

**UNIVERSIDADE DE SÃO PAULO**

Instituto de Ciências Matemáticas e de Computação

## Nutritional Awareness based on Image Object Detection and Web Scraping

**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**

## **Nutrional Awareness based on Image 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: Inteligência Artificial

Advisor: Prof. Dr. Fernando Pereira dos Santos

**Original version**

**São Carlos**

**2023**

I AUTHORIZE THE REPRODUCTION AND DISSEMINATION OF TOTAL OR PARTIAL COPIES OF THIS DOCUMENT, BY CONVENCIONAL OR ELECTRONIC MEDIA FOR STUDY OR RESEARCH PURPOSE, SINCE IT IS REFERENCED.

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>Nutritional Awareness based on Image Object Detection and Web Scraping / Marcelo Felipe Alves Souza ; advisor Prof. Dr. Fernando Pereira dos Santos. – São Carlos, 2023.</p> <p>54 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, advisor. II. Título.</p>
-------	---

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

## **ACKNOWLEDGEMENTS**

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.



## ABSTRACT

SOUZA, M.F.A. **Nutritional Awareness based on Image Object Detection and Web Scraping**. 2023. 54p. 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 dataset 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 instance 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 - like amount of Protein, which Vitamins and Minerals are present etc. 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.





## LIST OF FIGURES

Figure 1 – BMI by Year Worldwide — Own Authorship using WHO Data . . . . .	17
Figure 2 – BMI by Year in Brazil — Own Authorship using WHO Data . . . . .	18
Figure 3 – Diabetes Cases by Year in Brazil — Own Authorship using WHO Data	18
Figure 4 – Instance Segmentation by Comaniciu and Meer 2002, in (SZELISKI, 1986) . . . . .	21
Figure 5 – Place Detection by Philbin, Chum, Isard et al. 2007, in (SZELISKI, 1986)	22
Figure 6 – Potential Uses of YOLOv8 — Image from Roboflow Website . . . . .	22
Figure 7 – Architecture of YOLOv8 — Image by RangeKing (GitHub User) . . . .	23
Figure 8 – Segmentation Models of YOLOv8 — Image by Ultralytics . . . . .	23
Figure 9 – Architecture of Detectron2 — Image by Hiroto Honda (Medium User) .	24
Figure 10 – Potential Uses of Detectron2 — Image from Meta/Detectron2 Website	24
Figure 11 – IoU Calculation Formula - Image by Eric Hofesmann (Medium user) . .	25
Figure 12 – Example of Searching Information in HTML Code — Own Authorship using the FoodData Central website . . . . .	27
Figure 13 – Dataset Sample Picture (HAIR, 2023) . . . . .	28
Figure 14 – Histogram of Dataset — Own Authorship . . . . .	28
Figure 15 – Low Brightness and Contrast Sample (HAIR, 2023) . . . . .	28
Figure 16 – High Brightness and Contrast Sample (HAIR, 2023) . . . . .	29
Figure 17 – General Scheme of the Work — Im2Calories Article . . . . .	31
Figure 18 – Application for the End User — Im2Calories Article . . . . .	32
Figure 19 – First part of the execution flow — Own Authorship . . . . .	33
Figure 20 – Second part of the execution flow — Own Authorship using image from the website: <a href="https://www.apinchofhealthy.com/baked-chicken-breast/">https://www.apinchofhealthy.com/baked-chicken-breast/</a> .	33
Figure 21 – Example of Nutritional Label of White Rice - FoodData Central Website	36
Figure 22 – Example of Search for Rice - FoodData Central Website . . . . .	37
Figure 23 – Input and Output of Tool Integration — Own Authorship using image from the website: <a href="https://www.apinchofhealthy.com/baked-chicken-breast/">https://www.apinchofhealthy.com/baked-chicken-breast/</a> . . . . .	38
Figure 24 – Compiled Results YOLOv8n - Own Authorship . . . . .	41
Figure 25 – Segmentation Result of YOLOv8n Model — Own Authorship . . . . .	41
Figure 26 – Compiled YOLOv8s Results - Own Authorship . . . . .	42
Figure 27 – Segmentation Result of the YOLOv8s Model — Own Authorship . . .	43
Figure 28 – Full Application of the tools   Example 1 - Own Authorship . . . . .	47
Figure 29 – Full Application of the tools   Example 2 - Own Authorship . . . . .	48
Figure 30 – Full Application of the tools   Example 3 - Own Authorship . . . . .	48
Figure 31 – Full Application of the tools   Example 4 - Own Authorship . . . . .	48



## LIST OF TABLES

Table 1 – Food labels used in the Dataset of the study - Own Authorship . . . . .	30
Table 2 – Configuration of the Detectron2 model - Own Authorship . . . . .	35
Table 3 – Detectron2 Training Results . . . . .	40
Table 4 – Comparison between different models and conditions - Own Authorship	44
Table 5 – Comparison between high instance classes - Own Authorship . . . . .	46
Table 6 – Comparison between low instance classes - Own Authorship . . . . .	46
Table 7 – Application Results Analysis - Own Authorship . . . . .	49



## **LIST OF ABBREVIATIONS AND ACRONYMS**

FYP	Final Year Project
AI	Artificial Intelligence
WHO	World Health Organization
MBI	Mass Body Index
CNN	Convolutional Neural Network
ML	Machine Learning
CV	Computer Vision
IS	Instance Segmentation
OD	Object Detection
NLP	Natural Language Processing
IoU	Intersection over Union
mAP	mean Average Precision



## CONTENTS

<b>1</b>	<b>INTRODUCTION . . . . .</b>	<b>17</b>
<b>1.1</b>	<b>Hypotheses and Objectives . . . . .</b>	<b>19</b>
<b>1.2</b>	<b>Organization of the Text . . . . .</b>	<b>19</b>
<b>2</b>	<b>THEORETICAL FOUNDATION . . . . .</b>	<b>21</b>
<b>2.1</b>	<b>Computer Vision . . . . .</b>	<b>21</b>
2.1.1	Instance Segmentation . . . . .	21
2.1.2	Object Detection . . . . .	21
2.1.3	YOLOv8 Model . . . . .	22
2.1.4	Detectron2 Model . . . . .	24
2.1.5	Evaluation Metrics and Loss Functions . . . . .	24
2.1.5.1	Evaluation Metrics . . . . .	25
2.1.5.2	Loss Functions . . . . .	26
<b>2.2</b>	<b>Natural Language Processing . . . . .</b>	<b>26</b>
2.2.1	Scraping . . . . .	26
2.2.2	Selenium . . . . .	26
<b>2.3</b>	<b>Dataset . . . . .</b>	<b>27</b>
<b>3</b>	<b>RELATED ARTICLES . . . . .</b>	<b>31</b>
<b>4</b>	<b>METHODOLOGY . . . . .</b>	<b>33</b>
<b>4.1</b>	<b>Identification and Labeling of Foods . . . . .</b>	<b>34</b>
4.1.1	Neural Networks . . . . .	34
4.1.2	Model Evaluation . . . . .	34
4.1.3	Additional Validation with Brightness and Contrast Modifications in the Dataset . . . . .	35
<b>4.2</b>	<b>Nutritional Information Search . . . . .</b>	<b>35</b>
4.2.1	Scraping . . . . .	36
4.2.2	Integration of Tools and Expected Output . . . . .	37
<b>5</b>	<b>RESULTS ANALYSIS AND DISCUSSION . . . . .</b>	<b>39</b>
<b>5.1</b>	<b>Model Training . . . . .</b>	<b>39</b>
5.1.1	Detectron2 Training . . . . .	39
5.1.2	YOLOv8 Training . . . . .	39
5.1.2.1	YOLOv8n Model . . . . .	40
5.1.2.2	YOLOv8s Model . . . . .	41
<b>5.2</b>	<b>Comparison between Different Models and Lighting Conditions . . .</b>	<b>43</b>

5.2.1 General Results . . . . . 43

5.2.2 Results by Number of Instances . . . . . 44

**5.3 Integration of Results and Application . . . . . 46**

**6 CONCLUSIONS . . . . . 51**

**6.1 Next Steps . . . . . 51**

**References . . . . . 53**



## 1 INTRODUCTION

Obesity has been continuously increasing over the years, leading to consequences for both individual health and governments due to the high operational expenses associated with this issue (KUMANYIKA *et al.*, 2002). The Body Mass Index (BMI) is an indicator that assists in monitoring obesity. According to the CDC (Centers for Disease Control and Prevention), an individual is considered to have a normal weight if their BMI is between 18.5 and 25; pre-obese if it is between 25 and 30; and obese if it is over 30 (CDC, 2022). From Figure 1, it can be observed that the BMI across all continents is on the rise, according to data from the World Health Organization (WHO)<sup>1</sup>.

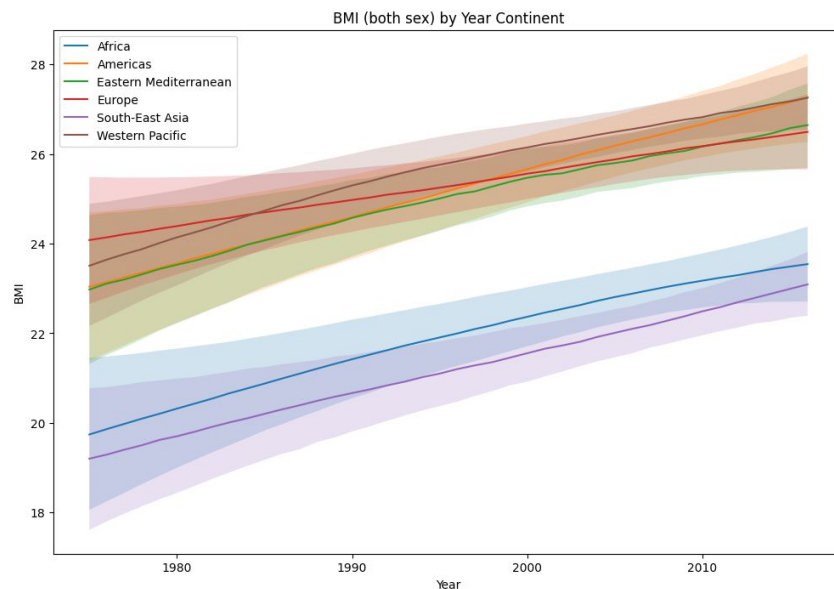


Figure 1 – BMI by Year Worldwide — Own Authorship using WHO Data

A similar analysis can be done for Brazil, as seen in Figure 2, reflecting the same increasing trend over the years, showing an average BMI in the overweight range (25 to 30) (CDC, 2022).

One of the conditions associated with obesity — and consequently with BMI — is Diabetes, especially Type II. The increasing trend of Diabetes in Brazil over the years can be observed in Figure 3, using data from the WHO<sup>2</sup>.

A tool that could assist individuals in achieving a healthier diet, and thereby reaching a BMI in the range of 18.5 to 25, is the effective use of nutritional labels. However, most consumers do not know how to use this tool correctly (COWBURN; STOCKLEY, 2005).

<sup>1</sup> <[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)

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

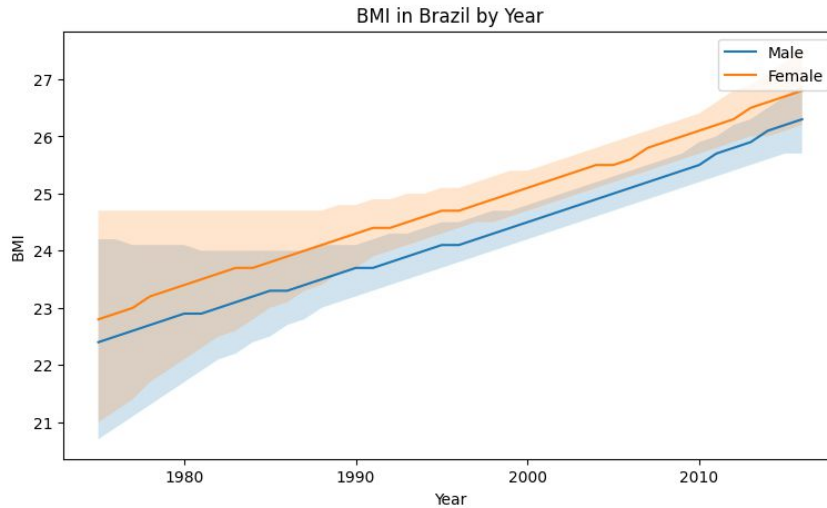


Figure 2 – BMI by Year in Brazil — Own Authorship using WHO Data

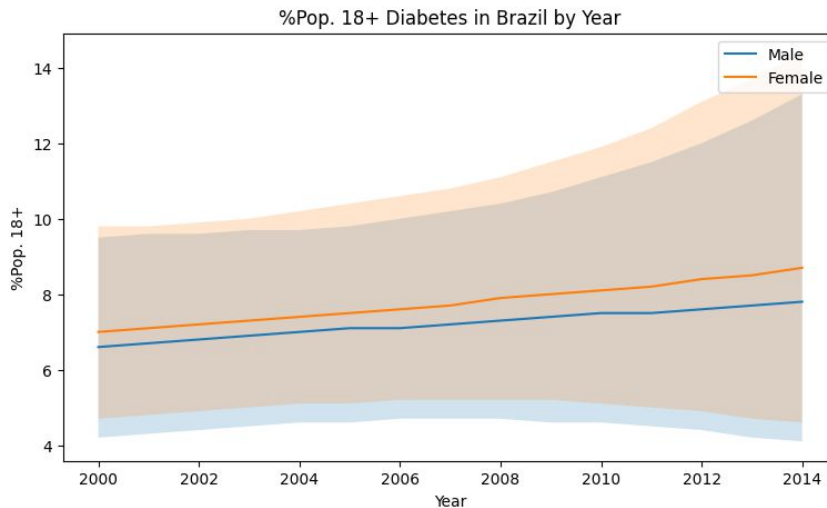


Figure 3 – Diabetes Cases by Year in Brazil — Own Authorship using WHO Data

Artificial Intelligence (AI) can be used for the identification and segregation of foods, as well as for the incorporation of their nutritional data, making the recognition process quick and intuitive, as mentioned in (MEYERS *et al.*, 2015). The use of Convolutional Neural Networks (CNN) can achieve high accuracy levels for this food identification and segregation task — around 90%, according to (POPLY; JOTHI, 2021). Additionally, AI can be used in conjunction with web information in a task known as Web Scraping, utilizing the Python library Selenium (MUTHUKADAN, 2023). Thus, as soon as a food item is recognized, this tool can be used to obtain detailed nutritional information, becoming a powerful combination of tools for a practical and efficient process of recognizing beneficial foods.

Other AI-related applications are tools for the prediction and management of Diabetes. According to (CHAKI *et al.*, 2022), Machine Learning (ML) algorithms can be trained for rapid detection of glaucoma caused by Diabetes, as well as for maintaining

optimal glycemic control (between 80 and 120 mg/mL of glucose in the blood).

## 1.1 Hypotheses and Objectives

As discussed in chapter 1, tools that assist individuals in correctly identifying nutritional labels would be extremely valuable for potentially reducing BMI and, consequently, obesity. In addition, a rapid evaluation of nutritional content would help Diabetes patients control their blood glucose levels, avoiding complications arising from an imbalance of glucose in the blood (ASSOCIATION, 2004).

The aim of this study is to train and evaluate two open-source Computer Vision (CV) algorithms — including the state-of-the-art Object Detection architecture YOLOv8 (You Only Look Once), (JOCHER; CHAURASIA; QIU, 2023) — to identify different types of food on a plate, as well as to associate these foods with nutritional components using Internet pages and a Natural Language Processing (NLP) technique called Scraping.

The algorithm will be trained using an existing image dataset, which forms one of the hypotheses of this study: the 103 types of food to be identified were created and identified by the author of the dataset (HAIR, 2023) — a fact that will be explained in subsequent chapters. Additionally, a significant limitation is that the Segmentation Masks (already present in the dataset) are not perfect — though adequate — containing imperfections that may result in a potentially worse outcome than expected. Another limitation is that the identification labels do not cover different food preparations (such as baked, grilled, fried, etc.). However, the time required to create a significantly large image dataset, with properly placed labels and masks, would be incompatible with the time proposed for this study. Therefore, the used dataset will be considered sufficient and convenient.

The ultimate goal of this study would be a tool for suggesting insulin dosage to Insulin-Dependent Diabetes patients based on an image of the food to be consumed — however, this task would require an implementation complexity that the proposed time for the completion of this work would not allow (future works). Therefore, a restriction imposed in this study is the limitation of the scope, encompassing only the task of identifying food on a plate, using the Instance Segmentation technique, along with the search for nutritional information on the web using the scraping technique.

## 1.2 Organization of the Text

This Final Year Project (FYP) is divided into 6 chapters, including this introduction, arranged as follows:

Chapter 1: Introduction regarding how nutrition can help control BMI, as well as obesity and Diabetes, introducing AI methods as potential tools for this purpose.

Chapter 2: The conceived system is addressed theoretically, presenting the data used for this task, the Computer Vision (CV) and Natural Language Process (NLP) methods, as well as the metrics used.

Chapter 3: Presents related article and study similar to the proposed study.

Chapter 4: Elaboration of the proposed study, covering the configuration aspects of the libraries and dataset used. The objective of this chapter is to explore the step-by-step tasks that were processed, coded, and executed.

Chapter 5: Presentation of results, comparison of the proposed CV models, and analysis of the metrics to identify the best model. This chapter also presents the application of the scraping technique to the chosen CV model, where there is an interaction between the detected objects and the nutritional data obtained from the U.S. government website Food Data Central (AGRICULTURE, 2023).

Chapter 6: Conclusions, final considerations, and next steps of the developed study.

## 2 THEORETICAL FOUNDATION

### 2.1 Computer Vision

Instance Segmentation (IS) and Object Detection (OD) are Computer Vision (CV) tasks that involve analyzing images to identify and locate objects. Although they share similarities, each has distinct goals and approaches. Among the CV techniques mentioned by (SZELISKI, 1986), IS and OD are closest to the objective of this study.

#### 2.1.1 Instance Segmentation

IS involves identifying and delineating individual objects in an image, classifying each object by type, as demonstrated in Figure 4. This can be useful for tasks like object tracking over time or identifying their spatial relationships. IS can be performed using deep learning models, such as Mask R-CNN, which combines OD with semantic segmentation to generate object masks. Creating masks is a time-consuming and resource-intensive task, making IS a limited but powerful and accurate technique.

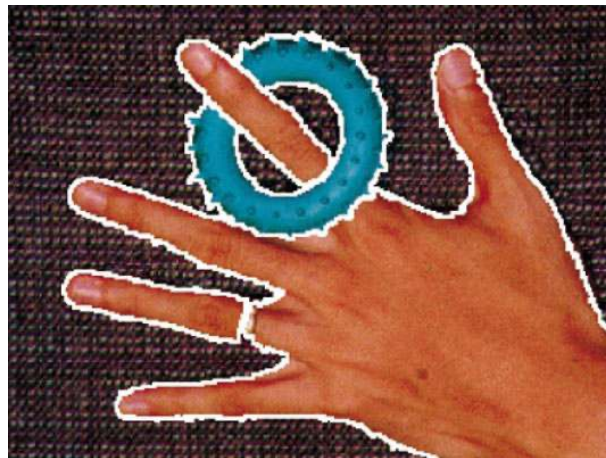


Figure 4 – Instance Segmentation by Comaniciu and Meer 2002, in (SZELISKI, 1986)

#### 2.1.2 Object Detection

OD involves identifying objects in an image and creating bounding boxes around them, classifying each object by type based on prior training - for example, in Figure 5. This can be useful for tasks such as surveillance, autonomous driving, or identifying people in images. OD can be performed using deep learning models, like YOLO (You Only Look Once), known for its speed and accuracy, and will be explored in the following sections of this study. Unlike IS, OD does not use masks, but annotations on the image. These annotations can be done in JSON (JavaScript Object Notation), XML (Extensible Markup

Language), or even in simple text files, indicating the coordinates of the object(s) in question for algorithm training.

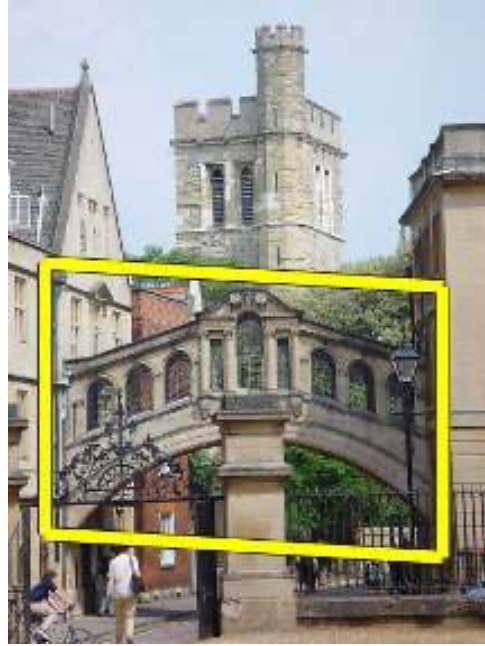


Figure 5 – Place Detection by Philbin, Chum, Isard et al. 2007, in (SZELISKI, 1986)

### 2.1.3 YOLOv8 Model

The YOLOv8 algorithm (JOCHER; CHAURASIA; QIU, 2023) is designed for use in CV, encompassing OD, IS, and Image Classification, as shown in Figure 6. It is considered state-of-the-art due to its pre-training image set, the number of parameters present, and the complexity of the architecture - represented in Figure 7, featuring convolutional layers and a C2f (Coarse-to-Fine) structure.



Figure 6 – Potential Uses of YOLOv8 — Image from Roboflow Website

Figure 8 shows different models available on the YOLOv8 platform for segmentation tasks. Accuracy between 36.7% to 53.4% for Average Precision in Box cases, and between 30.5% to 43.4% in Mask cases (reference results, obtained using the COCO2017 validation dataset, according to (JOCHER; CHAURASIA; QIU, 2023)) can be achieved - for this



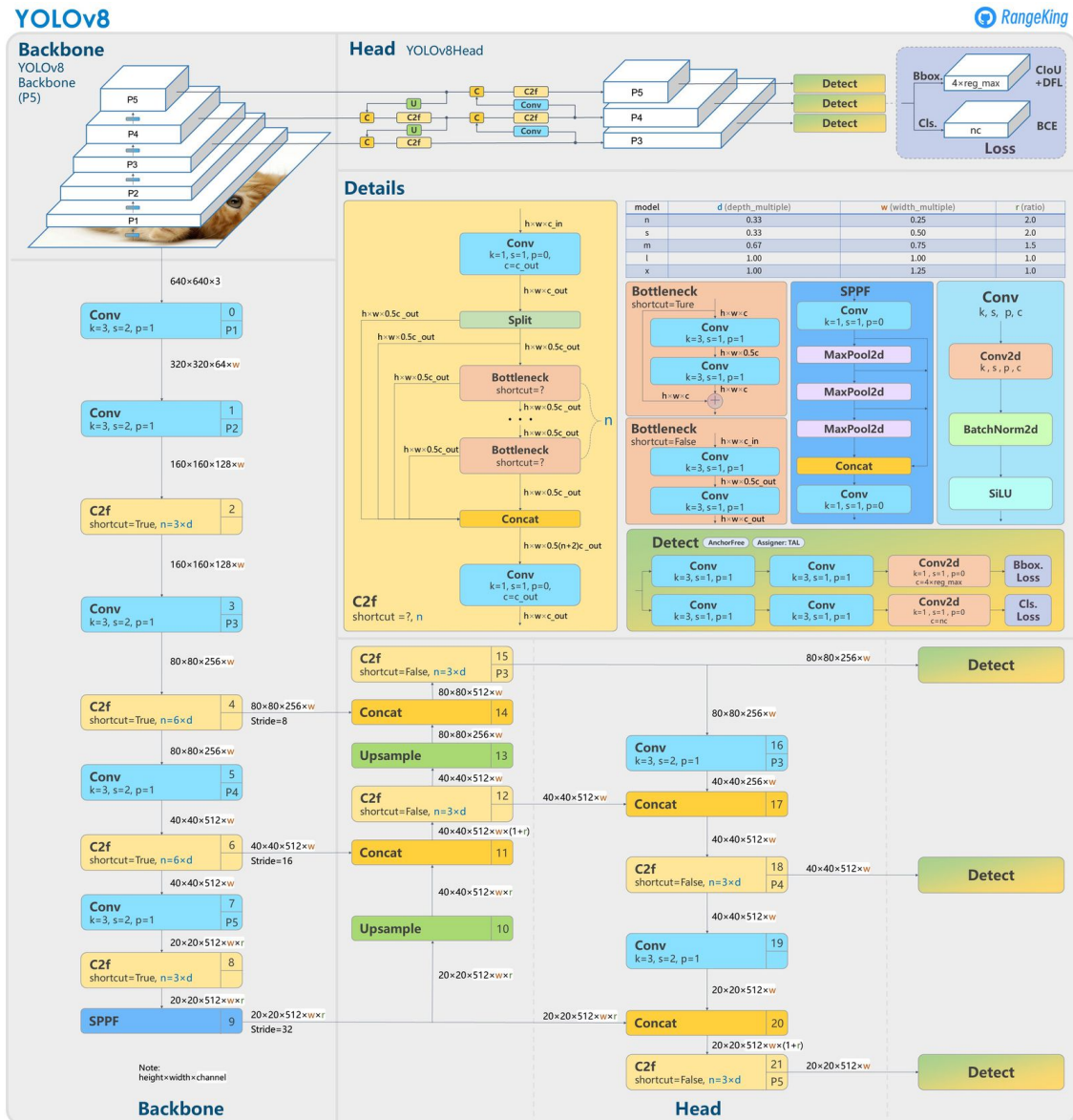


Figure 7 – Architecture of YOLOv8 — Image by RangeKing (GitHub User)

study, ‘n’ and ‘s’ models were used, with an expected accuracy of 36% and 30% for OD and IS of the smaller model, and 45% and 37% for the larger model.

Model	size (pixels)	mAP <sup>box</sup> <sub>50-95</sub>	mAP <sup>mask</sup> <sub>50-95</sub>	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

Figure 8 – Segmentation Models of YOLOv8 — Image by Ultralytics

### 2.1.4 Detectron2 Model

The CV algorithm Detectron2 (WU *et al.*, 2019), whose library is usually written in Python, is intended for CV tasks, like the YOLO architecture, but with a lesser capacity to achieve state-of-the-art results. Figure 9 and Figure 10 show that this model also offers various utilities, besides having a robust architecture of Recurrent Convolutional Neural Network with FPN (Feature Pyramid Network).

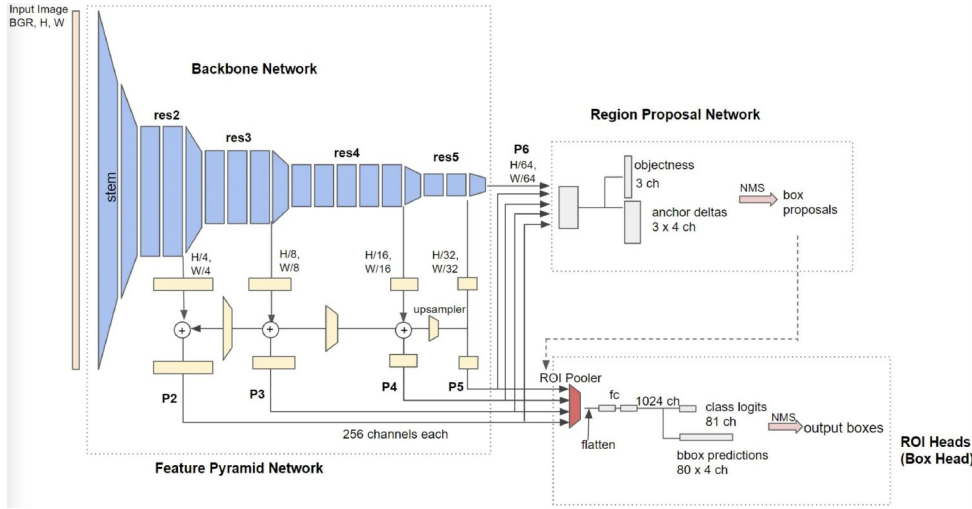


Figure 9 – Architecture of Detectron2 — Image by Hiroto Honda (Medium User)



Figure 10 – Potential Uses of Detectron2 — Image from Meta/Detectron2 Website

### 2.1.5 Evaluation Metrics and Loss Functions

Evaluation metrics and loss functions play an essential role in analyzing the performance of OD and IS models. Analyzing these metrics together provides a more comprehensive understanding of model performance, allowing identification of areas where the model can be improved or adjusted. Below are some of the main metrics that will be used in the Results section (5), according to (EVERINGHAM *et al.*, 2009) and (ULTRALYTICS, 2023).



### 2.1.5.1 Evaluation Metrics

- **Precision (P):** Precision is the ratio of true positive detections (correct detections) to the total number of positive predictions made by the model. Higher precision indicates that the model produces fewer false positives.
- **Recall (R):** Recall is the ratio of true positive detections to the total number of actual positive instances. Higher recall indicates that the model is better at detecting positive instances without missing them.
- **Intersection over Union (IoU):** Intersection over Union can be mathematically seen in Figure 11, representing the calculation dimension of this parameter. In the numerator, there is the overlap between the predicted image box and the original label or ground truth, and in the denominator, the union of these two parameters. Practically, IoU is a measure of location/accuracy between the predicted image box and the original label, varying from 0 to 1 — with 0 being no intersection between the predicted box and the true value, and 1 representing total intersection.


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


Figure 11 – IoU Calculation Formula - Image by Eric Hofesmann (Medium user)

- **Mean Average Precision - mAP:** mAP calculates the average of precision values at different recall levels across all classes. It is commonly used to assess Object Detection models, with higher values indicating better performance.
  1. **mAP50 or mAP95:** mAP50 represents the mean of the average precision (mAP) with an IoU threshold of 0.50 or 0.95 for the COCO test dataset. The 'B' index represents Box and relates the results to the Object Detection task, while the 'M' index represents Mask and relates the results to the Instance Segmentation task. Higher values for these metrics indicate better performance at IoU thresholds of 50
  2. **mAP50-95:** mAP50-95 refers to the mean of precision (mAP) with IoU thresholds ranging from 0.50 to 0.95, in steps of 0.05 (0.50, 0.55, ..., 0.95) for the PASCAL VOC test dataset. Higher values for this metric indicate better model performance across various IoU thresholds, meaning the model is more precise and robust.

#### 2.1.5.2 Loss Functions

- **Box Loss:** Box loss measures the difference between the predicted bounding box coordinates and the reference bounding box coordinates. Minimizing this loss helps the model predict bounding box coordinates more accurately.
- **Segmentation Loss (Seg loss):** Segmentation loss measures the difference between the segmentation masks predicted by the model and the reference segmentation masks. Segmentation loss is typically used in semantic segmentation and Instance Segmentation tasks, where the goal is to accurately predict a segmentation mask for each object in the image. Minimizing this loss helps the model predict more accurate segmentation masks.

## 2.2 Natural Language Processing

Natural Language Processing (NLP) is a vast field of study in Language, involving elements of syntax processing, semantics, as well as the identification of ambiguities and complex language elements (FINGER, 2021). In essence, NLP encompasses an immense toolkit for textual feature extraction. In the study developed, only the scraping technique was used.

### 2.2.1 Scraping

The technique of extracting texts from web pages by identifying elements of the HTML code is known as Web Scraping and is part of the NLP tools. Figure 12 shows an example of identifying an object of interest using the ‘Inspect’ tool, present in any modern browser.

### 2.2.2 Selenium

The Python Selenium library (MUTHUKADAN, 2023) is a widely used web automation tool that allows developers to interact programmatically with browsers, enabling to control and manipulate them to perform automated actions - such as filling out forms, clicking buttons, navigating through pages, and extracting web data. Selenium is especially useful for repetitive tasks involving web interface interaction, like automated testing, scraping, and user interaction simulation.

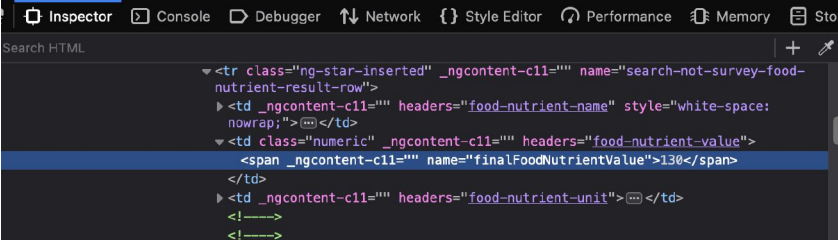
One of the main advantages of Selenium is its ability to work with different browsers, including Google Chrome, Mozilla Firefox, Safari, and Microsoft Edge. This allows developers to choose the browser that best suits their needs and use the same automation logic across different platforms. Moreover, Selenium supports multiple programming languages, including Python.

## 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:		100g	
Name	Amount	Unit	Deriv. By
Water	68.6	g	
Energy	130	kcal	<a href="#">Calculated</a>
Energy	544	kJ	

```

<tr class="ng-star-inserted" _ngcontent-c11="" name="search-not-survey-food-nutrient-result-row">
  <td _ngcontent-c11="" headers="food-nutrient-name" style="white-space: nowrap;"></td>
  <td class="numeric" _ngcontent-c11="" headers="food-nutrient-value">
    <span _ngcontent-c11="" name="finalFoodNutrientValue">130</span>
  </td>
  <td _ngcontent-c11="" headers="food-nutrient-unit"></td>
</tr>

```

Figure 12 – Example of Searching Information in HTML Code — Own Authorship using the FoodData Central website

However, Selenium also has some limitations. Firstly, its execution can be relatively slow compared to other automation solutions for scraping. Additionally, Selenium requires a running browser to perform automation actions, which can limit its scalability and use in non-graphical environments. Another limitation is the need to deal with dynamic elements on the page, like asynchronous content loading, which may require additional strategies to ensure actions are executed correctly.

### 2.3 Dataset

For the identification and segregation of foods on plates, the dataset on the Roboflow user's website (HAIR, 2023) was used — Figure 13 shows an example image from this dataset. The Histogram showing the frequency of food with a mask per image can be seen in Figure 14 — approximately 6500 images containing plates of food were used, with 4900 images for training and 1600 images for testing.

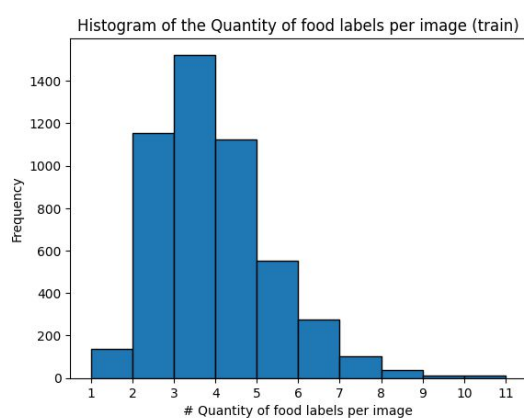
In addition to the original images, the model was also validated using this image dataset with changes in brightness and contrast, simulating variations in flash and photo luminosity, testing the model's robustness for this type of condition - represented in Figures 15 and 16.



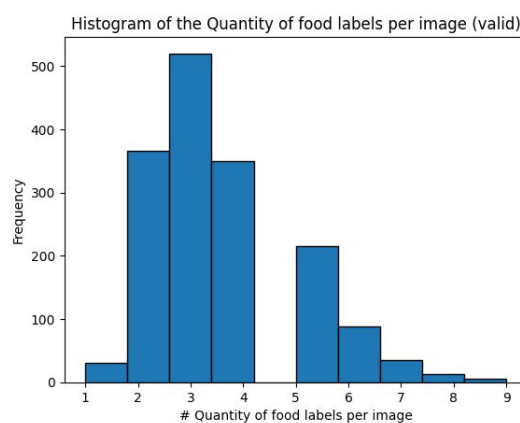
(a) Sample Picture

(b) Masked Sample Picture

Figure 13 – Dataset Sample Picture (HAIR, 2023)

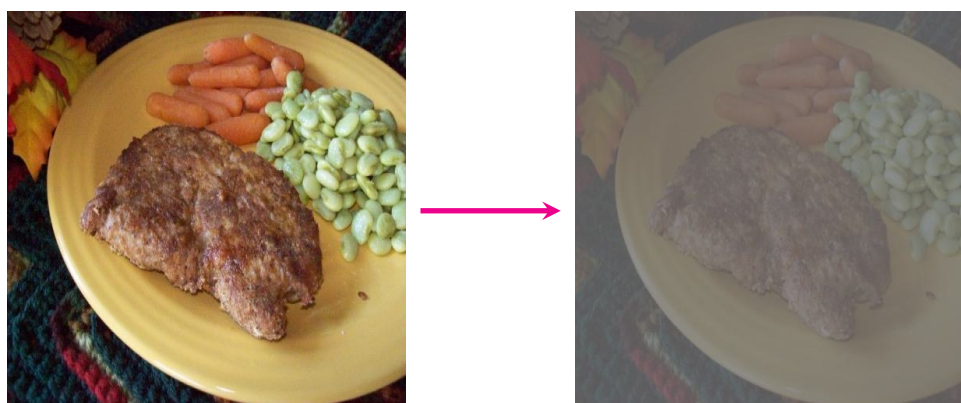


(a) Dataset - Training



(b) Dataset - Validation

Figure 14 – Histogram of Dataset — Own Authorship



(a) Sample Picture

(b) Low Brightness and Contrast Picture

Figure 15 – Low Brightness and Contrast Sample (HAIR, 2023)

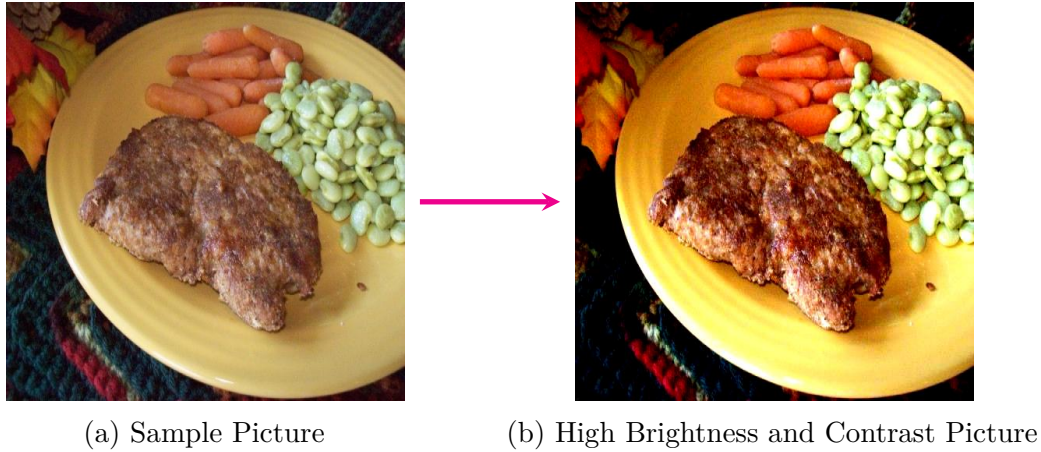


Figure 16 – High Brightness and Contrast Sample (HAIR, 2023)

Table 1 shows the 103 labels of the dataset used in this FYP. As already explained, being an existing dataset, not all labels represent the vast range of varieties of Brazilian cuisine. Still, it is noted that - generally speaking - these foods encompass the labels on a commonly considered food plate: rice, potatoes, chicken, fish, meat, types of vegetables, etc.

Two limitations to be discussed later: the dataset in question did not have image labels removed, as it already contained a relatively small number of samples for training and validation, implying training of image-mask pairs that are sometimes incorrect (i.e., image of a chicken, label of a fish); moreover, the dataset labels do not have variation in terms of food preparation, for example: roasted chicken, grilled chicken, and fried chicken; instead, there is only the generic term 'chicken'.

Table 1 – Food labels used in the Dataset of the study - Own Authorship

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 RELATED ARTICLES

This thesis employs techniques similar to those used in (MEYERS *et al.*, 2015), which integrates a Multi-Label CNN for the identification of food on a plate and the extraction of nutritional information, such as calories. Figure 17 provides an overview of the related work, which combines food detection, volume quantification of each food item, conversion of nutritional information via a dataset, and finally a mobile application that compiles all these actions for the end user.

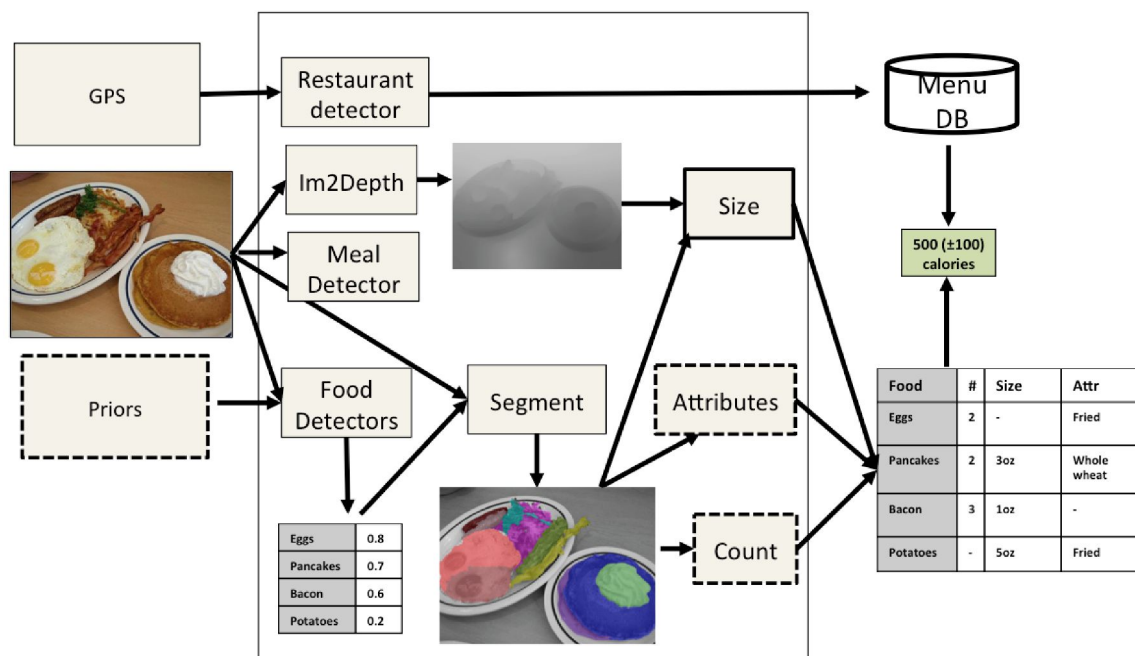


Figure 17 – General Scheme of the Work — Im2Calories Article

The chosen CV architecture was the GoogLeNet CNN, and the database used was the Food201-MultiLabel - which initially was the Food101, a public database containing 101 classes. It became the Food201 after adding another 100 classes, and finally, the Food201-Segmented after the author added masks for the Instance Segmentation task - totaling 35,000 training images and 15,000 test images. After training, the model underwent refinement by analyzing restaurant menus available in the Google Places API (Application Programming Interface). In general, food detection achieved mAP results of 80% for classes within the Food101 (original database) and 20% for classes outside of Food101, resulting in an average mAP value of 50%.

Finally, the volume estimation was done using a 3D food database from Google called GFood3d. For calorie estimation, the author used a nutritional database from the U.S. Government (USDA National Nutrient Database - NNDB). The final application was



created through the steps of detection, volume estimation, and calorie calculation; the input is a photo sent by the user and the output is the identification of the foods along with the calculation of calories, which can be observed in Figure 18.

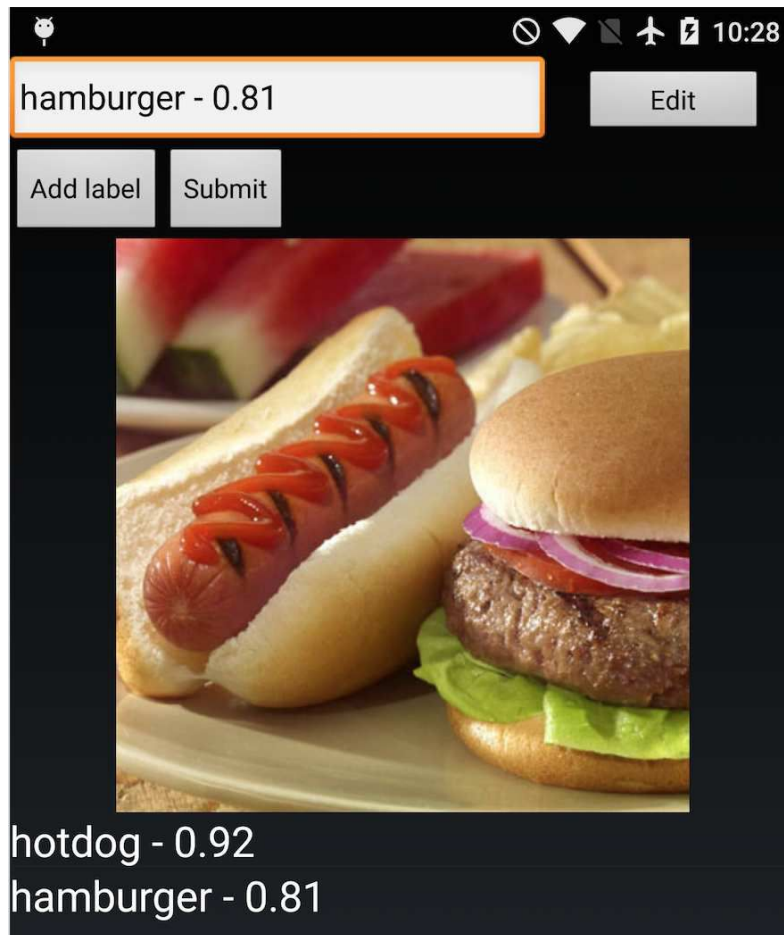


Figure 18 – Application for the End User — Im2Calories Article



## 4 METHODOLOGY

Figures 19 and 20 present the concepts of the workflow for this study. The first workflow (Figure 19) deals with the selection of the model to be used - involving the training and evaluation of different Instance Segmentation (IS) models (YOLOv8 and Detectron2), followed by additional validation with the original dataset (4900 images used in training and 1600 in validation) undergoing brightness and contrast disturbances. The second workflow (Figure 20) addresses the more applicable aspect of the selected model, where the identified labels are sent to the scraping library (Selenium), returning the nutritional labels that will be incorporated into the final image, along with Bounding Boxes of the detected food types.

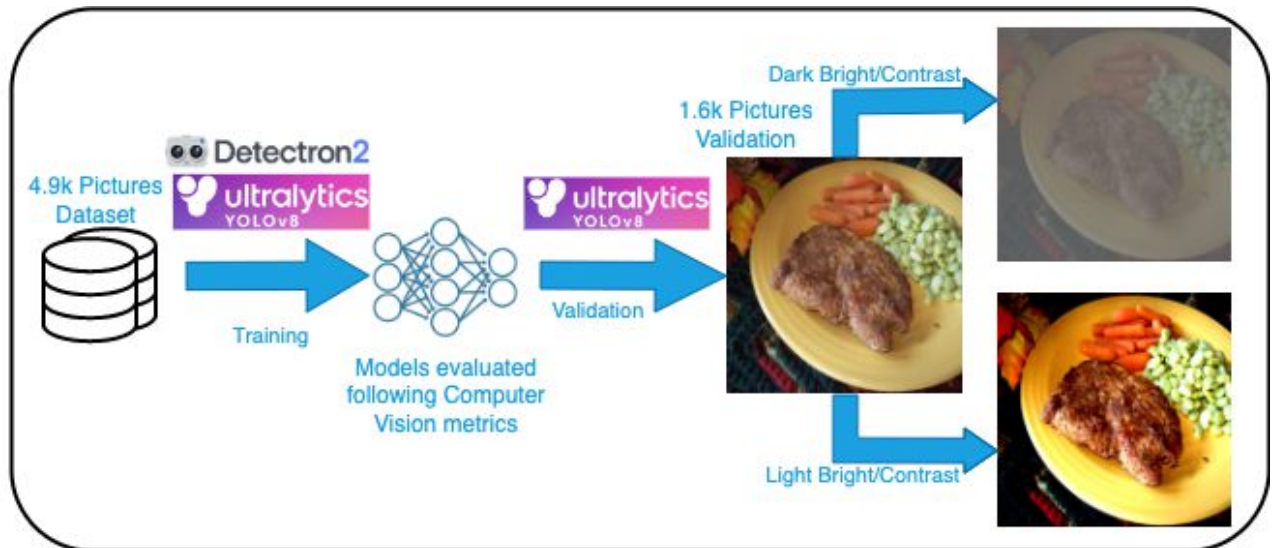


Figure 19 – First part of the execution flow — Own Authorship

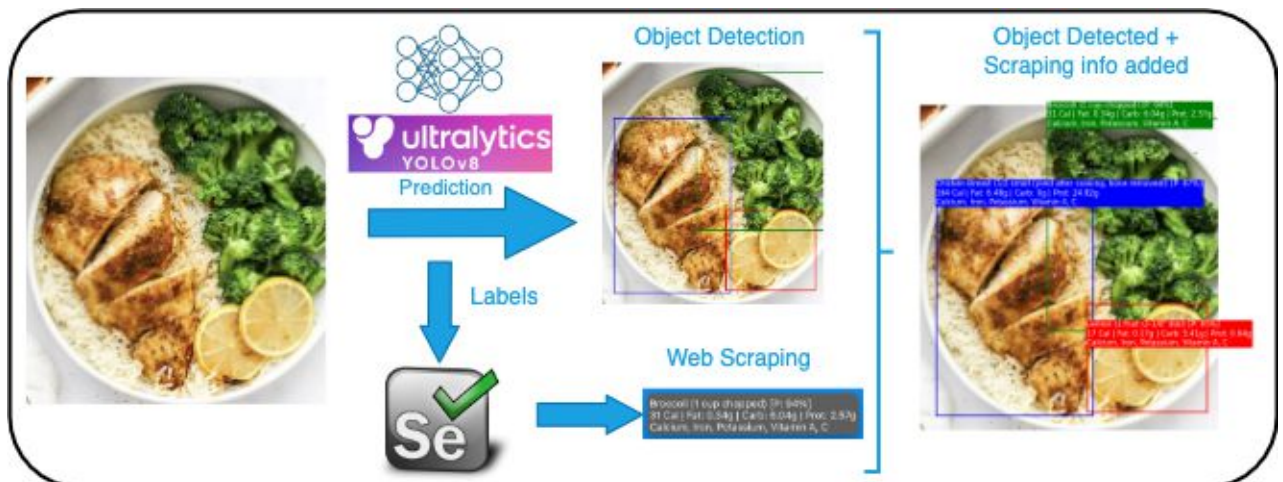


Figure 20 – Second part of the execution flow — Own Authorship using image from the website: <https://www.apinchofhealthy.com/baked-chicken-breast/>

In general terms, during the execution of the flow, it is expected that the trained model will be able to recognize different foods in an image of a food plate (CV technique - IS and OD), providing labels to bring nutritional information of the identified foods (NLP technique - Scraping), compiling this entire process into a final image. Thus, from just one input - a photo of a food plate - the model will process the information and the output will contain the detected food(s), as well as a brief description of the main nutritional information (Calories, Fat, Carbohydrates, Proteins, Vitamins, and Minerals).

## 4.1 Identification and Labeling of Foods

The task of Food Identification and Labeling uses CV techniques: IS and OD. The goal of this stage is to use state-of-the-art pre-trained models with the refinement of this network (fine-tuning) from a customized dataset with thousands of images of food plates, followed by their label and mask.

### 4.1.1 Neural Networks

Two models were used for the task of food identification and labeling: Detectron2 (WU *et al.*, 2019) and YOLOv8 (JOCHER; CHAURASIA; QIU, 2023). Both were trained with the same dataset (HAIR, 2023) for the segmentation task. For the YOLOv8 model, two variations were used with different numbers of parameters used in the pre-training of these models: YOLOv8n-seg (3.4 million parameters) and YOLOv8s-seg (11.8 million parameters). Both are available on the Ultralytics GitHub page<sup>1</sup>. No parameter was modified in relation to the standard for training the models, only the number of epochs was specified to 30.

The Detectron2 model was trained using the base configuration file, present on the model's GitHub page<sup>2</sup>: `mask_rcnn_R_101_FPN_3x.yaml` - a Recurrent Neural Network R-101 trained with the checkpoint file<sup>3</sup>. The main configurations overwritten for training this model can be seen in Table 2.

### 4.1.2 Model Evaluation

The Model Evaluation occurred using the proposed metrics and presented in chapter 2.1.5 - specifically the metrics of Precision, Recall, and mAP50/mAP50-95 for the task of IS and OD, i.e., the indices (B) and (M) of these metrics. The model that presented the best result - whether the highest value in the three metrics or the most consistently high among them - was considered the model to follow for the next tasks of this study. Both models (Detectron2 and YOLOv8) already have automatic tools for generating

---

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

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

<sup>3</sup> <[https://github.com/facebookresearch/detectron2/blob/main/MODEL\\_ZOO.md](https://github.com/facebookresearch/detectron2/blob/main/MODEL_ZOO.md)>

Table 2 – Configuration of the Detectron2 model - Own Authorship

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

these metrics at the end of the training and validation stages, whose values are used for comparison.

#### 4.1.3 Additional Validation with Brightness and Contrast Modifications in the Dataset

In order to verify the robustness of the IS method, the model selected in chapter 4.2.2 underwent additional validation: the original dataset had its Brightness and Contrast varied for a simulation of environments where the photograph of the food plate is obtained without lighting or with a lot of sunlight, for example. This new range of images was validated using the same metrics used for model selection. For this, the Python library called Pillow (CLARK, 2023) was used. Below is a snippet of the code used.

---

```
# Using the Pillow library
from PIL import Image, ImageEnhance
# Brightness adjustment based on a reference factor
image = Image.open(image_path)
enhancer = ImageEnhance.Brightness(image)
return enhancer.enhance(brightness_factor)

# Contrast adjustment based on a reference factor
image = Image.open(image_path)
enhancer = ImageEnhance.Contrast(image)
return enhancer.enhance(contrast_factor)
```

---

## 4.2 Nutritional Information Search

The task of Nutritional Information Search applies CV techniques along with the NLP technique called Scraping. In this stage, the main goal is to provide something close to the final experience for the user, i.e., the output should provide immediately useful information. In the case of this study, this stage involves the CV model selected in the previous step - which will make image predictions -, followed by scraping of the predicted labels and incorporation of this information into a return image for the user.

### 4.2.1 Scraping

The Python library called Selenium (MUTHUKADAN, 2023) was employed for the scraping task. The FoodData Central website (AGRICULTURE, 2023) was selected to be used as a provider of information and nutritional labels. In Figure 21, it can be seen how this information is arranged: a nutritional label containing the main data of Fat, Carbohydrates, Proteins, main Vitamins and Minerals present, as well as the Calories for a given portion.

**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	

Figure 21 – Example of Nutritional Label of White Rice - FoodData Central Website

The code snippet below shows an example of how information from the FoodData Central website (AGRICULTURE, 2023) is searched using the scraping technique. In this code, the label, the value, and the units of the Nutrients are obtained by codes in *HTML* and *JavaScript* languages.

```
# Using the Selenium library to find the Nutrients on the FoodData Central page
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

```

---

It was necessary to create a correspondence dictionary between the predicted labels of YOLOv8 and the search keys of the FoodData Central website, since the website presented a vast variety of items under the same name. Figure 22 shows an example of this search when looking for the term 'rice' and finding 132 results. Thus, the dictionary aims to remove ambiguity between these labels. Below is a sample:

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

Figure 22 – Example of Search for Rice - FoodData Central Website

---

```

# Sample of the correspondence dictionary - key = YOLO label, value =
  FoodData Central search key
correspondence_dict = {'banana': 'Bananas, raw', 'bread': 'Bread, white
  wheat', 'broccoli': 'Broccoli, raw', 'cabbage': 'Cabbage, raw', 'cake':
  'Cake, sponge, commercially prepared'}}

```

---

A second necessary adjustment was the search for the identifier code (called 'fdc\_id') in a CSV file provided by the FoodData Central website, as the website's URL does not use the names of the foods, but this identifier code. This change was simple to make, as the provided file was properly formatted. A limitation of this step is that the identified food - and consequently recognized in the correspondence dictionary - is restricted to only one type of instance on the website. This is also due to the fact that the original dataset labeling did not consider the type of preparation of that food, so it is not possible for this information to be incorporated into the scraping activity. Thus, when identifying the 'rice' component, it is not possible to distinguish between brown rice, risotto rice, or white rice, for example.

#### 4.2.2 Integration of Tools and Expected Output

With the information from the Bounding Boxes (obtained by the IS prediction - YOLOv8), along with the nutritional information (obtained by the Scraping task -

Selenium), it was possible to combine these capabilities, providing the end user with an experience where, from a photo of a food plate, it was possible to identify each food present, as well as to know the relevant nutritional information - whether for diet control or for medical use related to the restriction of a certain component. Figure 23 shows the beginning and the end of this process.

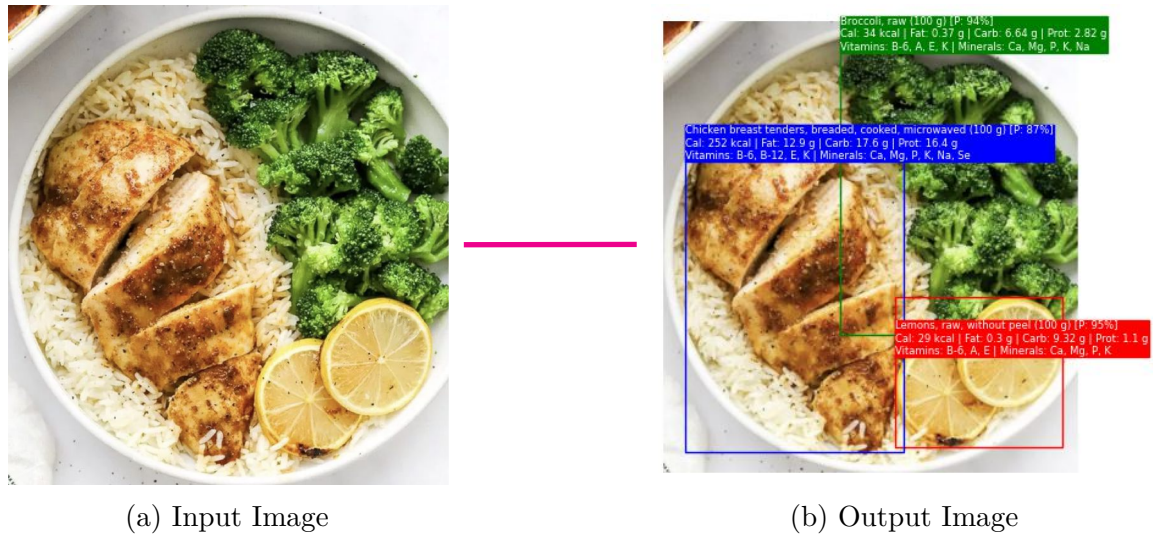


Figure 23 – Input and Output of Tool Integration — Own Authorship using image from the website: <https://www.apinchofhealthy.com/baked-chicken-breast/>

The final goal of this tool is to bring nutritional awareness in an easy, quick, and intuitive way, as opposed to the traditional method, which consists of reading the physical label of each component on the food plate.

## 5 RESULTS ANALYSIS AND DISCUSSION

### 5.1 Model Training

The following subsections address the training of the Detectron2 and YOLOv8 models (variations ‘n’ and ‘s’). Both YOLOv8 models showed better performance indicators than the Detectron2 model, as expected. The latter, being older and with a lesser detection capability, was included in the study to establish a baseline, from the simplest to the state-of-the-art model.

#### 5.1.1 Detectron2 Training

Table 3 shows the complete training results for Detectron2, both for the Object Detection (OD) task (column ‘Box’) and the Instance Segmentation (IS) task (column ‘Segment’). Generally, the results are low compared to those observed with the YOLOv8 architecture (which will be discussed in the following sections), with only a few precision and recall values above 10% - and this is only observed when the Intersection over Union (IoU) is at its lowest value (50%). This behavior was expected, as the Detectron2 model stopped being updated between 2020 and 2021.

Comparing the effect of detection area for an IoU of 50 to 95%, it is noted that for small areas the results are low for precision and recall in both tasks. When increasing the area to ‘large’, slightly better results are observed (9% in precision and 15% in recall - results in bold). This is because smaller areas imply greater detection sensitivity, making the task more complex, and the opposite for large areas.

Overall, the results were not satisfactory for the Detectron2 model, but it is a good candidate for establishing a baseline in comparison with the state-of-the-art YOLOv8 model.

#### 5.1.2 YOLOv8 Training

This subsection is divided into two parts, to separately analyze the results of the model with the lower number of parameters (YOLOv8n - 3.4 million) and the model with the higher number of parameters (YOLOv8s - 11.8 million). The results are presented in terms of the OD task - indicated by ‘Box’ or (B) - and IS task - indicated by ‘Segmentation’ or (M). Moreover, the two vertical axes were used in each graph, to optimize space, with the left axes of the left graphs showing the intervals of loss function values (dashed lines) during training and validation, while the right axis of the same graph shows the precision curve with a continuous line. For the right graphs, the intervals of recall values (continuous



Table 3 – Detectron2 Training Results

	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]	<b>0.117</b>	<b>0.118</b>
	[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]	<b>0.086</b>	<b>0.089</b>
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]	<b>0.151</b>	<b>0.152</b>
	[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]	<b>0.154</b>	<b>0.155</b>

line) are seen on the left axes, while the intervals of mAP50 and mAP50-95 values (dashed lines) are seen on the right axes.

#### 5.1.2.1 YOLOv8n Model

In Figure 24, the training results of the YOLOv8n model are observed. The model took approximately 24 hours for training in 30 epochs, using the previously presented food image dataset. The configuration used for training: MacBook 2021 with M1 Pro chipset with 10 CPU cores and 16 GPU cores, in the Jupyter and VSCode (version 1.8) environment.

Observing the graph containing the loss function values during the training and validation stages, it is noticed that for both tasks the trend of stability has not yet been reached, suggesting strong indications that the model could be run for more epochs before reaching the loss function limit - a fact limited by the current hardware. The precision showed unstable behavior, oscillating between values of 45 to 55% in the two tasks, ending at approximately 53%. This result is aligned with what is expected for version 8 of the YOLO model, as shown in chapter 2.1.3.

The mAP50-95 index shows that both tasks obtained results below expectations: according to the developer, version 8 of YOLO could achieve results of 30% for segmentation and 37% for detection (results obtained using the COCO2017 validation dataset, according to (JOCHER; CHAURASIA; QIU, 2023)), but 20% for segmentation and 23% for detection were obtained during the training stage. Moreover, as with the loss function, it is noted that the mAP50 and mAP50-95 metrics calculated for the two tasks have not yet obtained stable values, again indicating that training with more epochs could result in better metrics.

Figure 25 shows an example of validation of the IS task. It is noted that the original label (left figure) contains Noodles, a sauce and broccoli, while the figure with the



predicted labels (right figure) presents the same results, but repeating the identification of the Noodles and broccoli classes more than once, indicating that the mAP50-95 with a result of 20% may not achieve the ideal framing of the real label.

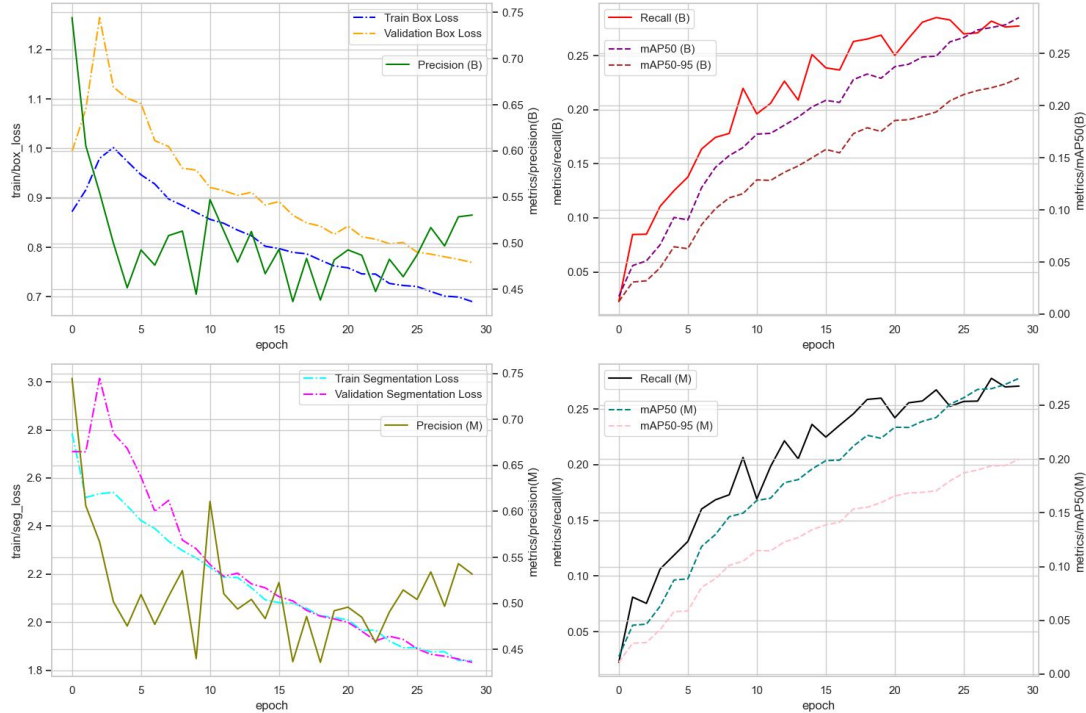
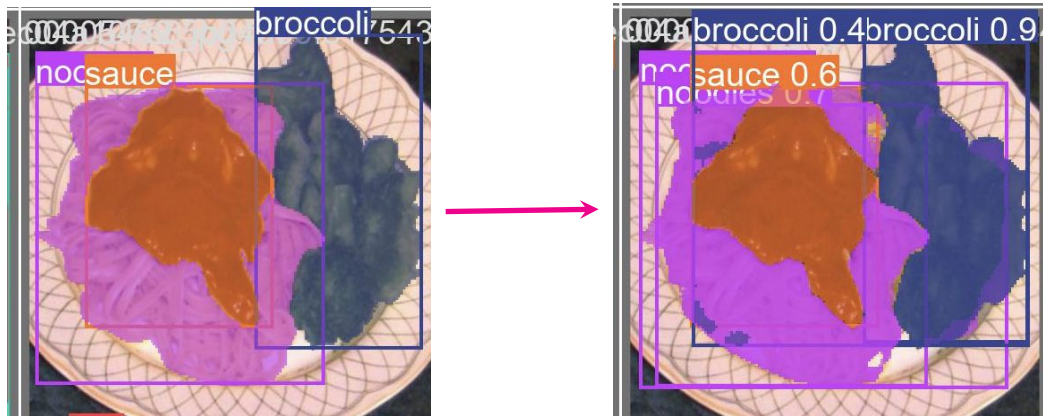


Figure 24 – Compiled Results YOLOv8n - Own Authorship



(a) Segmentation Example - Real Label

(b) Segmentation Example - Prediction

Figure 25 – Segmentation Result of YOLOv8n Model — Own Authorship

#### 5.1.2.2 YOLOv8s Model

Figure 26 shows the training results of the YOLOv8s model. The analysis is very similar to the previous model (YOLOv8n), both in form and results, therefore, the similarities will only be pointed out and the divergences will be explored more deeply.

This model also had a training of 30 epochs, taking more than 72 hours to complete. The same hardware configuration of the previous model was used for this model.

As with the previous model, the loss function of both tasks in the left graphs (dashed line), as well as the mAP50 and mAP50-95 (right graphs dashed line) show that the 30 epochs were not enough to reach the total stability of the model, indicating that a larger number of epochs could bring significant gains to the results. For the precision value in both tasks (left graphs continuous line), again similar to the previous model, the results showed an oscillation, ending in a precision of approximately 40% - a result lower than the YOLOv8n model, and consequently lower than expected for this model.

The mAP50-95 index, despite showing higher values for the YOLOv8s model (28%: detection; 26%: segmentation) compared to the YOLOv8n model during the training stage, still showed performance below expectations (above 45% for detection and 37% for segmentation, according to the developer of the YOLOv8 model (ULTRALYTICS, 2023)).

Figure 27 shows an example of validation of the IS task. In this example of the YOLOv8s model, the real labels (left figure) and the predicted labels (right figure) are exactly the same, and the predicted label does not show additional labels - which demonstrates that the mAP50 and mAP50-95 are indeed performing better.

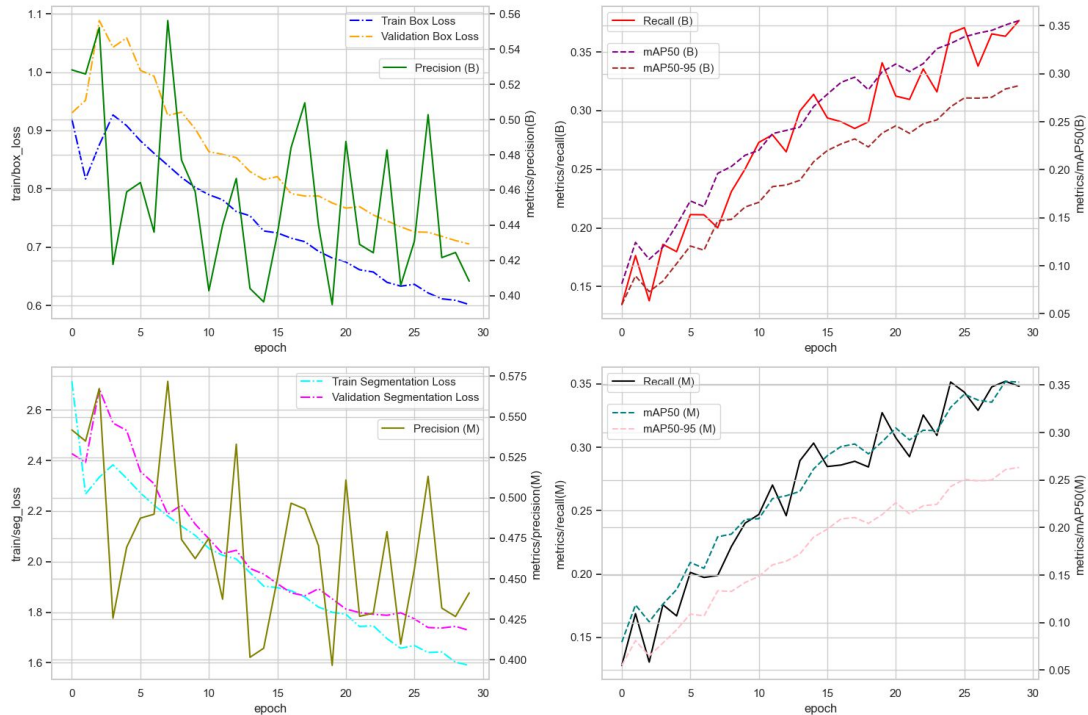


Figure 26 – Compiled YOLOv8s Results - Own Authorship

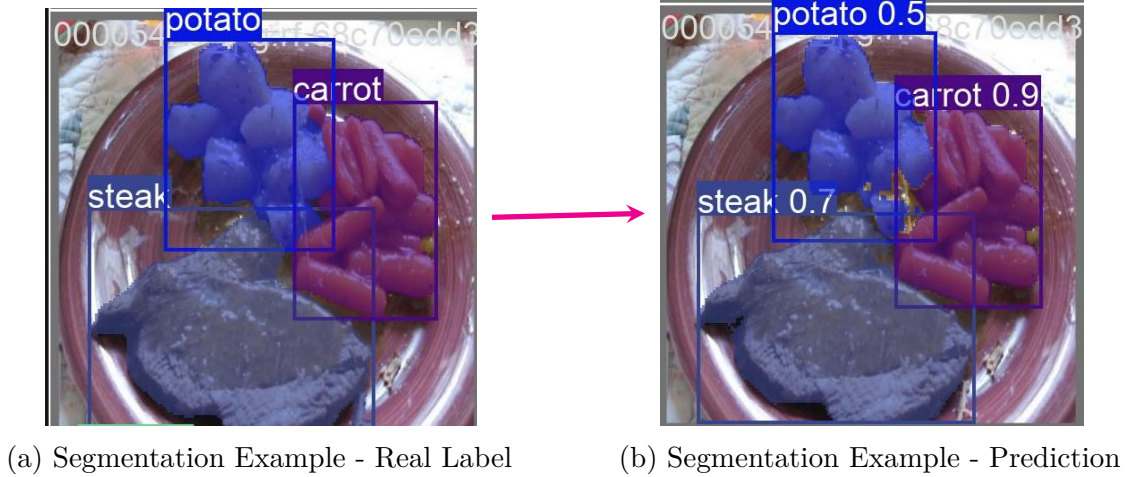


Figure 27 – Segmentation Result of the YOLOv8s Model — Own Authorship

## 5.2 Comparison between Different Models and Lighting Conditions

### 5.2.1 General Results

The models analyzed comprise five variants: Detectron2 with the original validation dataset, YOLOv8n with the original validation dataset, YOLOv8s with the original validation dataset, and YOLOv8s with brightness and contrast adjustments, represented as dark and light. For all models, the evaluated parameters were precision (P), recall (R), and the mean of the average precision (mAP50 and mAP50-95) for OD (index B) and IS (index M). The analysis below in Table 4 includes the validation datasets, executed and tested after the models' training.

The original Detectron2 presented more modest results in all metrics, except for the recall for IS (R(M)), where it performed slightly better than its corresponding precision value. It is noted that the precision and recall for OD and IS are quite close, indicating a balance between true positives and false positives in the tasks. However, its average precision, both in OD and IS, was significantly lower than that of the other models.

The YOLOv8n and YOLOv8s models with the original images showed a greater difference between precision and recall. In the YOLOv8n model, precision significantly exceeded recall for both OD and IS, suggesting that the model was more conservative, prioritizing the correction of true positives at the expense of detecting all possible positives. Conversely, the YOLOv8s showed a greater similarity between precision and recall values, with precision slightly higher than recall, demonstrating a more homogeneous balance between true positives and false positives.

When the dark and light versions of YOLOv8s are considered, there is a general decrease in performance metrics. In the dark scenario, precision and recall for both tasks, OD and IS, decreased considerably compared to the model with the original

validation dataset. This suggests that the model had difficulties adapting to images with reduced contrasts and brightness. In the light version, the decline in performance was less pronounced, with precision and recall for OD and IS showing a more modest reduction compared to the model with the original validation dataset. This may indicate that the model had a better ability to adapt to images with increased contrasts and brightness.

In terms of mAP50 and mAP50-95 - that is, in terms of separating overlapping instances - the original YOLOv8s and YOLOv8n models exhibited superior performance to Detectron2, both for OD and IS. The dark and light variants of YOLOv8s demonstrated a decrease in all mAP50 and mAP50-95 metrics compared to the model with the original validation dataset, with the dark model suffering a greater impact.

Information that has already been raised in previous chapters, but worth revisiting: the YOLOv8n model used 3.4 million parameters for training, while the YOLOv8s model used 11.8 million parameters - that is, common sense would lead one to think that the model with more parameters would bring the best results, but this was not seen in its entirety. Precision for both tasks showed a lower result in the 's' model, but all other indicators performed better in this model.

In summary, the YOLOv8 models demonstrated superior performance to Detectron2. Among the YOLOv8 models, the 's' model performed better overall. The dark and light variants of YOLOv8s with contrast and brightness adjustments had lower performance than the model with the original dataset, suggesting a sensitivity of the model to changes in lighting conditions. This analysis highlights the importance of considering lighting conditions when training and validating CV models, as well as the need for robust data augmentation strategies to improve the model's generalization to different lighting scenarios.

Table 4 – Comparison between different models and conditions - Own Authorship

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 Results by Number of Instances

This subsection presents an analysis of the results comparing two groups from the validation dataset: foods with many instances and foods with few instances. The question to be answered: do classes with a higher number of instances predominantly show better results?

In the high-instance group (Table 5), classes such as broccoli, lettuce, potato, and tomato were included. The YOLOv8n and YOLOv8s models with the original dataset images obtained better results in almost all metrics for these classes. For broccoli, YOLOv8s

with the original validation dataset demonstrated superiority, with the highest precision for both OD and IS tasks. The same pattern was observed for recall, mAP50, and mAP50-95 metrics. Evaluating lettuce, YOLOv8s with the original validation dataset again showed the highest precision and recall for both tasks. In terms of mAP50 and mAP50-95, this model was also superior, although with smaller margins compared to other classes. The potato class was best identified by YOLOv8s with the original dataset, with all metrics being superior compared to other models. Finally, for tomato, although YOLOv8n with the original dataset had the highest precision in the OD task, YOLOv8s with the original dataset outperformed in the other indices.

When comparing the performance of models in low-light (dark) or high-light (light) environments, there was a significant drop in metrics for all high-instance food classes. This suggests that these models are sensitive to lighting changes, which can be a challenge in real-world scenarios.

In the low-instance group (Table 6), which includes apple, grape, shrimp, and soy, a similar trend was observed. The YOLOv8s model with the original dataset generally presented the best metrics for apple and grape classes. However, the shrimp class was best identified by the YOLOv8n model with the original dataset. For the soy class, YOLOv8s with the original dataset again outperformed the others.

Lighting variations also negatively impacted the performance of the models for the low-instance classes. In particular, the drop in performance was more pronounced in the dark setting, possibly indicating greater sensitivity to low light conditions.

Comparing the overall results between classes with high and low instances, it was noted that the YOLOv8n and YOLOv8s models with the original dataset performed relatively well in all classes, regardless of the number of instances. However, a larger number of instances seems to have contributed to better overall performance, but this advantage was not significantly impactful considering that some classes had almost ten times fewer instances.

A possible explanation for why classes with a low number of instances showed considerable performance could be related to the shapes, colors, and textures of some foods, making CV model detection possible even with few training instances. However, this might not translate the same way under varying light conditions, suggesting the need to enhance the robustness of these models to such variations.

Finally, it's important to mention that the original Detectron2 model showed results only for precision and recall metrics, thus limiting a full comparison with other models. Additionally, for soy and shrimp, Detectron2 did not show any results. For the available indices, this model performed inferiorly compared to YOLOv8n and YOLOv8s models, and classes with a higher number of instances performed better than those with a fewer

number of instances.

Table 5 – Comparison between high instance classes - Own Authorship

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

Table 6 – Comparison between low instance classes - Own Authorship

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

### 5.3 Integration of Results and Application

The previous section explored the differences between Computer Vision (CV) models, resulting in the YOLOv8s model as the one that best achieved precision indicators along with instance separation. Therefore, this model was used in the subsequent stages of the execution flow. As explained in chapter 4, web scraping used a dictionary of equivalences for searching the predicted labels of YOLOv8s. Thus, the results of the scraping are always equivalent to the best possible outcome, making it impossible to evaluate the performance of this task.

For the integration and incorporation of the tools, the final code - here called the application - encompasses the following steps:



1. Use of the YOLOv8s model trained with data from the dishes to predict a new image;
2. Conversion of the labels obtained by the correspondence dictionary;
3. Use of the detected label for a search on the FoodData Central website using the Selenium library;
4. Formatting of the scraping results;
5. Incorporation of the detection boxes and the formatted scraping text into the final image.

Below are 5 examples where the application was performed from start to finish - Figures from 28 to 32 illustrate this process. All input images were taken from the website: <https://www.apinchofhealthy.com/>.

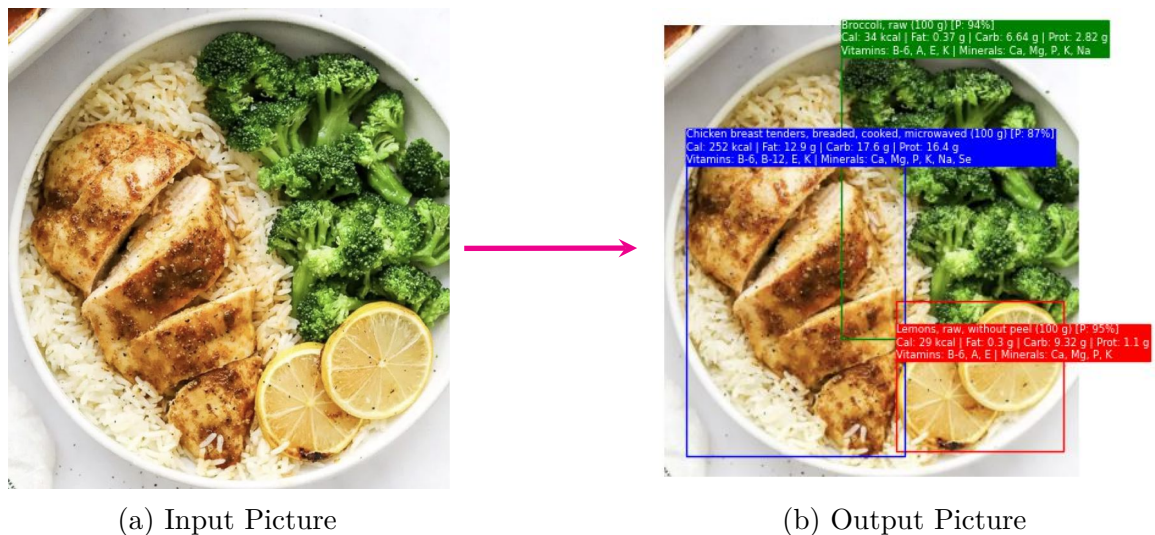


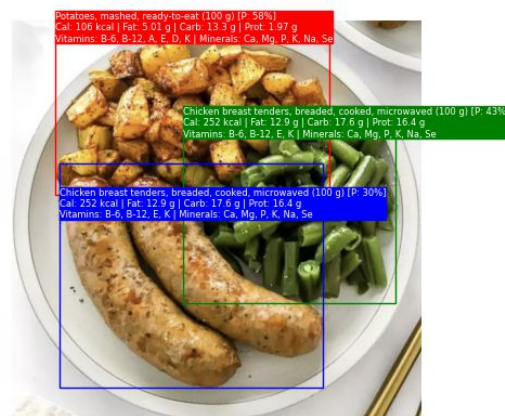
Figure 28 – Full Application of the tools | Example 1 - Own Authorship

The evaluation of the results can be seen in Table 7. The ‘C?’ columns show where the results were correct (marked with an ‘X’) and where the results were incorrect (without any marking). The examples range from excellent results - such as examples 1 and 3 - to very poor results - such as examples 2 and 5. Some of the observed problems:

- Incorrect Label Identification: Example 2 shows an image of Sausage being identified as Chicken Breast. This could be due to incorrect labeling in the original dataset for the image of the Sausage, the Chicken, or both. Another possible explanation for such errors could be the similarity in colors, textures, and shapes between the real and predicted labels. The solution to this problem would be a review of the



(a) Input Picture

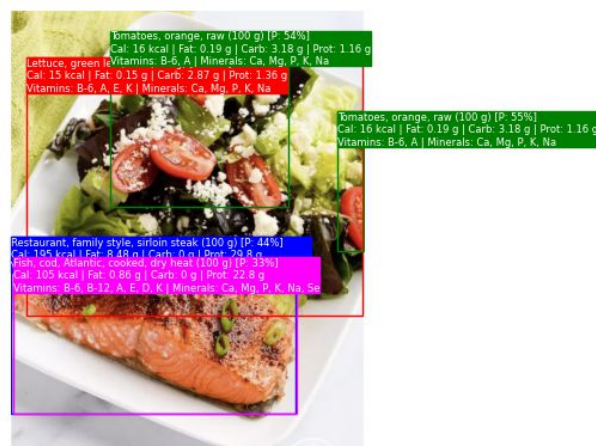


(b) Output Picture

Figure 29 – Full Application of the tools | Example 2 - Own Authorship

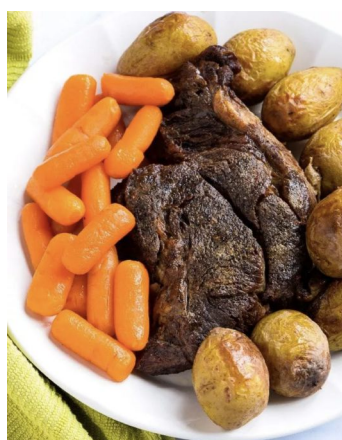


(a) Input Picture

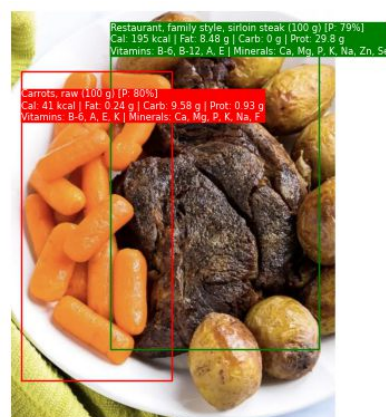


(b) Output Picture

Figure 30 – Full Application of the tools | Example 3 - Own Authorship



(a) Input Picture



(b) Output Picture

Figure 31 – Full Application of the tools | Example 4 - Own Authorship





Figure 32 – Full Application of the tools | Example 5 - Own Authorship

dataset, adjusting labels and masks according to the real image, along with adding new images with greater variation of foods.

- Label Overlap: This problem can be observed in examples 1, 3, and 4. In examples 1 and 4, it is noticeable that the identification of one of the foods was hampered by the overlap of surrounding foods - Rice in the case of example 1, and Potato in the case of example 4. This demonstrates that the value of mAP50 and mAP50-95 should be increased through optimizations in the YOLOv8 model.

Table 7 – Application Results Analysis - Own Authorship

	Food 1	C?	Food 2	C?	Food 3	C?	Food 4	C?	Total	Comment
Example 1	Broccoli	X	Chicken	X	Lemon	X	Rice		75%	Rice overlapped.
Example 2	Sausage		Green Beans		Potato	X			33%	
Example 3	Fish	X	Lettuce	X	Tomato	X			100%	Identified also a 'beef' label.
Example 4	Beef	X	Potato		Carrot	X			67%	Potato overlapped.
Example 5	Rice	X	Corn		Tomato	X	Beans		50%	

Overall, the practical application of this study was demonstrated by showing some examples of food detection and extraction of nutritional information. The limitation of this study occurs due to two aspects: lack of computational power for conducting new rounds of training - either of models with more parameters or of the models used, with variations in configuration -; and in the dataset: correction of incorrect masks and labels, and the addition of images, presenting greater variability for the foods and types of preparation.



## 6 CONCLUSIONS

Throughout this study, a Computer Vision (CV) solution aimed at providing nutritional awareness through food detection and the incorporation of nutritional labels was developed. The study addressed Object Detection (OD) and Instance Segmentation (IS) of foods using the following CV models: Detectron2, YOLOv8n, and YOLOv8s.

From the analysis of the obtained results, it was observed that the YOLOv8s model showed the best performance in OD and IS, with higher recall results and mAP50 and mAP50-95 metrics - only the precision result was higher in the YOLOv8n model. This suggests that YOLOv8s was the appropriate choice for the application of nutritional labels. This model was also evaluated using two variations in the validation dataset, altering light and contrast conditions to create the light and dark validation datasets. The results were almost unchanged for light images, but there was a nearly 50% decrease in food identification in dark images, demonstrating difficulties in executing the workflow in poorly lit environments.

The incorporation of web scraping enriched the results, obtaining additional nutritional information from the FoodData Central website. This integration contributed to increasing the usefulness of the information about the detected foods, bringing data on calories, protein, carbohydrates, fats, as well as vitamins and minerals.

Therefore, the solution developed in this study offers a significant contribution to nutritional awareness. Through the use of CV models, it was possible to detect and segment food, providing a useful tool to assist users in choosing healthier foods. The integration with web scraping resources expanded the available information, providing nutritional data.

It is hoped that the developed solution can contribute to promoting healthier eating habits and, consequently, to combating the growing trend of obesity and diabetes in Brazil and worldwide. For future work, what has been developed will serve as a foundation for creating an application aimed at calculating Insulin for Insulin-Dependent patients.

### 6.1 Next Steps

To optimize this study, increasing the main performance indicators explored here (Precision, Recall, mAP), it is suggested to increase the number of image-mask pairs containing food plates, as well as the use of more specific labels for variations in food preparation, especially for images with low lighting, providing greater robustness for detection in this condition. In addition to these points, it is highly recommended to use hardware with robust graphic processing, in order to explore larger YOLOv8 models (such

as the 'm' and 'l' models) and also the use of more epochs ( $> 30$ ).

As described in Chapter 1, the initial idea of this thesis would be the incorporation of Glycemic Index data, in order to become a tool that would assist insulin-dependent patients in estimating the amount of Insulin used per meal. For the execution of this task, it is suggested: to estimate the volume of each food - to calculate the quantity of each of the elements of the plate more precisely -, followed by the calculation of the increase in the Glycemic Index caused by the patient's food plate. Although it seems simple, this stage would have to consider Insulin sensitivity (a parameter that varies between individuals), and the Glycemic Index history of the last few hours.

## REFERENCES

- 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>>.