

UNIVERSIDADE DE SÃO PAULO

Instituto de Ciências Matemáticas e de Computação

Conscientização nutricional a partir de Detecção de Objetos em Imagens e Raspagem de Dados

Marcelo Felipe Alves Souza

Monografia - MBA em Inteligência Artificial e Big Data

SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-USP

Data de Depósito:

Assinatura: _____

Marcelo Felipe Alves Souza

Conscientização nutricional a partir de Detecção de Objetos em Imagens e Raspagem de Dados

Monografia apresentada ao Departamento de Ciências de Computação do Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo - ICMC/USP, como parte dos requisitos para obtenção do título de Especialista em Inteligência Artificial e Big Data.

Área de concentração: Inteligência Artificial

Orientador: Prof. Dr. Fernando Pereira dos Santos

Versão original

São Carlos

2023

AUTORIZO A REPRODUÇÃO E DIVULGAÇÃO TOTAL OU PARCIAL DESTE TRABALHO, POR QUALQUER MEIO CONVENCIONAL OU ELETRÔNICO PARA FINS DE ESTUDO E PESQUISA, DESDE QUE CITADA A FONTE.

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi, ICMC/USP, com os dados fornecidos pelo(a) autor(a)

S856m	<p>Souza, Marcelo Felipe Alves</p> <p>Conscientização nutricional a partir de Detecção de Objetos em Imagens e Raspagem de Dados / Marcelo Felipe Alves Souza ; orientador Fernando Pereira dos Santos. – São Carlos, 2023.</p> <p>58 p. : il. (algumas color.) ; 30 cm.</p> <p>Monografia (MBA em Inteligência Artificial e Big Data) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, 2023.</p> <p>1. LaTeX. 2. abnTeX. 3. Classe USPSC. 4. Editoração de texto. 5. Normalização da documentação. 6. Tese. 7. Dissertação. 8. Documentos (elaboração). 9. Documentos eletrônicos. I. Santos, Fernando Pereira, orient. II. Título.</p>
-------	---

Marcelo Felipe Alves Souza

Nutrional Label based on Object Detection and Web Scraping

Monograph presented to the Departamento de Ciências de Computação do Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo - ICMC/USP, as part of the requirements for obtaining the title of Specialist in Artificial Intelligence and Big Data.

Concentration area: Artificial Intelligence

Advisor: Prof. Dr. Fernando Pereira dos Santos

Original version

São Carlos

2023

*Este trabalho é dedicado à Comunidade Científica
para o desenvolvimento e disseminação da Inteligência Artificial no Brasil.*

AGRADECIMENTOS

Ao professor e orientador Prof. Dr. Fernando Pereira dos Santos pelo apoio e paciência.

Aos professores, funcionários e colegas da Segunda Turma do MBA - Inteligência Artificial e Big Data, que contribuíram para o desenvolvimento do curso.

A minha esposa Milena Marcato da Silva, pelo estímulo e amor incondicional.

Ao meu leal amigo de quatro patas Dino.

A toda a minha família pelo companheirismo.

RESUMO

SOUZA, M.F.A. **Conscientização nutricional a partir de Detecção de Objetos em Imagens e Raspagem de Dados**. 2023. 58p. Monografia (MBA em Inteligência Artificial e Big Data) - Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, 2023.

Neste estudo utilizou-se Inteligência Artificial e Visão Computacional para criar uma ferramenta de conscientização nutricional, tendo como motivação o aumento da obesidade e diabetes no Brasil e no mundo. Duas arquiteturas da linguagem Python foram usadas para este fim, o Detectron2 (Facebook) e YOLOv8 (Ultralytics). A partir de modelos pré-treinados de ambas arquiteturas, utilizou-se um banco de dados contendo 4900 imagens para treinamento de refinamento e 1600 imagens para validação, a fim de identificar e segmentar alimentos em um prato. Além disso, uma etapa adicional de avaliação de desempenho usando distúrbios no contraste e brilho das imagens foi feito de modo a verificar a robustez dos resultados em condições adversas de iluminação. A arquitetura YOLOv8s mostrou o melhor desempenho na detecção de objetos e segmentação, com alta revocação e separação de instâncias (mAP50 e mAP50-95). Após a escolha do melhor modelo, foi adicionada a etapa de aquisição de informação nutricional a partir da técnica de raspagem (arquitetura Selenium), incorporando um resumo dessas informações à imagem segmentada. A ferramenta se mostrou promissora para auxiliar os usuários na escolha de alimentos mais saudáveis e o uso de inteligência artificial e visão computacional foi eficaz na identificação e segmentação de alimentos. Sugere-se expandir o estudo para incluir a estimativa do volume dos alimentos e o cálculo do Índice Glicêmico, bem como aumentar a variedade de tipos de preparo dos alimentos analisados.

Palavras-chave: Inteligência Artificial. Ciência de Dados. Python. Aprendizado de Máquinas. Visão Computacional. Raspagem. Selenium. YOLOv8. Nutrição. Detecção de Objetos. Segmentação de Instâncias. Banco de Dados de Alimentos.

ABSTRACT

SOUZA, M.F.A. **Nutritional Awareness based on Image Object Detection and Web Scraping**. 2023. 58p. Monograph (MBA in Artificial Intelligence and Big Data) - Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, 2023.

In this study, Artificial Intelligence and Computer Vision were used to create a nutritional awareness tool, motivated by the increase in obesity and diabetes in Brazil and worldwide. Two architectures of the Python language were used for this purpose, Detectron2 (Facebook) and YOLOv8 (Ultralytics). From pre-trained models of both architectures, a database containing 4900 images for fine-tuning training and 1600 images for validation was used to identify and segment food on a plate. Furthermore, an additional step of performance evaluation using disturbances in the contrast and brightness of the images was done in order to verify the robustness of the results in adverse lighting conditions. The YOLOv8s architecture showed the best performance in object detection and segmentation, with high recall and instance separation (mAP50 and mAP50-95). After choosing the best model, the step of acquiring nutritional information was added using the technique of scraping (Selenium architecture), incorporating a summary of this information into the segmented image. The tool proved to be promising to help users choose healthier foods and the use of artificial intelligence and computer vision was effective in identifying and segmenting foods. It is suggested to expand the study to include the estimation of food volume and the calculation of the Glycemic Index, as well as to increase the variety of types of preparation of the analyzed foods.

Keywords: Artificial Intelligence. Data Science. Python. Machine Learning. Computer Vision. Web Scraping. Selenium. YOLOv8. Nutrition. Object Detection. Instance Segmentation. Food Dataset.

LISTA DE FIGURAS

Figura 1 – IMC por ano no Mundo — Autoria Própria usando Dados da OMS . . .	21
Figura 2 – IMC por ano no Brasil — Autoria Própria usando Dados da OMS . . .	22
Figura 3 – Casos de Diabetes por ano no Brasil — Autoria Própria usando Dados da OMS	22
Figura 4 – Segmentação de Instâncias por Comaniciu e Meer 2002, em (SZELISKI, 1986)	25
Figura 5 – Detecção de Locais por Philbin, Chum, Isard et al. 2007, em (SZELISKI, 1986)	26
Figura 6 – Potenciais usos do YOLOv8 — Imagem do <i>website</i> Roboflow	26
Figura 7 – Arquitetura do YOLOv8 — Imagem de RangeKing (usuário do GitHub)	27
Figura 8 – Modelos de Segmentação do YOLOv8 — Imagem da Ultralytics	28
Figura 9 – Arquitetura do Detectron2 — Imagem de Hiroto Honda (usuário do Medium)	28
Figura 10 – Potenciais usos do Detectron2 — Imagem do <i>website</i> Meta/Detectron2	28
Figura 11 – Fórmula de Cálculo do IoU - Imagem de Eric Hofesmann (usuário do Medium)	29
Figura 12 – Exemplo de busca de informações no código HTML — Autoria Própria usando o <i>website</i> FoodData Central	31
Figura 13 – Amostra do Banco de Dados (HAIR, 2023)	32
Figura 14 – Histograma do Banco de Dados — Autoria Própria	32
Figura 15 – Amostra baixa iluminação do Banco de Dados (HAIR, 2023)	33
Figura 16 – Amostra alta iluminação do Banco de Dados (HAIR, 2023)	33
Figura 17 – Esquema Geral do trabalho — Artigo Im2Calories	35
Figura 18 – Aplicação para o Usuário Final — Artigo Im2Calories	36
Figura 19 – Primeira parte do fluxo de execução — Autoria Própria	37
Figura 20 – Segunda parte do fluxo de execução — Autoria Própria usando a imagem do <i>website</i> : https://www.apinchofhealthy.com/baked-chicken-breast/ .	37
Figura 21 – Exemplo de Rótulo Nutricional de Arroz Branco - <i>Website</i> FoodData Central	40
Figura 22 – Exemplo de busca por Arroz - <i>Website</i> FoodData Central	41
Figura 23 – Entrada e Saída da Junção das Ferramentas — Autoria Própria usando a imagem do <i>website</i> : https://www.apinchofhealthy.com/baked-chicken-breast/	42
Figura 24 – Resultados Compilados YOLOv8n - Autoria Própria	45
Figura 25 – Resultado de Segmentação do Modelo YOLOv8n — Autoria Própria .	45
Figura 26 – Resultados Compilados YOLOv8s - Autoria Própria	47

Figura 27 – Resultado de Segmentação do Modelo YOLOv8s — Autoria Própria . .	47
Figura 28 – Aplicação das Ferramentas Exemplo 1 — Autoria Própria	52
Figura 29 – Aplicação das Ferramentas Exemplo 2 — Autoria Própria	52
Figura 30 – Aplicação das Ferramentas Exemplo 3 — Autoria Própria	53
Figura 31 – Aplicação das Ferramentas Exemplo 4 — Autoria Própria	53
Figura 32 – Aplicação das Ferramentas Exemplo 5 — Autoria Própria	53

LISTA DE TABELAS

Tabela 1 – Rótulos dos Alimentos do Banco de Dados utilizados no estudo - Autoria Própria	34
Tabela 2 – Configuração do modelo Detectron2 - Autoria Própria	39
Tabela 3 – Resultados do treinamento do modelo Detectron2	44
Tabela 4 – Comparação entre os diferentes modelos e condições - Autoria Própria	48
Tabela 5 – Comparação entre classes com alto número de instâncias - Autoria Própria	50
Tabela 6 – Comparação entre classes com baixo número de instâncias - Autoria Própria	51
Tabela 7 – Resultados dos Exemplos de Aplicação - Autoria Própria	54

LISTA DE ABREVIATURAS E SIGLAS

TCC	Trabalho de Conclusão de Curso
IA	Inteligência Artificial
OMS	Organização Mundial da Saúde
IMC	Índice de Massa Corporal
RNC	Rede Neural Convolucional
AM	Aprendizado de Máquinas
VC	Visão Computacional
SI	Segmentação de Imagens
DO	Detecção de Objetos
PLN	Processamento de Língua Natural
IoU	<i>Intersection over Union</i>
mAP	<i>mean Average Precision</i>

SUMÁRIO

1	INTRODUÇÃO	21
1.1	Hipóteses e Objetivos	23
1.2	Organização do texto	24
2	FUNDAMENTAÇÃO TEÓRICA	25
2.1	Visão Computacional	25
2.1.1	Segmentação de Instâncias	25
2.1.2	Detecção de Objetos	25
2.1.3	Arquitetura YOLOv8	26
2.1.4	Modelo Detectron2	27
2.1.5	Métricas de Avaliação e Funções de Perda	28
2.1.5.1	Métricas de Avaliação	29
2.1.5.2	Funções de Perda	30
2.2	Processamento de Língua Natural	30
2.2.1	Raspagem	30
2.2.2	Selenium	31
2.3	Banco de Dados	32
3	TRABALHOS RELACIONADOS	35
4	METODOLOGIA	37
4.1	Identificação e Rotulação dos Alimentos	38
4.1.1	Redes Neurais	38
4.1.2	Avaliação dos Modelos	38
4.1.3	Validação Adicional com Modificações de Brilho e Contraste no Banco de Dados	39
4.2	Busca de Informação Nutricional	39
4.2.1	Raspagem	40
4.2.2	Junção das Ferramentas e Saída Esperada	42
5	ANÁLISE DOS RESULTADOS	43
5.1	Treinamento dos Modelos	43
5.1.1	Treinamento Detectron2	43
5.1.2	Treinamento YOLOv8	43
5.1.2.1	Modelo YOLOv8n	44
5.1.2.2	Modelo YOLOv8s	46
5.2	Comparação entre diferentes Modelos e Condições de Luminosidade	46

5.2.1	Resultados Gerais	46
5.2.2	Resultados pelo Número de Instâncias	49
5.3	Integração dos Resultados e Aplicação	50
6	CONCLUSÕES	55
6.1	Próximos passos	55
	Referências	57

1 INTRODUÇÃO

A obesidade tem aumentado de forma contínua ao longo dos anos, o que leva a consequências tanto para a saúde dos indivíduos, quanto para os Governos, devido aos altos custos operacionais inerentes ao problema em questão (KUMANYIKA *et al.*, 2002). O Índice de Massa Corporal (IMC), ou do inglês *Body Mass Index (BMI)* é um indicador que auxilia o monitoramento da obesidade. Segundo o CDC (sigla em inglês para *Centers for Disease Control and Prevention*), considera-se que um indivíduo tem peso normal caso seu IMC varie entre 18,5 e 25; pré-obeso, se os valores estiverem entre 25 e 30; obeso, se os valores forem maiores do que 30 (CDC, 2022). A partir da Figura 1 é possível observar que o IMC de todos os continentes estão em ascendência segundo os dados da Organização Mundial da Saúde (OMS)¹.

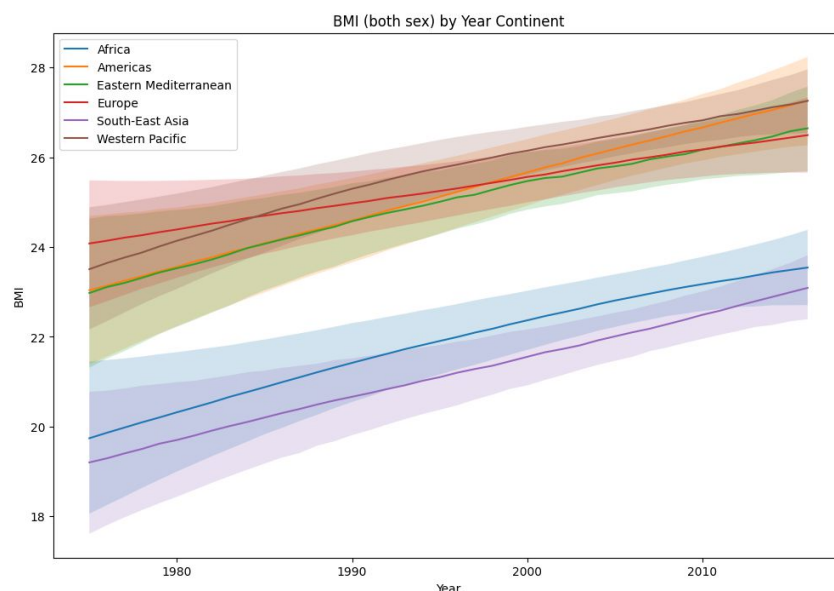


Figura 1 – IMC por ano no Mundo — Autoria Própria usando Dados da OMS

O mesmo tipo de análise pode ser feita para o Brasil, vide Figura 2, refletindo a mesma tendência de aumento ao longo dos anos, mostrando uma média de IMC configurada por sobrepeso (25 a 30) (CDC, 2022).

Uma das condições associadas à obesidade — e, consequentemente, com o IMC — é a Diabetes, especialmente Tipo II. Pode-se observar a tendência de aumento da Diabetes no Brasil ao longo dos anos (WHO, 2015) na Figura 3, utilizando os dados da OMS².

Uma ferramenta que pode auxiliar os indivíduos a atingir uma alimentação mais saudável e, com isso, atingir um IMC próximo da faixa de 18,5 a 25 seria a utilização

¹ <[https://www.who.int/data/gho/data/indicators/indicator-details/GHO/mean-bmi-\(kg-m\)-\(age-standardized-estimate\)](https://www.who.int/data/gho/data/indicators/indicator-details/GHO/mean-bmi-(kg-m)-(age-standardized-estimate))> (WHO, 2017)

² <<https://ncdportal.org/CountryProfile/GHE110/BRA>> (WHO, 2015)

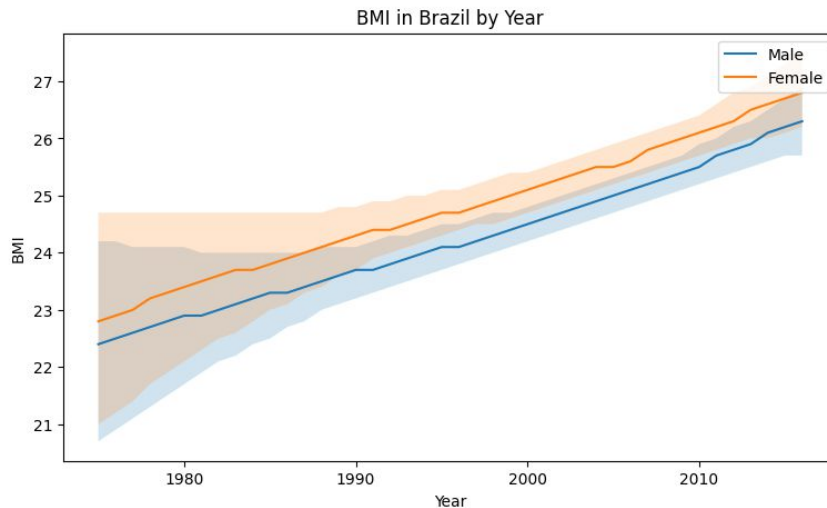


Figura 2 – IMC por ano no Brasil — Autoria Própria usando Dados da OMS

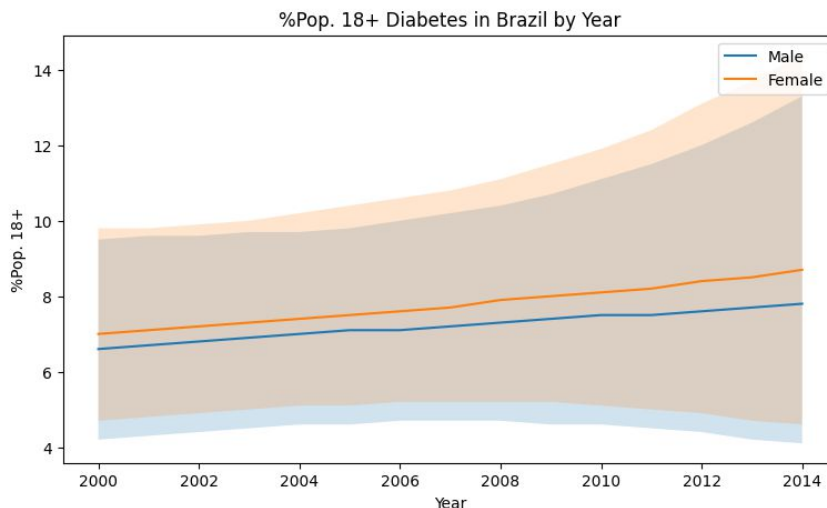


Figura 3 – Casos de Diabetes por ano no Brasil — Autoria Própria usando Dados da OMS

eficiente dos rótulos nutricionais, porém grande parte dos consumidores não sabe utilizar esse artifício corretamente (COWBURN; STOCKLEY, 2005).

O uso de Inteligência Artificial (IA) pode ser usado para a identificação e segregação de alimentos, bem como para a incorporação de dados nutricionais destes, de modo a tornar o processo de reconhecimento rápido e intuitivo, conforme (MEYERS *et al.*, 2015). O uso de Redes Neurais Convolucionais (RNC) pode trazer altos valores de precisão para esta tarefa de identificação e segregação de alimentos — da ordem de 90%, segundo (POPLY; JOTHI, 2021). Além disso, aliado à tarefa de identificação de alimentos, pode-se usar a IA juntamente com as informações da *web*, numa tarefa conhecida como raspagem (do inglês *Web Scraping*), usando a conhecida biblioteca do Python Selenium (MUTHUKADAN, 2023). Dessa forma, assim que um alimento seja reconhecido, pode-se usar esse recurso para a obtenção de informações nutricionais detalhadas, tornando-se uma poderosa combinação de ferramentas para que esse processo de reconhecimento de quais alimentos trarão

benefícios seja realizado de forma prática e em poucas etapas.

Outras aplicações relacionadas à IA são ferramentas para predição e manutenção de Diabetes. Segundo (CHAKI *et al.*, 2022), algoritmos de Aprendizado de Máquinas (AM) podem ser treinados para uma rápida detecção de glaucoma advindo da Diabetes, além do próprio processo de manutenção do índice glicêmico numa faixa de controle ótimo (entre 80 e 120 mg/mL de glicose no sangue).

1.1 Hipóteses e Objetivos

Conforme exposto no capítulo 1, ferramentas que auxiliem os indivíduos para a correta identificação dos rótulos nutricionais seriam de grande valia para uma potencial redução do IMC e, conseqüentemente, da obesidade. Além disso, uma rápida avaliação do conteúdo nutricional ajudaria pacientes de Diabetes com o controle glicêmico, evitando complicações provenientes de um desbalanço da glicose no sangue (ASSOCIATION, 2004).

O objetivo deste estudo é treinar e avaliar dois algoritmos livres (*open source*) de Visão Computacional (VC) — entre eles, a arquitetura estado-da-arte em Detecção de Objetos YOLOv8, sigla do inglês: *You Only Look Once*, (JOCHER; CHAURASIA; QIU, 2023) —, de modo a identificar diferentes tipos de alimento em um prato de comida — bem como associar esses alimentos aos componentes nutricionais via páginas da Internet usando a técnica de Processamento de Língua Natural (PLN) chamada Raspagem.

O algoritmo será treinado utilizando um banco de dados de imagens já existente, o que configura uma das hipóteses desse estudo: os 103 tipos de alimentos a serem identificados foram criados e identificados pelo autor do banco (HAIR, 2023) — fato que será explanado nos capítulos subsequentes. Além disso, um ponto relevante é que as Máscaras de Segmentação (já presentes no banco de dados) não são perfeitas - porém, adequadas -, contendo imperfeições que acarretarão em um resultado potencialmente pior que o esperado. Outra limitação deve-se aos rótulos de identificação não contemplarem diferentes tipos de preparo de alimentos (como assado, grelhado, frito etc.). Entretanto, o tempo designado para a criação de um banco de imagens significativamente grande, com rótulos e máscaras devidamente colocados, seria incompatível com o tempo proposto para a execução deste estudo. Dessa forma, o banco utilizado será considerado suficiente e conveniente.

O objetivo final deste estudo seria uma ferramenta para sugestão da dosagem de insulina aos pacientes de Diabetes Insulinodependentes a partir de uma imagem contendo o prato de comida a ser consumido — porém, essa tarefa exigiria uma complexidade de implementação que o tempo proposto para a conclusão deste trabalho não comportaria (trabalhos futuros). Dessa forma, uma restrição imposta neste estudo é a delimitação do escopo, englobando somente a tarefa de identificação dos alimentos em um prato de comida,

usando a técnica de Segmentação de Instâncias, juntamente com a busca de informações nutricionais na rede pelo uso da técnica de raspagem.

1.2 Organização do texto

Este Trabalho de Conclusão de Curso (TCC) está distribuído em 6 capítulos, incluindo esta introdução, dispostos conforme a descrição que segue:

Capítulo 1: Introdução relativa à maneira como a nutrição pode auxiliar no controle do IMC, bem como da obesidade e da Diabetes, trazendo métodos de IA como potenciais artifícios para este fim.

Capítulo 2: O sistema concebido é abordado de forma teórica, apresentando os dados utilizados para esta tarefa, os métodos de VC e PLN, bem como as métricas utilizadas.

Capítulo 3: São apresentados trabalhos relacionados à tarefa em questão.

Capítulo 4: Elaboração da proposta de como o estudo foi realizado, cobrindo os aspectos de configuração das bibliotecas e banco de dados utilizados. O objetivo desse capítulo é explorar o passo a passo das tarefas que foram processadas, codificadas e executadas.

Capítulo 5: Apresentação dos resultados, comparação dos modelos de VC propostos e análise das métricas para identificação do melhor modelo. Nesse capítulo, também é apresentada a aplicação da técnica de raspagem ao modelo de VC escolhido, onde há a interação entre os objetos detectados e os dados nutricionais obtidos a partir do *website* do governo Americano Food Data Central (AGRICULTURE, 2023).

Capítulo 6: Conclusões, considerações finais e próximos passos do estudo desenvolvido.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 Visão Computacional

A Segmentação de Instâncias (SI) e a Detecção de Objetos (DO) são tarefas de VC que envolvem a análise de imagens para identificar e localizar objetos. Embora compartilhem algumas semelhanças, cada uma tem objetivos e abordagens distintas. Entre as técnicas de VC citadas por (SZELISKI, 1986), observa-se que SI e DO são as que mais se aproximam do objetivo deste estudo.

2.1.1 Segmentação de Instâncias

A SI envolve a identificação e delimitação de objetos individuais em uma imagem, classificando cada objeto por tipo, tal como demonstrado na Figura 4. Isso pode ser útil para tarefas como rastreamento de objetos ao longo do tempo ou identificação de suas relações espaciais. A SI pode ser realizada usando modelos de aprendizado profundo, como *Mask R-CNN*, que combina DO com segmentação semântica para gerar máscaras de objetos. Criar máscaras é uma tarefa que consome tempo e recursos, tornando a atividade de SI limitada — apesar de ser uma técnica poderosa e de alta acurácia.

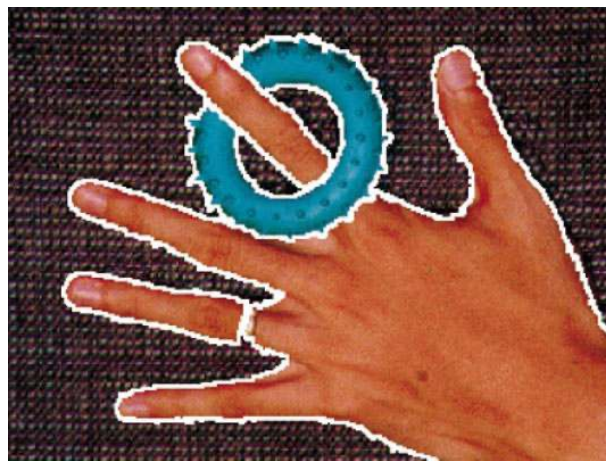


Figura 4 – Segmentação de Instâncias por Comaniciu e Meer 2002, em (SZELISKI, 1986)

2.1.2 Detecção de Objetos

A DO envolve a identificação de objetos em uma imagem e a criação de caixas delimitadoras ao redor deles, classificando cada objeto por tipo, conforme prévio treinamento - por exemplo, na Figura 5. Isso pode ser útil para tarefas como vigilância, direção autônoma ou identificação de pessoas em imagens. A DO pode ser realizada usando modelos de aprendizado profundo, como o YOLO (*You Only Look Once*), conhecido por sua velocidade e precisão, e que será explorado nas próximas seções deste estudo. A DO, ao contrário da

SI, não utiliza máscaras, mas anotações na imagem. Essas anotações podem ser realizadas em arquivos JSON (*JavaScript Object Notation*), XML (*Extensible Markup Language*) ou mesmo em arquivos simples de texto, indicando as coordenadas do(s) objeto(s) em questão para o treinamento do algoritmo.

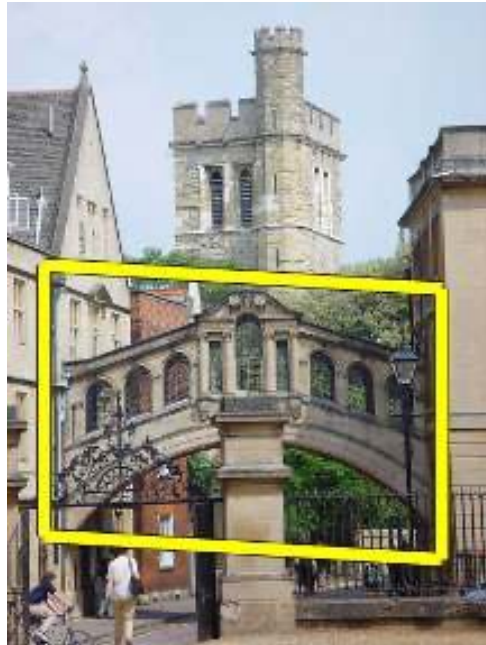


Figura 5 – Detecção de Locais por Philbin, Chum, Isard et al. 2007, em (SZELISKI, 1986)

2.1.3 Arquitetura YOLOv8

O algoritmo YOLOv8 (JOCHER; CHAURASIA; QIU, 2023) destina-se ao uso em VC, englobando a DO, a SI e a Classificação de Imagens, como mostrado na Figura 6. É considerada estado-da-arte pelo seu conjunto de imagens utilizado durante o pré-treinamento do modelo, pela quantidade de parâmetros presentes e pela complexidade da arquitetura - representada na Figura 7, contando com camadas de convolução e de C2f (do inglês *Coarse-to-Fine*, ou seja, uma estrutura de refinamento de parâmetros).

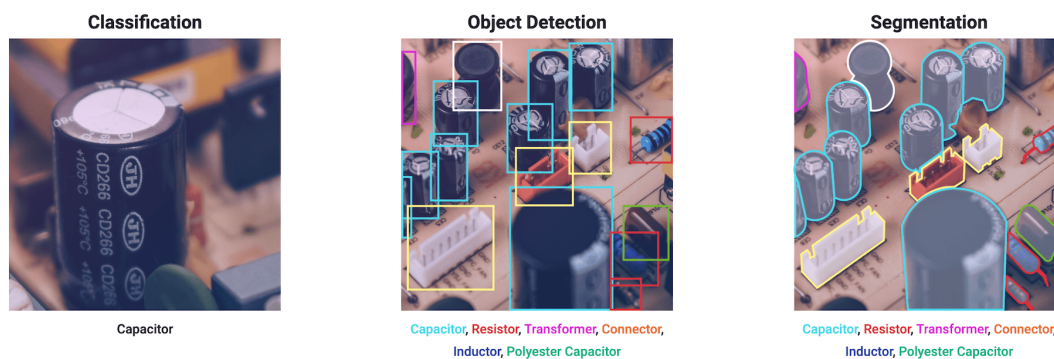


Figura 6 – Potenciais usos do YOLOv8 — Imagem do *website* Roboflow

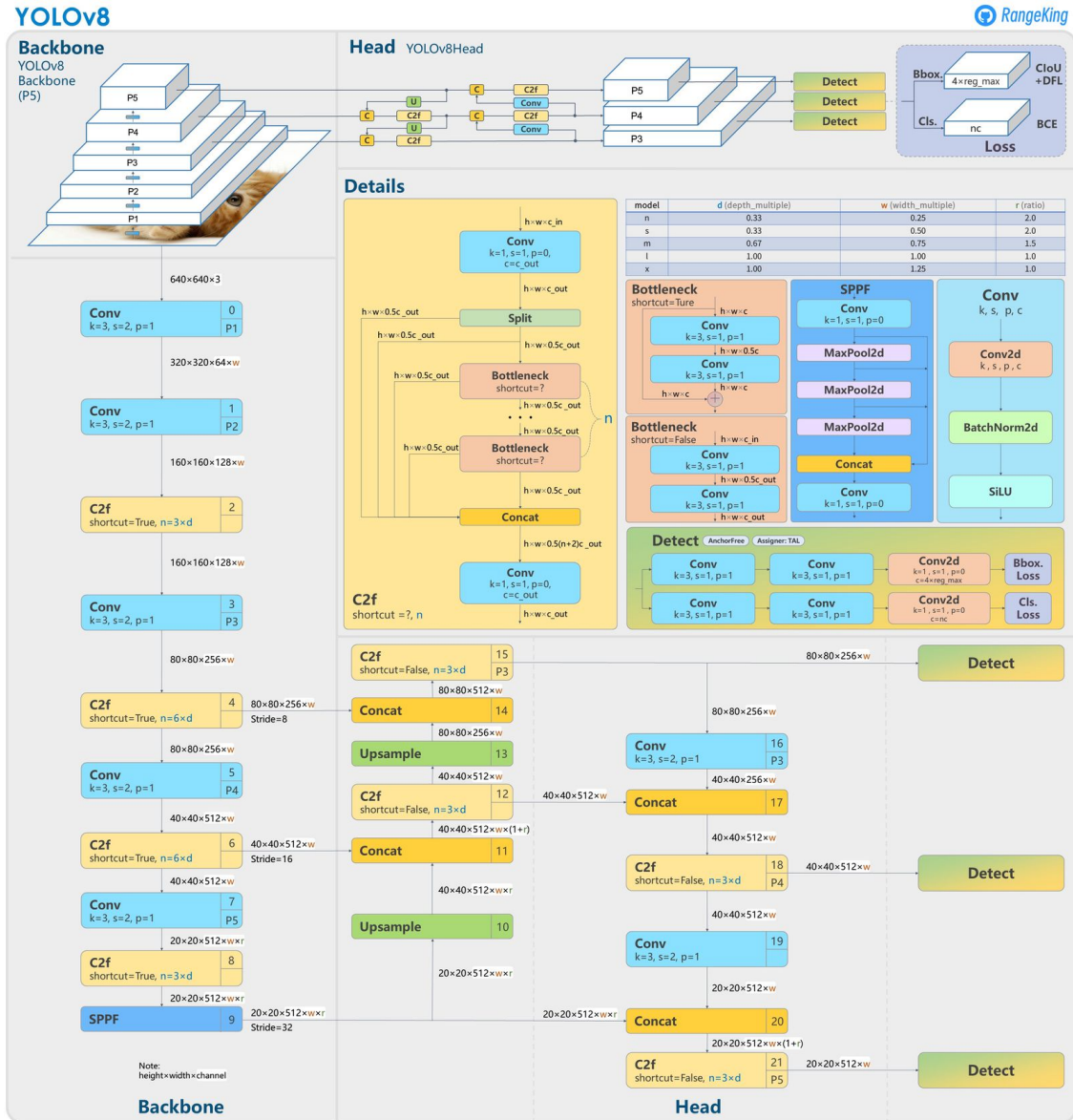


Figura 7 – Arquitetura do YOLOv8 — Imagem de RangeKing (usuário do GitHub)

A Figura 8 mostra diferentes modelos existentes na plataforma YOLOv8 para a tarefa de segmentação. É possível atingir uma acurácia entre 36,7% para 53,4% para a Média dos Valores de Precisão no caso de Caixas, e entre 30,5% para 43,4% no caso de Máscaras (resultados de referência, obtidos usando o banco de dados de validação COCO2017, segundo(JOCHER; CHAURASIA; QIU, 2023)) - para o caso desse estudo, os modelos ‘n’ e ‘s’ foram utilizados, com uma previsão de acurácia de 36% e 30% para DO e SI do menor modelo, e 45% e 37% para o maior modelo.

2.1.4 Modelo Detectron2

O algoritmo de VC Detectron2 (WU *et al.*, 2019), cuja biblioteca é geralmente escrita em Python, destina-se a tarefas de VC, tal como a arquitetura YOLO, porém com menor capacidade de atingir resultados estado-da-arte. Pela Figura 9 e Figura 10

Model	size (pixels)	mAP ^{box} 50-95	mAP ^{mask} 50-95	Speed CPU ONNX (ms)	Speed A100 TensorRT (ms)	params (M)	FLOPs (B)
YOLOv8n-seg	640	36.7	30.5	96.1	1.21	3.4	12.6
YOLOv8s-seg	640	44.6	36.8	155.7	1.47	11.8	42.6
YOLOv8m-seg	640	49.9	40.8	317.0	2.18	27.3	110.2
YOLOv8l-seg	640	52.3	42.6	572.4	2.79	46.0	220.5
YOLOv8x-seg	640	53.4	43.4	712.1	4.02	71.8	344.1

Figura 8 – Modelos de Segmentação do YOLOv8 — Imagem da Ultralytics

percebe-se que esse modelo também oferece diversas utilidades, além de contar com uma robusta arquitetura de Rede Neural Convolutacional Recorrente com FPN (do inglês *Feature Pyramid Network*).

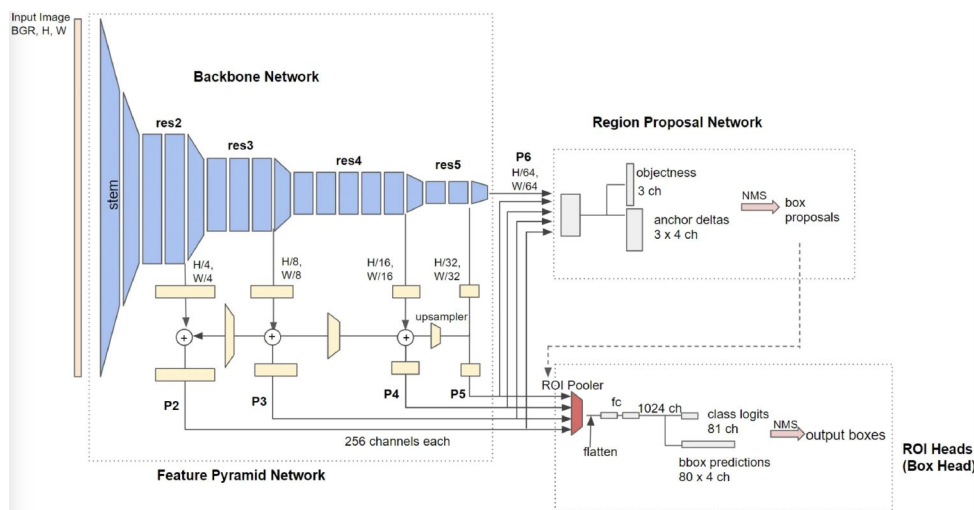


Figura 9 – Arquitetura do Detectron2 — Imagem de Hiroto Honda (usuário do Medium)



Figura 10 – Potenciais usos do Detectron2 — Imagem do *website* Meta/Detectron2

2.1.5 Métricas de Avaliação e Funções de Perda

As métricas de avaliação e funções de perda desempenham um papel essencial na análise de desempenho de modelos de DO e SI. Analisar essas métricas em conjunto fornece uma compreensão mais completa do desempenho do modelo, permitindo identificar

áreas onde o modelo pode ser melhorado ou ajustado. Abaixo estão algumas das principais métricas que serão utilizadas na seção de Resultados (5), segundo (EVERINGHAM *et al.*, 2009) e (ULTRALYTICS, 2023).

2.1.5.1 Métricas de Avaliação

- **Precisão (P):** A precisão (do inglês *precision*) é a razão entre as detecções verdadeiras positivas (detecções corretas) e o número total de previsões positivas realizadas pelo modelo. Uma precisão mais alta indica que o modelo produz menos falsos positivos.
- **Revocação (R):** A revocação (do inglês *recall*) é a razão entre as detecções verdadeiras positivas e o número total de instâncias positivas reais. Uma maior revocação indica que o modelo é melhor na detecção de instâncias positivas sem perdê-las.
- **Interseção sobre União:** A Interseção sobre União (do inglês *Intersection over Union* - ou IoU), pode ser vista matematicamente na Figura 11, representando a dimensão de cálculo desse parâmetro - observa-se, no numerador, a sobreposição entre a caixa de imagem predita no numerador e o rótulo original, ou verdade fundamental (do inglês *ground truth*), e a união entre esses dois parâmetros no denominador. Na prática, o IoU é uma medida de localização/acurácia entre a caixa de imagem predita e o rótulo original, podendo variar de 0 a 1 - sendo 0 nenhuma interseção entre a caixa predita e o valor verdadeiro, e 1 representando a interseção total.


$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$


Figura 11 – Fórmula de Cálculo do IoU - Imagem de Eric Hofesmann (usuário do Medium)

- **Média dos Valores de Precisão - mAP:** O mAP (do inglês *mean Average Precision*) é uma métrica resumida que calcula a média dos valores de precisão em diferentes níveis de revocação em todas as classes. É comumente usado para avaliar modelos de DO, sendo que valores altos para esta métrica indicam melhor desempenho.

1. **mAP50 ou mAP95:** O mAP50 representa a média do valor médio de precisão (mAP) com um limiar de IoU de 0,50 ou 0,95 para o conjunto de dados de teste do COCO. O índice 'B' representa Caixa (do inglês *Box*) e relaciona os

resultados à tarefa de DO, enquanto o índice ‘M’ representa Máscara (do inglês *Mask*) e relaciona os resultados à tarefa de SI. Valores altos para as métricas em questão indicam melhor desempenho nos limiares de IoU de 50% ou 95%, respectivamente.

2. **mAP50-95**: O mAP50-95 refere-se à média da precisão (mAP) com limiares de IoU variando entre 0,50 e 0,95, com um passo de 0,05 (0,50, 0,55, ..., 0,95) para o conjunto de dados de teste do PASCAL VOC. Valores altos para esta métrica indicam um melhor desempenho do modelo em vários limiares de IoU, o que significa que o modelo é mais preciso e robusto.

2.1.5.2 Funções de Perda

- **Perda de caixa (*Box loss*)**: A perda de caixa mede a diferença entre as coordenadas previstas da caixa delimitadora e as coordenadas da caixa delimitadora de referência. Minimizar essa perda ajuda o modelo a prever coordenadas de caixa delimitadora com mais precisão.
- **Perda de Segmentação (*Seg loss*)**: A perda de segmentação mede a diferença entre as máscaras de segmentação previstas pelo modelo e as máscaras de segmentação de referência. A perda de segmentação é tipicamente usada em tarefas de segmentação semântica e SI, onde o objetivo é prever uma máscara de segmentação precisamente para cada objeto na imagem. Minimizar essa perda ajuda o modelo a prever máscaras de segmentação mais precisas.

2.2 Processamento de Língua Natural

O PLN é um enorme campo de estudo da Linguagem, podendo contar com elementos de processamento de sintaxe, semântica, bem como identificação de ambiguidades e elementos complexos da linguagem (FINGER, 2021). Em suma, PLN engloba um imenso arcabouço ferramental para que a extração de características textuais possam ocorrer. No estudo desenvolvido somente a técnica de raspagem foi utilizada.

2.2.1 Raspagem

A técnica de extração de textos de páginas da *web* a partir da identificação de elementos do código HTML é conhecido como raspagem (do inglês *Web Scraping*), estando inserida dentro nas ferramentas do PLN. A Figura 12 mostra um exemplo de identificação de um objeto de interesse usando a ferramenta “Inspecionar”, presente em qualquer Navegador da atualidade.

Rice, white, medium-grain, cooked, unenriched

SR Legacy, released in April 2018, is the final release of this data type and will not be updated. FoodData Central.

Data Type: SR Legacy Food Category: Cereal Grains and Pasta FDC ID: 168930
FDC Published: 4/1/2019

Nutrients	Measures
-----------	----------

Portion:

Name	Amount	Unit	Deriv. By
Water	68.6	g	
Energy	130	kcal	Calculated
Energy	544	kJ	

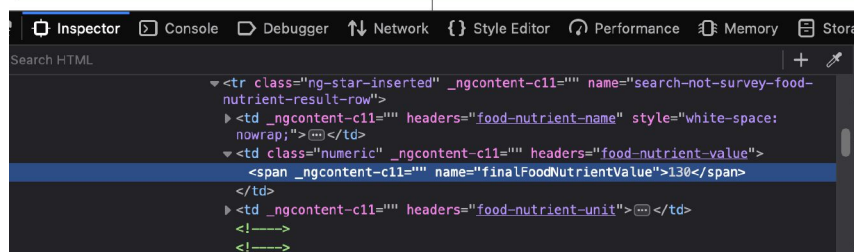


Figura 12 – Exemplo de busca de informações no código HTML — Autoria Própria usando o *website* FoodData Central

2.2.2 Selenium

A biblioteca Python Selenium (MUTHUKADAN, 2023) é uma ferramenta de automação *web* amplamente utilizada que permite interagir com navegadores de forma programática, permitindo aos desenvolvedores controlá-los e manipulá-los para realizar ações automatizadas - como preencher formulários, clicar em botões, navegar por páginas e extrair dados da *web*. O Selenium é especialmente útil para tarefas repetitivas que envolvem interação com interfaces *web*, como testes automatizados, raspagem e simulação de interações do usuário.

Uma das principais vantagens do Selenium é a sua capacidade de trabalhar com diferentes navegadores, incluindo Google Chrome, Mozilla Firefox, Safari e Microsoft Edge. Isso permite que os desenvolvedores escolham o navegador que melhor se adapte às suas necessidades e utilizem a mesma lógica de automação em diferentes plataformas. Além disso, o Selenium oferece suporte para várias linguagens de programação, incluindo Python.

No entanto, o Selenium também apresenta algumas limitações. Em primeiro lugar, sua execução pode ser relativamente lenta quando comparada a outras soluções de automação para raspagem. Além disso, o Selenium requer um navegador em execução para realizar as ações de automação, o que pode limitar sua escalabilidade e uso em ambientes

sem interface gráfica. Outra limitação é a necessidade de lidar com elementos dinâmicos na página, como carregamento assíncrono de conteúdo, que podem exigir estratégias adicionais para garantir que as ações sejam executadas corretamente.

2.3 Banco de Dados

Para a realização da identificação e segregação de alimentos em pratos de comida, o banco de dados presente no *website* Roboflow do usuário (HAIR, 2023) foi utilizado — a Figura 13 mostra um exemplo de imagem desse banco. O Histograma mostrando a frequência o número de alimentos com máscara por imagem pode ser visto na Figura 14 — aproximadamente 6500 imagens contendo pratos de comida foram utilizadas, sendo 4900 imagens para treinamento e 1600 imagens para teste.

Além das imagens originais, o modelo também foi validado usando esse banco de imagens com alterações no brilho e contraste, de modo a simular variações no *flash* e luminosidade da foto, testando a robustez do modelo para esse tipo de condição - representado na Figura 15 e Figura 16.

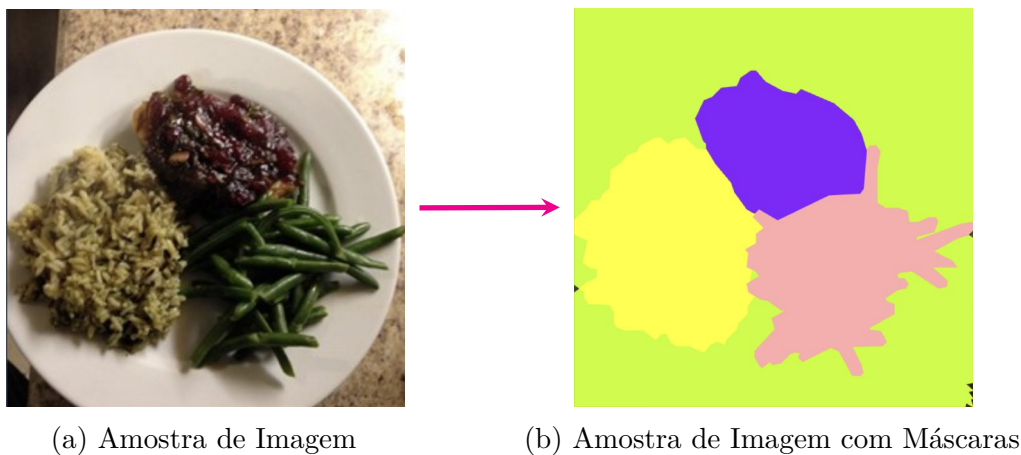


Figura 13 – Amostra do Banco de Dados (HAIR, 2023)

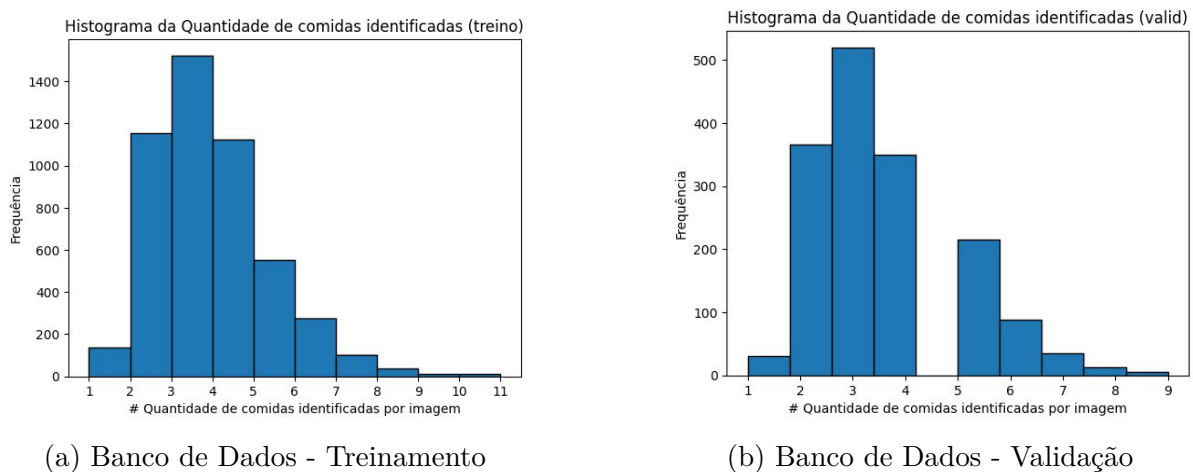


Figura 14 – Histograma do Banco de Dados — Autoria Própria



(a) Amostra de Imagem



(b) Imagem com menor Brilho e Contraste

Figura 15 – Amostra baixa iluminação do Banco de Dados (HAIR, 2023)



(a) Amostra de Imagem



(b) Imagem com maior Brilho e Contraste

Figura 16 – Amostra alta iluminação do Banco de Dados (HAIR, 2023)

A Tabela 1 mostra os 103 rótulos do Banco de Dados utilizado neste TCC. Conforme já explanado, por se tratar de um Banco de Dados já existente, nem todos os rótulos representam a vasta gama de variedades da culinária brasileira. Ainda assim, percebe-se que - em linhas gerais - esses alimentos contemplam os rótulos em um prato de comida considerado comum: arroz, batata, frango, peixe, carne, tipos de verdura, etc.

Duas limitações a serem discutidas posteriormente: o banco de dados em questão não teve imagens de rótulos retirados, uma vez que já continha uma quantidade relativamente pequena de amostras para treinamento e validação, implicando no treinamento de pares imagens-máscaras que por vezes são incorretos (i.e. imagem de um frango, rótulo de um peixe); além disso, os rótulos do banco de dados não possuem variação em termos do preparo do alimento, por exemplo: frango assado, frango grelhado e frango frito; ao invés disso, existe apenas o termo genérico ‘frango’.

Tabela 1 – Rótulos dos Alimentos do Banco de Dados utilizados no estudo - Autoria Própria

french beans	corn	lettuce	raspberry
almond	crab	mango	red beans
apple	cucumber	melon	rice
apricot	date	milk	salad
asparagus	dried cranberries	milkshake	sauce
avocado	egg tart	noodles	sausage
bamboo shoots	egg	okra	seaweed
banana	eggplant	olives	shellfish
bean sprouts	enoki mushroom	onion	shiitake
biscuit	fig	orange	shrimp
blueberry	fish	other ingredients	snow peas
bread	french fries	oyster mushroom	soup
broccoli	fried meat	pasta	soy
cabbage	garlic	peach	spring onion
cake	ginger	peanut	steak
candy	grape	pear	strawberry
carrot	green beans	pepper	tea
cashew	hamburg	pie	tofu
cauliflower	hanamaki baozi	pineapple	tomato
celery stick	ice cream	pizza	walnut
cheese butter	juice	popcorn	watermelon
cherry	kelp	pork	white button mushroom
chicken duck	king oyster mushroom	potato	white radish
chocolate	kiwi	pudding	wine
cilantro mint	lamb	pumpkin	wonton dumplings
coffee	lemon	rape	

3 TRABALHOS RELACIONADOS

Este TCC utiliza técnicas semelhantes às utilizadas em (MEYERS *et al.*, 2015), que agrega uma RNC Multi-Rótulo para a identificação de alimentos em um prato de comida e a extração de informações nutricionais, como calorias, por exemplo. A Figura 17 mostra uma visão geral do trabalho relacionado, que une detecção dos alimentos, quantificação de volume de cada alimento, conversão das informações nutricionais via banco de dados, e finalmente uma aplicação móvel que compila todas as ações ao usuário final.

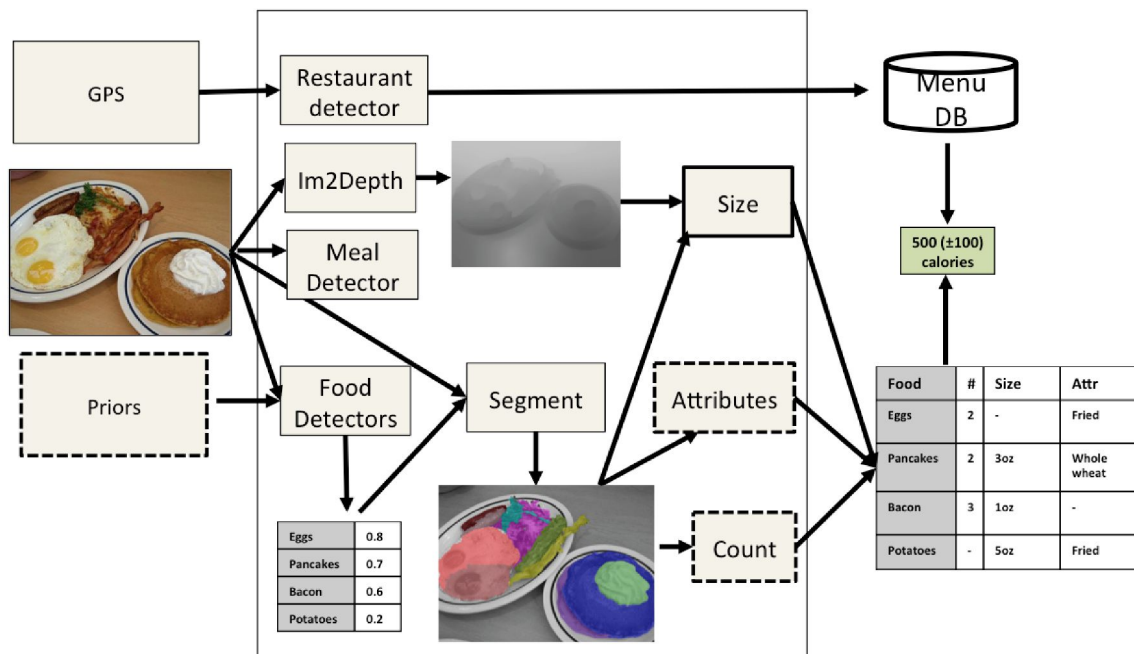


Figura 17 – Esquema Geral do trabalho — Artigo Im2Calories

A arquitetura de VC escolhida foi o GoogLeNet CNN, o banco de dados foi o *Food201-MultiLabel* - que inicialmente era o *Food101*, um banco de dados público contendo 101 classes, tornando-se o *Food201* após a adição de mais 100 classes, e finalmente tornando-se o *Food201-Segmented* após o autor realizar a adição das máscaras para a tarefa de SI - totalizando 35 mil imagens de treinamento e 15 mil imagens para teste. Após o treinamento, o modelo passou por um refinamento ao analisar os menus de restaurantes presentes na API (sigla do inglês *Application Programming Interface*) do Google Places. De modo geral, a detecção de alimentos atingiu resultados de mAP de 80% para classes dentro do *Food101* (banco original) e 20% para classes fora do *Food101*, resultando num valor médio de mAP equivalente a 50%.

Por fim, a estimativa de volume foi feita usando um banco de dados de alimentos 3D do Google chamada *GFood3d*. Para a estimativa de calorias, o autor utilizou um banco

de dados nutricional do Governo dos Estados Unidos (*USDA National Nutrient Database - NNDB*). A aplicação final foi feita através das etapas de detecção, estimativa de volume e cálculo de calorias; a entrada se dá por uma foto enviada pelo usuário e a saída é a identificação dos alimentos juntamente com o cálculo das calorias, o que pode ser observado na Figura 18.

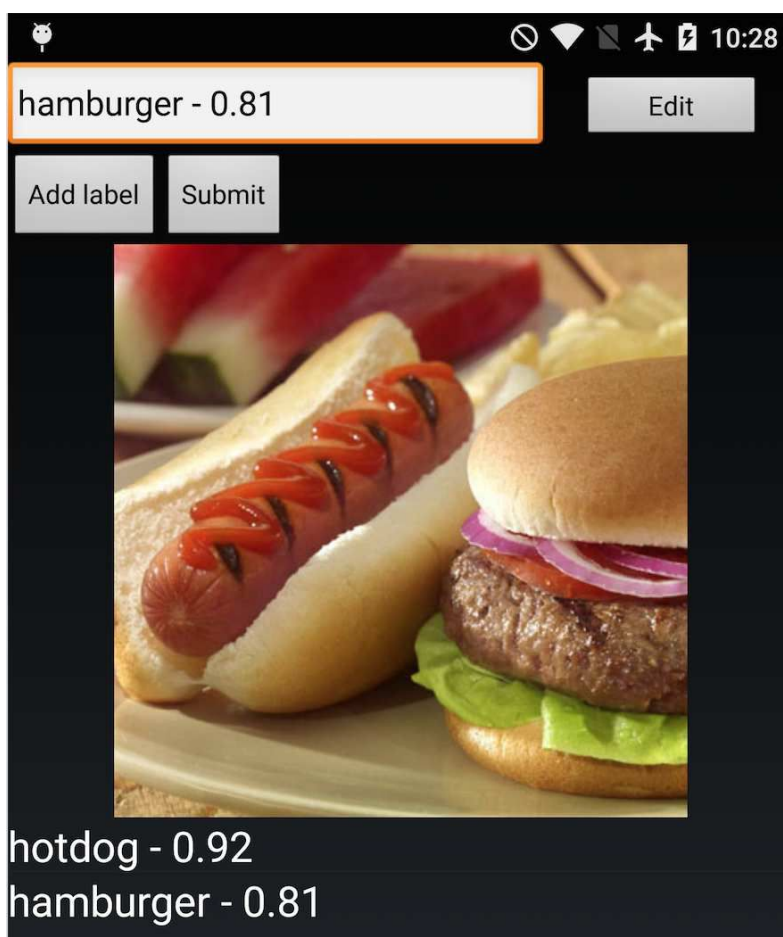


Figura 18 – Aplicação para o Usuário Final — Artigo Im2Calories

4 METODOLOGIA

A Figura 19 e a Figura 20 apresentam os conceitos do fluxo deste estudo. O primeiro fluxo (Figura 19) trata da seleção do modelo a ser utilizado - para tanto, conta com o treinamento e a avaliação de diferentes modelos de SI (YOLOv8 e Detectron2), seguido por uma validação adicional com o Banco de Dados original (4900 imagens utilizadas no treinamento e 1600 na validação) sofrendo perturbações de brilho e contraste. O segundo fluxo (Figura 20) trata do aspecto mais aplicável do modelo selecionado, onde os rótulos identificados são enviados para a biblioteca de raspagem (Selenium), retornando os rótulos nutricionais que serão incorporados à imagem final, juntamente com as Caixas Delimitadoras.

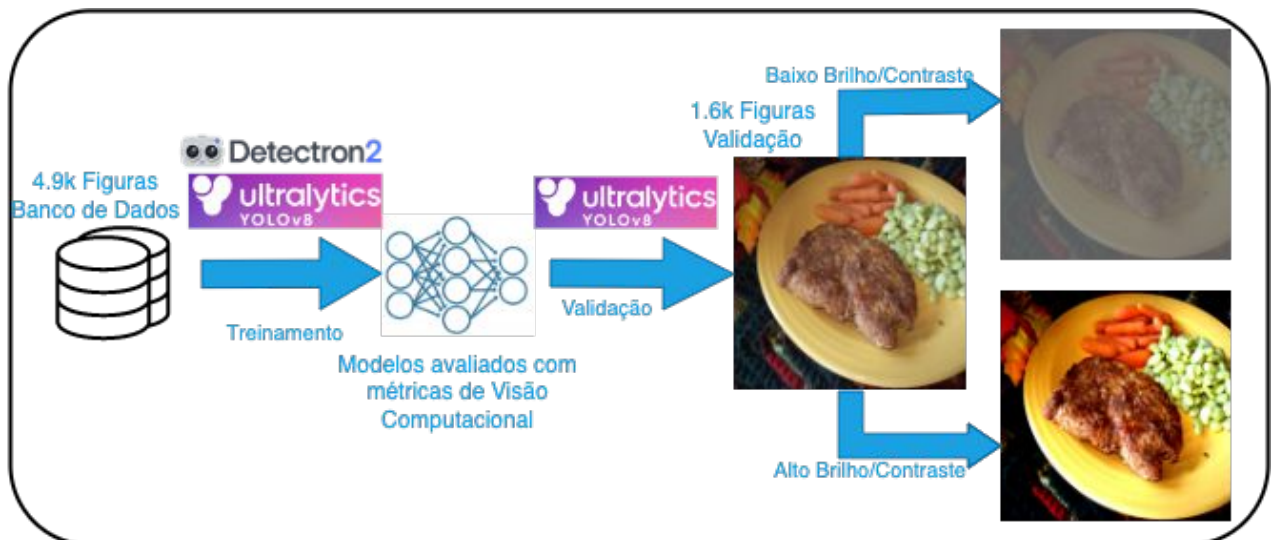


Figura 19 – Primeira parte do fluxo de execução — Autoria Própria

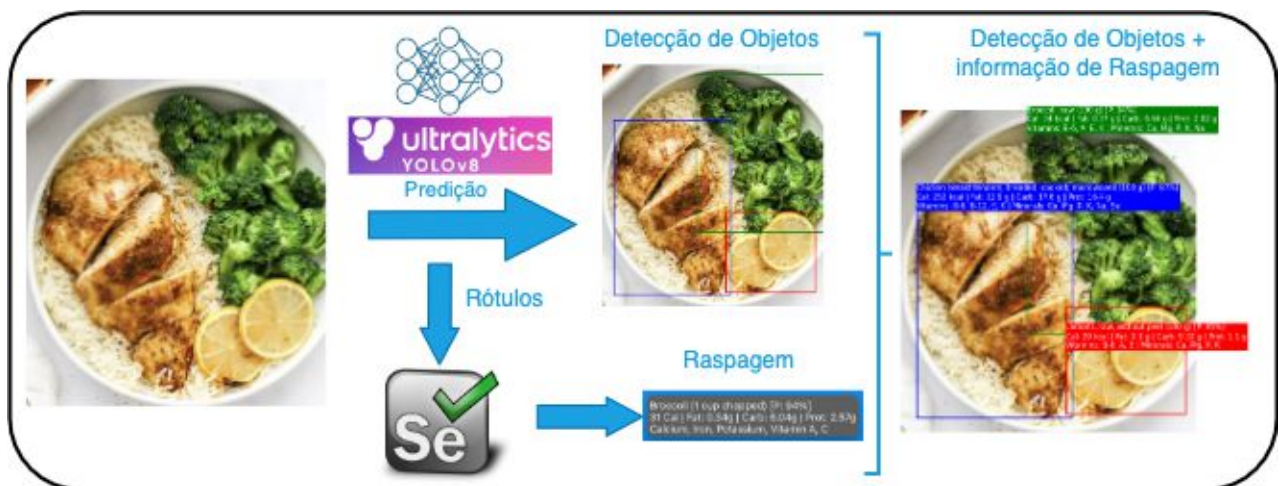


Figura 20 – Segunda parte do fluxo de execução — Autoria Própria usando a imagem do *website*: <https://www.apinchofhealthy.com/baked-chicken-breast/>

Em linhas gerais, durante a execução do fluxo espera-se que o modelo treinado seja capaz de reconhecer diferentes alimentos em uma imagem de um prato de comida (técnica de VC - SI e DO), fornecendo rótulos para que se possa trazer informações nutricionais dos alimentos identificados (técnica de PLN - Raspagem), compilando todo esse processo em uma imagem final. Dessa forma, a partir de apenas uma entrada - foto de um prato de comida -, o modelo processará as informações e a saída contará com o(s) alimento(s) detectados, bem como uma breve descrição das principais informações nutricionais (Calorias, Gordura, Carboidratos, Proteínas, Vitaminas e Minerais).

4.1 Identificação e Rotulação dos Alimentos

A tarefa de Identificação e Rotulação dos Alimentos usa as técnicas de VC: SI e DO. O objetivo dessa etapa é usar modelos pré-treinados do estado-da-arte com o refinamento desta rede (*fine-tuning*) a partir de um banco de dados customizado com milhares de imagens de pratos de comida, seguidas de seu rótulo e máscara.

4.1.1 Redes Neurais

Dois modelos foram usados para a tarefa de identificação e rotulação dos alimentos: Detectron2 (WU *et al.*, 2019) e YOLOv8 (JOCHER; CHAURASIA; QIU, 2023). Ambos foram treinados com o mesmo Banco de Dados (HAIR, 2023) para a tarefa de segmentação. Para o modelo YOLOv8 utilizou-se duas variações com diferentes quantidades de parâmetros utilizados no pré-treinamento desses modelos: YOLOv8n-seg (3,4 milhões de parâmetros) e YOLOv8s-seg (11,8 milhões de parâmetros). Ambos estão disponibilizados na página GitHub da Ultralytics¹. Nenhum parâmetro foi modificado em relação ao padrão para o treinamento dos modelos, apenas o número de épocas foi especificado para 30.

O modelo Detectron2 foi treinado usando o arquivo base de configuração, presente na página GitHub do próprio modelo ²: `mask_rcnn_R_101_FPN_3x.yaml` - uma Rede Neural Recorrente R-101 treinada com o arquivo checkpoint ³. As principais configurações sobrescritas para o treinamento desse modelo estão na Tabela 2.

4.1.2 Avaliação dos Modelos

A Avaliação dos Modelos ocorreu usando as métricas propostas e apresentadas no capítulo 2.1.5 - especificamente as métricas de Precisão, Revocação e mAP50/mAP50-95 para a tarefa de SI e DO, ou seja, os índices (B) e (M) dessas métricas. O modelo que apresentou um melhor resultado - seja o maior valor nas três métricas ou o mais consistentemente alto entre elas - foi considerado o modelo a seguir para as próximas

¹ <<https://github.com/ultralytics/ultralytics>>

² <<https://github.com/facebookresearch/detectron2/tree/main/configs/COCO-InstanceSegmentation>>

³ <https://github.com/facebookresearch/detectron2/blob/main/MODEL_ZOO.md>

Tabela 2 – Configuração do modelo Detectron2 - Autoria Própria

SOLVER.IMS_PER_BATCH	2
SOLVER.BASE_LR	0.00025
SOLVER.MAX_ITER	5000
MODEL.DEVICE	'cpu'
MODEL.ROI_HEADS.BATCH_SIZE_PER_IMAGE	512
MODEL.ROI_HEADS.NUM_CLASSES	103
TEST.EVAL_PERIOD	2
PATIENCE	5000

tarefas deste estudo. Ambos os modelos (Detectron2 e YOLOv8) já possuem ferramentas automáticas para a geração dessas métricas ao final da etapa de treinamento, cujos valores são utilizados para comparação.

4.1.3 Validação Adicional com Modificações de Brilho e Contraste no Banco de Dados

A fim de verificar a robustez do método de SI, o modelo selecionado no capítulo 4.2.2 foi submetido a uma validação adicional: o Banco de Dados original teve seu Brilho e Contraste variados para uma simulação de ambientes onde a fotografia do prato de comida é obtida sem iluminação, ou com muita luz do sol, por exemplo. Essa nova gama de imagens foi validada usando as mesmas métricas usadas para a seleção do modelo. Para tanto, a biblioteca do Python denominada *Pillow* (CLARK, 2023) foi utilizada. Abaixo encontra-se um trecho do código utilizado.

```
# Utilizando a biblioteca Pillow
from PIL import Image, ImageEnhance

# Ajuste de Brilho baseado em um fator de referencia
image = Image.open(image_path)
enhancer = ImageEnhance.Brightness(image)
return enhancer.enhance(brightness_factor)

# Ajuste de Contraste baseado em um fator de referencia
image = Image.open(image_path)
enhancer = ImageEnhance.Contrast(image)
return enhancer.enhance(contrast_factor)
```

4.2 Busca de Informação Nutricional

A tarefa de Busca de Informação Nutricional aplica as técnicas de VC juntamente com a técnica de PLN chamada Raspagem. Nessa etapa, o principal objetivo é promover algo próximo da experiência final para o usuário, ou seja, a saída deverá prover informações

de utilidade imediata. No caso desse estudo, essa etapa conta com o modelo de VC selecionado na etapa anterior - que fará a predição de imagens -, seguido de raspagem dos rótulos preditos e incorporação dessa informação em uma imagem de retorno para o usuário.

4.2.1 Raspagem

A biblioteca Python denominada Selenium (MUTHUKADAN, 2023) foi empregada para a tarefa de raspagem. O *website* FoodData Central (AGRICULTURE, 2023) foi selecionado para ser utilizado como provedor de informações e rótulos nutricionais. Na Figura 21 pode-se verificar como essas informações estão dispostas: um rótulo nutricional contendo os principais dados de Gordura, Carboidratos, Proteínas, principais Vitaminas e Minerais presentes, bem como as Calorias para uma dada porção.

Rice, white, medium-grain, cooked, unenriched

SR Legacy, released in April 2018, is the final release of this data type and will not be

Data Type: SR Legacy Food Category: Cereal Grains and Pasta FDC ID: 1
FDC Published: 4/1/2019

Nutrients Measures

Portion: 100g

Name	Amount	Unit	Deriv. By
Water	68.6	g	
Energy	130	kcal	Calculated
Energy	544	kJ	
Protein	2.38	g	
Total lipid (fat)	0.21	g	
Ash	0.21	g	
Carbohydrate, by difference	28.6	g	Calculated
Calcium, Ca	3	mg	
Iron, Fe	0.2	mg	
Magnesium, Mg	13	mg	

Figura 21 – Exemplo de Rótulo Nutricional de Arroz Branco - *Website* FoodData Central

O excerto de código abaixo mostra um exemplo de como informações do *website* FoodData Central (AGRICULTURE, 2023) são buscadas usando a técnica de raspagem. Nesse código, o rótulo, o valor e as unidades dos Nutrientes são obtidos por códigos na linguagem *HTML* e *JavaScript*.

```
# Utilizacao da biblioteca Selenium para encontrar os Nutrientes na pagina
da FoodData Central

nutrients = driver.find_elements('xpath', '[@name="finalFoodNutrientName"]')
values = driver.find_elements('xpath', '[@name="finalFoodNutrientValue"]')
units = driver.find_elements('xpath', '[@name="finalFoodNutrientUnit"]')

for nutrient, value, unit in zip(nutrients, values, units):
    if nutrient.text in required_nutrients and
        required_nutrients[nutrient.text] != 'Calories':
        nutrition_info[food_name][required_nutrients[nutrient.text]] =
            value.text
```

Foi necessária a criação de um dicionário de correspondências entre os rótulos preditos do YOLOv8 e as chaves de busca do *website* FoodData Central, uma vez que o *website* apresentava uma variedade enorme de itens com o mesmo nome. A Figura 22 mostra um exemplo dessa busca, quando se procura pelo termo arroz (*rice*) e encontra-se 132 resultados. Logo, o dicionário tem o objetivo de remover a ambiguidade entre esses rótulos. Abaixo pode-se verificar uma amostra:

NDB Number	Description	SR Food Category
25071	Rice crackers	Snacks
19052	Snacks, rice cakes, brown rice, buckwheat	Snacks
19413	Snacks, rice cakes, brown rice, corn	Snacks
19414	Snacks, rice cakes, brown rice, multigrain	Snacks
19416	Snacks, rice cakes, brown rice, rye	Snacks
19051	Snacks, rice cracker brown rice, plain	Snacks
32002	Rice and vermicelli mix, rice pilaf flavor, unprepared	Meals, Entrees, and Side Dishes

Figura 22 – Exemplo de busca por Arroz - *Website* FoodData Central

```
# Amostra do dicionario de correspondencias - chave = rotulo YOLO, valor =
chave de busca FoodData Central

correspondence_dict = {'banana': 'Bananas, raw', 'bread': 'Bread, white
wheat', 'broccoli': 'Broccoli, raw', 'cabbage': 'Cabbage, raw', 'cake':
'Cake, sponge, commercially prepared'}}
```

Uma segunda adequação necessária foi a busca do código identificador (chamado de 'fdc_id') em um arquivo CSV disponibilizado pelo *website* FoodData Central, uma vez que a URL da página desse *website* não usava os nomes dos alimentos, e sim esse código identificador. Essa alteração foi simples de ser feita, uma vez que o arquivo disponibilizado estava com a formatação adequada. Uma limitação para essa etapa é que o alimento identificado - e por sua vez reconhecido no dicionário de correspondências - fica restrito a

somente um tipo de instância do *website*. Isso também se deve ao fato de que a rotulagem do banco de dados original não considerou o tipo de preparo daquele alimento, portanto não é possível que essa informação seja incorporada à atividade de raspagem. Assim sendo, ao se identificar o componente ‘arroz’, não é possível uma distinção entre o arroz integral, arroz arbóreo ou arroz branco, por exemplo.

4.2.2 Junção das Ferramentas e Saída Esperada

Com as informações das coordenadas das Caixas Delimitadoras (obtidas pela predição da SI - YOLOv8), juntamente com as informações nutricionais (obtidas pela tarefa de Raspagem - Selenium), foi possível combinar as capacidades, trazendo para o usuário final uma experiência na qual a partir de uma foto de um prato de comida foi possível identificar cada alimento presente, bem como conhecer as informações nutricionais relevantes - seja para controlar a dieta, ou seja, para uma utilização médica relativa à restrição de um certo componente. A Figura 23 mostra o início e o fim desse processo.

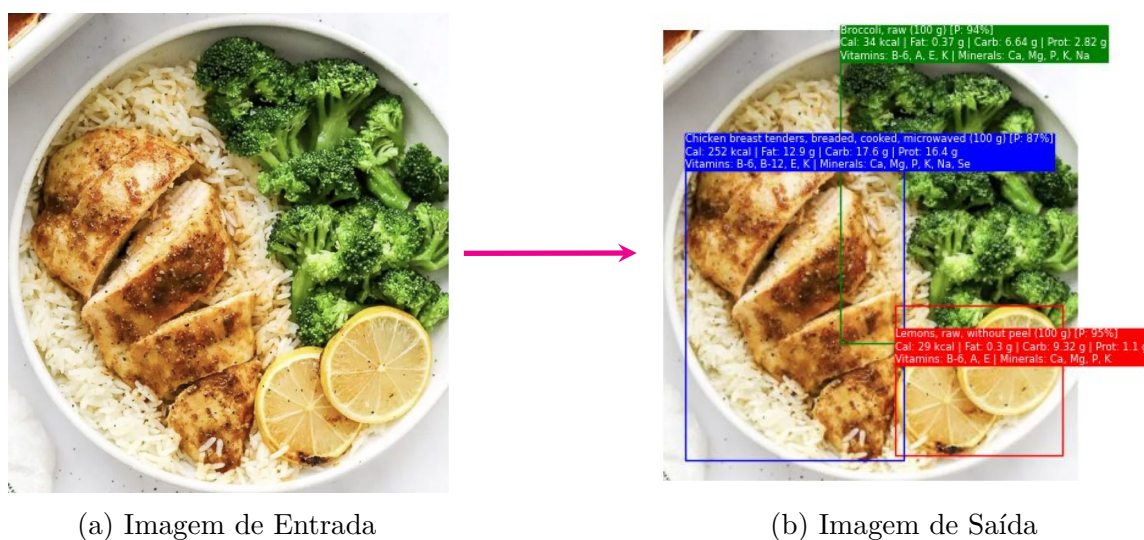


Figura 23 – Entrada e Saída da Junção das Ferramentas — Autoria Própria usando a imagem do *website*: <https://www.apinchofhealthy.com/baked-chicken-breast/>

O objetivo final dessa ferramenta é trazer a consciência nutricional de uma forma fácil, rápida e intuitiva, de modo contrário à forma tradicional, que consiste na leitura do rótulo físico de cada componente do prato de comida.

5 ANÁLISE DOS RESULTADOS

5.1 Treinamento dos Modelos

As próximas subseções abordam o treinamento dos modelos Detectron2 e YOLOv8 (variações ‘n’ e ‘s’). Ambos os modelos da arquitetura YOLOv8 apresentaram todos os indicadores de resultado melhor que o modelo Detectron2, como esperado. Esse último, mais antigo e com uma menor capacidade de detecção, foi adicionado ao estudo para montar uma linha de base, do modelo mais simples até o modelo estado-da-arte.

5.1.1 Treinamento Detectron2

A Tabela 3 mostra o resultado completo do treinamento do Detectron2, tanto para a tarefa de DO (coluna ‘*Box*’), quanto para a SI (coluna ‘*Segment*’). De modo geral, os resultados estão baixos em comparação aos resultados observados com a arquitetura YOLOv8 (que serão discutidos na próximas seções), com somente alguns valores de precisão e revocação acima de 10% - e isso é somente observado quando a Intersecção sobre a União (IoU) tem o seu menor valor (50%). Esse comportamento já era esperado, uma vez que o modelo Detectron2 parou de ser atualizado entre 2020 e 2021.

Ao se comparar o efeito da área de detecção para uma IoU de 50 a 95%, percebe-se que para áreas pequenas os resultados são baixos para a precisão e revocação em ambas as tarefas. Ao se aumentar a área para ‘*large*’ observa-se resultados ligeiramente melhores (9% na precisão e 15% na revocação - resultados em negrito). Isso se deve ao fato de que áreas menores implicam em uma maior sensibilidade de detecção, tornando a tarefa mais complexa, e o oposto para áreas grandes.

De modo geral, os resultados não foram satisfatórios para o modelo Detectron2, porém ele é um bom candidato para se montar uma linha de base na comparação com o modelo estado-da-arte YOLOv8.

5.1.2 Treinamento YOLOv8

Essa subseção foi subdividida em duas partes, a fim de analisar separadamente os resultados do modelo de menor quantidade de parâmetros (YOLOv8n - 3,4 milhões) e do modelo com maior quantidade de parâmetros (YOLOv8s - 11,8 milhões). Os resultados se apresentam na forma da tarefa de DO - indicada por ‘*Box*’ ou (B) - e SI - indicada por ‘*Segmentation*’ ou (M). Além disso, os dois eixos verticais foram utilizados em cada um dos gráficos, de modo a otimizar o espaço, sendo que os eixos à esquerda dos gráficos da esquerda apresentam os intervalos de valores da função perda (linhas tracejadas) durante as etapas de treinamento e de validação, enquanto o eixo da direita desse mesmo gráfico

Tabela 3 – Resultados do treinamento do modelo Detectron2

	Indicator	Box	Segment
Average Precision (AP)	[IoU=0,50:0,95 area=all maxDets=100]	0,083	0,084
	[IoU=0,50 area=all maxDets=100]	0,117	0,118
	[IoU=0,75 area=all maxDets=100]	0,093	0,092
	[IoU=0,50:0,95 area=small maxDets=100]	0	0
	[IoU=0,50:0,95 area=medium maxDets=100]	0,027	0,024
	[IoU=0,50:0,95 area=large maxDets=100]	0,086	0,089
Average Recall (AR)	[IoU=0,50:0,95 area=all maxDets=1]	0,144	0,146
	[IoU=0,50:0,95 area=all maxDets=100]	0,151	0,152
	[IoU=0,50:0,95 area=all maxDets=100]	0,151	0,152
	[IoU=0,50:0,95 area=small maxDets=100]	0	0
	[IoU=0,50:0,95 area=medium maxDets=100]	0,057	0,061
	[IoU=0,50:0,95 area=large maxDets=100]	0,154	0,155

apresenta a curva da precisão com uma linha contínua. Para os gráficos à direita, os intervalos de valores da revocação (linha contínua) podem ser vistos nos eixos à esquerda, enquanto os intervalos de valores das métricas mAP50 e mAP50-95 (linhas tracejadas) podem ser vistos nos eixos à direita.

5.1.2.1 Modelo YOLOv8n

Na Figura 24 observa-se os resultados do treinamento do modelo YOLOv8n. O modelo levou aproximadamente 24 horas para o treinamento em 30 épocas, utilizando o banco de dados das imagens de alimentos já apresentado. A configuração utilizada para o treinamento: MacBook 2021 chipset M1 Pro com 10 células de CPU e 16 células de GPU, no ambiente Jupyter e VSCode (versão 1.8).

Ao observar o gráfico contendo os valores da função perda durante as etapas de treinamento e de validação percebe-se que para ambas as tarefas a tendência de estabilidade ainda não foi atingida, sugerindo fortes indícios de que o modelo poderia ser rodado por mais épocas antes de se atingir o limite da função perda - fato limitado pelo atual *hardware*. A precisão apresentou um comportamento instável, oscilando entre valores de 45 a 55% nas duas tarefas, chegando ao final com aproximadamente 53%. Esse resultado está alinhado com o esperado para a versão 8 do modelo YOLO, conforme mostrado no capítulo 2.1.3.

O índice mAP50-95 mostra que ambas as tarefas obtiveram resultados abaixo do esperado: segundo o desenvolvedor, a versão 8 do YOLO poderia atingir resultados de 30% para segmentação e 37% para detecção (resultados obtidos usando o banco de dados de validação COCO2017, segundo(JOCHER; CHAURASIA; QIU, 2023)), porém obteve-se 20% para a segmentação e 23% para a detecção durante a etapa de treinamento. Além disso, assim como ocorreu com a função perda, percebe-se que as métricas mAP50 e mAP50-95 calculadas para as duas tarefas ainda não obtiveram valores estáveis, novamente indicando que um treinamento com mais épocas poderia resultar em melhores métricas.

A Figura 25 mostra um exemplo de validação da tarefa de SI. Percebe-se que o rótulo original (figura da esquerda) contém macarrão do tipo *Noodles*, um molho (‘*sauce*’) e brócolis, enquanto a figura com os rótulos preditos (figura da direita) apresenta os mesmos resultados, porém repetindo a identificação das classes de *Noodles* e de brócolis por mais de uma vez, indicando que o mAP50-95 com resultado de 20% pode não conseguir o enquadramento ideal do rótulo real.

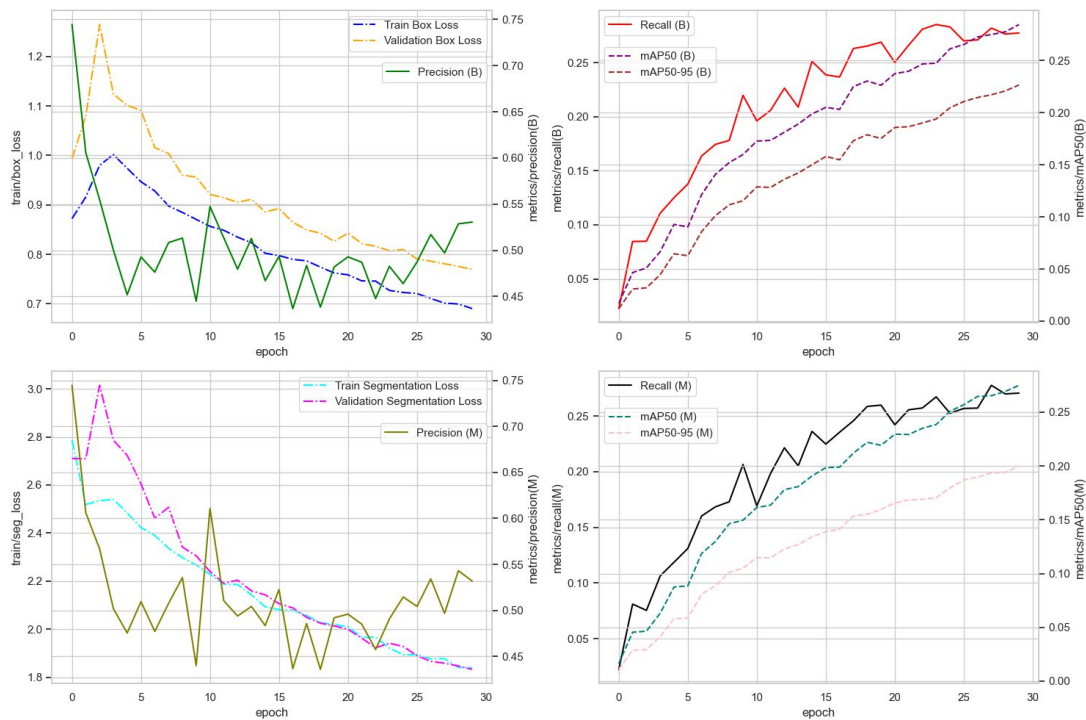
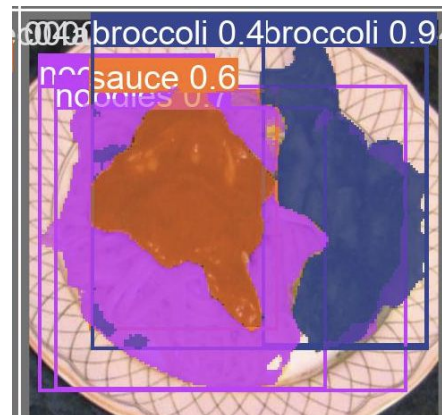


Figura 24 – Resultados Compilados YOLOv8n - Autoria Própria



(a) Exemplo Segmentação - Rótulo Real



(b) Exemplo Segmentação - Predição

Figura 25 – Resultado de Segmentação do Modelo YOLOv8n — Autoria Própria

5.1.2.2 Modelo YOLOv8s

A Figura 26 mostra os resultados do treinamento do modelo YOLOv8s. A análise é bem similar ao modelo anterior (YOLOv8n), tanto na forma quanto nos resultados, portanto, as similaridades serão apenas apontadas e as divergências serão exploradas mais a fundo. Esse modelo também teve um treinamento de 30 épocas, levando mais de 72 horas para ser concluído. A mesma configuração de *hardware* do modelo anterior foi utilizada para esse modelo.

Como no modelo anterior, a função perda de ambas as tarefas nos gráficos da esquerda linha tracejada, bem como o mAP50 e mAP50-95 (gráficos da direita linha tracejada) mostram que as 30 épocas não foram suficientes para se atingir a total estabilidade do modelo, indicando que um número maior de épocas poderia trazer ganhos significativos aos resultados. Para o valor de precisão nas duas tarefas (gráficos da esquerda linha contínua), novamente similar ao modelo anterior, os resultados apresentaram uma oscilação, finalizando em uma precisão de aproximadamente 40% - resultado inferior ao modelo YOLOv8n, e, conseqüentemente, mais baixo que o esperado para esse modelo.

O índice mAP50-95, apesar de apresentar valores mais altos para o modelo YOLOv8s (28%: detecção; 26%: segmentação) em comparação com o modelo YOLOv8n durante a etapa de treinamento, ainda mostraram um desempenho abaixo do esperado (acima de 45% para detecção e 37% para segmentação, segundo o desenvolvedor do modelo YOLOv8 (ULTRALYTICS, 2023)).

A Figura 27 mostra um exemplo de validação da tarefa de SI. Nesse exemplo do modelo YOLOv8s, os rótulos reais (figura esquerda) e os rótulos preditos (figura direita) estão exatamente com os mesmos valores, e o rótulo predito não mostra rótulos adicionais - o que demonstra que o mAP50 e o mAP50-95 estão de fato com um desempenho melhor.

5.2 Comparação entre diferentes Modelos e Condições de Luminosidade

5.2.1 Resultados Gerais

Os modelos analisados compreendem cinco variantes: Detectron2 com o banco de dados de validação original, YOLOv8n com o banco de dados de validação original, YOLOv8s com o banco de dados de validação original e YOLOv8s com ajustes de contraste e brilho, representados como *dark* (escuro) e *light* (claro). Para todos os modelos, os parâmetros avaliados foram a precisão (P), a revocação (R) e a média do valor médio de precisão (mAP50 e mAP50-95) para DO (índice B) e SI (índice M). A análise abaixo na Tabela 4 contempla os bancos de dados de validação, executados e testados posteriormente do treinamento dos modelos.

O Detectron2 original apresentou resultados mais modestos em todas as métricas, com exceção da revocação para SI (R(M)), onde se obteve um desempenho levemente

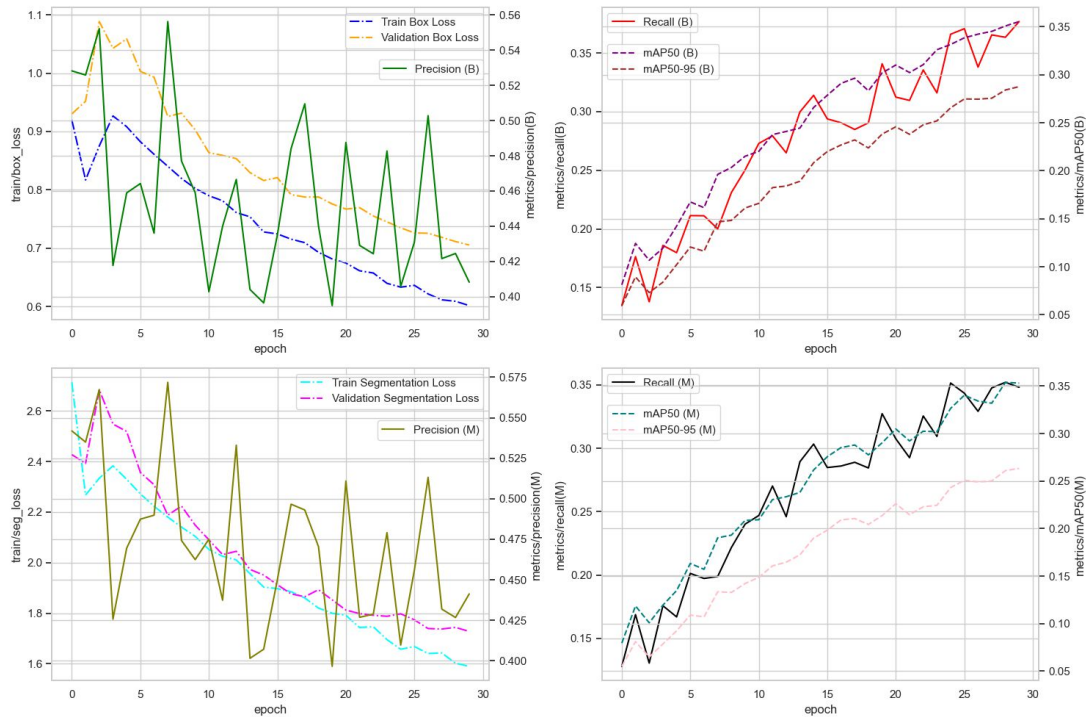
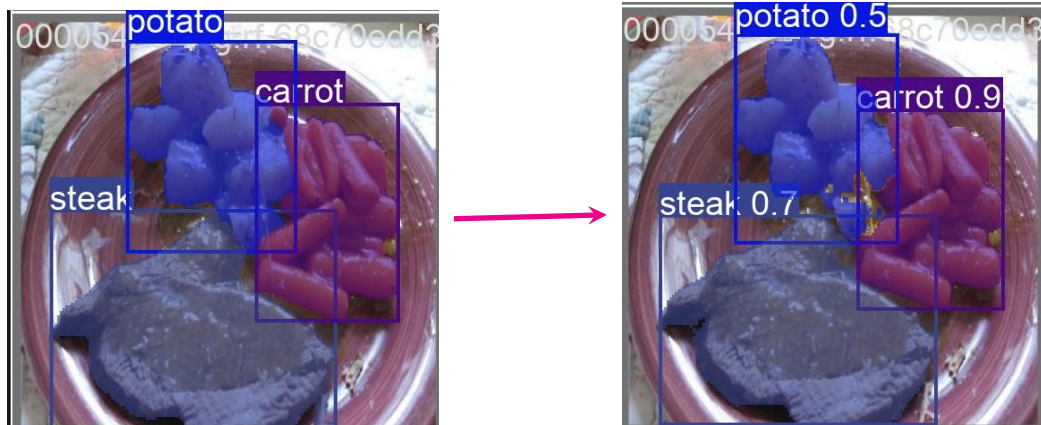


Figura 26 – Resultados Compilados YOLOv8s - Autoria Própria



(a) Exemplo Segmentação - Rótulo Real

(b) Exemplo Segmentação - Predição

Figura 27 – Resultado de Segmentação do Modelo YOLOv8s — Autoria Própria

superior ao de seu valor de precisão correspondente. Nota-se que a precisão e revocação para DO e SI estão bastante próximas, indicando um equilíbrio entre verdadeiros positivos e falsos positivos nas tarefas. Entretanto, sua média de precisão, tanto na DO quanto na SI, foi significativamente mais baixa que a dos outros modelos.

Os modelos YOLOv8n e YOLOv8s com as imagens originais apresentaram uma maior diferença entre a precisão e a revocação. No modelo YOLOv8n, a precisão excedeu significativamente a revocação tanto para DO quanto para SI, o que sugere que o modelo foi mais conservador, priorizando a correção dos verdadeiros positivos em detrimento da detecção de todos os possíveis positivos. Em contrapartida, o YOLOv8s apresentou uma

maior similaridade entre os valores de precisão e revocação, com a precisão ligeiramente superior à revocação, o que demonstra um balanço mais homogêneo entre os verdadeiros positivos e os falsos positivos.

Quando as versões escuro e claro do YOLOv8s são consideradas, nota-se uma diminuição geral nas métricas de desempenho. No cenário escuro, a precisão e revocação para ambas as tarefas, DO e SI, diminuíram consideravelmente em relação ao modelo com o banco de dados de validação original. Isso sugere que o modelo teve dificuldades para se adaptar a imagens com contrastes e brilho reduzidos. Na versão claro, a queda no desempenho foi menos acentuada, com a precisão e a revocação para DO e SI mostrando uma redução mais modesta em relação ao modelo com o banco de dados de validação original. Isto pode indicar que o modelo teve uma melhor capacidade de adaptação a imagens com contrastes e brilho aumentados.

Em termos de mAP50 e mAP50-95 - ou seja, em termos de separação de instâncias sobrepostas -, os modelos YOLOv8s e YOLOv8n originais exibiram um desempenho superior ao Detectron2, tanto para a DO quanto para a SI. As variantes escuro e claro do YOLOv8s demonstraram um decréscimo em todas as métricas de mAP50 e mAP50-95 em comparação ao modelo com o banco de dados de validação original, com o modelo escuro sofrendo um impacto maior.

Uma informação que já foi levantada em capítulos anteriores, mas que vale ser revisitada: o modelo YOLOv8n utilizou 3,4 milhões de parâmetros para o treinamento, enquanto o modelo YOLOv8s utilizou 11,8 milhões de parâmetros - ou seja, o senso comum induz a se pensar que o modelo com mais parâmetros traria os melhores resultados, porém isso não foi visto na sua totalidade. A precisão de ambas as tarefas apresentou um resultado inferior no modelo 's', porém todos os outros indicadores apresentaram desempenho superior nesse modelo.

Em resumo, os modelos YOLOv8 demonstraram um desempenho superior ao Detectron2. Dentre os modelos do YOLOv8, o modelo 's' apresentou um desempenho melhor de modo geral. As variantes do YOLOv8s com alterações de contraste e brilho (escuro e claro) tiveram um desempenho inferior ao modelo com o banco de dados original, sugerindo uma sensibilidade do modelo a mudanças nas condições de iluminação. Esta análise destaca a importância de considerar as condições de iluminação ao treinar e validar modelos de VC, bem como a necessidade de estratégias robustas de incremento de dados para melhorar a generalização do modelo a diferentes cenários de iluminação.

Tabela 4 – Comparação entre os diferentes modelos e condições - Autoria Própria

Model	Class	Instances	P (B)	R (B)	mAP50 (B)	mAP50-95 (B)	P (M)	R (M)	mAP50 (M)	mAP50-95 (M)
Detectron2 original	all	5830	0,117	0,151	0,0273	0,086	0,118	0,152	0,0241	0,089
YOLOv8n original	all	5830	0,531	0,277	0,285	0,227	0,53	0,272	0,275	0,200
YOLOv8s original	all	5830	0,408	0,377	0,355	0,288	0,441	0,349	0,353	0,263
YOLOv8s dark	all	5830	0,400	0,188	0,171	0,132	0,395	0,182	0,162	0,114
YOLOv8s light	all	5830	0,349	0,337	0,294	0,231	0,39	0,31	0,295	0,206

5.2.2 Resultados pelo Número de Instâncias

Nesta subseção é apresentada a análise dos resultados quando se compara dois grupos do banco de dados de validação: alimentos com muitas instâncias e alimentos com poucas instâncias. A questão a ser respondida: classes com maior número de instâncias apresentam majoritariamente melhores resultados?

No grupo de alta quantidade de instâncias (Tabela 5), incluem-se as classes de brócolis, alface, batata e tomate. Os modelos YOLOv8n e YOLOv8s com as imagens do banco de dados original obtiveram melhores resultados em quase todas as métricas para essas classes. Para o brócolis, YOLOv8s com o banco de dados de validação original demonstrou superioridade, com a precisão mais alta tanto para a tarefa de DO quanto para a SI. O mesmo padrão foi observado para as métricas de revocação, mAP50 e mAP50-95. Ao avaliar a alface, YOLOv8s com o banco de dados de validação original apresentou novamente a maior precisão e revocação para ambas as tarefas. Em termos de mAP50 e mAP50-95, esse modelo também foi superior, embora com margens menores em comparação com as outras classes. A classe batata foi mais bem identificada pelo YOLOv8s com o banco de dados original, com as todas as métricas superiores em comparação aos outros modelos. Finalmente, para o tomate, embora YOLOv8n com o banco de dados original tenha tido a maior precisão na tarefa de DO, o YOLOv8s com o banco de dados original superou nos demais índices.

Ao comparar o desempenho dos modelos em ambientes de baixa iluminação (escuro) ou alta iluminação (claro) observou-se uma queda significativa nas métricas para todas as classes de alimentos com alta quantidade de instâncias. Isso sugere que esses modelos são sensíveis a alterações de luminosidade, o que pode ser um desafio em cenários reais.

No grupo de baixa quantidade de instâncias (Tabela 6), que inclui maçã, uva, camarão e soja, uma tendência semelhante foi observada. O modelo YOLOv8s com o banco de dados original, em geral, apresentou as melhores métricas para as classes maçã e uva. No entanto, a classe camarão foi mais bem identificada pelo modelo YOLOv8n com o banco de dados original. Para a classe soja, o YOLOv8s com o banco de dados original novamente superou os demais.

As variações de iluminação também afetaram negativamente o desempenho dos modelos para as classes com baixa quantidade de instâncias. Em particular, a queda no desempenho foi mais pronunciada na configuração escuro, possivelmente indicando uma maior sensibilidade a baixas condições de luz.

Ao comparar os resultados gerais entre classes com alta e baixa quantidade de instâncias, notou-se que os modelos YOLOv8n e YOLOv8s com o banco de dados original apresentaram um desempenho relativamente bom em todas as classes, independentemente do número de instâncias. No entanto, uma quantidade maior de instâncias parece ter

contribuído para um melhor desempenho geral, porém, apesar de melhor, pouco significativo tendo em vista que algumas das classes tinham quase dez vezes menos instâncias.

Uma possível explicação do porquê as classes com instâncias de baixo valor apresentaram um desempenho considerável pode ser relacionado com as formas, cores e texturas de alguns alimentos, tornando a detecção do modelo de VC possível ainda que com poucas instâncias de treino. No entanto, isso pode não se traduzir da mesma forma em condições de luz variáveis, sugerindo a necessidade de aprimorar a robustez desses modelos a tais variações.

Finalmente, é importante mencionar que o modelo Detectron2 original apresentou resultados apenas para as métricas de precisão e revocação, limitando assim a comparação completa com os outros modelos. Além disso, para a soja e o camarão, o Detectron2 não apresentou nenhum resultado. Para os índices disponíveis, este modelo teve desempenho inferior em relação aos modelos YOLOv8n e YOLOv8s, e as classes com maior número de instâncias tiveram desempenho superior às classes com menor número de instâncias.

Tabela 5 – Comparação entre classes com alto número de instâncias - Autoria Própria

Model	Class	Instances	P (B)	R (B)	mAP50 (B)	mAP50-95 (B)	P (M)	R (M)	mAP50 (M)	mAP50-95 (M)
Detectron2 original	broccoli	239	0,587	—	—	—	0,586	—	—	—
YOLOv8n original	broccoli	239	0,73	0,866	0,858	0,721	0,711	0,841	0,845	0,676
YOLOv8s original	broccoli	239	0,739	0,895	0,882	0,764	0,758	0,887	0,882	0,71
YOLOv8s dark	broccoli	239	0,702	0,395	0,537	0,416	0,684	0,385	0,506	0,350
YOLOv8s light	broccoli	239	0,664	0,833	0,812	0,649	0,677	0,782	0,797	0,565
Detectron2 original	lettuce	139	0,078	—	—	—	0,064	—	—	—
YOLOv8n original	lettuce	139	0,419	0,446	0,386	0,208	0,388	0,41	0,34	0,156
YOLOv8s original	lettuce	139	0,417	0,518	0,433	0,28	0,463	0,49	0,415	0,203
YOLOv8s dark	lettuce	139	0,228	0,381	0,198	0,109	0,205	0,345	0,158	0,075
YOLOv8s light	lettuce	139	0,385	0,439	0,382	0,224	0,42	0,375	0,324	0,156
Detectron2 original	potato	255	0,191	—	—	—	0,20	—	—	—
YOLOv8n original	potato	255	0,513	0,686	0,597	0,479	0,504	0,667	0,572	0,446
YOLOv8s original	potato	255	0,551	0,749	0,657	0,549	0,574	0,722	0,632	0,522
YOLOv8s dark	potato	255	0,296	0,502	0,350	0,277	0,281	0,482	0,325	0,248
YOLOv8s light	potato	255	0,488	0,663	0,584	0,473	0,505	0,604	0,558	0,445
Detectron2 original	tomato	280	0,197	—	—	—	0,227	—	—	—
YOLOv8n original	tomato	280	0,594	0,557	0,539	0,423	0,613	0,571	0,545	0,391
YOLOv8s original	tomato	280	0,532	0,629	0,603	0,483	0,586	0,607	0,596	0,439
YOLOv8s dark	tomato	280	0,350	0,543	0,459	0,371	0,366	0,568	0,463	0,331
YOLOv8s light	tomato	280	0,454	0,595	0,491	0,382	0,506	0,575	0,480	0,339

5.3 Integração dos Resultados e Aplicação

A seção anterior explorou as diferenças entre os modelos de VC, resultando no modelo YOLOv8s como o que melhor atingiu os indicadores de precisão juntamente com a separação de instâncias. Dessa forma, esse modelo foi utilizado nas sessões posteriores do fluxo de execução. Conforme explicado no capítulo 4, a raspagem utilizou um dicionário de equivalências para a busca dos rótulos preditos do YOLOv8s. Dessa forma, os resultados da raspagem sempre são equivalentes ao melhor resultado possível, não sendo possível avaliar o desempenho dessa tarefa.

Para a junção e a incorporação das ferramentas, o código final - aqui chamado de aplicação - contempla as seguintes etapas:

Tabela 6 – Comparação entre classes com baixo número de instâncias - Autoria Própria

Model	Class	Instances	P (B)	R (B)	mAP50 (B)	mAP50-95 (B)	P (M)	R (M)	mAP50 (M)	mAP50-95 (M)
Detectron2 original	apple	39	0,077	—	—	—	0,073	—	—	—
YOLOv8n original	apple	39	0,401	0,077	0,139	0,105	0,421	0,077	0,141	0,0857
YOLOv8s original	apple	39	0,487	0,292	0,279	0,165	0,535	0,256	0,281	0,158
YOLOv8s dark	apple	39	0,048	0,026	0,053	0,030	0,048	0,026	0,044	0,026
YOLOv8s light	apple	39	0,249	0,256	0,170	0,117	0,279	0,231	0,157	0,104
Detectron2 original	grape	27	0,058	—	—	—	0,065	—	—	—
YOLOv8n original	grape	27	0,477	0,575	0,526	0,427	0,494	0,593	0,553	0,435
YOLOv8s original	grape	27	0,469	0,623	0,578	0,502	0,514	0,63	0,599	0,505
YOLOv8s dark	grape	27	0,179	0,333	0,199	0,160	0,158	0,296	0,189	0,138
YOLOv8s light	grape	27	0,552	0,519	0,526	0,464	0,613	0,481	0,504	0,413
Detectron2 original	shrimp	31	0	—	—	—	0	—	—	—
YOLOv8n original	shrimp	31	0,377	0,097	0,131	0,0944	0,383	0,097	0,144	0,0873
YOLOv8s original	shrimp	31	0,366	0,226	0,26	0,213	0,447	0,235	0,265	0,145
YOLOv8s dark	shrimp	31	0,381	0,120	0,101	0,083	0,383	0,121	0,098	0,062
YOLOv8s light	shrimp	31	0,411	0,097	0,183	0,133	0,588	0,093	0,189	0,111
Detectron2 original	soy	12	0	—	—	—	0	—	—	—
YOLOv8n original	soy	12	0,574	0,5	0,572	0,493	0,577	0,5	0,624	0,424
YOLOv8s original	soy	12	0,629	0,583	0,641	0,576	0,653	0,583	0,673	0,564
YOLOv8s dark	soy	12	0,309	0,083	0,174	0,161	0,307	0,083	0,174	0,138
YOLOv8s light	soy	12	0,363	0,333	0,362	0,344	0,381	0,257	0,372	0,33

1. Utilização do modelo YOLOv8s treinado com os dados dos pratos para a predição de uma nova imagem;
2. Conversão dos rótulos obtidos pelo dicionário de correspondências;
3. Utilização do novo rótulo para uma busca no *website* FoodData Central utilizando a biblioteca Selenium;
4. Formatação dos resultados da raspagem;
5. Incorporação das caixas de detecção e do texto de raspagem formatado na imagem final.

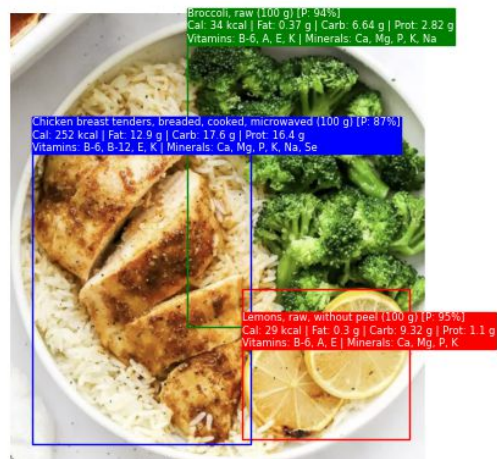
Abaixo pode-se observar 5 exemplos em que a aplicação foi feita do início ao fim - as Figuras de 28 a 32 ilustram esse processo. Todas as imagens utilizadas como entrada foram retiradas do *website*: <<https://www.apinchofhealthy.com/>>.

A avaliação dos resultados pode ser observada na Tabela 7. As colunas ‘C?’ mostram onde os resultados foram corretos (colocando-se um ‘X’) e onde os resultados foram incorretos (sem nenhuma marcação). Os exemplos contemplam desde resultados ótimos - como os exemplos 1 e 3 - assim como resultados muito ruins - como os exemplos 2 e 5. Alguns dos problemas observados:

- Identificação Incorreta de Rótulo: O exemplo 2 mostra uma imagem de Linguixa sendo identificada como Peito de Frango. Esse fato pode ser explicado pela incorreta identificação de rótulos no banco de dados original para a imagem da Linguixa, do Frango, ou de ambos. Outra possível explicação para erros desse tipo seria a semelhança de cores, texturas e formatos entre o rótulo real e o predito. A correção para esse problema seria uma revisão no banco de dados, ajustando os rótulos e



(a) Imagem de Entrada

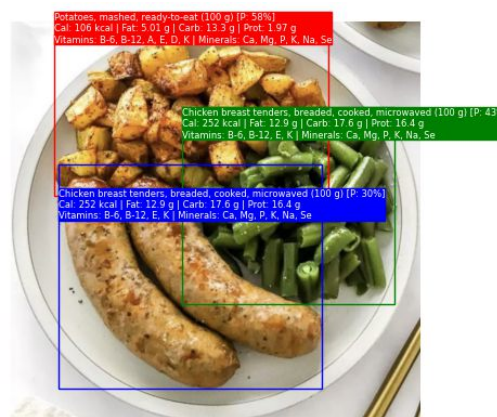


(b) Imagem de Saída

Figura 28 – Aplicação das Ferramentas | Exemplo 1 — Autoria Própria



(a) Imagem de Entrada



(b) Imagem de Saída

Figura 29 – Aplicação das Ferramentas | Exemplo 2 — Autoria Própria

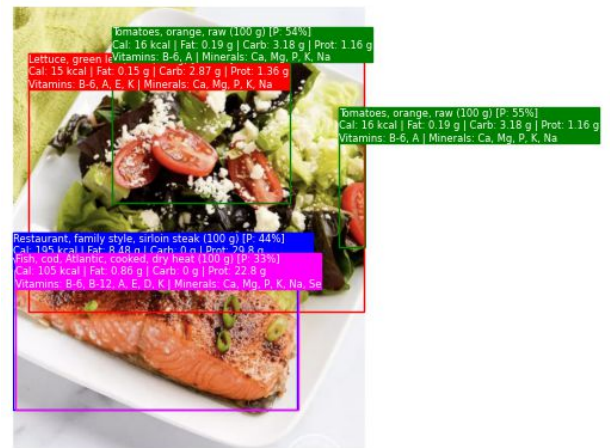
máscaras conforme a imagem real, juntamente com a adição de novas imagens com uma maior variação dos alimentos.

- Sobreposição de Rótulos: Esse problema pode ser observado nos exemplos 1, 3 e 4. Nos exemplos 1 e 4, percebe-se que um dos alimentos teve sua identificação dificultada graças à sobreposição de alimentos ao redor - Arroz, no caso do exemplo 1, e Batata, no caso do exemplo 4. Isso demonstra que o valor de mAP50 e mAP50-95 deve ser aumentado por otimizações no modelo YOLOv8.

De modo geral, a Aplicação prática deste estudo foi demonstrada exibindo alguns exemplos de detecção de alimentos e extração de informações nutricionais. A limitação deste estudo ocorre por duas vertentes: falta de poder computacional para a realização de novas rodadas de treinamentos - seja de modelos com mais parâmetros, seja dos modelos



(a) Imagem de Entrada



(b) Imagem de Saída

Figura 30 – Aplicação das Ferramentas | Exemplo 3 — Autoria Própria



(a) Imagem de Entrada

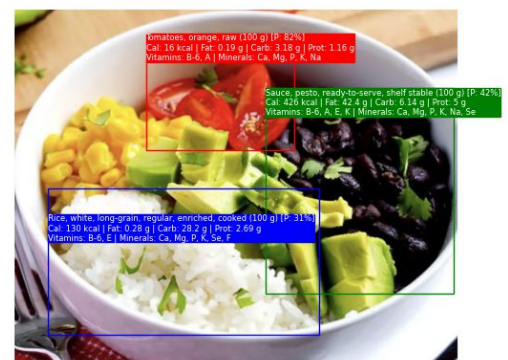


(b) Imagem de Saída

Figura 31 – Aplicação das Ferramentas | Exemplo 4 — Autoria Própria



(a) Imagem de Entrada



(b) Imagem de Saída

Figura 32 – Aplicação das Ferramentas | Exemplo 5 — Autoria Própria

utilizados, com variações na configuração -; e no banco de dados: correção das máscaras e rótulos incorretos, e incremento de imagens, apresentando uma maior variabilidade para

Tabela 7 – Resultados dos Exemplos de Aplicação - Autoria Própria

	Alimento 1	C?	Alimento 2	C?	Alimento 3	C?	Alimento 4	C?	Total	Comentário
Exemplo 1	Brócolis	X	Frango	X	Limão	X	Arroz		75%	Arroz ficou sobreposto.
Exemplo 2	Linguiça		Vagem		Batata	X			33%	
Exemplo 3	Peixe	X	Alface	X	Tomate	X			100%	Também identificou 'carne'.
Exemplo 4	Carne	X	Batata		Cenoura	X			67%	Batata ficou sobreposta.
Exemplo 5	Arroz	X	Milho		Tomate	X	Feijão		50%	

os alimentos e tipos de preparo dos mesmos.

6 CONCLUSÕES

No decorrer deste estudo foi desenvolvida uma solução de VC que visa fornecer a conscientização nutricional através da detecção de alimentos e da incorporação de rótulos nutricionais. O estudo abordou a DO e a SI de alimentos utilizando os seguintes modelos de VC: Detectron2, YOLOv8n e YOLOv8s.

A partir da análise dos resultados obtidos, observou-se que o modelo YOLOv8s apresentou o melhor desempenho na DO e SI, com resultados mais altos de revocação e métricas mAP50 e mAP50-95 - apenas o resultado de precisão foi maior no modelo YOLOv8n. Isso sugere que o YOLOv8s foi a escolha adequada para a aplicação dos rótulos nutricionais. Esse modelo também foi avaliado utilizando-se duas variações no banco de dados de validação, alterando-se as condições de luz e contraste para a criação do banco de validação claro e escuro. Os resultados foram quase inalterados para as imagens claras, porém observou-se um prejuízo na identificação de alimentos das imagens escuras de quase 50%, demonstrando dificuldades para a execução do fluxo em ambientes pouco iluminados.

A incorporação da raspagem permitiu enriquecer os resultados, obtendo informações nutricionais adicionais do *website* FoodData Central. Essa integração contribuiu para aumentar a utilidade das informações sobre os alimentos detectados, trazendo informações de calorias, quantidade de proteínas, carboidratos e gordura, além das vitaminas e minerais.

Portanto, a solução desenvolvida neste estudo oferece uma contribuição significativa para a conscientização nutricional. Através do uso de modelos de VC foi possível detectar e segmentar alimentos, proporcionando uma ferramenta útil para auxiliar os usuários na escolha de alimentos mais saudáveis. A integração com recursos de raspagem ampliou as informações disponíveis, fornecendo dados nutricionais.

Espera-se que a solução desenvolvida possa contribuir para a promoção de hábitos alimentares mais saudáveis e, conseqüentemente, para o combate à crescente tendência de obesidade e diabetes no Brasil e no mundo. Para trabalhos futuros, o que foi desenvolvido servirá de alicerce para a criação de uma aplicação objetivando o cálculo de Insulina para pacientes Insulinodependentes.

6.1 Próximos passos

Para a otimização deste estudo, aumentando os principais indicadores de desempenho aqui explorados (Precisão, Revocação, mAP), sugere-se o incremento de pares imagens-máscaras contendo pratos de comida, bem como a utilização de rótulos mais específicos para variações de preparo para um mesmo alimento, principalmente de imagens com pouca iluminação, proporcionando maior robustez para a detecção nessa condição. Além

desses pontos, é altamente recomendado utilizar um *hardware* com um processamento gráfico robusto, a fim de se explorar modelos do YOLOv8 maiores (como o modelo 'm' e 'l') e também a utilização de mais épocas (> 30).

Conforme descrito na Capítulo 1, a ideia inicial desse TCC seria a incorporação de dados de Índice Glicêmico, de modo a tornar-se uma ferramenta que auxiliasse pacientes insulino-dependentes a terem uma estimativa da quantidade de Insulina utilizada por refeição. Para a execução dessa tarefa, sugere-se: estimativa de volume de cada alimento - para o cálculo da quantidade de cada um dos elementos do prato de forma mais precisa -, seguido do cálculo do aumento do Índice Glicêmico causado pelo prato de comida do paciente. Apesar de parecer simples, essa etapa teria que considerar a sensibilidade a Insulina (um parâmetro que varia entre indivíduos), e o histórico do Índice Glicêmico das últimas horas.

REFERÊNCIAS

- AGRICULTURE, F. C. U. D. of. **FoodData Central**. 2023. Available at: <<https://fdc.nal.usda.gov/>>.
- ASSOCIATION, A. D. Nutrition principles and recommendations in diabetes. **Diabetes care**, Am Diabetes Assoc, v. 27, n. suppl_1, p. s36–s36, 2004.
- CDC, C. f. D. C. **Defining Adult Overweight & Obesity**. 2022. Available at: <<https://www.cdc.gov/obesity/basics/adult-defining.html>>.
- CHAKI, J. *et al.* Machine learning and artificial intelligence based diabetes mellitus detection and self-management: A systematic review. **Journal of King Saud University-Computer and Information Sciences**, Elsevier, v. 34, n. 6, p. 3204–3225, 2022.
- CLARK, J. A. **Pillow**. 2023. Available at: <<https://github.com/python-pillow/Pillow>>.
- COWBURN, G.; STOCKLEY, L. Consumer understanding and use of nutrition labelling: a systematic review. **Public health nutrition**, Cambridge University Press, v. 8, n. 1, p. 21–28, 2005.
- EVERINGHAM, M. *et al.* The pascal visual object classes (VOC) challenge. **International Journal of Computer Vision**, Springer Science and Business Media LLC, v. 88, n. 2, p. 303–338, set. 2009. Available at: <<https://doi.org/10.1007/s11263-009-0275-4>>.
- FINGER, M. Inteligência artificial e os rumos do processamento do português brasileiro. **Estudos Avançados**, FapUNIFESP (SciELO), v. 35, n. 101, p. 51–72, abr. 2021. Available at: <<https://doi.org/10.1590/s0103-4014.2021.35101.005>>.
- HAIR, L. Open Source Dataset, **food v18 Dataset**. Roboflow, 2023. <<https://universe.roboflow.com/lawrence-hair-wpavf/food-v18>>. Visited on 2023-04-09. Available at: <<https://universe.roboflow.com/lawrence-hair-wpavf/food-v18>>.
- JOCHER, G.; CHAURASIA, A.; QIU, J. **YOLO by Ultralytics**. 2023. Available at: <<https://github.com/ultralytics/ultralytics>>.
- KUMANYIKA, S. *et al.* Obesity prevention: the case for action. **International journal of obesity**, Nature Publishing Group, v. 26, n. 3, p. 425–436, 2002.
- MEYERS, A. *et al.* Im2calories: Towards an automated mobile vision food diary. *In: Proceedings of the IEEE international conference on computer vision*. [S.l.: s.n.], 2015. p. 1233–1241.
- MUTHUKADAN, B. **Selenium**. 2023. Available at: <<https://github.com/SeleniumHQ/selenium>>.
- POPLY, P.; JOTHI, J. A. A. Refined image segmentation for calorie estimation of multiple-dish food items. *In: IEEE. 2021 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS)*. [S.l.: s.n.], 2021. p. 682–687.

SZELISKI, R. **Computer Vision: Algorithms and Applications**. [*S.l.: s.n.*]: Springer, 1986.

ULTRALYTICS. **Metrics for YOLOv8**. 2023. Available at: <<https://docs.ultralytics.com/reference/yolo/utils/metrics/>>.

WHO, W. H. O. **Noncommunicable Diseases Profile – Brazil Diabetes**. 2015. Available at: <<https://ncdportal.org/CountryProfile/GHE110/BRA>>.

WHO, W. H. O. **Mean BMI 2017**. 2017. Available at: <[https://www.who.int/data/gho/data/indicators/indicator-details/GHO/mean-bmi-\(kg-m\)-\(age-standardized-estimate\)](https://www.who.int/data/gho/data/indicators/indicator-details/GHO/mean-bmi-(kg-m)-(age-standardized-estimate))>.

WU, Y. *et al.* **Detectron2**. 2019. <<https://github.com/facebookresearch/detectron2>>.