11-2021

GourmetNet: Food Segmentation Using Multi-Scale Waterfall Features With Spatial and Channel Attention

Udit Sharma
us2848@rit.edu

GourmetNet: Food Segmentation Using Multi-Scale Waterfall Features With Spatial and Channel Attention

Udit Sharma

GourmetNet: Food Segmentation Using Multi-Scale Waterfall Features With Spatial and Channel Attention

Udit Sharma

November 2021

A Thesis Submitted
in Partial Fulfillment
of the Requirements for the Degree of
Master of Science
in
Computer Engineering

RIT | Kate Gleason College of Engineering

Department of Computer Engineering

GourmetNet: Food Segmentation Using Multi-Scale Waterfall Features With Spatial and Channel Attention

Udit Sharma

Committee Approval:

Dr. Andreas Savakis, *Advisor* Date
Department of Computer Engineering

Dr. Andres Kwasinski, *Committee Member* Date
Department of Computer Engineering

Dr. Alexander Loui, *Committee Member* Date
Department of Computer Engineering

Acknowledgments

I would like to thank the following people who have supported and guided me during my research.

Firstly, I would like to thank my advisor Dr. Andreas Savakis whose knowledge and insight into the subject steered me through the research. His dedication and love for the subject inspire me. I am grateful to Dr. Andres Kwasinski and Dr. Alexander Loui for providing their guidance and feedback throughout my research and being part of my committee. I would also like to thank Dr. Raymond Ptucha who placed trust in me during my early days and provided me an opportunity to work on the domain that I love. I would also like to express my gratitude towards Bruno Artacho, a Ph.D. student with Dr. Savakis in the Vision and Image Processing lab for sharing his valuable suggestions and recommendations.

My biggest thanks to my parents and sister for always motivating me to keep working. Last but not the least, I would like to thank my friends, Aakash and Ishita, from the bottom of my heart for their unwavering support and encouragement.

To Mom and Dad, without whose sacrifices, this was not possible.

Abstract

Deep learning and Computer vision are extensively used to solve problems in wide range of domains from automotive and manufacturing to healthcare and surveillance. Research in deep learning for food images is mainly limited to food identification and detection. Food segmentation is an important problem as the first step for nutrition monitoring, food volume and calorie estimation. This research is intended to expand the horizons of deep learning and semantic segmentation by proposing a novel single-pass, end-to-end trainable network for food segmentation. Our novel architecture incorporates both channel attention and spatial attention information in an expanded multi-scale feature representation using the WASPv2 module. The refined features will be processed with the advanced multi-scale waterfall module that combines the benefits of cascade filtering and pyramid representations without requiring a separate decoder or postprocessing. The code is made available at: `https://github.com/uditsharma29/GourmetNet`

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction

In the last decade, the application of deep learning and computer vision have grown exponentially to the point that it is used in a range of application domains. Automotive industry is applying computer vision and reinforcement learning to build self-driving cars [1], [2], [3]. Manufacturing plants are using computer vision bots to build critical equipment and eliminate infrastructure faults which might arise due to human errors [4]. Retail giants, Walmart and Amazon are using computer vision, deep learning and 3D reconstruction technologies for self checkout and theft detection [5] [6]. The financial services sector are adopting computer vision to resolve billing disputes and facial recognition to allow users to withdraw money from ATMs. There are diverse applications of computer vision in security and surveillance domain as well where it is being used for facial recognition, speeding vehicle detection and illegal parking detection [7], [8].

Even though the applications of deep learning have a presence in most of the domains, methods for food segmentation are still lagging in development and this thesis aims to advance the state-of-the-art in this field. Existing methods for food analysis primarily focus on food and ingredient recognition. In this fast-paced world, it is difficult to maintain a healthy lifestyle which is causing a number of illnesses.

According to experts [9], it is predicted that 38% of the adults will be overweight and 20% will be obese by 2030. Due to the rising obesity rates, the awareness around diet management and nutrition has increased. Obesity causes chronic illnesses like diabetes and other heart diseases that are getting increasingly common among the younger generation which is alarming. Fortunately, obesity and other diet related illnesses are preventable if we are more aware and make informed decisions using the tools available to maintain a healthy diet.

This thesis presents GourmetNet for semantic segmentation of food images using deep learning techniques. The GourmetNet results outperform the current state-of-the-art and can be used as a reliable input to volume estimation and calorie estimation tasks.

Semantic segmentation is an important computer vision task that has advanced significantly due to deep learning techniques. Food segmentation methods would enable a variety of capabilities including nutrition monitoring [10] [11] [12], food volume estimation [13] [14], calorie estimation [15] [16], ingredient detection [17] [18], recipe generation [19] [20] and quality control of food preparation.

The application of nutrition monitoring using smartphones can significantly benefit from accurate food segmentation by alleviating the user from manually entering food labels and portion size for each meal. In this context, the user takes a picture of the meal and food segmentation model placed underneath automatically detects each food item and provides an estimate of the portion size. This information can be further used to assess the nutritional content of a meal and monitor the nutrition intake of an individual over a time period in order to provide recommendations for dietary improvements for health benefits. This scenario is supportive of the World Health Organization's Sustainable Development Goals (SDGs) to achieve improved nutrition, ensure sustainable consumption patterns, ensure healthy lives and promote well-being for all at all ages.

Food segmentation is a challenging problem due to high intra-class variability and low inter-class variability. A food element can be presented in a widely diverse set of shapes, sizes, colors, and combinations with other ingredients. The examples in first row of Figure 1.1 shows a food class 'Chicken Rice' in the UECFoodPix dataset. The presentation in each of the instances is different from the other making it difficult to find definite and characteristic patterns for a food class. Similarly, we observe low inter-class variability among some classes examples of which are shown in bottom row in Figure 1.1. Images in classes 'Spaghetti' and 'Fried Noodles' are very close in appearance to each other just like the classes 'Croissants' and 'Roll Bread'. Another characteristic of food analysis, is that some food items are routinely paired, allowing the network to infer correlations between the occurrence of different classes.

Figure 1.1: Top row: Examples of high intra-class variability in food images ('Chicken Rice'). Bottom row: Examples of low inter-class variability ('Spaghetti' & 'Fried Noodles' and 'Croissants' & 'Roll Bread'.

We propose a single-stage network for food segmentation, that is end-to-end trainable and generates state-of-the-art results without requiring multiple iterations, intermediate supervision or postprocessing. Our method is inspired by recent advances in multi-scale feature representations [21], [22] and dual attention methods [23] to create a contextual multi-scale framework that improves the pixel-level detection of different foods for segmentation.

The main aspect of our novel architecture is the extraction of both channel attention and spatial attention information for an expanded multi-scale feature representation using the advanced Waterfall Atrous Spatial Pooling (WASPv2) module [22]. The WASPv2 module generates multi-scale representations by increasing the Field-of-View (FOV) for the network while better describing shapes, colors, and textures from images, resulting in a significant improvement in accuracy for food segmentation.

Figure 1.2: Sample GourmetNet results on the UNIMIB2016 dataset

Examples of food segmentation obtained with GourmetNet are shown in Figure 1.2. Our method predicts the location of multiple food classes and performs segmentation of multiple food items based on contextual information due to the multi-scale feature representation. The contextual approach allows our network to include information from the entire image, including all channels and shapes, and consequently

does not require post analysis based on statistical or geometric methods, e.g., there is no need to use the computationally expensive Conditional Random Fields (CRF's).

1.2 Contributions

The main contributions of this thesis are as follows:

- Developed a single-pass, end-to-end trainable, multi-scale approach integrated with channel and attention modules for refinement of features for enhanced contextual learning.
- Proposed an integration of channel and attention modules with waterfall spatial pyramids, which is expected to result in increased performance due to an improved extraction of information combined with the multi-scales approach to produce a larger FOV.
- Performed tests and reported results on three publicly available food segmentation datasets, namely UNIMIB 2016, UEC FoodPix and the FoodSeg103.
- Published the work to MDPI Sensors journal, Nov 2021 edition.

1.3 Document Structure

The rest of the document is structured as follows: Chapter 2 discusses the background and current state of research in neural networks, convolutional neural network for semantic segmentation and food segmentation. Specifically, we discuss the application of multi-scale features and attention mechanisms for image segmentation and how these innovations have helped advance the domain. Chapter 3 discusses the proposed methodology that includes an explanation on the choice of backbone, atrous spatial pooling module and the dual attention mechanisms employed in our architecture. Chapter 4 will elucidate the datasets used and the relevant implementation details

such as hyperparameter tuning, loss functions and evaluation metrics used in our experiments. Chapter 5 discusses the results obtained, its comparison with the current state of the art, class-wise analysis and discussion on failure cases. Finally, we provide a conclusion outlining the key takeaways as well as provide some directions for future work in the domain in Chapter 6.

Chapter 2

Background

2.1 Convolutional Neural Netowrks

Deep learning is a subset of machine learning, which is essentially a neural network with more than two hidden layers. Convolutional Neural Networks (CNN) are a class of neural networks which are inspired by the working of the visual cortex in the human brain and use convolution and pooling operations to extract features from the input. A typical CNN consists of a convolutional layer, pooling layer, activation layer and a few fully connected layers as depicted in Figure 2.1. The convolutional layer applies a filter to the input to create an activation map that summarizes the presence of detected features in the input. The filter slides through the entire image with a prespecified stride and applies convolution on the entire image. Pooling operations are useful in reducing the spatial dimensions of the activation maps by clubbing together a group of pixels in a patch with the summary statistic of those pixel values. Max pooling performs a max operation on the nearby pixels and average pooling takes the mean of the nearby pixels are the most popular pooling operations. Activation layers apply a non-linearity in the output and helps the network to learn more complex representations. Fully connected layers form the last few layers of the network and are basically a simple feed forward neural network. The output from the last pooling layer is flattened and fed to the fully connected layers.

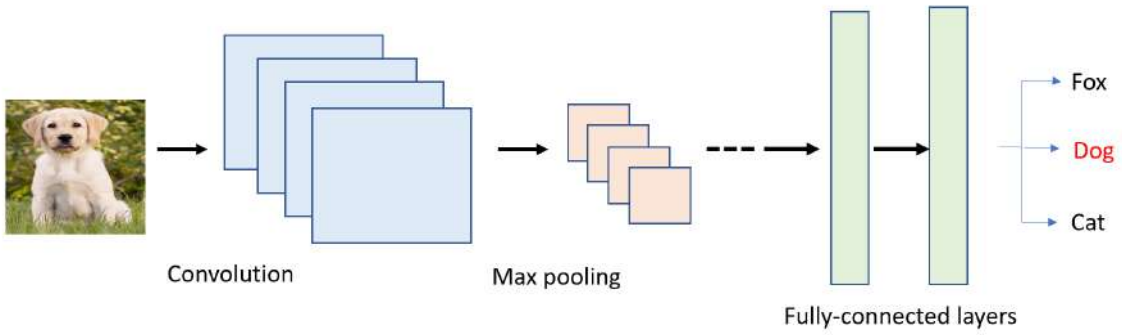

Figure 2.1: A high level diagram showing essential components of a typical Convolutional Neural Network.

CNNs gained massive popularity because of their ability to recognize complex underlying patterns without any prespecified rules. This was complemented by the improvements in computing power and the availability of large datasets. Major breakthrough was achieved in 2012 when AlexNet [24] won the ILSVRC2012 challenge [25] by outperforming the traditional machine learning models by a huge margin. They adopted 5 convolutional layers and 3 fully connected layers, used a fixed input size of 224x224 and trained on ImageNet dataset. This work further led to a surge in interest and belief in CNNs. VGG [26] hypothesize that increasing the number of layers will help the model to learn more complex representations. They build a network with 19 layers and using filters of size 3x3 to achieve 2nd best results in the ILSVRC2014 challenge. Szegedy et. al. [27] introduced the GoogLeNet which is also a Deep CNN consisting of 22 layers. In contrast with the VGG network ,they utilize 1x1 filters and performed global average pooling instead of the fully connected layers. GoogLeNet was the winner of ILSVRC2014 ImageNet challenge. Simply increasing layers further was not improving the results anymore as the network was becoming too large to train and it suffered from the vanishing gradient problem. He et. al. [28] came up with a novel approach using skip connections which allowed the network to skip some layers. The hypothesis is that letting stacked layers learn a residual mapping is easier than

learning the underlying mapping directly. They backed their hypothesis by building and successfully training a ResNet network with upto 152 layers.

The developments discussed till now largely concerns with classification tasks but the applications of CNNs are not limited to classification. They have been adapted to perform other complex tasks such as object detection, semantic segmentation, pose estimation, etc. Since this research is about semantic segmentation on food images, we review the most notable developments in semantic segmentation and food segmentation in the next sections.

2.2 Semantic Segmentation

Researchers have implemented different approaches for semantic segmentation, an overview of which is shown in Figure 2.2. Semantic segmentation methods have improved significantly following the breakthrough introduction of the Deconvolution Network [29] and Fully Convolutional Networks (FCN) [30], where the traditional fully connected layers at the end of the network were replaced by deconvolutional stages, allowing the network to output a higher resolution response, and enabling the high accuracy implementation of the semantic segmentation task.

The U-Net architecture [31] extended the convolution-deconvolution framework by concatenating features from the convolution layers with their counterparts in the deconvolution part of the network. The architecture consists of an expansive path and a contracting path. The contracting path incorporates convolutional layers and is rich in features while the expansive path incorporates deconvolutional layers and concatenates the features and spatial resolution using a series of up-convolution operations and then concatenation with the result from the contracting path. Using an encoder-decoder approach, SegNet [32] used the initial layers of the VGG backbone [26] in the encoder stage with up-sampling deconvolution layers in the decoder stage. SegNet was further developed in [33] to include Bayesian techniques to model uncer-

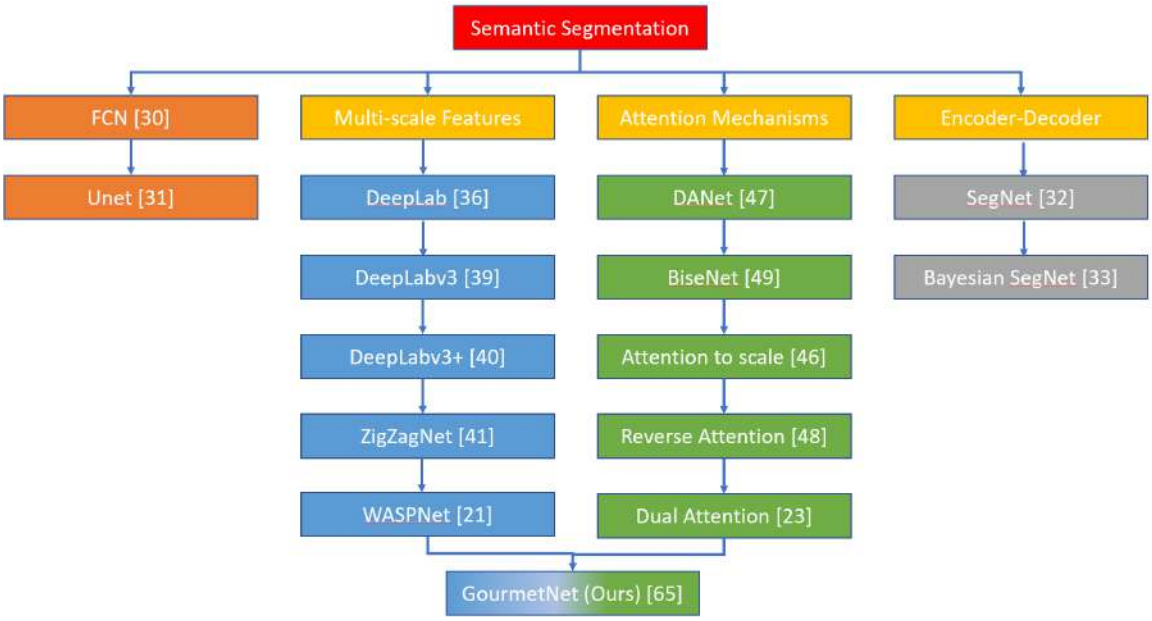

Figure 2.2: Overview of current semantic segmentation methods.

tainty. Aiming to expand the learning context of the network, Pyramid Scene Parsing (PSPnet) [34] combined scene parsing with semantic segmentation. The Efficient Network (ENet) approach [35] sought to develop a real-time semantic segmentation method, resulting in a significant improvement in processing speed compared to other methods.

2.2.1 DeepLab family

DeepLab [36] is a popular architecture that gained a significant improvement in performance by utilizing the Atrous Spatial Pooling Pyramid (ASPP) module which leverages the use of atrous convolutions [37] and Spatial Pyramid Pooling (SPP) [38]. Atrous convolutions, also known as dilated convolutions are a type of convolutions that are used to increase the field of view to incorporate larger context. While performing this type of convolution, we place holes between adjacent elements in the filter which help us to serve two purposes: we have the control over the resolution of the feature maps and we are able to capture a larger context of the image. Figures

2.3 shows the basic difference between a standard convolution and a atrous convolution. In the case of a standard convolution, each output is the linear combination of k adjacent pixels where k is the kernel size. On the other hand, when an atrous convolution is performed with rate r, $r-1$ zeros are inserted between adjacent filter values which effectively enlarges the kernel size of a k×k filter to

$$k_e = k + (k-1) * (r-1) \tag{2.1}$$

There is no increase in the number of parameters or the amount of computation. For example, a 3x3 filter will have the same receptive field as a 5x5 filter while having the same number of parameters as a 3x3 filter.

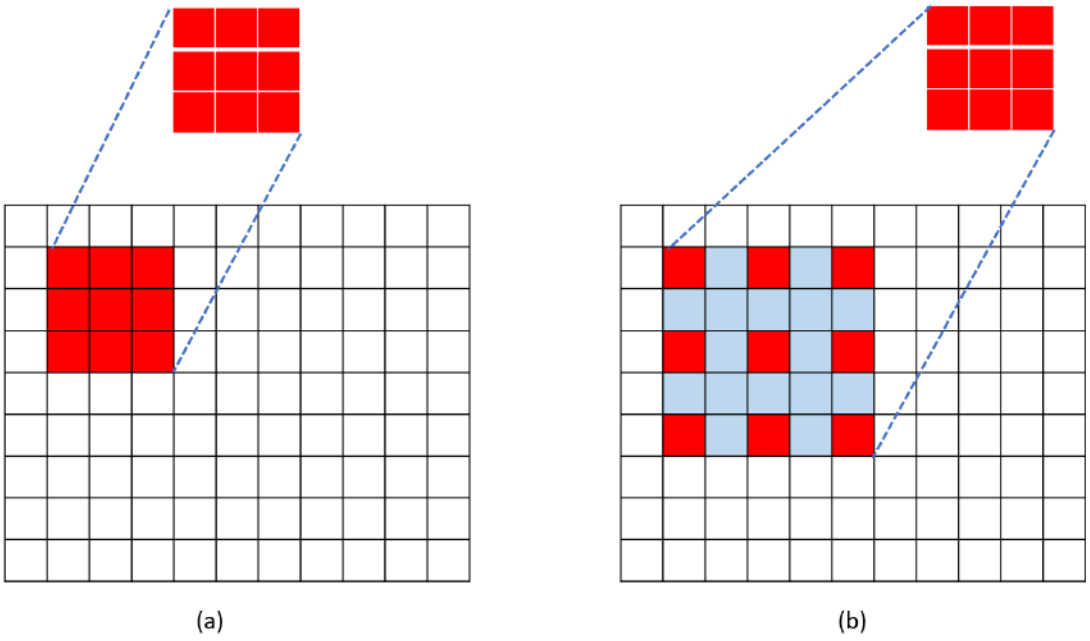

Figure 2.3: (a) Standard convolutions using a 3 x 3 kernel (b) Atrous Convolutions using a 3 x 3 kernel and a dilation rate of 2.

ASPP incorporates branches with different rates of dilation for their convolutions, increasing its field of view and better learning global context. Specifically, the DeepLab [36] architecture uses dilation rates of 6, 12, 18 and 24 with each branch

arranged in parallel as shown in Figure 2.4. The input from the backbone is fed to the 4 branches where atrous convolutions are performed using a 3x3 kernel with different dilation rates. The higher the dilation rate the higher is the field of view that is captured by the branch. Finally, outputs from each branch are concatenated and the resulting output is upsampled to match the input dimensions. DeepLabv3 [39] improved this approach by applying atrous convolutions in a cascade manner, progressively increasing the dilation rates through the layers. This makes the architecture computationally lighter. A further improvement was reported in the DeepLabv3+ [40] which, instead of performing an upsampling operation after the ASPP module, adds a simple but effective decoder to the architecture in DeepLabv3. The DeepLabv3+ also introduced the use of separable convolutions to decrease the computational cost of the network without a significant drop in performance.

2.2.2 Waterfall Multi-scale Features

Building on the research and application of ASPP in DeepLab, Artacho [21] proposed an improvement to the module by proposing the Waterfall Atrous Spatial Pooling (WASP) module. The WASP module, shown in Figure 2.5, leverages the reduced size of cascaded atrous convolutions while maintaining the larger FOV through multi-scale features in the pyramid configuration resulting in the best of both worlds for feature extraction.

Arranging the atrous spatial pooling module in a pyramid fashion addresses the issue of high memory requirement of the parallel configuration by reducing the number of parameters by over 20% while also improving segmentation performance. As shown in Figure 2.5, the input from the backbone is only fed to the atrous convolution block. Instead of passing the features from the backbone, the output of the first atrous convolution block is passed to the next block resulting in the cascaded structure.

An improved version of the WASP module, the WASPv2 module was proposed for

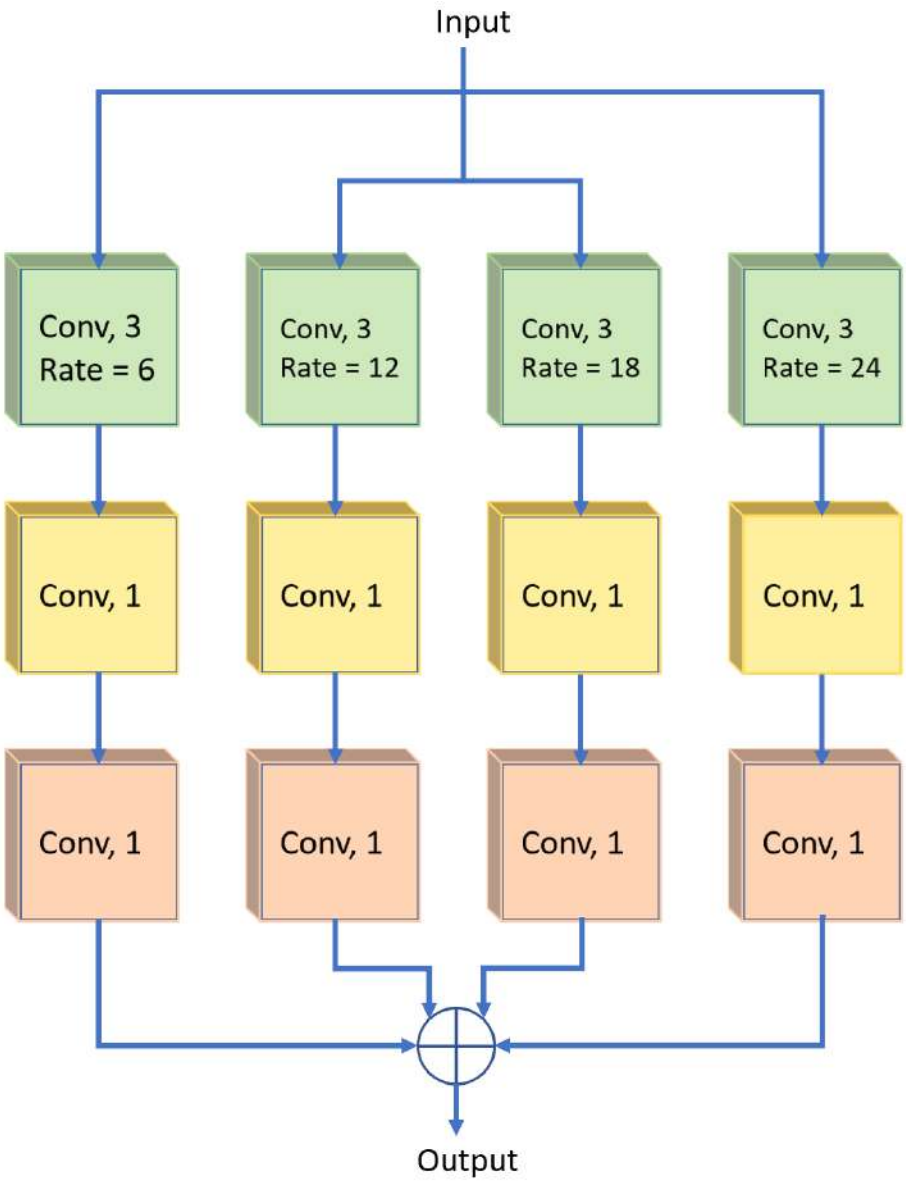

Figure 2.4: Architecture for the Atrous Spatial Pooling Module (ASPP)

the task of multi-person pose estimation in the the OmniPose framework [22]. The novel module combines the learning of the multi-scale features using the waterfall approach while also making use of low-level features from the backbone to embed spatial information, maintain the high resolution throughout its layers. WASPv2 shows increased performance for pose estimation and further reduction in computational cost, presenting promising potential to be applied for semantic segmentation.

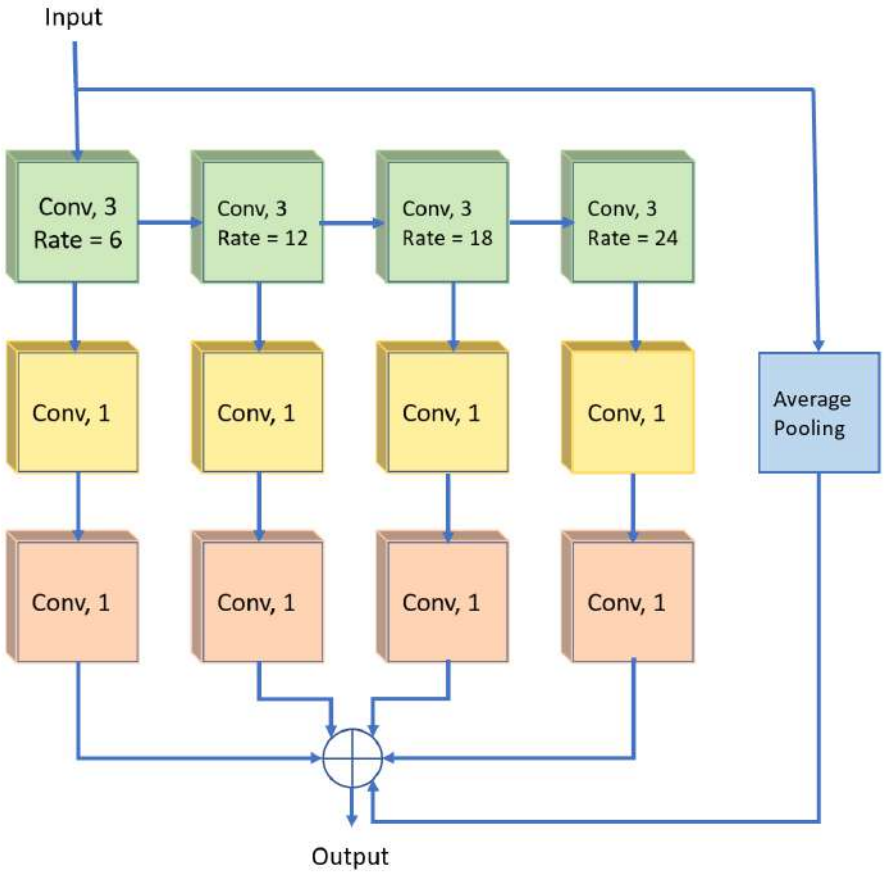

Figure 2.5: Architecture for the Waterfall Atrous Spatial Pooling Module (WASP)

The architecture of WASPv2 is discussed in detail in Chapter 3 as we adopt a modified WASPv2 in our proposed method.

Driven by the success of using multi-scale features, ZigZagNet [41] and ACNet [42] proposed the use of intermediate features combined with high level features from the network backbone, creating a multi-level features context for the decoder.

2.2.3 Attention mechanisms

The use of attention for machine learning tasks was first proposed in [43] and further expanded in [44] by introducing attention to computer vision tasks. The introduction of the transformer model [45] brought a major breakthrough in Natural Language Processing (NLP). The multi-head self-attention layer in the transformer aligns words

in a sequence with others, calculating a representation of the sequence.

The use of attention to improve semantic segmentation methods was explored by [46], training attention heads across scales for semantic segmentation. Similarly, the Dual Attention Network (DANet) [47] uses the channel and spatial attention to improve the network's understanding of the global context for the image, while [48] performs the reverse operation for attention, also aiming to better understand the entire context of the image.

Expanding on attention decoders, BiSeNet [49] fuses two branches for low and high level features bilaterally aiming to construct a real-time approach for segmentation. In similar fashion, the Dual Attention Decoder [23] applies the low-level features to perform its attention module on high level features while creating a channel mask to its low-level features. Our method leverages the promising use of attention decoders to further improve its multi-scale approach.

2.3 Food segmentation

Food segmentation methods were initially developed using image processing and computer vision techniques. Local variation and normalized graph cut [50] were used by [51] to extract the segmentation, while [52] focused its evaluation on the coloring and shape of the food items for its segmentation using JSEG [53]. The biggest challenge in food segmentation and consequently volume estimation are due to the high intra-class variability regarding texture, density, colors, and shapes in food images.

The introduction of deep learning methods has proven to be more effective than rule based methods for food segmentation. Initial applications for food segmentation with deep learning include the mobile application of im2calories [16], having a long list of non-integrated steps for the food segmentation task. It relies on the GoogleNet model [27] to detect instances of food, followed by another GoogleNet trained to detect the food type, and finally performs the classification through DeepLab [36].

Bolanos et al. [54] used the GoogleNet [27] architecture to first predict food and no-food regions and then work on the predicted bounding boxes to classify each bounding box with the food class. CNNs are used to detect the food borders of already identified food items assisted by the growing/merging technique in [55]. Wang et al. [56] employed a graph-based segmentation approach for binary food segmentation, leveraging the class activation map output from VGG-16 as prior knowledge trained on food datasets. Shimoda et al. [57] proposed a new architecture for food image segmentation using food region proposals obtained by selective search and bounding box clustering. This is a region segmentation technique using Region-CNN and does not require pixel-wise annotations. DepthNet [58] accomplishes instance segmentation of food images using Mask R-CNN. The work also provides an extension of their method to incorporate volume estimation.

Besides introducing the UEC Foodpix dataset, [59] also proposed a multi-step approach for food segmentation applying YoloV2 [60] for food detection followed by segmentation using the DeepLabv3 method [**?**].

Slightly increasing the integration of networks and approaching the task of food segmentation, [61] applies an encoder-decoder architecture to perform binary segmentation on food images. The method combines the first 3 layers of the ResNet-101 [28] and a decoder. DeepLab [36] and SegNet [32] architectures are adopted by [62] and [63] respectively to perform semantic segmentation on the UNIMIB 2016 dataset [64].

Chapter 3

Methodolgy

In this section, we provide a detailed description of the attention modules that refine the features from the backbone before passing them to the spatial pooling module for extracting multi-scale representations.

3.1 Proposed Method

The proposed GourmetNet [65] framework, illustrated in Figure 3.1, is a single pass, end-to-end trainable network for food segmentation. Inspired by [23], we introduce attention mechanisms for refining the features from the feature extractor with the multi-scale feature extraction of the WASPv2 module. GourmetNet re-purposes the dual attention module to extract context prior to the multi-scale feature extraction and decoder stage from the WASPv2 module and the spatial pooling modules.

We determine that attention is more useful when it operates on features coming directly from the backbone, as opposed to its application after feature extraction during the spatial pooling modules. This is done because features from the backbone are richer in information and the attention modules have more to work with. Further, GourmetNet combines the improvements in feature representations from WASPv2 and the attention extraction of information from both channel and spatial attention modules.

The processing pipeline of GourmetNet is shown in Figure 3.1. The low-level

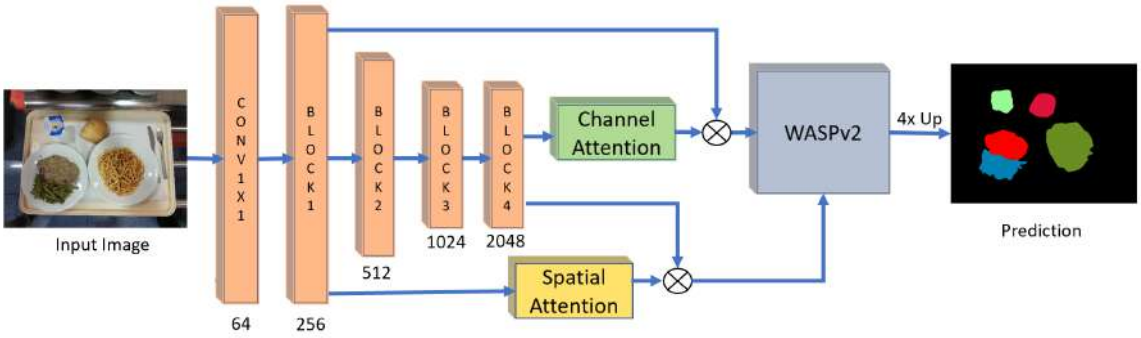

Figure 3.1: The proposed GourmetNet architecture for food segmentation. The input image is fed through a modified ResNet backbone and the features are refined by the spatial and channel attention modules before the multi-scale WASPv2 module which produces the output semantic segmentation result.

features are extracted from the input image through a first block of the modified ResNet feature extractor and includes a dilated last block for the generation of a large FOV. The high-level features are the output of the last block of the modified ResNet feature extractor. The low-level and high-level features are then processed through the attention modules that aim to extract the spatial understanding from the low-level features and richer understanding of channel information from the high-level features.

Low-level features are used to create a mask for refining spatial information because they have a larger resolution by virtue of being early on in the network. Since the low-level features are closer to the actual resolution of the image, they have a better spatial understanding. As we go deeper into the network, the resolution of feature maps is reduced and spatial information is lost. We perform spatial attention to mitigate this issue and to reinforce spatial context into the high level features.

High-level features are the output from the final block of the backbone. They are obtained after 4 blocks of convolutions and pooling operations and hence are rich in information about the objects in the image. Therefore, we leverage the high-level features to create the channel attention mask which is applied on the low-level features.

In conclusion, both the low and high-level features complete each other as low-level features help the high-level features to understand the spatial position of the subject, while high-level features support the low-level features in discerning the features of the subject.

3.1.1 Backbone

We employ the ResNet backbone modified with atrous convolutions as done in [36]. For feature extraction, the first 4 blocks of ResNet-101 are used. However, the last block is modified for multi-scale feature learning. Instead of using regular convolutions, this block uses atrous convolutions. Further, each convolution in this block uses different rates of dilation to capture multi-scale context. The output size of the feature maps is determined by the output stride. For an output stride of s, the output is reduced by s times from the original image. Having a higher output stride affects the quality of dense predictions but reduces the size of the model. For practical reasons, we use an output stride of 16 in our experiments.

3.1.2 Attention Modules

GourmetNet utilizes two attention modules to generate masks and refine the low-level and high-level features extracted from the modified ResNet backbone. The placement of the attention modules in the GourmetNet framework is illustrated in Figure 3.1. The spatial attention branch uses the low-level features from the backbone to create a mask containing spatial information to refine the high-level features prior to the waterfall module. The channel attention branch uses the high-level features to create a mask containing channel information from the feature maps, and applies it to refine the the low-level features.

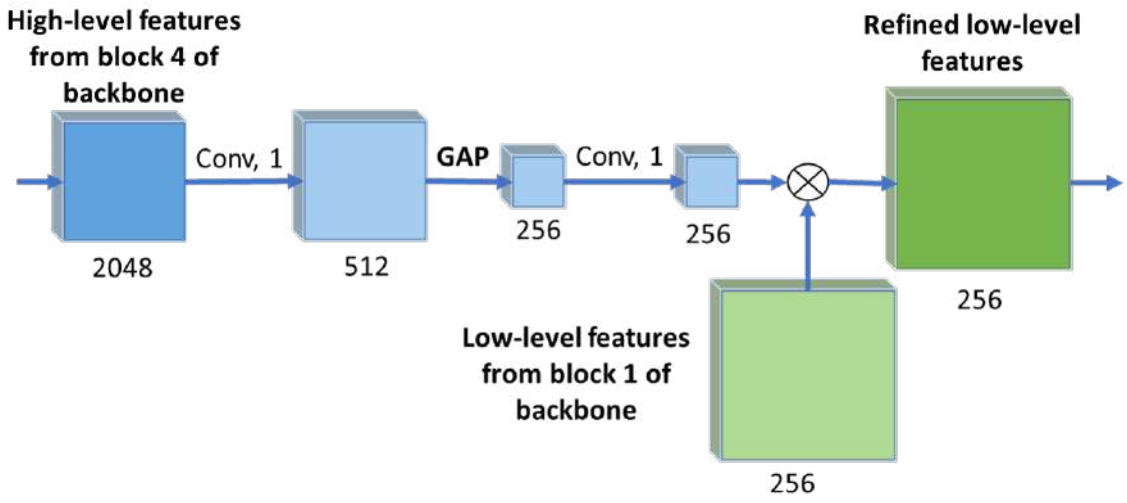

Figure 3.2: Channel attention module architecture.

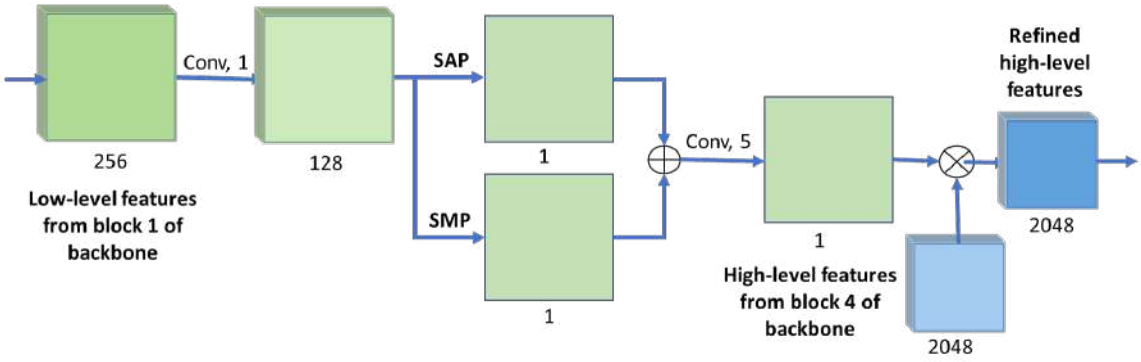

Figure 3.3: Spatial attention module architecture.

3.1.2.1 Channel Attention

Channel attention utilizes high-level features which consist of 2048 feature maps with width and height reduced by a factor of four compared to the original dimensions of the input image. Our modified channel attention module progressively reduces the number of feature maps to 256. These maps produce the channel attention mask used as one of the inputs to the WASPv2 module after pixel-wise multiplication with the low-level features from the backbone.

The channel attention module architecture is shown in Figure 3.2. The 2048

high-level feature maps from the modified ResNet backbone are processed with 1×1 convolutions to reduce the number of feature maps to 512, followed by a global average pooling layer and another 1×1 convolution stage, reducing the number of feature maps to 256. The output of the module is then multiplied pixel-wise with the low-level features from the backbone, producing the refined low-level features with 256 channels. The channel attention module operation can be expressed as follows:

$$f_{rl} = f_l * (K_1 \circledast AP(K_1 \circledast f_h) \tag{3.1}$$

where $\circledast$ represents convolution, f_{rl} represents the refined low-level features, f_l are the low-level features extracted from block 1 of the backbone, $*$ represents element-wise multiplication, K_1 is a kernel of size 1×1, AP denotes Average Pooling, and f_h represents the high-level features extracted from backbone. The dimensions of the channel mask are $1 \times 1 \times c$ where c is the number of channels in the low-level feature space. This mask is broadcast to all the pixels in the low-level feature maps.

3.1.2.2 Spatial Attention

Spatial attention utilizes low-level features that are extracted from the first block of the modified ResNet backbone, by converting features maps into the spatial attention mask. This mask is then used to refine the high-level backbone features using element-wise multiplication.

The spatial attention module is shown in Figure 3.3. It receives the 256 channels of low-level features from the first block of the modified ResNet backbone, and reduces them to 128 channels via 1×1 convolution. This is followed by a set of two parallel pooling operations, one for spatial average pooling (SAP) and one for spatial max pooling (SMP). The outputs of both spatial pooling operations are then concatenated and processed through a 5×5 convolution in order to extract spatial information with

a larger FOV. The output of the module is then multiplied pixel-wise with the high-level features from the backbone, producing the refined high-level features with 2048 channels. The mathematical representation of the spatial attention module can be described as follows:

$$f_{rh} = f_h * (K_5 \circledast (SAP(K_1 \circledast f_l) \oplus SMP(K_1 \circledast f_l))) \tag{3.2}$$

where $\circledast$ represents convolution, f_{rh} represents the refined high-level features, f_h are the high-level features extracted from the backbone, * represents element-wise multiplication, K_1 and K_5 are kernels of size 1×1 and 5×5 respectively, SAP and SMP denote Spatial Average Pooling and Spatial Max pooling operations, respectively, $\oplus$ is a concatenation operation, and f_l represents the low-level features extracted from block 1 of the backbone. The dimensions of the generated spatial mask are $h \times w \times 1$ where h and w are the height and width of the low-level feature maps. The same mask is broadcast across all feature maps in the high-level features space.

3.1.3 Multi-Scale Waterfall Features

Following the refinement of the low-level and high-level features via the attention modules, we perform multi-scale feature extraction and decoding through the WASPv2 module [22]. The WASPv2 module, depicted in Figure 3.4, increases the FOV by applying a set of atrous convolutions with dilation rates of $[1, 6, 12, 18]$ assembled in a waterfall configuration.

The waterfall architecture utilizes progressive filtering in an efficient cascade architecture, while maintaining the multi-scale FOV found in the spatial pyramid configurations. The refined low-level features are concatenated with the high-level features to obtain a multi-scale representation with increased FOV. The final layers with 1×1 convolutions acts as an inbuilt decoder, generating the final segmentation maps for our

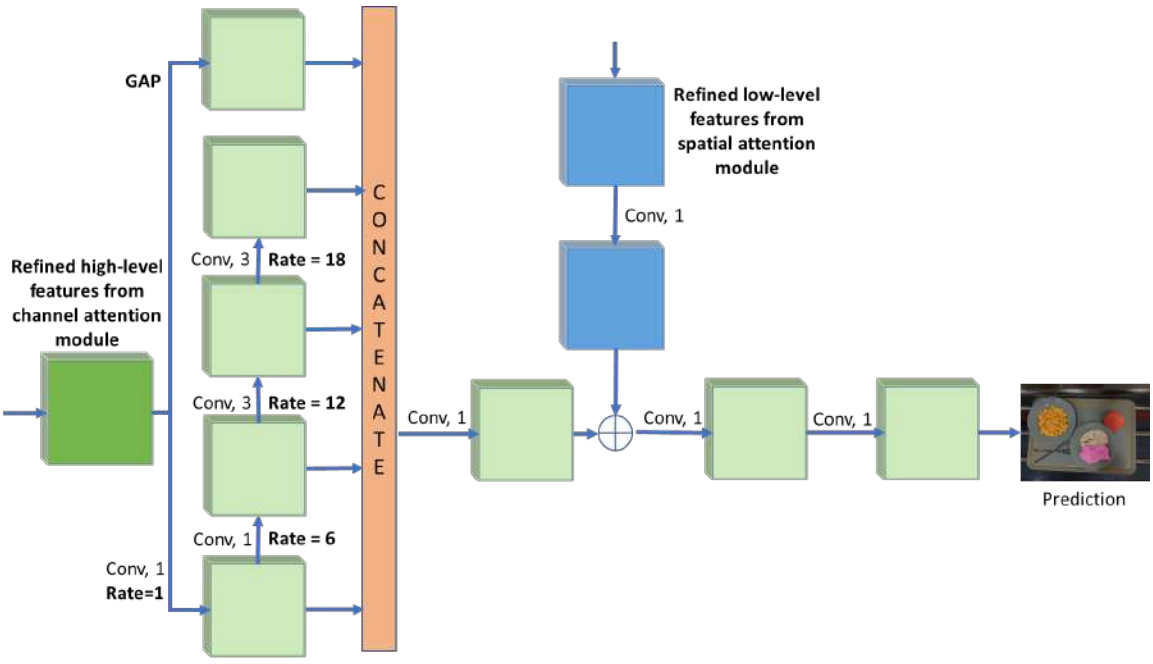

Figure 3.4: The advanced waterfall (WASPv2) module architecture with channel attention and spatial attention refined features.

GourmetNet model without requiring a separate decoder module or postprocessing.

Chapter 4

Implementation Details

This section discusses the datasets we used to perform our experiments, the hyperparameter settings such as learning rate, batch size and decay used in the experiments and the metrics used to evaluate the performance of the model.

4.1 Datasets

We perform food segmentation experiments with GourmetNet on three datasets: the UECFoodPix dataset [59], UNIMIB2016 dataset [64] and the FoodSeg103 [66]

4.1.1 UEC FoodPix

The UEC FoodPix dataset [59] is a large scale dataset for food segmentation. It consists of 9,000 images for training and 1,000 images for testing, labelled with manually annotated masks to segment 102 food categories. Due to its origin and nature, this dataset has frequent occurrences of Japanese dishes. The main challenges of the UEC FoodPix dataset include the presence of multiple food types in the same plate without a significant separation, diverse camera angles, various arrangements of the plates, and variation of the image size. This is a harder dataset as it contains images of food in which there are more than one food items per plate. Sample images and their corresponding ground truth segmentation masks are shown in Figure 4.1. Annotations for the UECFoodPix dataset were generated using a coarse automated tool

and manually refined by the authors.

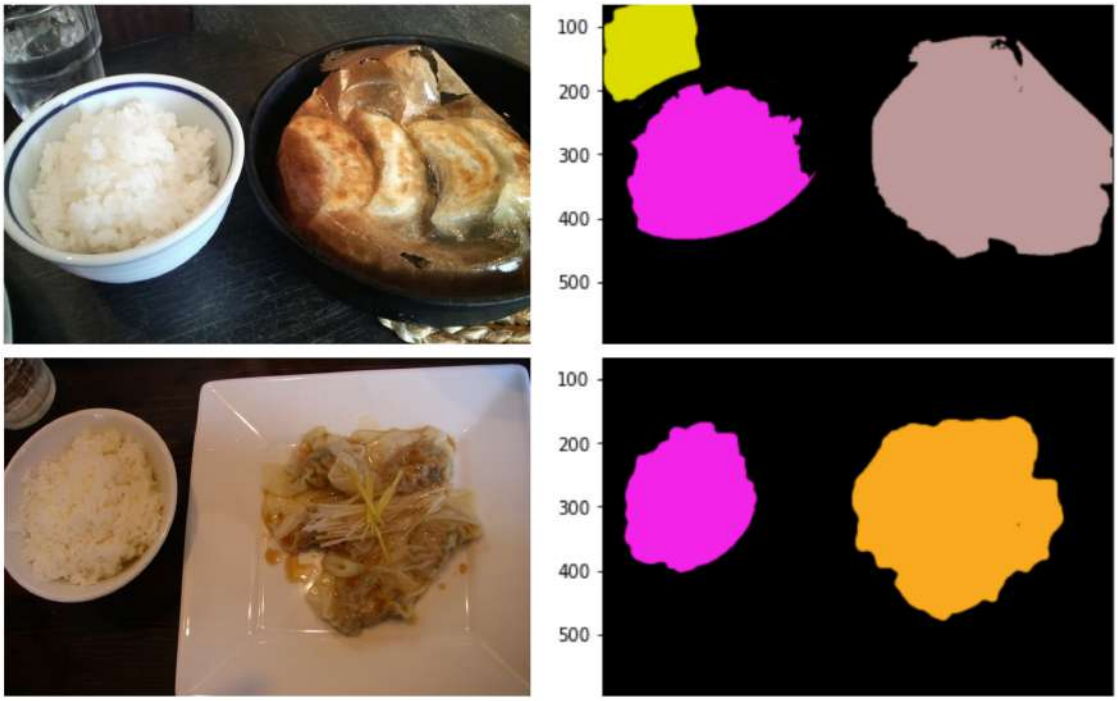

Figure 4.1: Sample images and corresponding annotated segmentation masks in the UEC FoodPix dataset.

4.1.2 UNIMIB 2016

The UNIMIB 2016 dataset [64] is a popular food dataset, especially for the tasks of food classification and recognition. It consists of 1,027 tray images, consists of 73 different food categories with a total of 3,616 food instances. This dataset provides food region information as polygons that can be converted to masks for performing semantic segmentation. Due to its origin and nature, UNIMIB 2016 contains a variety of Western food items, with a large proportion of Italian dishes. The images are shot in a controlled environment of a canteen. Most images contain several plates on a tray with each plate containing one food item. All images are shot from a constant angle and at the same high resolution (3,264×2,448). The dataset is divided into 650 images for training and 360 images for testing. Annotations were created using an automated

tool [67] to generate polygons using the Douglas-Peucker algorithm [68] . A draw back of this method is the more coarse borders resultant from the polygon method. Figure 4.2 shows some example images and the corresponding masks generated from polygon annotations.

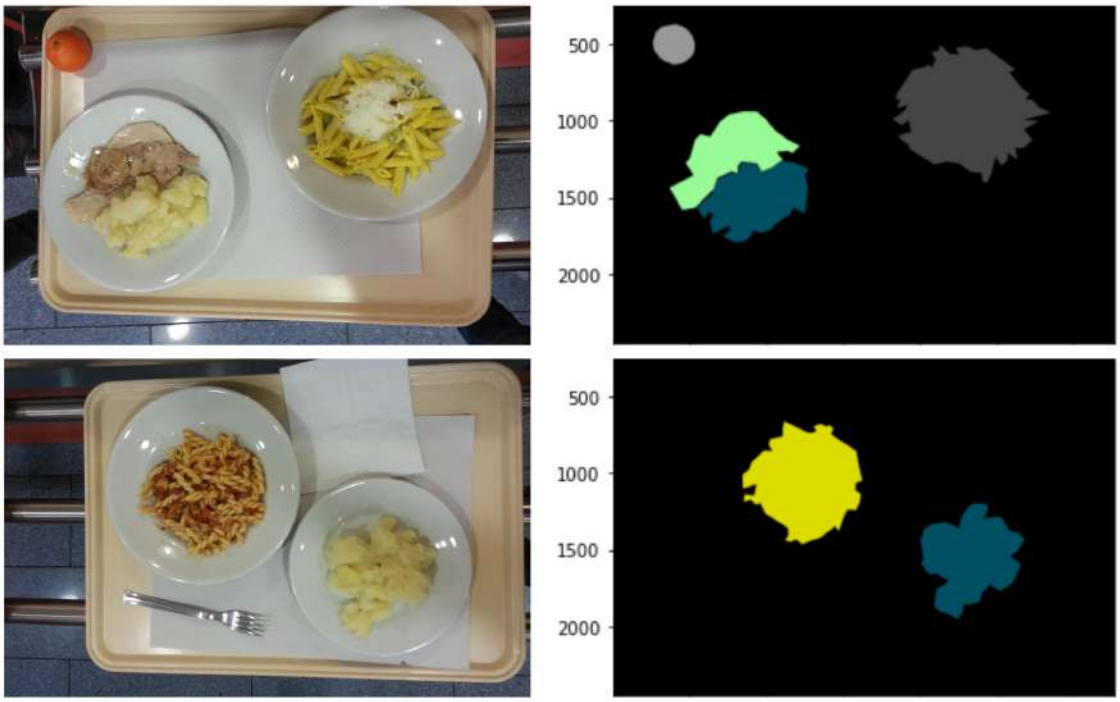

Figure 4.2: Sample images and corresponding annotated segmentation masks in the UNIMIB 2016 dataset.

4.1.3 FoodSeg103

The FoodSeg103 dataset [66] consists of 7,118 images with manual annotations for 103 different ingredients. Each image is annotated with 6 ingredients on average. The images are sourced from an existing recipe dataset called the Recipe1M [20]. The Recipe1M dataset has 900k images and more than 1500 ingredient annotations but most of these ingredients are present in very few images. The authors have taken the most occurring 103 ingredients to build the new dataset. Following conditions are applied to select images: Image should have at least 2 ingredients and

no more than 16 ingredients. Further, the ingredients should be visible and easy to annotate. Contrary to the UEC FoodPix and UNIMIB2016 datasets which provided dish-level annotations, the FoodSeg103 dataset provides ingredient-level annotations. Sample images and their corresponding ground truth segmentation masks from the FoodSeg103 dataset are shown in figure 4.3.

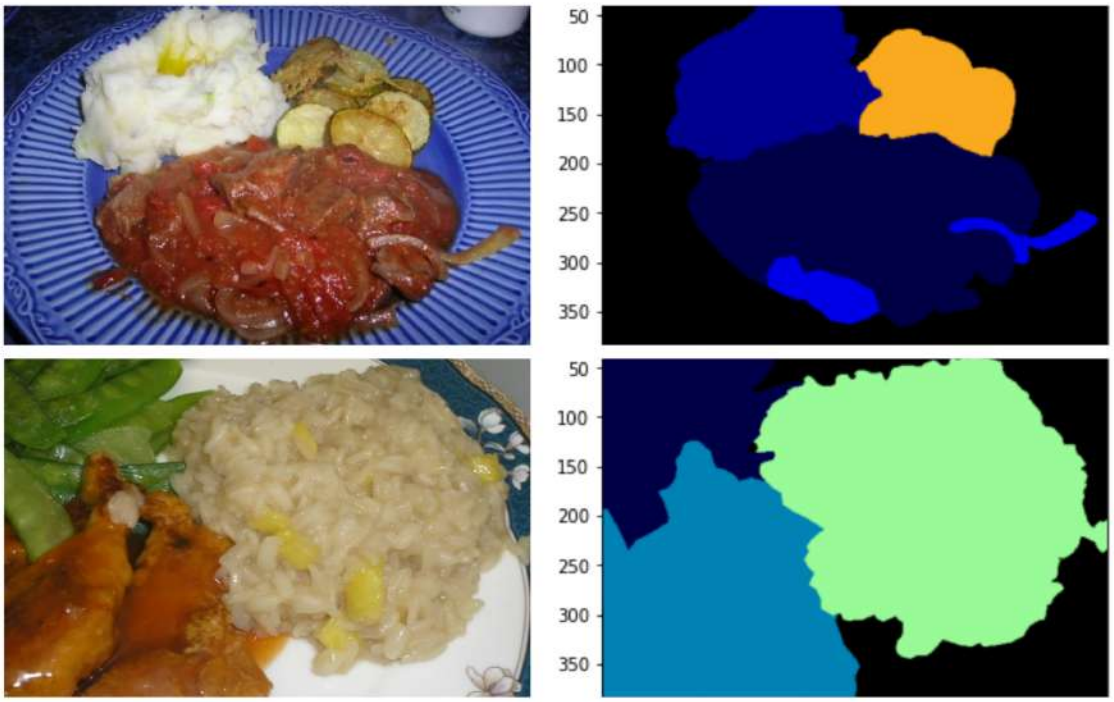

Figure 4.3: Sample images and corresponding annotated segmentation masks in the FoodSeg103 dataset.

4.2 Parameter Setting

We train GourmetNet in all experiments for 100 epochs by applying a batch size of 8. We implement a multi-step learning rate routine with a base learning rate of 10^{-5} and steps of 0.3 at epochs 40 and 70. The model is trained with the Cross-Entropy (CE) loss using the Stochastic Gradient Descent (SGD) optimizer [69]. The weight decay is set to $5 * 10^{-4}$ and momentum to 0.9 [70]. All experiments were performed

using PyTorch on Ubuntu 16.04. The workstation has an Intel i5-2650 2.20GHz CPU with 16GB of RAM and an NVIDIA Tesla V100 GPU.

The experiments are performed with an input size of 320×320 for the UEC Food-Pix [59] dataset, image size of 480×360 for the UNIMIB2016 [64] dataset and a resolution of 512x512 is used for the FoodSeg103 dataset in order to match resolution with prior literature during performance comparisons. Since the code for the dual attention decoder is not publicly available, we have written our own code based on the architecture described in [23].

4.3 Evaluation Metrics

The evaluation of the GourmetNet experiments were based on the Mean Intersection over Union (mIOU), a standard metric used for semantic segmentation. As shown in Figure 4.4 and in the context of our problem, Intersection over Union is the ratio of overlap between the prediction and ground truth and union of prediction and ground truth for each class. The IOU can be mathematically represented as follows:

$$IOU = \frac{TP}{TP + FP + FN} \tag{4.1}$$

where TP, FP and FN represent True Positives, False Positives and False Negatives, respectively. The mIOU is obtained by the simple average score of IoU for all classes and instances in the dataset.

4.4 Loss function - Cross-entropy loss

A loss function in a neural network is used to adjust the weight values after each iteration. We penalize the model for incorrect predictions, guiding it in the right direction in the process. GourmetNet uses the Cross Entropy loss which is a popular loss function for classification problems. Semantic segmentation can be treated as a

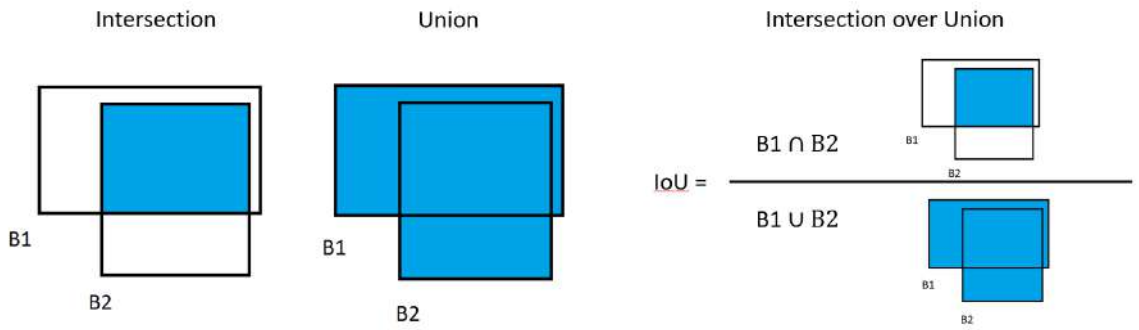

Figure 4.4: Visual representation of the Intersection over Union metric.

classification problem because we classify each pixel in the image to a class. Cross entropy can be calculated for binary classification as shown in the equation below:

$$-(y \log(p) + (1 - y) \log(1 - p)) \tag{4.2}$$

where y is the ground truth label and p is the probability predicted for the class. Cross entropy loss increases as the prediction diverges from the ground truth and decreases when the predicted value is closer to the ground truth value. Figure 4.5 shows the variation of loss with predicted values. As we approach closer to the correct value, the loss decreases slowly. On the other hand, when the prediction is off from the ground truth by a large margin, it penalizes the model heavily.

Semantic segmentation is a multi-class classification problem. Therefore, we need to calculate the cross-entropy for all the class labels and sum them up to get the total loss as represented in the equation below:

$$-\sum_{c=1}^{M} y_{o,c} \log(p_{o,c}) \tag{4.3}$$

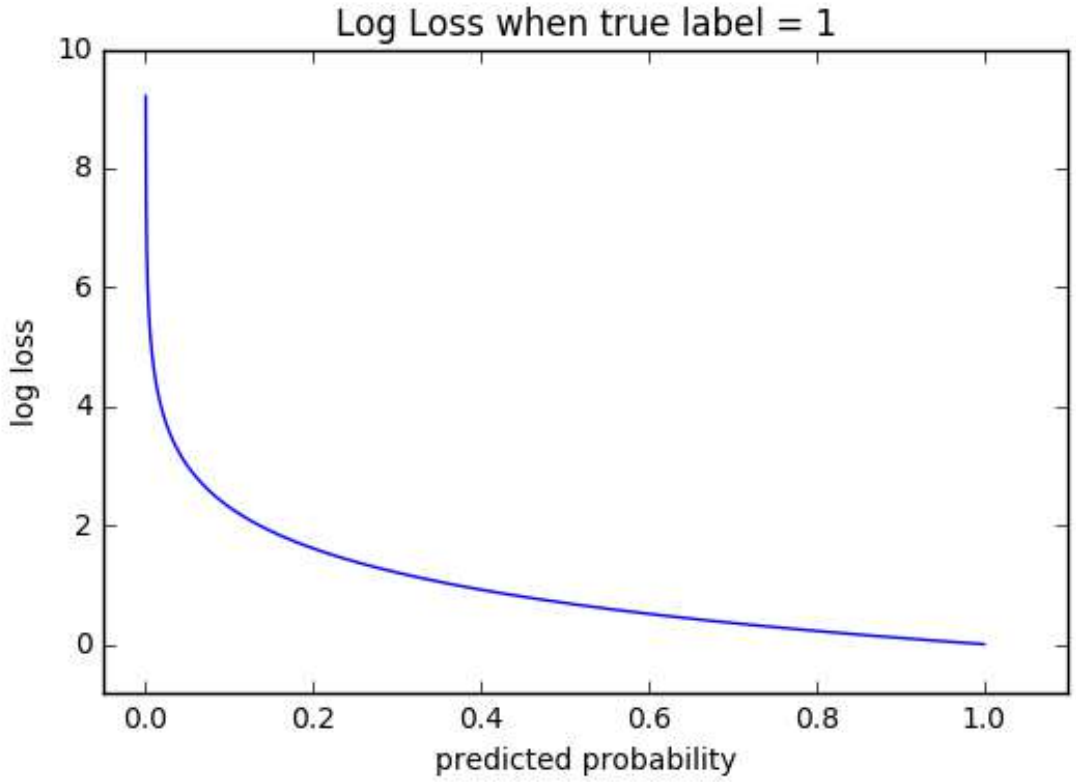

Figure 4.5: Depiction of the relation between Log loss and predicted probability.

Chapter 5

Results

We evaluate GourmetNet on the UEC FoodPix, UNIMIB2016 and the FoodSeg103 datasets, and compare our results with other methods and the current state-of-the-art.

5.1 Ablation Studies

During our experiments, we performed a series of ablation studies to analyze the performance gains due to different components of GourmetNet. Tables 5.1, 5.2 and 5.3 present our ablation results on the UNIMIB2016, UEC FoodPix datasets and FoodSeg103 datasets respectively. In these ablation studies GourmetNet is used with the following options: no module, Dual Attention Decoder [23], ASPP [36], WASP [21], WASPv2 [22], our Channel Attention and Spatial Attention modules and channel and spatial modules coupled with the Dual Attention Decoder [23]. All of the experiments are performed with a modified ResNet-101 backbone for feature extraction.

The results of Table 5.1 show that the mIOU performance of GourmetNet progressively increases with the inclusion of the multi-scale modules and attention modules. The WASPv2 presented the largest gain to the network as a single contribution, increasing the mIOU by 1.6% (from 68.25% to 69.17%). The dual attention decoder results in a 0.8% mIOU increase when added to the network in combination to the WASPv2 module to 70.29%. When individually utilizing our modified channel attention and spatial attention modules in addition to the WASPv2 module, the mIOU

Dual Attention	Channel Attention	Spatial Attention	ASPP	WASP	WASPv2	GFLOPs	#Params	mIOU
						87.20	47.95M	68.25 %
✓						51.56	45.58M	69.44%
✓			✓			54.60	59.41M	69.73%
✓				✓		46.98	47.49M	69.25%
✓					✓	48.81	47M	70.29%
					✓	47.02	46.9M	69.17%
	✓				✓	53.62	48.7M	70.28%
		✓			✓	72	46.9M	70.58%
	✓	✓			✓	78.6	48.8M	**71.79%**
✓	✓	✓			✓	78.6	49M	69.79%

Table 5.1: Results of GourmetNet ablation experiments for various configurations on the UNIMIB2016 dataset. The segmentation accuracy is indicated by the mIOU score, while the model complexity is described by the number of parameters and GFLOPS.

Dual Attention	Channel Attention	Spatial Attention	ASPP	WASP	WASPv2	GFLOPs	#Params	mIOU
						51.33	47.95M	62.33%
✓						30.21	45.58M	62.48%
✓			✓			31.89	59.41M	62.49%
✓				✓		27.47	47.49M	61.95%
✓					✓	28.91	47M	63.14%
					✓	27.5	46.9M	63.54%
	✓				✓	31.4	48.7M	64.30%
		✓			✓	42.3	46.9M	64.29%
	✓	✓			✓	46.2	48.8M	**65.13%**
✓	✓	✓			✓	31.9	49M	63.92%

Table 5.2: Results of GourmetNet ablation experiments for various configurations on the UEC FoodPix dataset. The segmentation accuracy is indicated by the mIOU score, while the model complexity is described by the number of parameters and GFLOPS.

increased to 70.28% and 70.58%, respectively. The most effective configuration was found to be the inclusion of both our modified channel and spatial attention modules in addition to the WASPv2 module, resulting in the highest mIOU of 71.79% for the UNIMIB2016 dataset, a significant increase of 2.06% compared to the results obtained with Dual Attention and ASPP.

Table 5.2 shows the performance of GourmetNet for the UEC FoodPix dataset with the same variations in its components. Consistent to the results for the previous dataset, GourmetNet shows a progressive increase in performance with the addition

Dual Attention	Channel Attention	Spatial Attention	ASPP	WASP	WASPv2	GFLOPs	#Params	mIOU
✓						30.21	45.58M	32.27%
✓			✓			31.89	59.41M	32.6%
✓				✓		27.47	47.49M	32.65%
✓					✓	28.91	47M	33.99%
					✓	27.5	46.9M	33.62%
	✓				✓	31.4	48.7M	34.41%
		✓			✓	42.3	46.9M	34.12%
	✓	✓			✓	46.2	48.8M	**34.66%**

Table 5.3: Results of GourmetNet ablation experiments for various configurations on the FoodSeg103 dataset. The segmentation accuracy is indicated by the mIOU score, while the model complexity is described by the number of parameters and GFLOPS.

of each component. The best results achieve an mIOU of 65.13% when incorporating both Channel and Spatial attention modules in addition to the WASPv2 module.

For completeness, we perform the experiment where we combine both the dual attention decoder [23] and the channel and spatial attention modules in our proposed configuration. This configuration was not optimal, as we observe that the performance diminishes by 1.8% from 65.13% by our proposed architecture to 63.92% for the UEC FoodPix dataset (Table 5.2). In this configuration, we apply attention twice: once before the waterfall module and once in the dual attention decoder. However, the WASPv2 module performs better without the dual attention decoder, as indicated in the results of Table 5.2. A similar observation was made from the results of the UNIMIB 2016 dataset in Table 5.1.

The results of experiments for the FoodSeg103 dataset are shown in Table 5.3. Similar to the other datasets, we observe that the performance increases progressively as multi-scale features and attention mechanisms are added to the network. Our proposed method, leveraging the use of both the attention blocks and the multi-scale features from WASPv2 produces the best performance. We reinforce the effectiveness of the dual attention blocks by proving that using one or none of the blocks achieve an inferior performance.

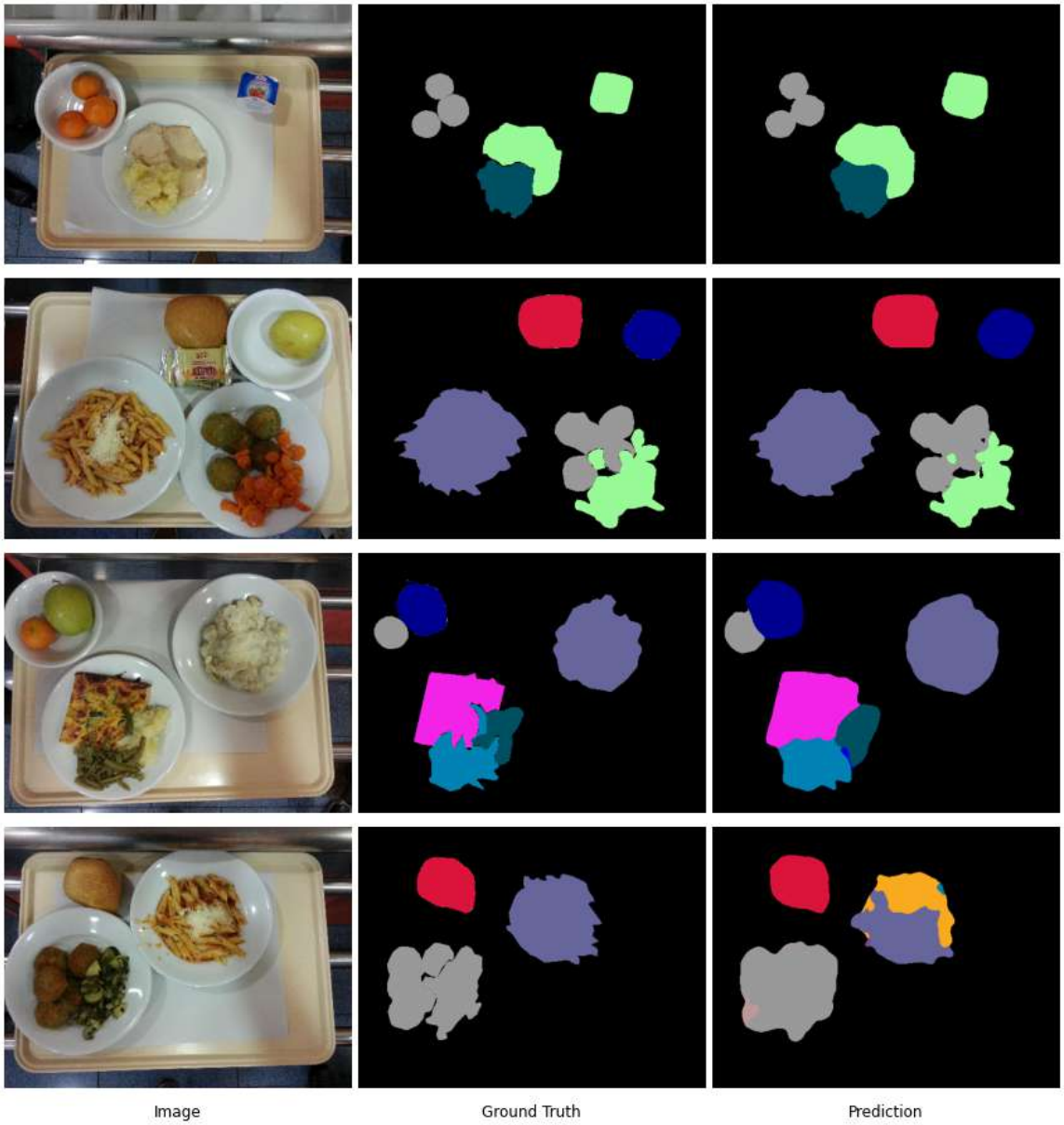

Figure 5.1: Segmentation examples using GourmetNet for the UNIMIB2016 dataset.

Comparing the results for each experiment in Tables 5.1 and 5.2, we observe that the absolute values of mIoU are better in Table 5.1. This is due to two reasons. First, the images in the UECFoodPix dataset may contain more than one food item per plate. This fact makes it harder for the model to differentiate the boundaries of the different food items lying over or beside each other in the same plate. In contrast, the UNIMIB2016 contains only one item per plate in most of the images in the dataset.

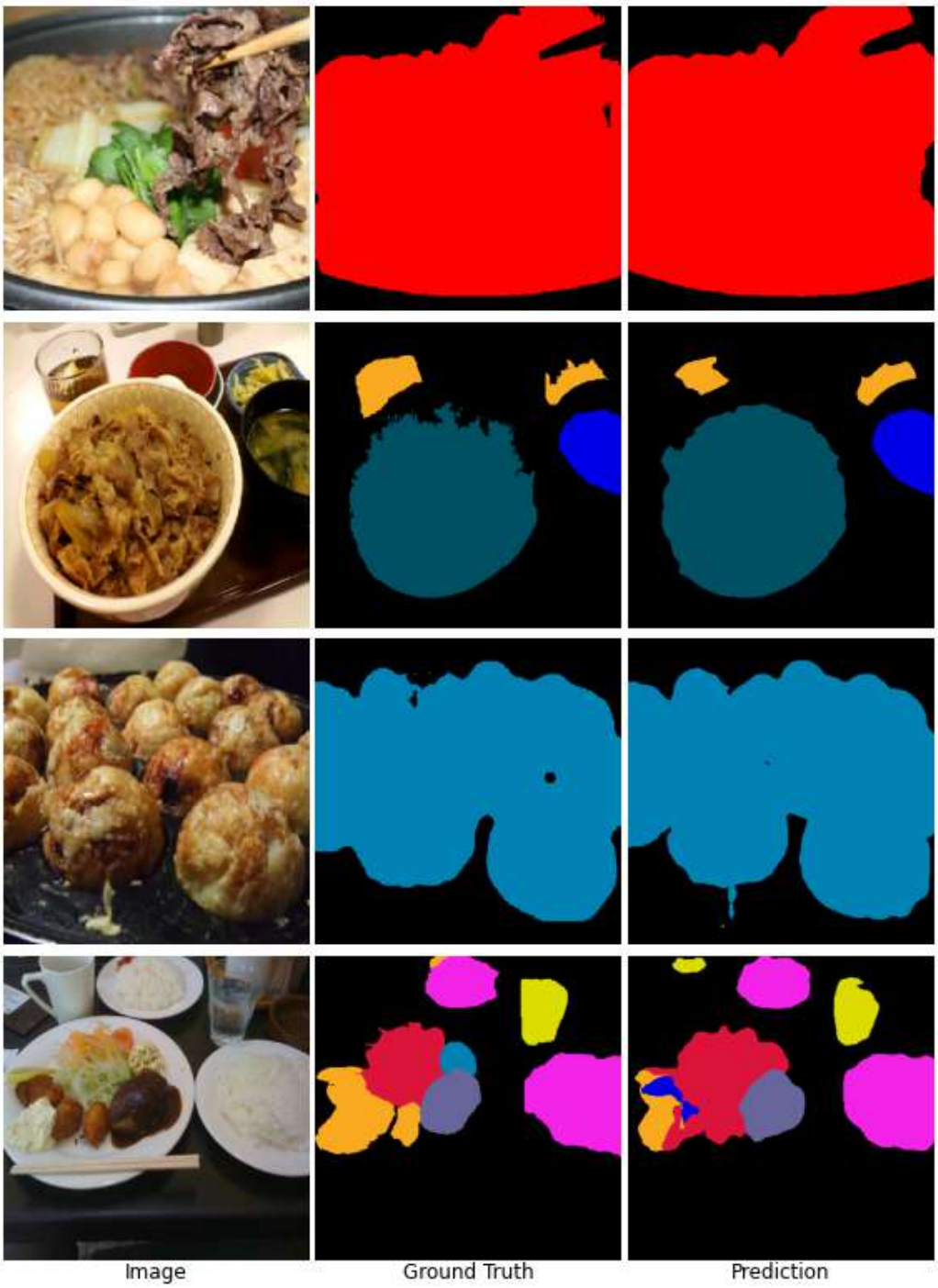

Figure 5.2: Successful examples from the UEC FoodPix dataset. Left column - Images, middle column - Ground truth masks, right column - GourmetNet predictions.

Further the plates are set sufficiently apart which helps the model in recognizing the pattern. Another reason is related to how the dataset is collected. The UNIMIB2016

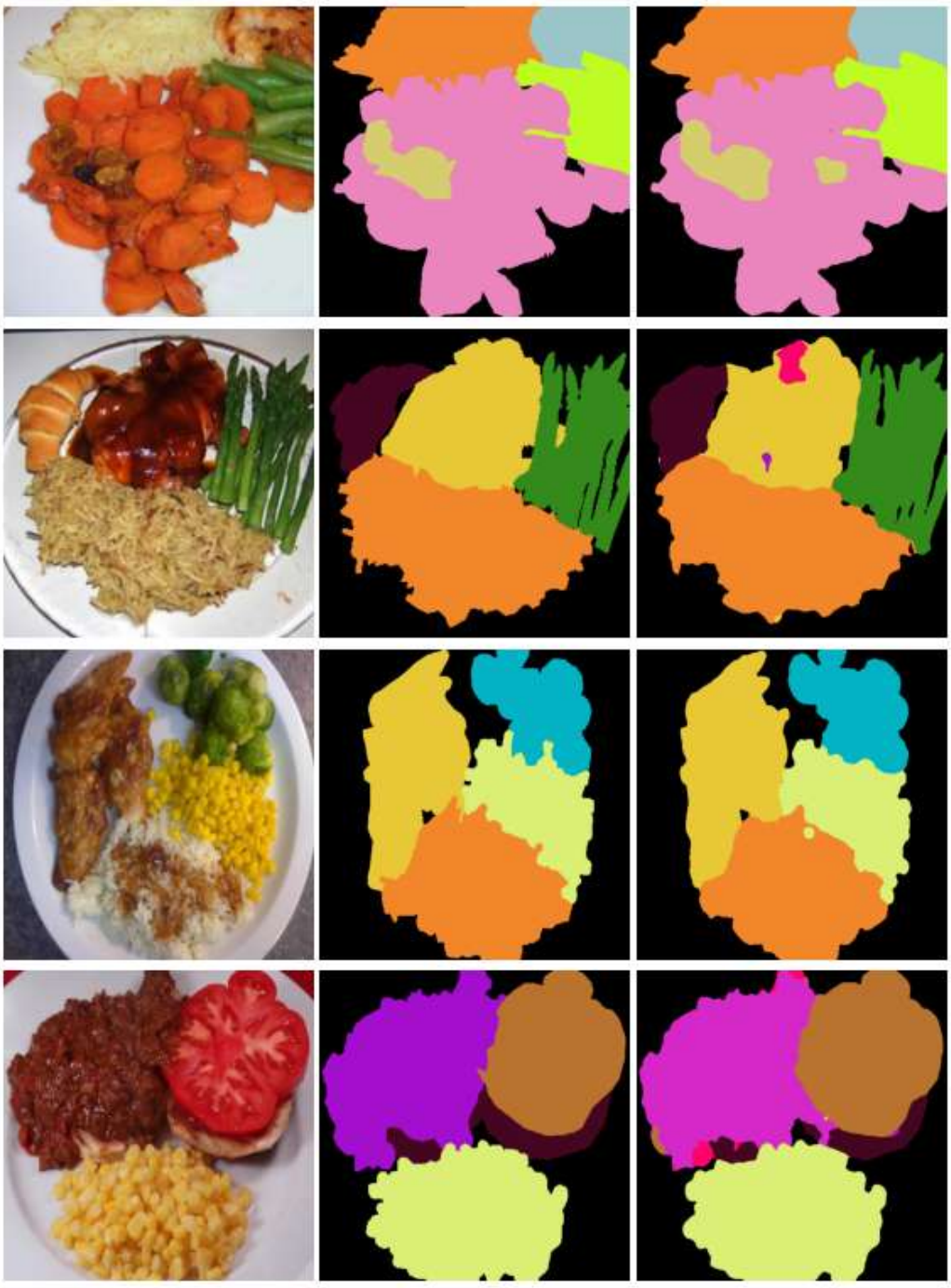

Figure 5.3: Successful examples from the FoodSeg103 dataset. Left column - Images, middle column - Ground truth masks, right column - GourmetNet predictions.

dataset is shot in a controlled environment of a lab. The same camera, lighting, camera angle and background setting is used to capture all images in the dataset

while the UECFoodPix dataset has images with different camera angles, backgrounds, lighting and resolution. These factors introduce necessary variation in the dataset and make it harder for the model to recognize patterns. Finally, the UECFoodPix dataset consists of 102 food classes against 73 in the UNIMIB2016 dataset which results in a higher chance of the model getting confused between classes.

Similarly, the results in Table 5.3 are numerically lower than in Tables 5.1 and 5.2 because the FoodSeg103 dataset contains ingredient-level annotations as opposed to dish-level annotations in the UNIMIB2016 and the UECFoodPix datasets. It is harder to recognize and localize each ingredient from the dish than the dish itself. Figure 5.3 displays some of the successful segmentation results on the FoodSeg103 dataset. We observe that our trained mode is good at recognizing food boundaries between classes and is able to identify some irregular shapes well.

To assess the GourmetNet model complexity, we present the GFLOPS and number of parameters for each configuration. These results show that the top performing WASPv2 module requires fewer parameters and is more computationally efficient than the popular ASPP architecture. The addition of the channel and spatial attention modules slightly increases the number of parameters but significantly increases the computational load.

5.2 Comparison to State-of-the-art

Following our ablation studies, we compared our GourmetNet method with the current state-of-the-art for food segmentation, when results were available. We also included results using top performing methods for semantic segmentation, such as DeepLabv3+ and WASPnet. The IOU results obtained for the UNIMIB2016 dataset are shown in Table 5.4. GourmetNet achieves top performance, showing significant mIOU gains in comparison to other methods. For the UNIMIB 2016 dataset, GourmetNet achieves 71.79% mIOU, compared to 68.87% achieved by DeepLabv3+,

which is a 4.2% improvement. It is important to note that References [62] and [63] do not provide details about their experimental settings while the training for [21] and [40] were performed from scratch using the code available online.

Method	mIOU
DeepLab [62]	43.3%
SegNet [63]	44%
WASPnet [21]	67.50%
DeepLabv3+ [40]	68.87%
GourmetNet (Ours)	**71.79**%

Table 5.4: GourmetNet results and comparison with SOTA methods for the UNIMIB2016 dataset.

Example results for the UNIMIB2016 dataset are shown in Figure 5.1. These examples illustrate that GourmetNet successfully identifies the location of food groups with accuracy for challenging scenarios including food items that share irregular borders and shapes. Challenging conditions include the detection of food items that overlap but are described by a single segmentation mask, e.g., pasta containing grated cheese on it.

Method	mIOU
UEC FoodPix [59]	55.55%
DeepLabv3+ [40]	61.54%
WASPnet [21]	62.09%
GourmetNet (Ours)	**65.13**%

Table 5.5: GourmetNet results and comparison with SOTA methods for the UEC FoodPix dataset.

We next performed testing on the UEC FoodPix dataset, which is more challenging due to occurrences of multiple food items in proximity, different angles, and different resolutions for training and testing images. The mIOU results are shown in Table 5.5. GourmetNet outperforms the current state-of-the-art achieving 65.13% mIOU, a significant performance increase of 5.8% compared to DeepLabv3+ and 17.2% compared to the dataset baseline set by [59]. Similar to the results for the UNIMIB dataset in

Table 5.4, the training for [21] and [40] were performed from scratch using the code available online. The examples in Figure 5.2 demonstrate successful segmentations for the UEC FoodPix dataset. These examples show that GourmetNet deals effectively with food accuracy, localization, and shape.

Method	mIOU
CCNet [71]	35.5%
FPN [72]	27.8%
SeTR [73]	41.3%
GourmetNet (Ours)	**34.7**%

Table 5.6: GourmetNet results and comparison with SOTA methods for the FoodSeg103 dataset.

State-of-the-art results on the FoodSeg103 dataset are presented in Table 5.6. GourmetNet outperforms the results from the FPN [72] but was unable to match the performance with the SeTR [73]. This was due to two reasons: other works employ the processing of recipe information as additional parameters for the semantic segmentation. Secondly, they use a larger resolution and 8 times more computational resources than this research.

5.3 Food Classes Performance Analysis

Table 5.7 lists the performance of GourmetNet for different food classes at both ends of the performance spectrum for the UEC FoodPix dataset. Food items that present constant shape and color, that are displayed with separation from other items, present a more solid consistency and achieve a higher mIOU from the GourmetNet model. Examples of classes containing these characteristics are croquette and pancakes. Another important factor for high accuracy is the fact that the class is visually distinct from the other classes, i.e. udon noodle and goya chanpuru. Food classes that are routinely served in a separate bowl, such as mixed rice, also achieve a high mIOU score.

Food name	mIOU	Food name	mIoU
Croquette	92.16%	Fried Fish	16.29%
Pancake	91.67%	Tempura	17.46%
Udon Noodle	88.67%	Vegetable Tempura	18.23%
Goya Chanpuru	88.61%	Salmon Meuniere	30.28%
Mixed Rice	87.54%	Chip Butty	31.03%

Table 5.7: Comparison and analysis of food segmentation performance class-wise for the UEC FoodPix dataset. The left section mentions classes with the highest mIOU while the right section mentions the classes with the lowest mIOU.

On the low performing side of Table 5.7, classes that present food items in close proximity to other food items have the lowest scores. For example, fried fish has a significant overlap and cross-error with other fried food items. A similar cross-error is observed for tempura and vegetable tempura, as well as chip butty being more routinely mistaken with other types of chips from the dataset. Another source of error is the presence of sauces or garnishing, altering the shape and color of the food item, and consequently increasing its variability. One example of this occurrence is salmon meunière.

Food name	mIOU	Food name	mIoU
Broccoli	85.79%	Eggplant	4.25%
Corn	81.64%	Cashew	6.22%
Green beans	80.56%	Cheese Butter	7.28%
Carrot	79.74%	Crab	7.6%
Strawberry	77.51%	Red beans	13.52%

Table 5.8: Comparison and analysis of food segmentation performance class-wise for the FoodSeg103 dataset. The left section mentions classes with the highest mIOU while the right section mentions the classes with the lowest mIOU.

It is interesting to analyze the class-wise performance for the FoodSeg103 dataset since it is an ingredient dataset. The best performing classes are shown on the left side of Table 5.8 while the worst performing classes are shown on the right side of the same table. 'Broccoli' and 'Strawberry', owing to their distinct shape, color and low variance in appearance, were the easiest to recognize while we observe that green beans and carrots were served as sides and cut roughly in the same way in most of

the images. 'Corn' is served in 2 different ways: the entire corn or boiled corn kernels served as sides. The model is able to identify these variations.

The model struggles to identify some ingredients like 'Cashew' due to their size and their nature of being used as an ingredient in dishes making them harder to recognize. 'Cheese butter' has a high intra-class variability as it is used in different forms based on the type of cuisine. 'Eggplant' has a lower occurrence in the training set which is not enough to recognize the variance of the class.

5.4 Failure cases

While GourmetNet produces state-of-the-art results for food segmentation, just like any other deep learning model, it has its limitations mainly owing to the challenging problem of food segmentation. Food segmentation is challenging problem due to different food types overlapping and are placed in close proximity with different items composing a single dish, e.g., a bowl of soup containing vegetables and tofu in its broth. Figure 5.4 shows some instances when our method did not too well. The boundaries of different ingredients in the burger (1st row) were captured correctly but there was an error in guessing the class correctly. The model guessed the filling as 'steak' instead of 'pork'. This confusion reinforces our hypotheses of low intra-class variablity among food classes. Low occurrence of some classes in the training set caused the model to miss the 'candy' in the 2nd row. Occlusions are one of the common reasons for classifications in food segmentation. This is illustrated in last row of Figure 5.4 where the model is struggling to draw boundaries between different food items as they are placed randomly on top of each other.

Some failure cases produced from the UEC FoodPix dataset are illustrated in Figure 5.5. In some cases, model is unable to recognize the correct food item while in others it is unable to detect the food item itself. For example, in the 2nd and last row, we observe that the model is unable to detect some food items. Poor lighting

conditions and camouflage with background could be some plausible reasons of missed detections. The model tried to identify more classes in row 1 because of some other classes visible on 'Ramen Noodle'. The model confused 'Cabbage roll' with 'Fish shaped pancake' on 3rd row. Such cases can easily be confused even by a human being as the two classes are visually similar to each other. This shows that there is room for improvement in food segmentation.

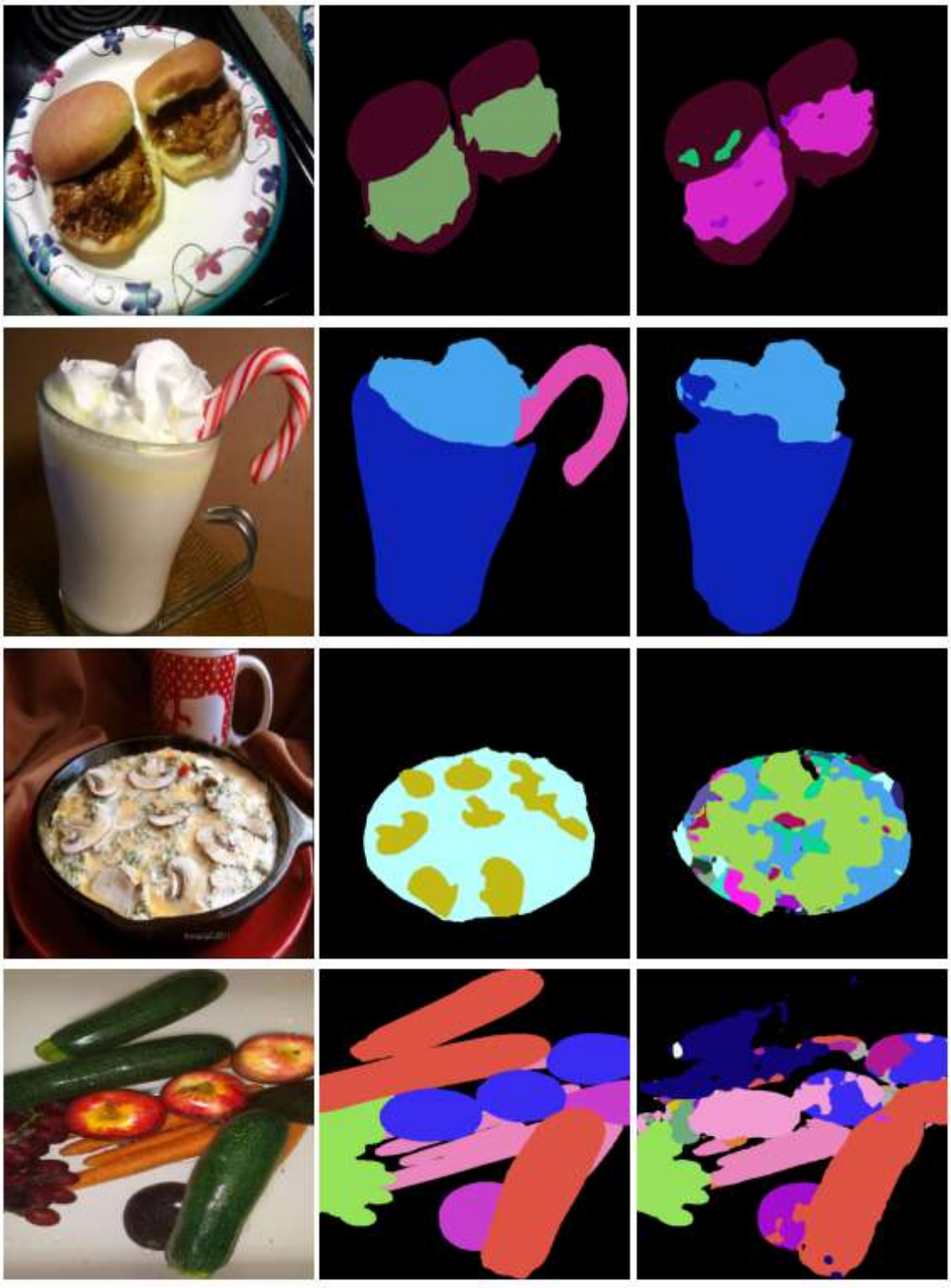

Figure 5.4: Failure cases from the FoodSeg103 dataset. Left column - Images, middle column - Ground truth masks, right column - GourmetNet predictions.

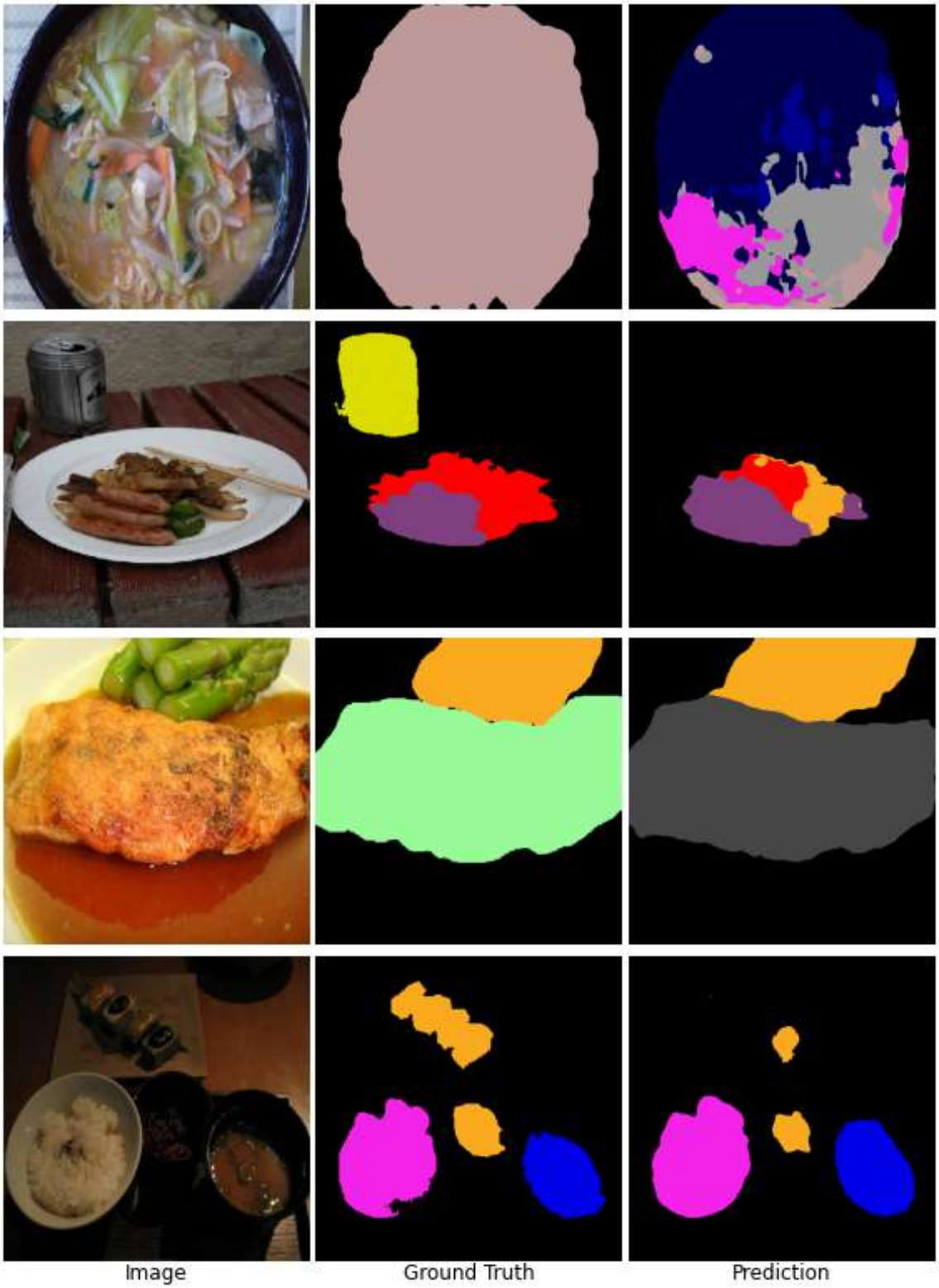

Figure 5.5: Failure cases from the UEC FoodPix dataset.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

We presented GourmetNet, a novel, end-to-end trainable architecture for food segmentation. GourmetNet incorporates the benefits of feature refinement from the channel and spatial attention modules with the improved multi-scale feature representations of the WASPv2 module. We performed extensive experiments to establish the effectiveness of this model. Our model outperforms the current state-of the-art for the UEC FoodPix dataset by 10%. We also compare our model with the DeepLabv3+ [40] and report an improvement of 4.2% on the UNIMIB2016 dataset and 5.8% on the UECFoodPix dataset. The goal of GourmetNet is to achieve improved food segmentation accuracy, consequently improving the performance of related tasks, such as automatic nutrition monitoring, food volume estimation, recipe extraction, or meal preparation.

6.2 Future Work

The GourmetNet framework can be improved by making the process more computationally efficient and increasing segmentation accuracy, so that food segmentation can be incorporated in a larger system for food volume estimation for dietary recommendations or assistance for meal preparation.

More concretely, after their success in Natuaral Language Processing (NLP) tasks, transformers are being adapted to be used with images. Recent advances have shown that Vision Transformer [74] produces state-of-the-art performance with images for some tasks. Vision transformer splits an image into patches, arranges these patches linearly and adds positional embedding as input to the transformational embedding. The embeddings are used as input to the transformer encoder consisting of multi-headed self attention, MLP and layer norm blocks. Our architecture can be modified as shown by Wu et.al. [75] where a CNN is used to extract low-level features and the Vision Transformer is used to extract the high-level features. Therefore, the vision transformer is used as a feature extractor and rest of the architecture can be used as it is. Although the vision transformer has proved to outperform CNNs by using smaller number of parameters, it is harder to train and requires large amounts of data for training. If these limitations are overcome, it would be interesting to to note the results.

Bibliography

[1] M. Bojarski, D. D. Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang, X. Zhang, J. Zhao, and K. Zieba, "End to end learning for self-driving cars," 2016.

[2] S. Shalev-Shwartz, S. Shammah, and A. Shashua, "On a formal model of safe and scalable self-driving cars," *CoRR*, vol. abs/1708.06374, 2017.

[3] B. Paden, M. Čáp, S. Z. Yong, D. Yershov, and E. Frazzoli, "A survey of motion planning and control techniques for self-driving urban vehicles," *IEEE Transactions on Intelligent Vehicles*, vol. 1, no. 1, pp. 33–55, 2016.

[4] H. Würschinger, M. Mühlbauer, M. Winter, M. Engelbrecht, and N. Hanenkamp, "Implementation and potentials of a machine vision system in a series production using deep learning and low-cost hardware," *Procedia CIRP*, vol. 90, pp. 611–616, 2020.

[5] F. Femling, A. Olsson, and F. Alonso-Fernandez, "Fruit and vegetable identification using machine learning for retail applications," in *2018 14th International Conference on Signal-Image Technology Internet-Based Systems (SITIS)*, 2018, pp. 9–15.

[6] D. A. Mora Hernandez, O. Nalbach, and D. Werth, "How computer vision provides physical retail with a better view on customers," in *2019 IEEE 21st Conference on Business Informatics (CBI)*, vol. 01, 2019, pp. 462–471.

[7] A. Afiq, M. Zakariya, M. Saad, A. Nurfarzana, M. Khir, A. Fadzil, A. Jale, W. Gunawan, Z. Izuddin, and M. Faizari, "A review on classifying abnormal behavior in crowd scene," *Journal of Visual Communication and Image Representation*, vol. 58, pp. 285–303, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1047320318303146

[8] W.-K. Lee, C.-F. Leong, W.-K. Lai, L.-K. Leow, and T.-H. Yap, "Archcam: Real time expert system for suspicious behaviour detection in atm site," *Expert Systems with Applications*, vol. 109, pp. 12–24, 2018.

[9] T. Kelly, W. B. Yang, C.-S. Chen, K. Reynolds, and J. He, "Global burden of obesity in 2005 and projections to 2030," *International Journal of Obesity*, vol. 32, pp. 1431–1437, 2008.

[10] F. Kong and J. Tan, "Dietcam: Automatic dietary assessment with mobile camera phones," *Pervasive and Mobile Computing*, vol. 8, no. 1, pp. 147–163, 2012.

[11] Y. Kawano and K. Yanai, "Foodcam: A real-time food recognition system on a smartphone," *Multimedia Tools Appl.*, vol. 74, no. 14, p. 5263–5287, Jul. 2015.

[12] C. Liu, Y. Cao, Y. Luo, G. Chen, V. Vokkarane, and Y. Ma, "Deepfood: Deep learning-based food image recognition for computer-aided dietary assessment," in *Proceedings of the 14th International Conference on Inclusive Smart Cities and Digital Health - Volume 9677*, ser. ICOST 2016. Berlin, Heidelberg: Springer-Verlag, 2016, p. 37–48.

[13] M. Puri, Z. Zhu, Q. Yu, A. Divakaran, and H. Sawhney, "Recognition and volume estimation of food intake using a mobile device," in *2009 Workshop on Applications of Computer Vision (WACV)*, 2009, pp. 1–8.

[14] J. Dehais, M. Anthimopoulos, S. Shevchik, and S. Mougiakakou, "Two-view 3d reconstruction for food volume estimation," *IEEE Transactions on Multimedia*, vol. 19, no. 5, pp. 1090–1099, 2017.

[15] R. Tanno, T. Ege, and K. Yanai, "Ar deepcaloriecam v2: food calorie estimation with cnn and ar-based actual size estimation," *Proceedings of the 24th ACM Symposium on Virtual Reality Software and Technology*, pp. 1–2, 11 2018.

[16] A. Myers, N. Johnston, V. Rathod, A. Korattikara, A. Gorban, N. Silberman, S. Guadarrama, G. Papandreou, J. Huang, and K. Murphy, "Im2calories: Towards an automated mobile vision food diary," in *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 1233–1241.

[17] W. Min, L. Liu, Z. Luo, and S. Jiang, "Ingredient-guided cascaded multi-attention network for food recognition," in *Proceedings of the 27th ACM International Conference on Multimedia*, ser. MM '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 1331–1339.

[18] J. Li, R. Guerrero, and V. Pavlovic, "Deep cooking: Predicting relative food ingredient amounts from images," *Proceedings of the 5th International Workshop on Multimedia Assisted Dietary Management*, vol. abs/1910.00100, 2019.

[19] A. Salvador, M. Drozdzal, X. Giro-i Nieto, and A. Romero, "Inverse cooking: Recipe generation from food images," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 10 445–10 454.

[20] J. Marín, A. Biswas, F. Ofli, N. Hynes, A. Salvador, Y. Aytar, I. Weber, and A. Torralba, "Recipe1m+: A dataset for learning cross-modal embeddings for cooking recipes and food images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 1, pp. 187–203, 2021.

[21] B. Artacho and A. Savakis, "Waterfall atrous spatial pooling architecture for efficient semantic segmentation," *Sensors*, vol. 19, no. 24, p. 5361, 2019.

[22] B. Artacho and A. E. Savakis, "Omnipose: A multi-scale framework for multi-person pose estimation," *ArXiv*, vol. abs/2103.10180, 2021.

[23] C. Peng and J. Ma, "Semantic segmentation using stride spatial pyramid pooling and dual attention decoder," *Pattern Recognition*, vol. 107, p. 107498, 06 2020.

[24] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, ser. NIPS'12. Curran Associates Inc., 2012, p. 1097–1105.

[25] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.

[26] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *International Conference on Learning Representations (ICLR)*, 2015.

[27] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 1–9.

[28] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[29] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," in *IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 1520–1528.

[30] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 3431–3440.

[31] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*. Cham: Springer International Publishing, 2015, pp. 234–241.

[32] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 39, no. 12, pp. 2481–2495, 2017.

[33] A. Kendall, V. Badrinarayanan, and R. Cipolla, "Bayesian segnet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding," *arXiv preprint arXiv:1511.02680*, 2015.

[34] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," *CoRR*, vol. abs/1612.01105, 2016.

[35] A. Paszke, A. Chaurasia, S. Kim, and E. Culurciello, "Enet: A deep neural network architecture for real-time semantic segmentation," *CoRR*, vol. abs/1606.02147, 2016.

[36] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 40, no. 4, pp. 834–848, 2018.

[37] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," *arXiv preprint arXiv:1511.07122*, 2015.

[38] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 37, no. 9, pp. 1904–1916, 2015.

[39] L. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," *CoRR*, vol. abs/1706.05587, 2017.

[40] L. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," *CoRR*, vol. abs/1802.02611, 2018.

[41] D. Lin, D. Shen, S. Shen, Y. Ji, D. Lischinski, D. Cohen-Or, and H. Huang, "Zigzagnet: Fusing top-down and bottom-up context for object segmentation," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 7482–7491.

[42] J. Fu, J. Liu, Y. Wang, Y. Li, Y. Bao, J. Tang, and H. Lu, "Adaptive context network for scene parsing," *CoRR*, vol. abs/1911.01664, 2019.

[43] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *3rd International Conference on Learning Representations, ICLR, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2015.

[44] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio, "Show, attend and tell: Neural image caption generation with visual attention," in *Proceedings of the 32nd International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 37. Lille, France: PMLR, 07–09 Jul 2015, pp. 2048–2057.

[45] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *CoRR*, vol. abs/1706.03762, 2017.

[46] L.-C. Chen, Y. Yang, J. Wang, W. Xu, and A. L. Yuille, "Attention to scale: Scale-aware semantic image segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 3640–3649.

[47] J. Fu, J. Liu, H. Tian, Y. Li, Y. Bao, Z. Fang, and H. Lu, "Dual attention network for scene segmentation," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 3141–3149.

[48] Q. Huang, C. Wu, C. X. amd Ye Wang, and C. J. Kuo, "Semantic segmentation with reverse attention," in *Proceedings of the British Machine Vision Conference (BMVC)*, G. B. Tae-Kyun Kim, Stefanos Zafeiriou and K. Mikolajczyk, Eds., September 2017, pp. 18.1–18.13.

[49] C. Yu, J. Wang, C. Peng, C. Gao, G. Yu, and N. Sang, "Bisenet: Bilateral segmentation network for real-time semantic segmentation," in *European Conference in Computer Vision (ECCV)*. Springer International Publishing, 2018, pp. 334–349.

[50] J. Shi and J. Malik, "Normalized cuts and image segmentation," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1997, pp. 731–737.

[51] Y. He, N. Khanna, C. Boushey, and E. Delp, "Image segmentation for image-based dietary assessment: A comparative study," 07 2013, pp. 1–4.

[52] S. Aslan, G. Ciocca, and R. Schettini, *On Comparing Color Spaces for Food Segmentation*, 09 2017, pp. 435–443.

[53] Y.-G. Wang, J. Yang, and Y.-C. Chang, "Color–texture image segmentation by integrating directional operators into jseg method," *Pattern Recognition Letters*, vol. 27, no. 16, pp. 1983–1990, 2006.

[54] M. Bolaños and P. Radeva, "Simultaneous food localization and recognition," *2016 23rd International Conference on Pattern Recognition (ICPR)*, pp. 3140–3145, 2016.

[55] J. Dehais, M. Anthimopoulos, and S. Mougiakakou, "Food image segmentation for dietary assessment," in *Proceedings of the 2nd International Workshop on Multimedia Assisted Dietary Management*, ser. MADiMa '16. New York, NY, USA: Association for Computing Machinery, 2016, p. 23–28. [Online]. Available: https://doi.org/10.1145/2986035.2986047

[56] Y. Wang, F. Zhu, C. J. Boushey, and E. J. Delp, "Weakly supervised food image segmentation using class activation maps," in *2017 IEEE International Conference on Image Processing (ICIP)*, 2017, pp. 1277–1281.

[57] W. Shimoda and K. Yanai, "Cnn-based food image segmentation without pixel-wise annotation," vol. 9281, 09 2015, pp. 449–457.

[58] Y. Lu, D. Allegra, M. Anthimopoulos, F. Stanco, G. M. Farinella, and S. Mougiakakou, "A multi-task learning approach for meal assessment," in *Proceedings of the Joint Workshop on Multimedia for Cooking and Eating*

Activities and Multimedia Assisted Dietary Management, ser. CEA/MADiMa '18. New York, NY, USA: Association for Computing Machinery, 2018, p. 46–52. [Online]. Available: https://doi-org.ezproxy.rit.edu/10.1145/3230519.3230593

[59] T. Ege, W. Shimoda, and K. Yanai, "A new large-scale food image segmentation dataset and its application to food calorie estimation based on grains of rice," in *International Workshop on Multimedia Assisted Dietary Management (MADiMa)*. Association for Computing Machinery, 2019, p. 82–87.

[60] J. Redmon and A. Farhadi, "Yolo9000: Better, faster, stronger," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 6517–6525.

[61] K. J. Pfisterer, R. Amelard, A. Chung, B. Syrnyk, A. MacLean, and A. Wong, "Fully-automatic semantic segmentation for food intake tracking in long-term care homes," *ArXiv*, vol. abs/1910.11250, 2019.

[62] S. Aslan, G. Ciocca, and R. Schettini, "Semantic food segmentation for automatic dietary monitoring," in *2018 IEEE 8th International Conference on Consumer Electronics - Berlin (ICCE-Berlin)*, 2018, pp. 1–6.

[63] ——, "Semantic segmentation of food images for automatic dietary monitoring," in *2018 26th Signal Processing and Communications Applications Conference (SIU)*, 2018, pp. 1–4.

[64] G. Ciocca, P. Napoletano, and R. Schettini, "Food recognition: A new dataset, experiments, and results," *IEEE Journal of Biomedical and Health Informatics*, vol. 21, no. 3, pp. 588–598, 2017.

[65] U. Sharma, B. Artacho, and A. Savakis, "Gourmetnet: Food segmentation using multi-scale waterfall features with spatial and channel attention," *Sensors*, vol. 21, p. 7504, 11 2021.

[66] X. Wu, X. Fu, Y. Liu, E.-P. Lim, S. C. Hoi, and Q. Sun, "A large-scale benchmark for food image segmentation," in *Proceedings of ACM international conference on Multimedia*, 2021.

[67] G. Ciocca, P. Napoletano, and R. Schettini, "IAT - image annotation tool: Manual," *CoRR*, vol. abs/1502.05212, 2015. [Online]. Available: http://arxiv.org/abs/1502.05212

[68] D. H. Douglas and T. K. Peucker, "Algorithms for the reduction of the number of points required to represent a digitized line or its caricature," *Cartographica: The International Journal for Geographic Information and Geovisualization*, vol. 10, pp. 112–122, 1973.

[69] S. Ruder, "An overview of gradient descent optimization algorithms," *arXiv preprint arXiv:1609.04747*, 2016.

[70] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, "On the importance of initialization and momentum in deep learning," in *Proceedings of the 30th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research (PMLR), vol. 28, no. 3, 17–19 Jun 2013, pp. 1139–1147.

[71] Z. Huang, X. Wang, L. Huang, C. Huang, Y. Wei, and W. Liu, "Ccnet: Criss-cross attention for semantic segmentation," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 603–612.

[72] A. Kirillov, R. B. Girshick, K. He, and P. Dollár, "Panoptic feature pyramid networks," *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6392–6401, 2019.

[73] S. Zheng, J. Lu, H. Zhao, X. Zhu, Z. Luo, Y. Wang, Y. Fu, J. Feng, T. Xiang, P. H. S. Torr, and L. Zhang, "Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers," *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6877–6886, 2021.

[74] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," *CoRR*, vol. abs/2010.11929, 2020. [Online]. Available: https://arxiv.org/abs/2010.11929

[75] B. Wu, C. Xu, X. Dai, A. Wan, P. Zhang, M. Tomizuka, K. Keutzer, and P. Vajda, "Visual transformers: Token-based image representation and processing for computer vision," *CoRR*, vol. abs/2006.03677, 2020. [Online]. Available: https://arxiv.org/abs/2006.03677