliação de todas as classes, tanto em relação a *precision* quanto ao *recall*. Dessa forma, foi o modelo selecionado para a construção do detector. A [Figura 16](#) demonstra a vantagem do YOLOv5 em comparação com o SSD em um exemplo de folheto com detecções de descrições. Na detecção com o SSD podem ser observadas várias descrições não localizadas, que foram detectadas pelo YOLOv5, assim como várias detecções para um mesmo objeto, o que não acontece com o segundo modelo nos mesmos casos.

Figura 16 – Detecção de descrição



Fonte: Elaborada pelo autor

5.3 Regras para aperfeiçoamento do modelo

Após a decisão do detector a ser utilizado, o YOLOv5, foram analisadas as detecções resultantes do modelo, os erros mais frequentes e o contexto onde eles ocorreram, com o objetivo de avaliar uma forma de minimizar essas falhas e melhorar os resultados do modelo. Dessa forma, foram desenvolvidas algumas regras, que consistem em algoritmos que procuram reduzir erros que foram observados de forma mais frequente em situações que serão demonstradas a seguir.

5.3.1 Regra 1 - Remover ofertas contidas em ofertas

Esse primeiro algoritmo tem como objetivo identificar e remover ofertas repetidas em um mesmo produto, partindo do princípio de que um anúncio só deve possuir um *bounding box* de oferta. Esse passo é importante não apenas para minimizar os erros de ofertas duplicadas, ilustrados pela Figura 17, mas também para facilitar a identificação de qual oferta faz parte uma outra categoria identificada, como preço, imagem e descrição, etapa que se faz necessária para regras posteriores.

Figura 17 – Ofertas duplicadas



Fonte: Elaborada pelo autor

Algoritmo 1: Regra 1

Dados: dataset com detecções de ofertas

Resultado: lista de detecções excluídas

ofertas = dataset apenas com detecções de ofertas;

excluídos = [];

para $i = 0$ até total de ofertas **faça**

 box1 = ofertas[i];

para $j = 0$ até total de ofertas **faça**

 box2 = ofertas[j];

se box1 é diferente do box2 e box2 não está na lista de excluídos **então**

se $\text{intersecao}(\text{box1}, \text{box2}) > 0.5$ **então**

se $\text{box1.confidence} > \text{box2.confidence}$ **então**

 adiciona box2 aos excluídos;

senão

 adiciona box1 aos excluídos;

fim

fim

fim

fim

fim

- Nessa regra são avaliadas ofertas que possuem mais de 50% da sua área em interseção com outra oferta. Quando um par dessas ofertas é detectado, a que possui o menor *score* de confiança é excluída das detecções. O Algoritmo 1 demonstra esse processo.

5.3.2 Regra 2 - Remover descrição, preço e imagens não contidos em uma oferta

Esse passo tem como objetivo identificar e remover *bounding boxes* de descrições, preços ou imagens não contidos em uma detecção de oferta. Essa regra foi desenvolvida ao observar que alguns objetos do folheto, como textos ou ilustrações externas aos produtos estavam sendo identificadas pelo modelo como preço, imagem ou descrição. A Figura 18 ilustra o caso de um texto de dicas presente no folheto que foi identificado pelo detector como a descrição de um produto.

- Nessa regra foi avaliada a interseção entre os *bounding boxes* de ofertas com os das detecções das outras categorias. Quando uma detecção de preço, descrição ou imagem não possui mais de 50% da sua área em interseção com alguma oferta, ela é excluída das detecções resultantes. Essa lógica pode ser observada no Algoritmo 2.

Figura 18 – Classes fora de ofertas



Fonte: Elaborada pelo autor

Algoritmo 2: Regra 2

Dados: dataset com detecções de ofertas e dataset com detecções das outras classes

Resultado: lista de detecções excluídas

detecoes = dataset com todos os *bouding boxes* exceto ofertas;

ofertas = dataset apenas com detecções de ofertas;

excluídos = [];

para $i = 0$ até total de *detecoes* **faça**

 excluir = Verdadeiro;

 box1 = *detecoes*[i];

para $j = 0$ até total de *ofertas* **faça**

 box2 = *ofertas*[j];

se *intersecao*(box1, box2) > 0.5 **então**

 excluir = Falso;

 interrompa;

fim

fim

se *excluir* = Verdadeiro **então**

 adiciona box1 aos excluídos;

fim

fim

5.3.3 Regra 3 - Remover *bouding boxes* de mesma categoria com grande intersecção dentro de uma oferta

Esse passo foi desenvolvido após observar muitas ocorrências de um único objeto receber mais de um *bouding box* de mesma categoria. Na Figura 19 estão expostos dois casos desse mesmo problema, um de descrições duplicadas e outro de preços duplicados, ambos apresentam uma detecção correta e com maior *score* de confiança e outra com menor pontuação de confiança e abrangendo uma área menor do objeto buscado. Nessa regra é necessário ter ciência da oferta a qual classe detectada faz parte, o que foi possível a partir da utilização da regra 1.

Figura 19 – Erro de categorias duplicadas



Fonte: Elaborada pelo autor

- Nessa regra são verificadas as detecções de mesma categoria com grande sobreposição dentro de uma oferta. Quando uma detecção tem interseção maior que 50% com outra da mesma classe e que esteja presente na mesma oferta, a que possuir o menor *score* de confiança é removida. O processo ocorre com detecções de preços, descrições e imagens. O Algoritmo 3 corresponde à essa regra.

Algoritmo 3: Regra 3

Dados: dataset com detecções de classes que não sejam ofertas

Resultado: lista de detecções excluídas

deteccoes = dataset com todos os *bounding boxes* exceto ofertas;

excluídos = [];

para $i = 0$ até total de *deteccoes* **faça**

 box1 = deteccoes[i];

para $j = 0$ até total de *deteccoes* **faça**

 box2 = deteccoes[j];

se box1 e box2 são diferentes e fazem parte da mesma oferta e pertencem a mesma categoria e não estão na lista de excluídos **então**

se $\text{intersecao}(\text{box1}, \text{box2}) > 0.5$ **então**

se box1.confidence > box2.confidence **então**

 adiciona box2 aos excluídos;

senão

 adiciona box1 aos excluídos;

fim

fim

fim

fim

fim

5.3.4 Regra 4 - Remover preço ou descrição quando existe grande interseção entre os dois e existe outro da mesma categoria na oferta

O objetivo desse passo é remover preços e descrições com grande intersecção em uma mesma oferta. Essa regra foi desenvolvida ao identificar diversos casos, ilustrados pela Figura 20 e Figura 21, em que o modelo detectou preço como descrição, ou vice-versa. Para determinar esses casos de erro, calculamos a intersecção entre preços e descrições e incluímos a condição de apenas remover a detecção caso exista outro *bounding box* de mesma categoria na oferta, para impedir a remoção de detecções corretas e que possivelmente apenas contém uma grande proximidade.

- Nessa regra são buscadas as interseções de mais de 50% da área entre preço e descrição. O *bounding box* (preço ou descrição) com menor confiança é marcado e busca-se então outra detecção da mesma classe na mesma oferta. Quando encontrada, o *bounding box* marcado

Figura 20 – Erro de preço



Fonte: Elaborada pelo autor

Figura 21 – Erro de descrição



Fonte: Elaborada pelo autor

é então removido. Caso contrário, os dois objetos são mantidos para evitar a remoção de detecções corretas. Esse processo pode ser observado no Algoritmo 4.

Algoritmo 4: Regra 4

Dados: dataset com detecções de preços, dataset com detecções de descrições
Resultado: lista de detecções excluídas
preços = dataset apenas com detecções de preços;
descrições = dataset apenas com detecções de descrições;
excluídos = [];

para $i = 0$ até total de preços **faça**
 box1 = preços[i];
 para $j = 0$ até total de descrições **faça**
 box2 = descrições[j];
 se box1 e box2 fazem parte da mesma oferta e não estão na lista de excluídos
 então
 se intersecao(box1, box2) > 0.5 **então**
 se box1.confidence > box2.confidence **então**
 se (número de descrições na mesma oferta com confidence >=
 box2.confidence) > 1 **então**
 adiciona box2 aos excluídos;
 fim
 senão
 se (número de preços na mesma oferta com confidence >=
 box1.confidence) > 1 **então**
 adiciona box1 aos excluídos;
 fim
 fim
 fim
 fim
 fim
fim

5.3.5 Regra 5 - Remover imagem ou descrição quando existe grande interseção entre as duas e existe outra da mesma categoria na oferta

Esse passo é semelhante ao anterior, mas analisa as detecções de imagem e descrição. O objetivo é remover imagens e descrições com grande interseção em uma mesma oferta. Essa regra foi desenvolvida ao identificar casos (Figura 22) em que o modelo detectou imagem como descrição, ou vice-versa, possivelmente por identificar textos nas imagens dos produtos. Incluímos a condição de apenas remover a detecção caso exista uma de mesma classe na oferta.

- Nessa regra são buscadas as interseções de mais de 50% da área entre imagem e descrição. O *bounding box* (imagem ou descrição) com menor confiança é marcado e busca-se então outra detecção da mesma classe na mesma oferta. Quando encontrada, o *bounding box* marcado é então removido. Caso contrário, os dois objetos são mantidos para evitar a remoção de detecções corretas e que possivelmente contém uma grande proximidade entre elas. O Algoritmo 5 expressa essa regra.

Figura 22 – Erro de descrição em imagem



Fonte: Elaborada pelo autor

Algoritmo 5: Regra 5

Dados: dataset com detecções de imagens, dataset com detecções de descrições

Resultado: lista de detecções excluídas

imagens = dataset apenas com detecções de imagens;

descrições = dataset apenas com detecções de descrições;

excluídos = [];

para $i = 0$ até total de imagens **faça**

 box1 = imagens[i];

para $j = 0$ até total de descrições **faça**

 box2 = descrições[j];

se box1 e box2 fazem parte da mesma oferta e não estão na lista de excluídos

então

se intersecao(box1, box2) > 0.5 **então**

se box1.confidence > box2.confidence **então**

se (número de descrições na mesma oferta com confidence >= box2.confidence) > 1 **então**

 adiciona box2 aos excluídos;

fim

senão

se (número de imagens na mesma oferta com confidence >= box1.confidence) > 1 **então**

 adiciona box1 aos excluídos;

fim

fim

fim

fim

fim

fim

5.3.6 Resultados após aplicação das regras

Para a validação das regras desenvolvidas, foram utilizadas as detecções do YOLOv5, pois foi o modelo que apresentou melhores resultados anteriormente. As regras foram aplicadas

e as métricas foram calculadas novamente, podendo ser visualizadas na [Tabela 9](#). Observa-se um aumento significativo na *precision* de todas as classes, especialmente no preço e descrição, que possuíam menores valores comparados às outras duas, vide [Tabela 7](#). Dessa forma, decidimos manter a utilização das regras no nosso detector.

Tabela 9 – Métricas Yolo pós regras

	Precision	Recall	F-score	AP
descrição	95,91%	92,54%	94,19%	90,01%
imagem	96,91%	96,91%	96,91%	96,13%
oferta	96,34%	97,92%	97,12%	96,55%
preço	94,52%	93,85%	94,18%	92,92%

5.4 Nova fase de rotulação

Com os resultados obtidos após a aplicação das regras, buscamos novas estratégias para aprimoramento do modelo. Nessa fase optamos por rotular novas imagens para utilização no treinamento e teste a fim de verificar o comportamento dos resultados após essas adições. A Seção 5.4.1 apresenta os resultados após o aumento da base de treino, a Seção 5.4.2 apresenta as métricas calculadas com a nova base de teste e a Seção 5.4.3 expõe os valores com uma base de teste com supermercados diferentes dos utilizados no treinamento.

5.4.1 Base de treinamento

Com o propósito de avaliar o impacto do aumento da base de treino nos resultados do modelo, foi realizada uma nova fase de rotulação com 250 imagens. Como o software que utilizamos para anotação dos dados, o Daturks, aceita imagens pré-rotuladas, utilizamos o próprio detector para apoiar o processo. Inicialmente submetemos ao modelo essas novas imagens e os resultados foram exportados para um formato compatível com o Daturks. O próximo passo foi corrigir os erros, rotular os casos omissos e manter apenas as rotulações corretas. Com as imagens rotuladas, partimos para o novo treino, onde utilizamos os pesos obtidos anteriormente e os mesmos parâmetros.

Utilizando a mesma base de teste para o cálculo das métricas apresentadas na [Tabela 10](#). A mAP que era de 94,2% aumentou para 95,88%. A *precision* teve uma leve queda na descrição e aumento nas outras categorias, já o *recall* aumentou uma pequena porcentagem em todas as classes buscadas, ou seja, o modelo foi capaz de capturar mais *bounding boxes* esperados. Aplicando as regras, as métricas alcançaram um aumento considerável, que pode ser observado na [Tabela 11](#), a mAP foi de 93,9% para 95,9%.

Tabela 10 – Métricas com o novo treino

	Precision Antes	Precision Depois	Recall Antes	Recall Depois	F-score Antes	F-score Depois	AP Antes	AP Depois
descrição	84,85%	81,11%	93,10%	94,37%	88,78%	87,24%	90,34%	91,20%
imagem	94,29%	94,64%	97,35%	98,82%	95,80%	96,69%	96,55%	98,12%
oferta	95,09%	95,16%	97,92%	99,41%	96,48%	97,23%	96,55%	98,02%
preço	83,87%	86,88%	94,41%	97,07%	88,31%	91,69%	93,37%	96,17%

Tabela 11 – Métricas com a nova base de treino com aplicação das regras

	Precision	Recall	F-score	AP
descrição	95,58%	94,37%	94,97 %	91,51%
imagem	96,69%	98,82%	97,74%	98,12%
oferta	96,52%	99,11%	97,80%	97,73%
preço	94,82%	97,07%	95,93%	96,22%

5.4.2 Base de teste

Como a base de treino aumentou para 700 imagens, a base de teste formada por 50 folhetos já não correspondia nem a 10% da base total. Dessa forma, foi realizada uma nova fase de rotulação com mais 50 imagens para compor a base de teste. O processo foi similar ao da seção anterior, onde foi utilizado o modelo para auxiliar na rotulação e os ajustes necessários foram realizados.

O mAP resultante com a utilização da nova base de teste foi de 97,17%, superior aos das etapas anteriores, as outras métricas apresentaram em sua maioria variações positivas para todas as classes. Após a aplicação das regras, todas as medidas de *precision* e *recall* ficaram superiores a 96% (Tabela 13).

Tabela 12 – Métricas com a nova base de teste

	Precision	Recall	F-score	AP
descrição	82,67%	96,56%	89,07%	94,80%
imagem	95,24%	95,52%	96,85%	98,11%
oferta	95,58%	99,29%	97,40%	98,38%
preço	88,63%	97,97%	93,06%	97,39%

Tabela 13 – Métricas com a nova base de teste com aplicação das regras

	Precision	Recall	F-score	AP
descrição	97,15%	96,28%	96,71%	94,78%
imagem	97,49%	98,45%	97,97%	98,05%
oferta	97,70%	99,15%	98,42%	98,25%
preço	96,76%	97,54%	97,15%	97,04%

5.4.3 Novos supermercados

Até o momento estávamos utilizando para teste apenas folhetos dos mesmos supermercados que tiveram folhetos presentes no treinamento. Decidimos então rotular uma nova base de teste com 100 imagens de folhetos de supermercados que não possuíssem nenhum dos

seus folhetos no treino. O mesmo processo de rotulação das etapas anteriores foi aplicado e os resultados podem ser observados na [Tabela 14](#). Após a aplicação das regras, como muitas detecções incorretas são eliminadas, a *precision* apresenta um aumento considerável, tornando-se maior que 90% para todas as classes, isso pode ser observado na [Tabela 15](#).

Tabela 14 – Métricas com novos supermercados

	Precision	Recall	F-score	AP
descrição	79,99%	84,19%	82,04%	82,36%
imagem	96,41%	94,57%	95,48%	94,41%
oferta	95,25%	94,18%	94,71%	93,87%
preço	89,21%	87,97%	88,58%	87,20%

Tabela 15 – Métricas com novos supermercados após regras

	Precision	Recall	F-score	AP
descrição	95,01%	80,36%	87,08%	79,32%
imagem	99,06%	93,54%	96,22%	93,49%
oferta	98,92%	93,87%	96,33%	93,61%
preço	97,07%	86,47%	91,46%	86,08%

5.4.4 Experimento com ofertas

Nas seções anteriores, as melhores métricas foram encontradas especialmente na categoria oferta. Por conta disso, foi feito um experimento de detecção de preço, imagem e descrição utilizando apenas as seções do folheto onde as ofertas haviam sido detectadas. Como a estratégia não foi bem sucedida, descartamos a sua utilização no detector, porém os resultados encontrados podem ser verificados no [Apêndice B](#).

5.5 Resultados

Nessa seção vamos recapitular os resultados obtidos com o detector, que contou com a utilização de 700 imagens para seu treino e a aplicação das regras desenvolvidas. Os resultados são exibidos em duas situações, a primeira delas, com a utilização de supermercados conhecidos pelo modelo ([Tabela 16](#)), obteve resultados com precision e recall acima de 96% em todas as classes. A segunda situação ([Tabela 17](#)) expõe a capacidade do modelo de detectar corretamente os alvos em folhetos de supermercados desconhecidos, apresentando uma diminuição nas métricas aferidas para descrições e preços, em relação ao caso anterior.

Tabela 16 – Métricas com supermercados conhecidos

	Precision	Recall	F-score	AP
descrição	97,15%	96,28%	96,71%	94,76%
imagem	97,49%	98,45%	97,97%	98,05%
oferta	97,70%	99,15%	98,42%	98,25%
preço	96,76%	97,54%	97,15%	97,04%

Tabela 17 – Métricas com supermercados desconhecidos

	Precision	Recall	F-score	AP
descrição	95,01%	80,36%	87,07%	79,32%
imagem	99,06%	93,54%	96,22%	93,49%
oferta	98,92%	93,87%	96,33%	93,61%
preço	97,07%	86,47%	91,46%	86,08%

Para o detector proposto neste trabalho, a métrica mais importante é a *precision*, pois é mais importante que os produtos localizados nos folhetos acompanhem suas descrições e preços corretos, assegurando uma confiabilidade maior ao LudiiPrice. O *recall* também é uma métrica muito importante, pois desejamos que o máximo de produtos contidos nos folhetos possam alimentar a base do aplicativo, porém a falta de alguns produtos são menos prejudiciais à aplicação do que uma informação incorreta de alguns deles. Dessa forma, mesmo na base de supermercados desconhecidos, nosso modelo apresentou um desempenho que consideramos satisfatório para resolução da situação proposta no trabalho.

6

Considerações Finais

O presente trabalho surgiu para integrar o LudiiPrice, com a finalidade de facilitar a comparação de preços entre estabelecimentos que divulgam suas ofertas em folhetos. Para esse propósito, foram analisadas técnicas de *deep learning* para detectar produtos e preços em folhetos digitais de supermercados.

Inicialmente foi criado um *web crawler* para realizar o download de imagens de encartes de supermercados para a formação da base de dados. Após a execução do *web crawler* e seleção das imagens, partimos para o estágio de rotulação. Ao final da fase da rotulação atingimos o objetivo de construir uma base de dados, onde foram rotulados 900 folhetos de 20 supermercados diferentes com 13841 ofertas, 13900 imagens, 14175 descrições e 15829 preços.

O próximo passo foi a avaliação e treinamento de modelos de detecção. Realizamos o treinamento com 2 modelos diferentes e selecionamos o que obteve melhores resultados, o YOLOv5. Conseguimos aprimorar as métricas do modelo a partir da construção de algoritmos, que intitulamos de regras, resultando em *average precision* maior que 96% para todas as classes detectadas em supermercados que passaram pelo treinamento. As métricas também foram calculadas numa nova base de teste com supermercados diferentes dos que passaram pelo treinamento, com esses dados o modelo obteve 79% de AP para descrição, 86% para preço e 93% para imagem e oferta.

Apesar dos resultados apresentados na detecção de objetos nos folhetos de supermercados, podem ser utilizadas outras estratégias para o aperfeiçoamento das métricas em supermercados novos, especialmente para preços e descrições, além da exploração do reconhecimento de caracteres aplicado às imagens resultantes do nosso modelo para a finalização da integração do trabalho ao LudiiPrice.

Referências

CALEFATI, A. et al. Using convolutional neural networks for content extraction from online flyers. In: *DocEng 2016 - Proceedings of the 2016 ACM Symposium on Document Engineering*. [s.n.], 2016. p. 127–130. Cited By :1. Disponível em: <www.scopus.com>. Citado 3 vezes nas páginas 23, 24 e 25.

DRUZHKOVA, P.; KUSTIKOVA, V. A survey of deep learning methods and software tools for image classification and object detection. *Pattern Recognition and Image Analysis*, Springer, v. 26, n. 1, p. 9–15, 2016. Citado 2 vezes nas páginas 13 e 14.

GALLO, I.; ZAMBERLETTI, A.; NOCE, L. *Content extraction from marketing flyers*. [s.n.], 2015. v. 9256. 325–336 p. (Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), v. 9256). Cited By :4. Disponível em: <www.scopus.com>. Citado 2 vezes nas páginas 24 e 25.

GANDRA, A. Preços determinam decisão de compra dos consumidores, indica pesquisa. *Agência Brasil*, p. 2015–03, 2015. Citado na página 10.

GIRSHICK, R. Fast r-cnn. In: *Proceedings of the IEEE international conference on computer vision*. [S.l.: s.n.], 2015. p. 1440–1448. Citado na página 14.

GIRSHICK, R. et al. Region-based convolutional networks for accurate object detection and segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 38, n. 1, p. 142–158, 2016. Citado na página 14.

GOOGLE. *Colaboratory - Perguntas frequentes*. 2021. FAQ do Google Colaboratory. Disponível em: <<https://research.google.com/colaboratory/intl/pt-BR/faq.html>>. Acesso em: 12 out 2022. Citado na página 32.

HORVAT, M.; GLEDEC, G. A comparative study of yolov5 models performance for image localization and classification. Citado na página 15.

HOSANG, J. et al. What makes for effective detection proposals? *IEEE transactions on pattern analysis and machine intelligence*, IEEE, v. 38, n. 4, p. 814–830, 2015. Citado na página 14.

KAUSAR, M. A.; DHAKA, V.; SINGH, S. K. Web crawler: a review. *International Journal of Computer Applications*, Foundation of Computer Science, v. 63, n. 2, 2013. Citado na página 13.

KHANDELWAL, R. *Evaluating performance of an object detection model*. 2020. Disponível em: <<https://towardsdatascience.com>>. Citado na página 18.

LAMPERT, C. H.; BLASCHKO, M. B.; HOFMANN, T. Beyond sliding windows: Object localization by efficient subwindow search. In: *2008 IEEE Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2008. p. 1–8. Citado na página 15.

LIU, W. et al. Ssd: Single shot multibox detector. In: SPRINGER. *European conference on computer vision*. [S.l.], 2016. p. 21–37. Citado 2 vezes nas páginas 14 e 16.

- LUCERI, B. et al. The impact of store flyers on store performance: a format and customer related approach. *International Journal of Retail & Distribution Management*, Emerald Group Publishing Limited, 2014. Citado na página 10.
- MOSQUERA, H. P.; GENC, Y. Recognition and classifying sales flyers using semi-supervised learning. In: *2019 4th International Conference on Computer Science and Engineering (UBMK)*. [S.l.: s.n.], 2019. p. 1–6. Citado 3 vezes nas páginas 11, 21 e 22.
- PADILLA, R.; NETTO, S. L.; SILVA, E. A. B. da. A survey on performance metrics for object-detection algorithms. In: *2020 International Conference on Systems, Signals and Image Processing (IWSSIP)*. [S.l.: s.n.], 2020. p. 237–242. Citado na página 17.
- 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. Citado 3 vezes nas páginas 11, 14 e 15.
- REN, S. et al. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, v. 28, 2015. Citado na página 14.
- REZATOFIGHI, H. et al. Generalized intersection over union: A metric and a loss for bounding box regression. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. [S.l.: s.n.], 2019. p. 658–666. Citado na página 16.
- UIJLINGS, J. R. et al. Selective search for object recognition. *International journal of computer vision*, Springer, v. 104, n. 2, p. 154–171, 2013. Citado na página 14.
- ULTRALYTICS. *YOLOv5*. 2020. [Código-fonte]. Disponível em: <<https://github.com/ultralytics/yolov5>>. Acesso em: 12 out 2022. Citado 3 vezes nas páginas 15, 16 e 33.
- VENTURIERI, A.; SANTOS, J. d. Técnicas de classificação de imagens para análise de cobertura vegetal. *Sistemas de informações geográficas: aplicações na agricultura*, v. 2, p. 351–371, 1998. Citado na página 13.
- VINODABABU, S. *a PyTorch Tutorial to Object Detection*. 2018. [Código-fonte]. Disponível em: <<https://github.com/sgrvinod/a-PyTorch-Tutorial-to-Object-Detection>>. Acesso em: 12 out 2022. Citado na página 34.
- ZOU, Z. et al. Object detection in 20 years: A survey. *arXiv preprint arXiv:1905.05055*, 2019. Citado na página 14.

Apêndices

APÊNDICE

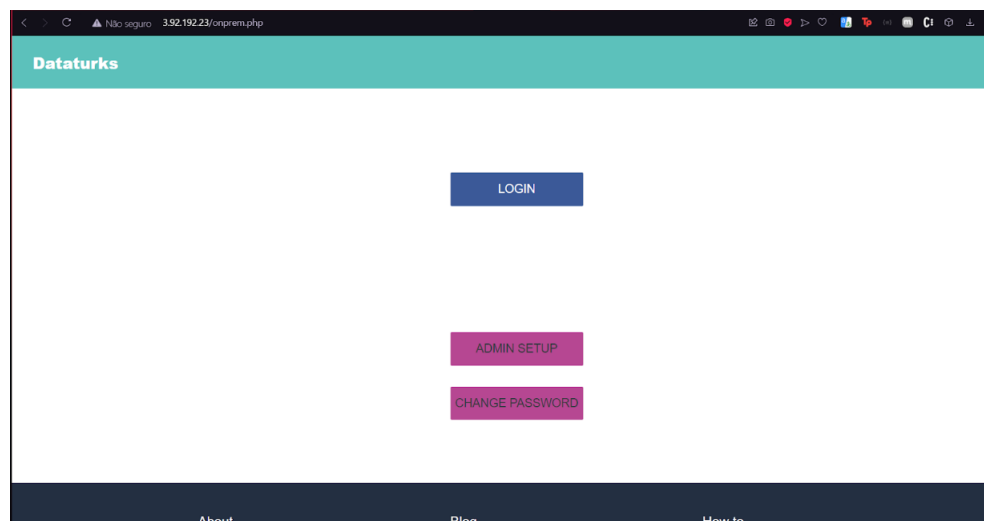


Manual para Rotulação

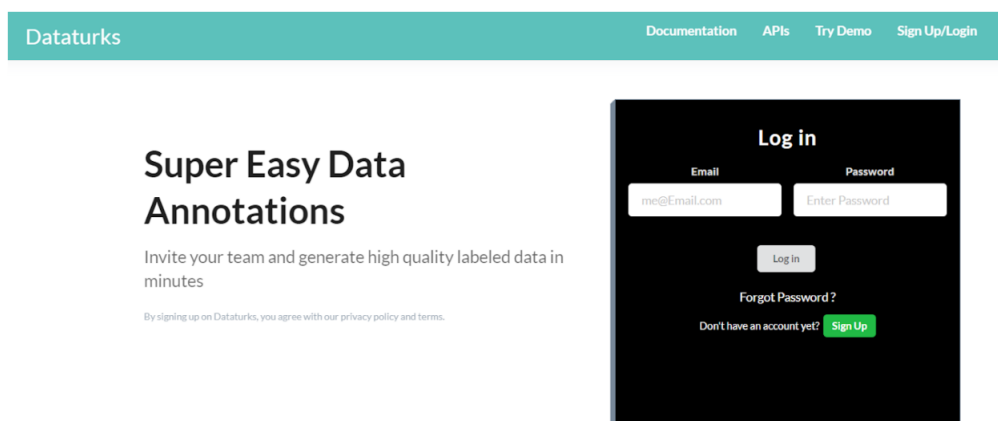
Dataturks - Passo a Passo

Acesso ao Dataturks

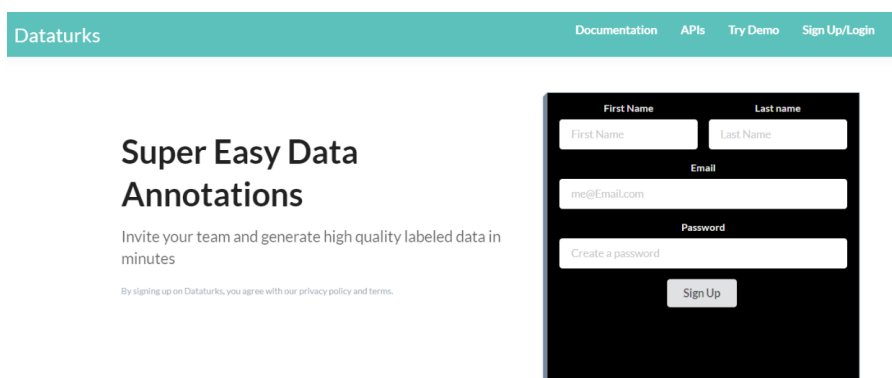
1. Acesse: <http://3.92.192.23>
2. Clique em LOGIN



3. No 1º acesso, clique em Sign Up. Caso contrário, digite email e senha para entrar

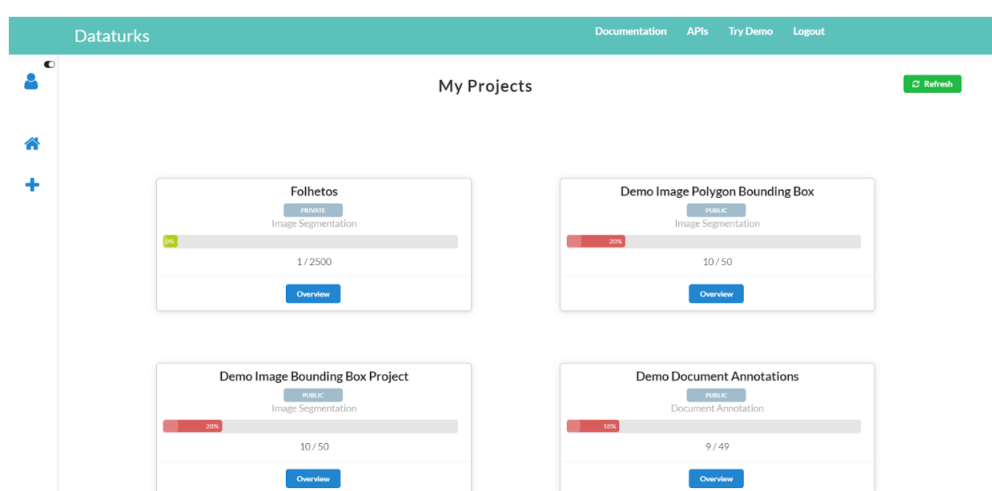


4. Realize o cadastro

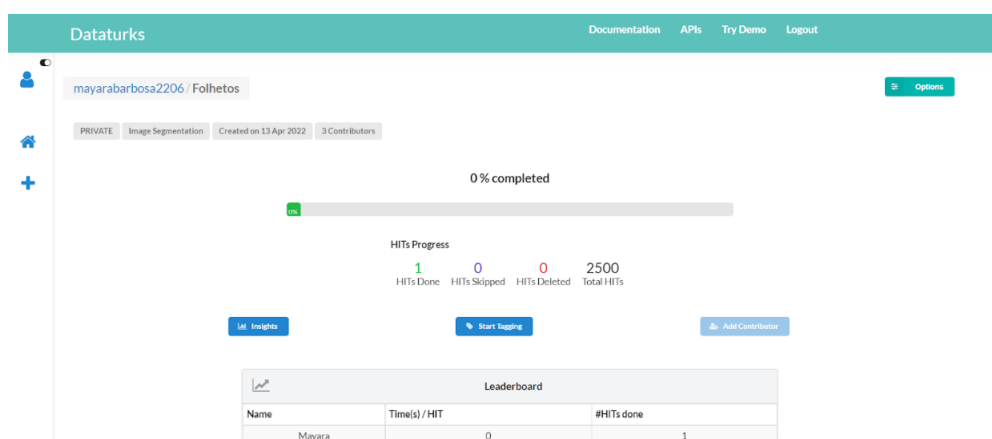


5. Para que você possa ser adicionado ao projeto, informe o email utilizado através do Whatsapp 79998560395 ou envie um email para mayara.barbosa@dcomp.ufs.br

6. Quando o email tiver sido adicionado, você poderá ver o projeto chamado Folhetos, clique nele para acessar



7. Clique em Start Tagging para começar a rotular



Leaderboard		
Name	Time(s) / HIT	#HITS done
Mayara	0	1

Rotulagem

Existem 4 categorias que devem ser avaliadas nas ofertas:

- **Preço:** o valor do produto
- **Descrição:** o título e informações sobre o produto;
- **Imagem:** a imagem do produto;
- **Oferta:** toda área relativa a uma oferta, que engloba preço, descrição e imagem;

Para cada oferta de um folheto, devem ser feitos retângulos para as categorias descritas anteriormente. Para isso, clique na categoria na coluna direita e selecione o objeto desta categoria na imagem, como pode ser visualizado na imagem a seguir, onde o preço do produto foi selecionado.



Repita o procedimento para todas as categorias encontradas em todas as ofertas do folheto. Quando completar a rotulagem do folheto completo, clique em Next ou Move To Done.



Observações

1. Para produtos que possuam mais de um preço, como por exemplo, preço de atacado e preço de varejo, selecione os dois com a tag preço. Na imagem a seguir podemos ver um caso que possui dois preços e também acompanha duas descrições, as duas também devem ser rotuladas com a tag descrição.



2. Não utilize nenhuma das tags para situações que fujam do padrão, por exemplo, a mesma imagem para dois produtos, mesma descrição para dois produtos, produto apenas com desconto e sem preço, um preço para duas descrições ou outras situações em que não seja possível rotular individualmente o produto. Nesses casos, deixe o produto sem rotulação. Abaixo segue exemplos que devem ser deixados sem rótulos.



APÊNDICE

B

Experimento de Detecção em Ofertas

Um experimento realizado para aperfeiçoar os resultados do modelo foi aplicar a detecção apenas nas seções do folheto onde as ofertas haviam sido detectadas. Como em todas as fases de detecção realizadas anteriormente, as melhores métricas foram encontradas na categoria de ofertas, decidimos utilizar o modelo para detectar os *bounding boxes* das outras classes, ou seja, preço, imagem e descrição, apenas no recorte obtido da oferta.

Foram utilizadas as 100 imagens de teste rotuladas anteriormente. Os resultados encontrados podem ser visualizados nas colunas à direita da [Tabela 18](#), já as colunas da esquerda representam os valores obtidos na última detecção. Houve um grande declínio nas métricas em relação a detecção passada, onde se utilizava o folheto completo. A [Figura 23](#) ilustra um exemplo de folheto com os resultados na detecção anterior e com o escopo reduzido às ofertas, onde uma parte dos *bounding boxes* são perdidos e outros passam a ser incorretos. Por conta disso, descartamos a aplicação desse passo no detector proposto.

Tabela 18 – Métricas com a detecção em ofertas

	Precision Antes	Precision Depois	Recall Antes	Recall Depois	F-score Antes	F-score Depois	AP Antes	AP Depois
descrição	82,66%	74,01%	96,56%	21,97%	89,07%	33,88%	94,80%	19,51%
imagem	95,24%	49,48%	95,52%	50,21%	96,85%	49,84%	98,11%	29,40%
preço	88,63%	66,51%	97,97%	26,31%	93,06%	37,70%	97,39%	21,38%

Figura 23 – Detecção de imagem

SUPER OFERTAS ATACADÃO **4** ATACADÃO Lugar de comprar barato

OFERTAS VÁLIDAS DE 16/11 A 30/11/2021

imagem 0.83 Café Maratá Tradicional 250g 5,19	imagem 0.67 Café Maratá Tradicional 500g 10,38	imagem 0.80 Café Maratá Tradicional 1kg 3,49
imagem 0.87 Cappuccino Maratá 6,49	imagem 0.81 Filtro de Papel para Café Maratá 3,29	imagem 0.69 Adoçante Líquido Maratá 2,29
imagem 0.86 Achocolatado em Pó Maratá 6,99	imagem 0.81 Chá Maratá Preto 2,59	imagem 0.92 Farinha Láctea Maratá 3,99

COMPRE NO PARCEIRÃO COM CARTÃO DE CRÉDITO OU VALE-ALIMENTAÇÃO

Consulte no site os cartões aceitos na loja mais próxima de você. Acesse atacadao.com.br/lojas

(a) Detecção Anterior

SUPER OFERTAS ATACADÃO **4** ATACADÃO Lugar de comprar barato

OFERTAS VÁLIDAS DE 16/11 A 30/11/2021

imagem 0.49 Café Maratá Tradicional 250g 5,19	imagem 0.39 Café Maratá Tradicional 500g 10,38	imagem 0.39 Café Maratá Tradicional 1kg 3,49
imagem 0.37 Cappuccino Maratá 6,49	imagem 0.40 Filtro de Papel para Café Maratá 3,29	imagem 0.29 Adoçante Líquido Maratá 2,29
imagem 0.33 Achocolatado em Pó Maratá 6,99	imagem 0.43 Chá Maratá Preto 2,59	imagem 0.56 Farinha Láctea Maratá 3,99

COMPRE NO PARCEIRÃO COM CARTÃO DE CRÉDITO OU VALE-ALIMENTAÇÃO

Consulte no site os cartões aceitos na loja mais próxima de você. Acesse atacadao.com.br/lojas

(b) Detecção em Ofertas

Fonte: Elaborada pelo autor