National College of Ireland

Evaluation of the Detectron2 framework for Instance Segmentation of Multi-Component Meal Images

MSc Research Project
Artificial Intelligence

Ayodeji Michael Adedeji
Student ID: x23170476

School of Computing
National College of Ireland

Supervisor: Anh Duong Trinh

National College of Ireland
Project Submission Sheet
School of Computing

National College of Ireland

Student Name:	Ayodeji Michael Adedeji
Student ID:	x23170476
Programme:	Artificial Intelligence
Year:	2024
Module:	MSc Research Project
Supervisor:	Anh Duong Trinh
Submission Due Date:	12/12/2024
Project Title:	Evaluation of the Detectron2 framework for Instance Segmentation of Multi-Component Meal Images
Word Count:	4590
Page Count:	20

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	
Date:	28th January 2025

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Evaluation of the Detectron2 framework for Instance Segmentation of Multi-Component Meal Images

Ayodeji Michael Adedeji
x23170476

Abstract

The task of instance segmentation of multi-component meal images presents a unique challenge due to the complex visual characteristics of certain food components, such as small areas, complex boundaries and the lack of visual distinction between food components. These challenges are further complicated by the diversity of food globally, with differences in preparation and presentation. This study uses a multi-stage segmentation approach with the Detectron2 framework on two multi-component meal image datasets, UECFOODPIXComplete and FoodSeg103, to assess the framework's suitability at the task of instance segmentation for multi-component meal images. Experimental results showed the framework achieved a mean average precision score of 36.4% for the UECFOODPIXComplete dataset and 20.9% for the FoodSeg103 dataset. This research highlights the framework's potential, challenges and areas of improvement at the task of instance segmentation of multi-component meal images.

1 Introduction

1.1 Background

The rise of advanced computer vision techniques has created more possibilities for innovative applications in various domains, including medicine Park et al. (2015), aerial imagery CV et al. (2023), nutritional and food analysis Wang et al. (2022). One of such advanced techniques, instance segmentation, an extension of object detection that focuses on the accurate segmentation of each object in an image, has become well-known for its ability to detect and segment objects in complex visual scenes. An example of its use is the segmentation of multi-component meal images, a challenging task with significant effects for automated dietary assessment, portion control and nutrition monitoring.

Unlike general object detection, instance segmentation in food images requires a complex understanding of overlapping food items, diverse textures, colours and irregular shapes. The complexity is further increased by varying lighting conditions and the variability in food presentations and cooking styles. Multi-component meal images consist of multiple food components, and require efficient segmentation techniques to accurately detect and segment each food component. This task is essential for applications such as calorie estimation, automated meal logging, and personalised diet planning, which rely on the accurate identification and quantification of food items.

Detectron2, an open-source platform developed by Facebook AI research, has emerged as a leading framework for object detection and instance segmentation tasks. It offers modular and state of the art architectures with effective customisation capabilities. With in-built support for advanced models such as Faster R-CNN and Mask R-CNN, Detectron2 has shown remarkable performance across various domains such aerial imagery, defect detection, mammography. However, its application in the domain of multi-component meal segmentation remains under-explored. This study seeks to address this gap by evaluating the performance of the framework at the task of instance segmentation in multi-component meal images.

This research aims to analyse the performance of the Detectron2 framework, an advanced object detection and segmentation framework when presented with the unique challenges of detecting and segmenting individual food components in multi-component meal images. The goal of this research is to improve automated food recognition technologies, which could help with practical applications like diet tracking and food logging. This study also intends to offer useful ideas for future research in the ever growing area of AI-assisted food analysis.

This research would analyse the performance of the Detectron2 framework on two multi-component meal images datasets namely, FoodSeg103 and UECFOODPIXComplete, two datasets that have been specifically designed for multi-component meal image segmentation research.

1.2 Objectives and Research Question

The main objective of this research is to evaluate the performance of the Detectron2 framework using metrics such as mean Average Precision (mAP) and Average Precision (AP) per-class. To achieve this, the Detectron2 framework would use the Mask R-CNN model, a model well-known for its instance segmentation capabilities, with a ResNet-101 Backbone for feature extraction. The model is trained on two datasets namely, FoodSeg103 and UECFoodPixComplete, with both datasets individually comprising of over 7,000 images and their respective masks. By assessing the effectiveness of the Detectron2 model in accurately segmenting multi-component meal images, this research aims to contribute to the advancements in automated food recognition technologies and providing insights for future research.

Based on the discussed objectives, this goal of this study is to answer the following research question: How effective is the Detectron2 framework at the task of instance segmentation in multi-component meal images, and what are its limitations at the task?

1.3 Structure of the Report

This report is divided into the following sections: Section one provides an overview of the research work, the research question and the objective of the research. Section two is composed of the various related works on instance segmentation, food image analysis and the applications of Detectron2. Section three is composed of the research methodology, it focuses on the datasets used and the Detectron2 model. Section four and five respectively detail the design specifications and implementation for the experiment. Section six contains the results and evaluation of the experiment, including a discussion of the results. Finally, section seven contains the conclusion and future work.

2 Related Work

Various works have been done on instance segmentation and food image analysis, this section provides insight on the previous and existing works, it also discusses the varied application of the Detectron2 framework.

2.1 Instance Segmentation

Image segmentation, a subsection of the computer vision, is sub-divided into three tasks: semantic segmentation, instance segmentation and panoptic segmentation. Semantic segmentation focuses on the assignment of corresponding labels to each object instance in images, with repeated object instances sharing the same labels Tang et al. (2021). Instance segmentation differs from semantic segmentation as it focuses on outlining each object with a bounding box or a segmentation mask, with repeated object instances being outlined separately. Panoptic segmentation combines both approaches as it assigns a semantic label and an instance id to each pixel in an image Kirillov et al. (2019).

Existing fully-supervised instance segmentation methods are divided into three categories based on their number of stages: single-stage, two-stage and multi-stage Gu et al. (2022). Two-stage instance segmentation methods are often characterised by sequentially predicting object bounding box and segmenting instances within the predicted object bounding box, a notable examples of this method is Mask R-CNN He et al. (2017). Despite its strengths, this approach had limitations such as increased inference time due its sequential process of two-stage instance segmentation methods, asides from this, bad performance at the bounding box prediction stage often influenced the segmentation stage. These limitations led to the development of single-stage and multi-stage instance segmentation methods.

Single-stage instance segmentation methods focus on the parallel processing of object detection and mask prediction. It had significant advantages as its parallel processing often led to faster inference time and higher scalability, as discussed by Lin et al. (2020). However, this single-stage method was still subject to one of the limitations of the two-stage framework: the dependence of the segmentation stage on accuracy of the bounding box predictions, notables examples of this method include YOLACT Bolya et al. (2019) and SoloV2 Wang et al. (2020).

Multi-stage instance segmentation methods focus on the sequential refinement of the results from both the bounding box prediction and segmentation stages Gu et al. (2022). The methods employed techniques such as quality scoring and segmentation mask refinement were used improve the performance. Some of the multi-stage methods include Hybrid Task Cascade, a multi-stage framework designed to approach instance segmentation by using information from both the object detection and segmentation processes in various stages to enhance the instance segmentation task Chen et al. (2019). It also includes a fully convolutional branch, enabling the method with the ability to distinguish hard foreground objects from cluttered background, another notable mention is Mask Scoring R-CNN, a method that employs a mask scoring technique to refine its generated masks Huang et al. (2019).

2.2 Food Image Analysis

Food image analysis is an exciting and growing field, driven by the need for systems that can segment and detect food under diverse conditions. The technology plays a key role in tasks such as diet tracking, calories estimation, and creation of personalized nutrition plans. However, the task of segmenting and detecting food items in an image is challenging due to the complexities such as the lack of diverse datasets, overlapping items, and multi-component meals with varying textures and appearances.

To address these challenges, researchers have not only experimented with different advanced models, with each offering its unique strengths, but have also worked to create informational datasets. Datasets such as FoodSeg103 Wu et al. (2021), UECFoodPix-Complete Okamoto and Yanai (2021), and Food101 Bossard et al. (2014), have been curated to fulfil the need for diverse and detailed datasets for food image analysis.

Wu et al. (2021) developed two food image datasets, FoodSeg103 and FoodSeg154, to address the lack of high quality food image datasets with accurate ingredient labels and pixel-wise masks. Additionally, they also developed a segmentation model using a multi-modality pre-training approach, which was then validated against three baseline semantic segmentation methods: dilated convolution, feature pyramid, and vision transformer. However, the dataset exhibited certain limitations, such as high intra-class variance i.e. the same food items appeared in diverse forms due to factors such as cooking styles and presentations. Furthermore, some food classes were significantly under-represented in the dataset leading to an imbalance in the dataset.

Wang et al. (2022) noted the limitations of Convolution Neural Network(CNN) at capturing the specific characteristics of food images for segmentation tasks and proposed the development of Swin Transformer-based pyramid network to enhance food segmentation tasks by capturing richer background and boundary information. The goal was to use a pyramid pooling module(PPM) that is capable of aggregating contextual information from various regions of images to improve the feature representation of global information in the images. The experiment was also conducted with the FoodSeg103 dataset, with the results demonstrating significant improvements over existing traditional CNN models.

With a focus on addressing the speed-accuracy trade-off and limitations of existing techniques at handling complex shapes, Nguyen et al. (2024) proposed a novel multi-task neural approach for instance counting, detection and segmentation of food components. The approach prioritised real-time performance while maintaining high segmentation quality by using a shared deconvolution sub-network, selective pixel labelling and parameter-free post-processing to reduce processing time and memory consumption. The experiment was conducted with the Mixed Dishes, UECFOODPIXComplete and FoodSeg103 datasets.

2.3 Detectron2 Framework

The Detectron2 framework, developed by Facebook AI Research, is a flexible framework designed for object detection and segmentation tasks. It supports different computer vision needs, such as object detection, semantic segmentation, instance segmentation and panoptic segmentation. It is equipped with models such as Faster R-CNN, DeepLabV3+, RetinaNet, Mask R-CNN and Panoptic FPN. Additionally, it offers advanced capabilities such as the ability to refine the segmentation stage of two-stage models, such as Mask R-CNN with PointRend to improve the mask precision Kirillov et al. (2020).

Detectron2 has been used across various domains, demonstrating its versatility and effectiveness in addressing domain-specific challenges. Butt et al. (2023) discussed the usage of Detectron2 in bullet hole detection and scoring targets in shooting sports, obtaining a mean Average Precision (mAP) value of 66.4%. In the security domain, Wadhwa et al. (2023) used the framework for crowd counting.

Other applications include work by Rashmi et al. (2024), transfer learning was used to fine-tune a Detectron2 model for cephalometric landmark annotation in radiographic analysis, addressing needs in the medical imaging domain. In the agricultural sector, Wang et al. (2023) used Detectron2 for the identification of rapeseed pods and its related attributes. In the manufacturing sector, Anupama et al. (2024) also explored its usage in car parts image segmentation.

The versatility of the Detectron2 framework in various domains, such as security, sports, medical imaging, agriculture and manufacturing. Its versatility prompted the need to evaluate its potential for multi-component meal image analysis, as the accurate segmentation of varied and overlapping food items is essential for improving automated food recognition technologies.

3 Methodology

This research uses the Detectron2 framework to design and implement a multi-stage instance segmentation model for multi-component meal images. The training uses two comprehensive food datasets: the FoodSeg103 and UECFoodPixComplete datasets. This section provides a thorough documentation of the critical aspects of the research methodology, including detailed descriptions of the datasets used, the preprocessing techniques applied in preparation for training, and the augmentation techniques used to enhance the model's capabilities. Additionally, it discusses details of the model configuration, including the model's architecture and the loss functions used to guide the model's learning. Finally, it discusses the training and validation process, including the metrics used to assess the model's performance.

3.1 Dataset

The effectiveness of segmentation models in real-world is reliant on the quality of the datasets used for training and evaluation. For this research, the FoodSeg103 and UEC-FoodPixComplete datasets were selected. The choice to use both datasets was driven by the presence of diverse food categories and combined with pixel-level annotations and real-world meal compositions. These factors, combined with being publicly available, made both datasets the best choice for the research study.

3.1.1 FoodSeg103

The FoodSeg103 was designed specifically for food segmentation tasks, it was created as a subset of a larger FoodSeg154 dataset Wu et al. (2021). The FoodSeg154 includes an additional subset of Asian images and annotations, while the FoodSeg103 dataset focuses on Western-style meals. The FoodSeg103 consists of 7,118 images and 103 ingredient categories with their corresponding segmentation masks. The images were sourced from an existing recipe dataset called Recipe1M Salvador et al. (2017).

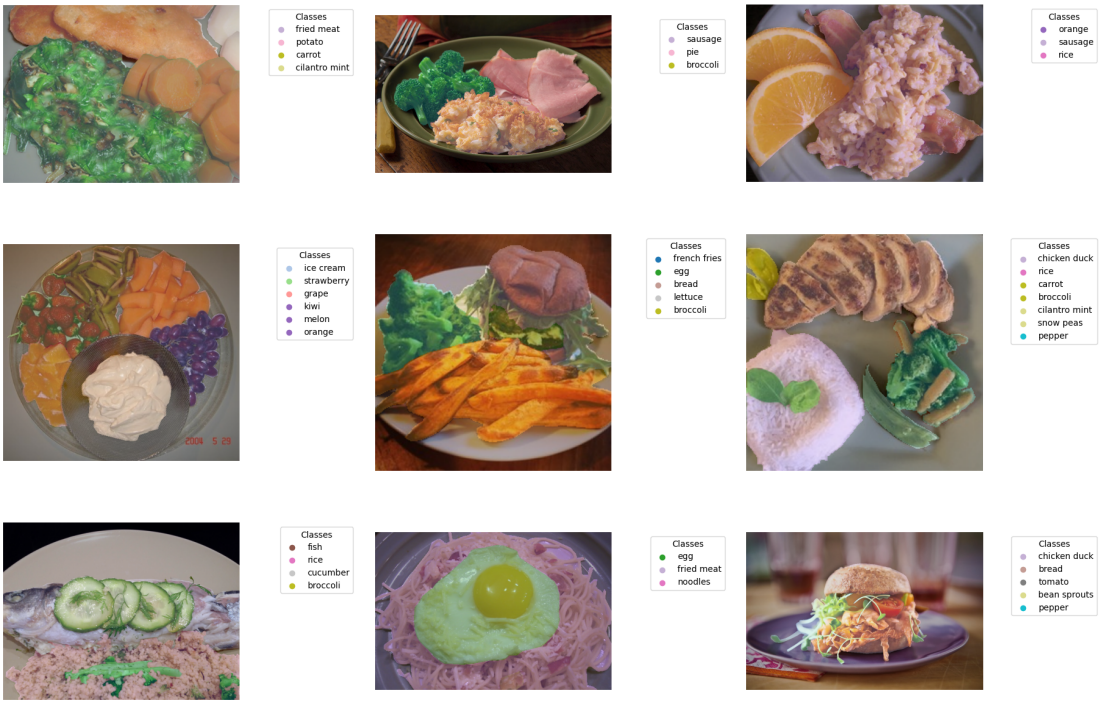

Figure 1: Meal images and individual food components in FoodSeg103

The images in the FoodSeg103 dataset were carefully selected based on the following criteria: 1) Each image should consist of at least two ingredients(from the same or different categories) with a maximum of 16 ingredients; and 2) The ingredients in the image should be clearly visible and easy to annotate.

The creation of the FoodSeg103 dataset involved the extraction of the ingredient categories from the Recipe1M dataset, a dataset with 800,000 food images, with each image paired with ingredient labels and cooking recipes. The refinement of the ingredients categories started with the selection of only the top-124 ingredient categories, before subsequently consolidating the categories into 103 categories.

The annotation process for the dataset was careful and involved professional annotators and researchers to ensure high quality annotations. Each image was labelled by one annotator, it involved the identification of the categories of ingredients in an image by a human annotator, each ingredients are then tagged with the appropriate category label before drawing the pixel-wise mask. The annotators were informed to ignore tiny image regions with areas covering less than 5% of the whole image. After the initial annotation phase, a refinement process that involved correction of mislabelled data, removal of categories present in fewer than five images and the merger of visually similar categories was undertaken. The categories were reduced from the initial 124 categories to 103, after the refinement.

FoodSeg103 captures various scenarios, ranging from images with clear boundaries between the food components to images with overlapping regions and complex compositions.

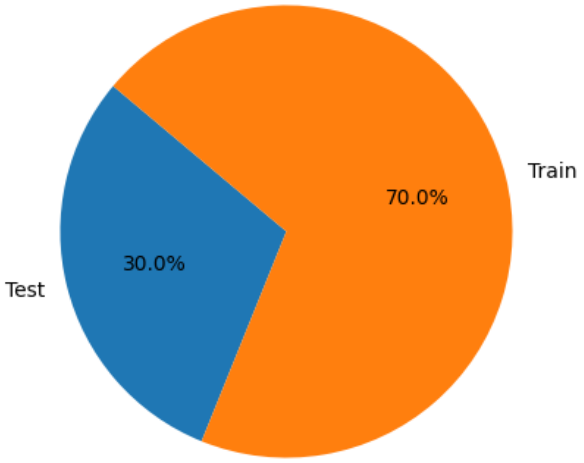

Figure 2: Distribution of Test and Train Datasets - FoodSeg103

The dataset is split into two subsets: the train set and the test set. The training set, which constitutes 70% of the entire dataset, is used for the model training and learning to extract features and learn patterns effectively. The test set, consists 10% of the dataset and serves as the evaluation set, is used to access the model's performance and to test the model's predictive capabilities to new data.

3.1.2 UECFoodPixComplete

The UECFoodPixComplete Okamoto and Yanai (2021) is a refined version of the UEC-FoodPix dataset Ege et al. (2019). Due to being generated automatically from the bounding box annotations, the segmentation masks in the UECFoodPix dataset contained incomplete segmentation masks on the boundaries of some food regions. The UECFood-PixComplete dataset was curated to address the limitations of the UECFoodPix dataset with the addition of high quality and complete segmentation masks. By improving the annotation quality and introducing pixel-wise segmentation, the UECFoodPixComplete dataset offers a better quality dietary image dataset for segmentation research.

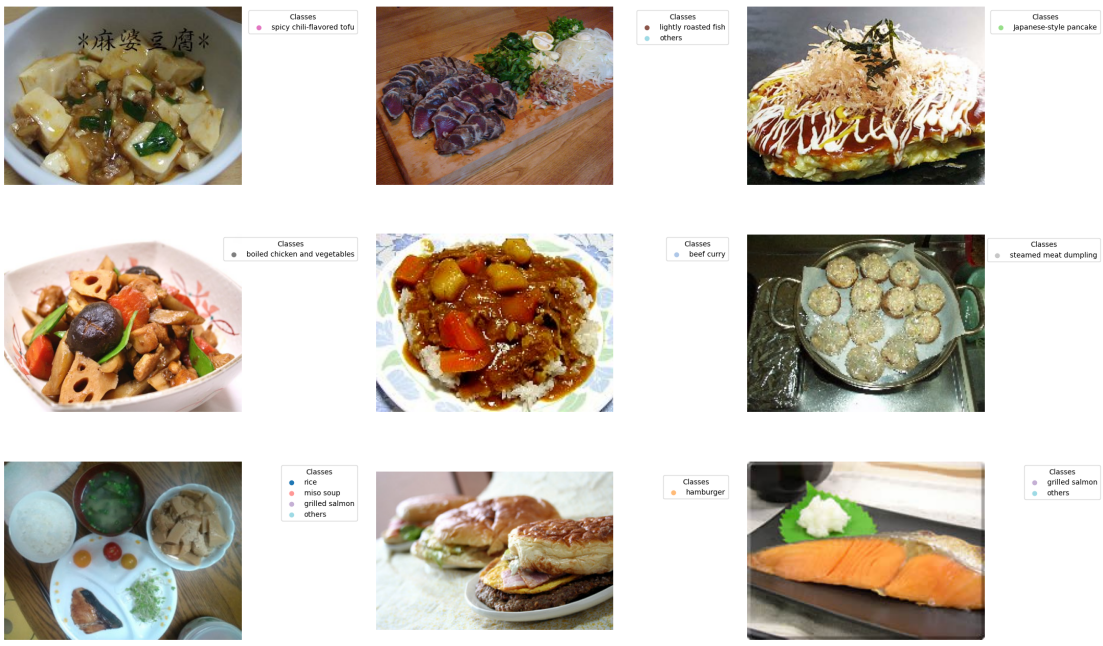

Figure 3: Meal images and individual food components in UECFoodPixComplete

The curation involved several careful steps including automated and manual methods. Initially, a manual approach that involved the use of a web-based pixel-wise annotation tool developed by Tangseng et al. (2017) was used to facilitate the synthesis and separation of food regions using super-pixels. Unlike the initial automated approach, this manual approach allowed for precise boundary refinement and correction of mislabelled regions.

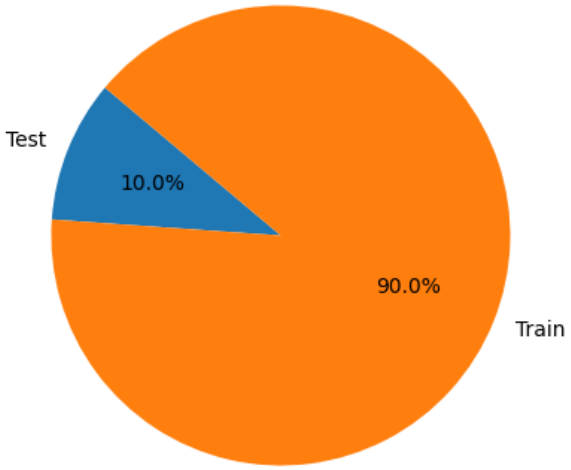

Figure 4: Distribution of Test and Train Datasets - UECFoodPixComplete

The UECFoodPixComplete dataset consists of 10,000 high quality food images with pixel-segmentation masks. The dataset consists of 103 food categories with the categories covering a diverse range of cuisines. It is also split into the train and test sets for effective

model development and evaluation. The train set consists of 90% of the dataset for the model training and learning, while 10% of the dataset is reserved for the test set to evaluate the model's performance.

3.2 Data Preprocessing & Augmentation

To ensure the input data is in the standardised format required for the Detectron2 framework and enhance the diversity and representativeness of the training data, different preprocessing and augmentation techniques were applied to it. Preprocessing ensures that our data is in the appropriate format for easy integration with the framework, while augmentation assists by introducing variability, enabling the model to generalize effectively to unseen conditions.

3.2.1 Data Preprocessing

The data preprocessing stage involved loading the image and their corresponding annotations, extracting information using contours from the segmentation masks, and filtering small or invalid contours. Each instance is encoded with its bounding box, segmentation mask, and category ID, complying with Detectron2's COCO-style dataset format. To accelerate the generation of the training and test COCO-style dataset, parallel processing was used. Finally, the preprocessed data and corresponding metadata were respectively registered in the DatasetCatalog and MetadataCatalog of the Detectron2 framework.

3.2.2 Data Augmentation

Various augmentation techniques were applied to increase the diversity of the model and improve the performance of the model by simulating different real-world scenarios. Firstly, images were resized to various scales while maintaining their aspect ratios, using a method that randomly selects one of several predefined shortest edge lengths. This approach standardises the image dimensions, and also introduces variability to prevent the model from over-fitting to specific image sizes.

Other augmentation techniques used include horizontal flipping to introduce variability in the object orientation, while brightness, contrast, saturation and hue adjustments using colour jittering were used to replicate changes in the lighting conditions and image quality. Random rotations were also used to replicate the slight misalignments that often occur when taking pictures. Random cropping is used to focus on smaller parts of images, helping the model learn to identify objects even when they are only partially visible. Finally, the addition of Gaussian noise helps simulate poor lighting conditions that are found in some real-world images, this would enable the model to handle lower-quality images.

The use of these techniques allows the model to adapt to different food presentations, lighting and environment; improving the model's ability to generalise to unseen data.

3.3 PointRend

To meet the multi-stage segmentation requirements, the instance segmentation model is developed using the Detectron2 framework. Within this framework, the PointRend neural network module is used to improve the precision of the instance segmentation task. For instance segmentation tasks, PointRend can be incorporated with existing instance

segmentation architectures like Mask R-CNN Kirillov et al. (2020). By incorporating PointRend with Mask R-CNN, a mask refinement stage is introduced, allowing the model to segment better, particularly for overlapping and smaller objects, as well as object boundaries and regions with fine details.

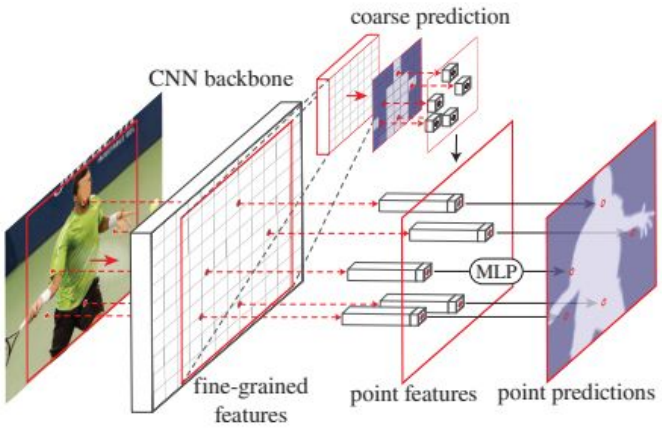

Figure 5: PointRend for Instance Segmentation Kirillov et al. (2020)

3.4 Loss Functions

To address the imbalanced class distributions in the dataset and improve the pixel-level segmentation accuracy, the following loss functions Dice loss, Focal loss and weighted cross-entropy loss were combined.

$$\text{total\_loss} = 0.5 \cdot \text{WCE} + 0.5 \cdot \text{Focal Loss} + \text{Dice Loss} \tag{1}$$

where WCE is the Weighted Cross-Entropy Loss function to address class imbalance, Focal Loss is the Focal Loss function to focus on classes that are difficult to classify, Dice Loss is the Dice Loss function for improve segmentation accuracy.

3.4.1 Dice Loss

Due to the imbalanced distribution of some classes in the dataset, the Dice loss function was in the computation of the total loss to optimise the overlap between the predicted and ground truth segmentation masks, ensuring more accurate segmentation for less-represented food components.

$$DL(y, \hat{p}) = 1 - \frac{2y\hat{p} + 1}{y + \hat{p} + 1} \tag{2}$$

where y is the ground truth binary mask and $\hat{p}$ is the predicted probability mask.

3.4.2 Focal Loss

The focal loss function is also used in the calculation of the total loss. It helps the model focus its learning on classes that are harder to classify and reduces the impact of correctly classified classes, improving the segmentation accuracy for underrepresented classes.

$$FL(p_t) = -\alpha_t(1-p_t)^{\gamma}\log(p_t) \tag{3}$$

where p_t is the predicted probability for the correct class. α_t is the balancing factor to handle class imbalance, set to 0.25. γ is the focusing parameter that reduces the loss for correctly classified examples, set to 2.0.

3.4.3 Weighted Cross-Entropy Loss

The default classification loss function, cross-entropy loss function is modified to account for the class imbalance by assigning different weights to each class. The weights were calculated based on the inverse frequency of the class occurrences in the dataset, to ensure that underrepresented classes were given more significance.

4 Design Specification

The model for this research was developed by fine-tuning a pre-trained PointRend model on the FoodSeg103 and UECFoodPixComplete datasets. PointRend, designed to improve segmentation accuracy, is capable of integrating with various segmentation architectures to perform segmentation tasks. The specific pre-trained model used for this research was configured to extend the Mask R-CNN architecture by adding a mask refinement stage, using a point-based approach with a point-head module.

The pre-trained model uses ResNet-50 as its backbone, ResNet-50 is a deep convolutional neural network with 50 layers. It is designed to allow models learn patterns and extract features from images. It functions as the feature extraction layer by focusing on the extraction of features using details such as colour, shape and edges. In the context of this research work, this layer allows the pre-trained model to learn patterns specific to the various food components in a meal image, allowing it to extract individual food components from the multi-component meal images.

The model also uses a Feature Pyramid Network layer to generate feature maps at different scales. This improves the model's ability to detect and segment the individual food components with varying sizes in the multi-component meal images. Additionally, the model's architecture also features a Region Proposal Network layer used to propose the probable regions with food components in the images, using the feature maps generated by the Feature Pyramid Network layer. The proposals generated by the Regional Proposal Network layer are then refined by an RoIAlign layer.

Subsequently, the model classifies the refined regions, refines the bounding boxes, generates the initial segmentation mask for the image and finally, PointRend refines the generated masks for a more accurate instance segmentation.

5 Implementation

The training involved fine-tuning the pre-trained model for 50,000 iterations with a batch size of 8. The selection of these values was influenced by the need to give the model sufficient time to learn, considering the complexity of the datasets, as well as resource utilisation and to allow for better convergence. The learning rate was set to 0.001 for faster convergence, with step decays applied at iterations 15,000 and 18,000 by a factor of 0.1 to allow for finer adjustments as training progressed. Additionally, early stopping

was used to avoid over-fitting; training was set to stop if the validation loss showed no improvement after 25 consecutive iterations. Some of the model training parameters are show in Table 1 below.

Parameter	**Value**
Batch Size	8
Base Learning Rate	0.001
Maximum Iteration	50,000
Learning Rate Decay	0.1
Learning Rate Steps	15,000 & 18,000

Table 1: Model Training Parameter

The experiments were conducted on Google Colab with the following specifications: Intel(R) Xeon(R) CPU @ 2.20GHz, 83.5GB of RAM, a single NVIDIA Tesla A100 with 40GB of memory, and 235.7 GB of disk space.

6 Evaluation

The metric used to monitor the model's performance for this instance segmentation task is the mean Average Precision(mAP). The mAP metric suits this experiment as it provides a good measure of the model's ability to detect, classify and segment the food components in a multi-component meal image. The mAP combines the precision and recall of the model's predictions to evaluate the model's performance, it is calculated at different Intersection over Union(IoU) thresholds. The IoU measures the overlap between the ground truth masks and predicted masks, the formula is shown below:

$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}} \tag{4}$$

The mAP metric is calculated as the mean of the average precision values at different recall levels for predictions that meet or exceed a series of IoU thresholds, usually ranging from 0.50 and 0.95 in increments of 0.05. The formula for the mAP is shown below:

$$\text{mAP} = \frac{1}{N} \sum_{c=1}^{N} \text{AP}(c), \tag{5}$$

where N is the number of food categories, and $\text{AP}(c)$ is the Average Precision for category c, calculated as:

$$\text{AP}(c) = \frac{1}{T} \sum_{t=1}^{T} \text{AP}_t(c), \tag{6}$$

where T is the number of IoU thresholds, and $\text{AP}_t(c)$ is the Average Precision for category c at IoU threshold t, defined as:

$$\text{AP}_t(c) = \int_0^1 \text{Precision}_t(r)\, dr, \tag{7}$$

with $\text{Precision}_t(r)$ being the precision at recall r for the given IoU threshold t.

6.1 Experiment 1: Detectron2 and FoodSeg103

The results for the experiments with the FoodSeg103 dataset are shown in Table 2. Plots of the total loss and mean average precision across various iterations are also displayed in Figures 6 and 7.

IoU Threshold	**Area**	**mAP%**
0.50	All	28.2
0.75	All	22.4
0.50-0.95	**All**	**20.9**
0.50-0.95	Small	10.0
0.50-0.95	Medium	19.1
0.50-0.95	Large	24.9

Table 2: Average Precision (AP) values at different IoU thresholds and areas

Category	**AP**	**Category**	**AP**	**Category**	**AP**
candy	6.085	egg tart	0.000	french fries	42.583
chocolate	33.626	biscuit	28.825	popcorn	21.124
pudding	0.000	ice cream	37.413	cheese butter	4.804
cake	20.461	wine	28.187	milkshake	34.400
coffee	58.321	juice	35.859	milk	28.504
tea	9.019	almond	16.754	red beans	2.075
cashew	5.631	dried cranberries	13.299	soy	30.249
walnut	24.576	peanut	1.040	egg	16.344
apple	19.721	date	0.000	apricot	2.356
avocado	7.340	banana	49.870	strawberry	52.356
cherry	31.355	blueberry	61.108	raspberry	38.140
mango	1.099	olives	17.080	peach	12.557
lemon	57.874	pear	3.858	fig	1.211
pineapple	16.630	grape	40.019	kiwi	27.332
melon	6.733	orange	36.399	watermelon	1.882
steak	29.649	pork	10.940	chicken duck	24.315
sausage	9.814	fried meat	4.479	lamb	6.188
sauce	36.394	crab	4.785	fish	15.657
shellfish	24.926	shrimp	13.301	soup	31.959
bread	36.456	corn	69.087	hamburg	0.000
pizza	6.955	hanamaki baozi	0.703	wonton dumplings	0.000
pasta	34.522	noodles	26.616	rice	47.060
pie	15.608	tofu	5.451	eggplant	10.835
potato	38.132	garlic	11.246	cauliflower	39.502
tomato	53.605	kelp	0.000	seaweed	10.668
spring onion	3.621	rape	20.832	ginger	3.868
okra	0.000	lettuce	19.850	pumpkin	23.882
cucumber	36.432	white radish	7.052	carrot	58.411
asparagus	30.613	bamboo shoots	0.000	broccoli	66.338
celery stick	44.590	cilantro mint	37.804	snow peas	8.751
cabbage	33.383	bean sprouts	4.311	onion	9.143
pepper	8.994	green beans	44.643	French beans	50.923
king oyster mushroom	0.000	shiitake	5.545	enoki mushroom	0.000
oyster mushroom	0.000	white button mushroom	10.404		

Table 3: Average Precision (AP) values per food category

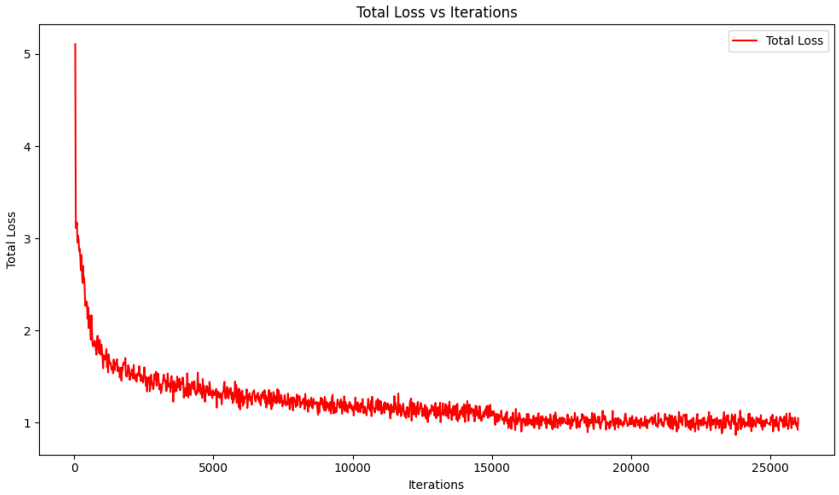

Figure 6: Total loss across various iterations

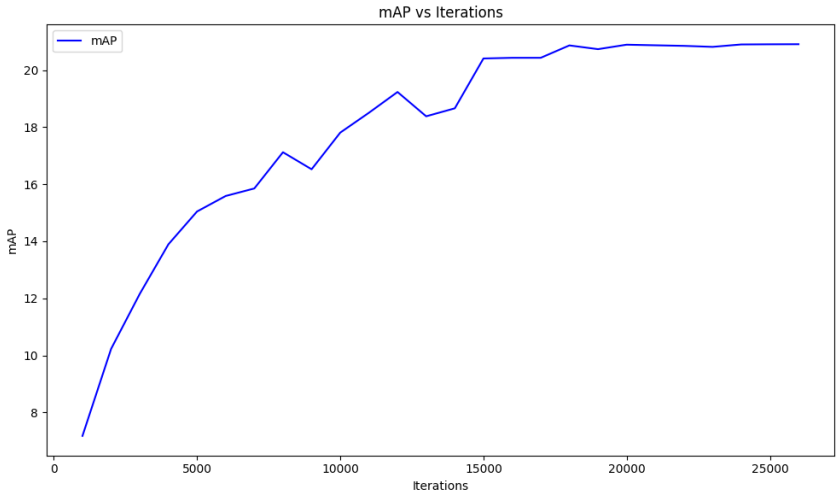

Figure 7: Segmentation mAP across various iterations

6.2 Experiment 2: Detectron2 and UECFoodPixComplete

The results for the experiments with the FoodSeg103 dataset are shown in Table 4. Plots of the total loss and mean average precision across various iterations are also displayed in Figures 8 and 9.

IoU Threshold	**Area**	**mAP%**
0.50	All	44.0
0.75	All	38.9
0.50-0.95	**All**	**36.4**
0.50-0.95	Small	0.7
0.50-0.95	Medium	40.8
0.50-0.95	Large	47.1

Table 4: Average Precision (AP) values at different IoU thresholds and areas

Category	AP	Category	AP	Category	AP
rice	55.619	eels on rice	64.000	pilaf	35.179
chicken-'n'-egg on rice	51.134	pork cutlet on rice	18.371	beef curry	49.678
sushi	24.562	chicken rice	29.162	fried rice	57.056
tempura bowl	35.128	bibimbap	44.890	toast	41.361
croissant	64.585	roll bread	28.784	raisin bread	46.379
chip butty	17.828	hamburger	56.590	pizza	55.557
sandwiches	44.453	udon noodle	66.084	tempura udon	40.077
soba noodle	42.613	ramen noodle	33.228	beef noodle	18.898
tensin noodle	38.724	fried noodle	30.880	spaghetti	26.809
Japanese-style pancake	38.238	takoyaki	15.901	gratin	61.709
sauteed vegetables	19.963	croquette	47.733	grilled eggplant	14.205
sauteed spinach	62.179	vegetable tempura	21.889	miso soup	51.039
potage	57.560	sausage	21.454	oden	29.176
omelet	13.252	ganmodoki	8.366	jiaozi	39.697
stew	50.671	teriyaki grilled fish	26.673	fried fish	22.378
grilled salmon	41.648	salmon meuniere	12.882	sashimi	13.787
grilled pacific saury	33.493	sukiyaki	23.560	sweet and sour pork	56.362
lightly roasted fish	4.985	steamed egg hotchpotch	40.285	tempura	16.807
fried chicken	25.094	sirloin cutlet	24.289	nanbanzuke	22.433
boiled fish	11.372	seasoned beef with potatoes	23.322	hambarg steak	10.144
beef steak	38.117	dried fish	68.941	ginger pork saute	31.605
spicy chili-flavored tofu	52.343	yakitori	45.480	cabbage roll	26.294
rolled omelet	24.980	egg sunny-side up	30.794	fermented soybeans	34.449
cold tofu	16.889	egg roll	44.148	chilled noodle	56.542
stir-fried beef and peppers	33.998	simmered pork	23.156	boiled chicken and vegetables	71.457
sashimi bowl	30.545	sushi bowl	48.628	fish-shaped pancake with bean jam	61.140
shrimp with chill source	40.180	roast chicken	13.391	steamed meat dumpling	43.294
omelet with fried rice	41.415	cutlet curry	9.911	spaghetti meat sauce	48.072
fried shrimp	15.112	potato salad	30.930	green salad	23.324
macaroni salad	37.463	Japanese tofu and vegetable chowder	31.598	pork miso soup	42.410
chinese soup	20.316	beef bowl	58.335	kinpira-style sauteed burdock	25.224
rice ball	11.250	pizza toast	64.615	dipping noodles	57.047
hot dog	67.596	french fries	26.904	mixed rice	62.195
goya chanpuru	39.285	beverage	45.501		

Table 5: Average Precision (AP) values per food category

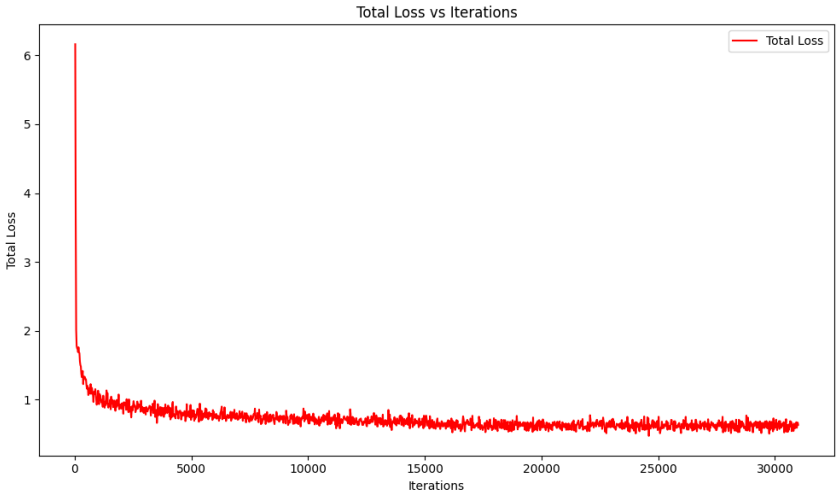

Figure 8: Total loss across various iterations

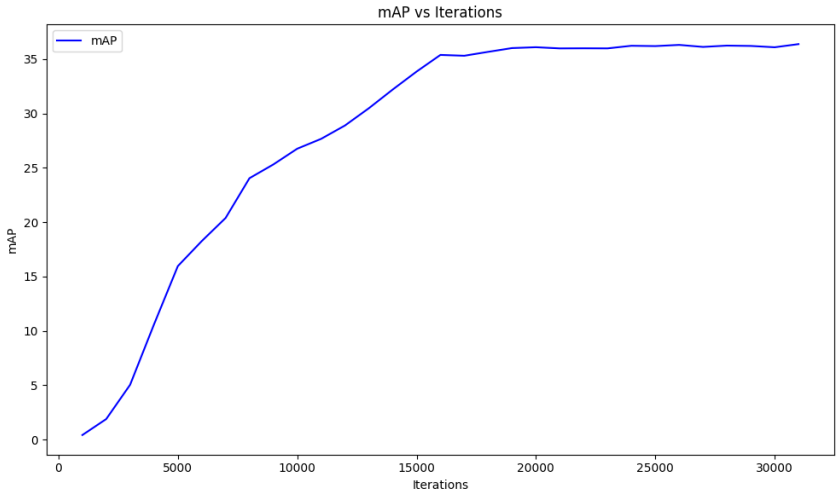

Figure 9: Segmentation mAP across various iterations

6.3 Discussion

The results obtained from the instance segmentation tasks for multi-component meal images using the Detectron2 framework on the FoodSeg103 and UECFOODPIXComplete dataset offered key insights about the effectiveness of the framework.

The model achieved an mAP value of 20.9% across all thresholds with the FoodSeg103 dataset. It also achieved an mAP value of 28.2% at IoU threshold of 0.50 and 22.2% at IoU threshold of 0.75. Comparatively, the model performed better with the UECFOODPIXComplete dataset, as it achieved an mAP value of 36.4% across all thresholds, 44% achieved at IoU threshold of 0.50 and 38.9% achieved at IoU threshold of 0.75. The disparity in the performance of the model with different datasets indicates that the model encountered difficulties with the FoodSeg103 dataset, with the most probable problem being the complexity of the FoodSeg103 dataset, as the meal images in the FoodSeg103 dataset relatively had more overlapping food components and complex compositions.

Furthermore, the model performed better for food components with large areas, as evidenced by the AP values at different sizes. With the FoodSeg103 dataset, an mAP value of 24.9% was achieved for food components with large area, 19.1% for food component with medium area and 10% for food components with small area. A similar pattern was observed with the UECFOODPIXComplete dataset, as the corresponding mAP values for food components with large, medium and small area are 47.1%, 40.8% and 0.7%, respectively. These metrics points to a significant limitation of the Detectron2 framework at handling food components with small areas, a critical factor in food analysis, considering the small size of certain food components.

An analysis of the category-specific average precision values also highlights the varying performance of the model across different food components. In the FoodSeg103 dataset, food components such as corn, broccoli and blueberry had high AP values, while the model failed to detect and segment food components such as egg tart, hamburg and wonton dumplings. Similarly, in the UECFOODPIXComplete dataset, food components such as dried fish, hot dog and udon noodle had high AP values, while food components such as lightly roasted fish and cutlet curry had low AP values. The varying performance across different food components can be attributed to the limitation of the model at detecting and segmenting categories with less distinctive visual features and complex

boundaries, despite the applied data augmentation techniques.

Finally, the results of these experiments highlights the model's limitations in dealing with food components with small area, less distinctive visual features and complex boundaries, leading to potential avenue of model improvement.

7 Conclusion and Future Work

This project focused on evaluating the performance of the Detectron2 framework at the task of instance segmentation for multi-component meal images. To achieve this, a multi-stage segmentation approach with a pre-trained PointRend model was used. The model extends the Mask R-CNN architecture with an additional mask refinement stage and was fine-tuned using the FoodSeg103 AND UECFOODPIXComplete datasets. The mAP and average precision per class metrics were used to evaluate the performance of the model.

The model obtained an mAP of 36.4% with the UECFOODPIXComplete dataset and 20.9% with the FoodSeg103 dataset. These results highlights the challenges and opportunities in the application of the framework to complex food datasets. While the framework showed the potential to become a viable tool at the task of detecting and segmenting multi-component meal images, its effectiveness is poor when handling food components with small areas, complex boundaries and lack visual distinctiveness.

To improve its viability, future work should focus on improvements to address these limitations, approaches such as using existing knowledge of food textures and shapes could significantly improve the performance, especially for challenging food components. These improvements would enable the framework handle the complexities associated with multi-component meal images more effectively and make way for extensive applications in food image analysis.

References

Anupama, M., Chhabra, K., Ghosh, A. and Thavva, R. S. R. (2024). Comparative analysis of deep learning models for car part image segmentation, *International Conference on Data Management, Analytics & Innovation*, Springer, pp. 267–279.

Bolya, D., Zhou, C., Xiao, F. and Lee, Y. J. (2019). Yolact: Real-time instance segmentation, *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 9157–9166.

Bossard, L., Guillaumin, M. and Van Gool, L. (2014). Food-101–mining discriminative components with random forests, *Computer vision–ECCV 2014: 13th European conference, zurich, Switzerland, September 6-12, 2014, proceedings, part VI 13*, Springer, pp. 446–461.

Butt, M., Glas, N., Monsuur, J., Stoop, R. and de Keijzer, A. (2023). Application of yolov8 and detectron2 for bullet hole detection and score calculation from shooting cards, *AI* **5**(1): 72–90.

Chen, K., Pang, J., Wang, J., Xiong, Y., Li, X., Sun, S., Feng, W., Liu, Z., Shi, J., Ouyang, W. et al. (2019). Hybrid task cascade for instance segmentation, *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4974–4983.

CV, A. et al. (2023). Deep learning-based instance segmentation of aircraft in aerial images using detectron2.

Ege, T., Shimoda, W. and Yanai, K. (2019). A new large-scale food image segmentation dataset and its application to food calorie estimation based on grains of rice, *Proceedings of the 5th international workshop on multimedia assisted dietary management*, pp. 82–87.

Gu, W., Bai, S. and Kong, L. (2022). A review on 2d instance segmentation based on deep neural networks, *Image and Vision Computing* **120**: 104401.

He, K., Gkioxari, G., Dollár, P. and Girshick, R. (2017). Mask r-cnn, *Proceedings of the IEEE international conference on computer vision*, pp. 2961–2969.

Huang, Z., Huang, L., Gong, Y., Huang, C. and Wang, X. (2019). Mask scoring r-cnn, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Kirillov, A., He, K., Girshick, R., Rother, C. and Dollar, P. (2019). Panoptic segmentation, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Kirillov, A., Wu, Y., He, K. and Girshick, R. (2020). Pointrend: Image segmentation as rendering, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Lin, F., Li, B., Zhou, W., Li, H. and Lu, Y. (2020). Single-stage instance segmentation, *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* **16**(3): 1–19.

Nguyen, H.-T., Cao, Y., Ngo, C.-W. and Chan, W.-K. (2024). Foodmask: Real-time food instance counting, segmentation and recognition, *Pattern Recognition* **146**: 110017.

Okamoto, K. and Yanai, K. (2021). Uec-foodpix complete: A large-scale food image segmentation dataset, *Pattern Recognition. ICPR International Workshops and Challenges: Virtual Event, January 10–15, 2021, Proceedings, Part V*, Springer, pp. 647–659.

Park, H., Kang, J., Kim, Y. O. and Lee, S. (2015). Automatical cranial suture detection based on thresholding method, *Journal of International Society for Simulation Surgery* **2**(1): 33–39.

Rashmi, S., Srinath, S., Deshmukh, S., Prashanth, S. and Patil, K. (2024). Cephalometric landmark annotation using transfer learning: Detectron2 and yolov8 baselines on a diverse cephalometric image dataset, *Computers in Biology and Medicine* **183**: 109318.

Salvador, A., Hynes, N., Aytar, Y., Marin, J., Ofli, F., Weber, I. and Torralba, A. (2017). Learning cross-modal embeddings for cooking recipes and food images, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Tang, Q., Liu, F., Zhang, T., Jiang, J. and Zhang, Y. (2021). Attention-guided chained context aggregation for semantic segmentation, *Image and Vision Computing* **115**: 104309.

Tangseng, P., Wu, Z. and Yamaguchi, K. (2017). Looking at outfit to parse clothing, *arXiv preprint arXiv:1703.01386* .

Wadhwa, M., Choudhury, T., Raj, G. and Patni, J. C. (2023). Comparison of yolov8 and detectron2 on crowd counting techniques, *2023 7th International Symposium on Innovative Approaches in Smart Technologies (ISAS)*, IEEE, pp. 1–6.

Wang, N., Liu, H., Li, Y., Zhou, W. and Ding, M. (2023). Segmentation and phenotype calculation of rapeseed pods based on yolo v8 and mask r-convolution neural networks, *Plants* **12**(18): 3328.

Wang, Q., Dong, X., Wang, R. and Sun, H. (2022). Swin transformer based pyramid pooling network for food segmentation, *2022 IEEE 2nd International Conference on Software Engineering and Artificial Intelligence (SEAI)*, IEEE, pp. 64–68.

Wang, X., Zhang, R., Kong, T., Li, L. and Shen, C. (2020). Solov2: Dynamic and fast instance segmentation, *Advances in Neural information processing systems* **33**: 17721–17732.

Wu, X., Fu, X., Liu, Y., Lim, E.-P., Hoi, S. C. and Sun, Q. (2021). A large-scale benchmark for food image segmentation, *Proceedings of the 29th ACM international conference on multimedia*, pp. 506–515.