Deep Learning

Food Recognition Challenge

Gee Jun Hui Leonidas Yunani
Marini Luca

University of Bologna
Master in Artificial Intelligence

Contents

1 Introduction

The Food Recognition Challenge is an AICrowd challenge whereby participants are asked to train a model to look at images of food items and detect the individual food items in them. This task of food tracking can be both of personal interest and of medical relevance. Medical studies have for some time been interested in the food intake of study participants but had to rely on food frequency questionnaires that are known to be imprecise. A trained food tacking model can be used instead to track the food intake of participants by simply taking a picture of what they consume.

Scene parsing, based on semantic segmentation, is a fundamental topic in computer vision. The goal is to assign each pixel in the image a category label. Since the problem is defined at the pixel level, determining image class labels only is not acceptable, but localising them at the original image pixel resolution is necessary [3]. For this project, multiple image segmentation models have been trained and evaluated to learn and understand the semantic segmentation process.

1.1 Dataset

The **AICrowd Food Recognition Challenge Dataset** consists of 273 classes of food along with their respective bounding boxes and semantic segmentation masks. The dataset is divided into 2 parts: training and validation sets.

- The training set consists of 24119 RGB images, with their corresponding 39328 annotations in MS-COCO format.
- The validation set consists of 1269 RGB images, with their corresponding 2053 annotations in MS-COCO format.

The top-5 and bottom-5 classes in the training set are determined and shown in Figures 1 and 2. It can be seen that food such as *water* and *bread-white* are unsurprisingly highly common among the food images, while food such as *veggie-burger* are much rarer in the food images.

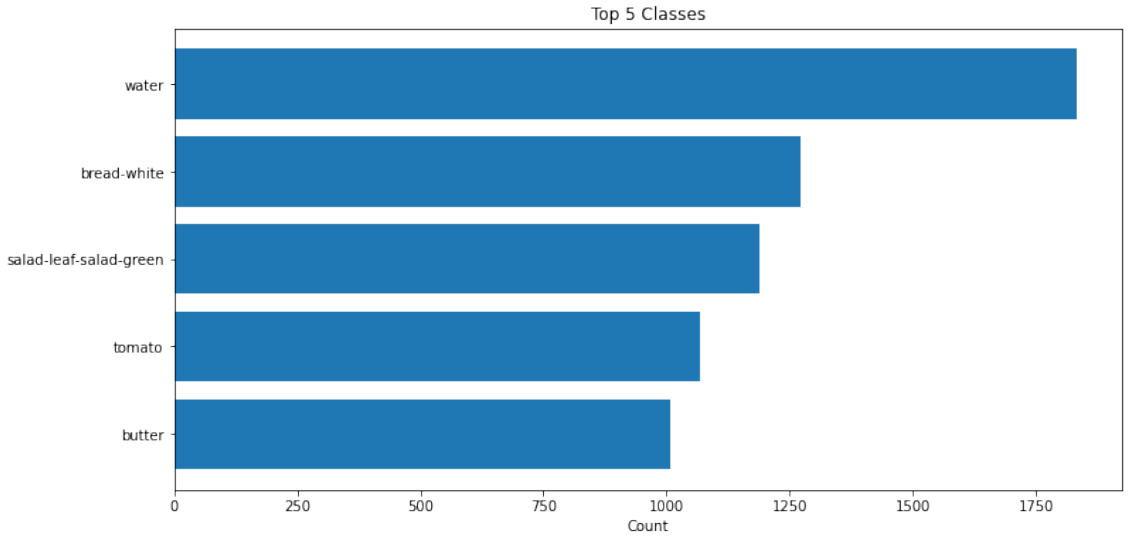

Figure 1: Top-5 food classes

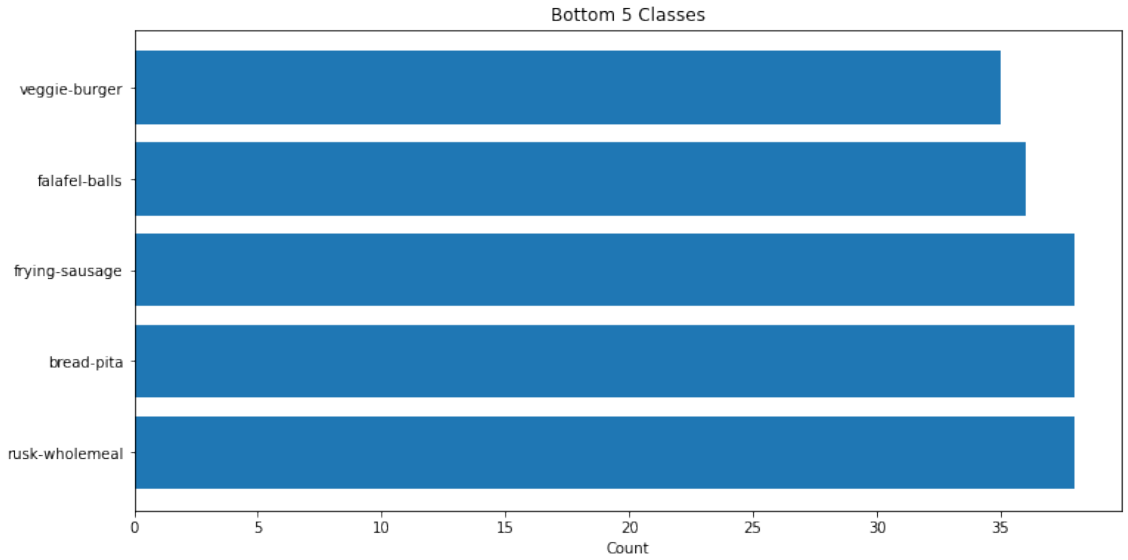

Figure 2: Bottom-5 food classes

2 Data Preparation

2.1 Multi-channel Mask

For multi-class semantic segmentation, a multi-channel mask must be created for each image. Each channel represents a binary mask of the food class in the image. The first channel of each mask is the background class. Therefore, for the challenge, a 274 (273 classes + background) channel mask is generated for each image.

Figure 3: Visualization of a binary class image and its ground truth mask

Figure 4: Visualization of a multi class image and its ground truth mask. Different colors in the mask represent multiple different food classes

It has been observed that there are imperfections in the dataset, hence for certain images, a multi-channel mask with no annotations can be obtained.

2.2 Image Augmentation

To further improve the performance of the model, a geometric augmentation is applied to the images and masks of the training set. The images and masks are flipped horizontally with a probability of 0.5. Additional augmentations, such as applying a Gaussian filter, were examined. However, empirical results show that such augmentations reduced the model's performance.

2.3 Image Generator

To further reduce the memory requirements, a custom generator is created to feed in the images and their multi-channel masks in batches. The chosen batch size is set to 300 images. This choice is important because a too small batch size will cause the model to plateau very early in the training phase. However, a batch size that is too large may cause the kernel to crash if the memory limit is exceeded.

3 Model Architecture

A PSPNet architecture with an EfficientNet-B7 backbone was chosen as the model for the food segmentation task. Before training, the number of training epochs was set to 10 and the encoder weights were frozen.

3.1 EfficientNet-B7

Input images are fed into the Feature Map, which is responsible for extracting out features from the images.

More precisely, the Feature Map is the neural network used as the backbone. In particular, we use EfficientNet-B7, whose baseline network was developed by using neural architecture search. The baseline was then scaled up to obtain a family of models, called EfficientNets, which achieve much better accuracy and efficiency than previous existing ConvNets [2].

EfficientNet-B7 is obtained by applying a compound scaling method that uniformly scales all three dimensions of the baseline with a fixed ratio (Figure 5).

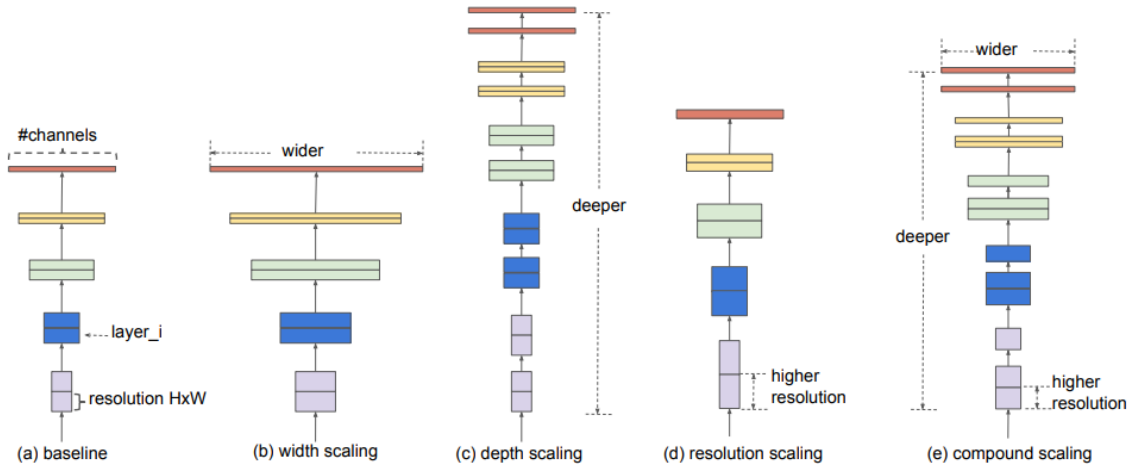

Figure 5: Model Scaling. (a) is a baseline network example; (b)-(d) are conventional scaling that only increases one dimension of network width, depth, or resolution. (e) is compound scaling method that uniformly scales all three dimensions with a fixed ratio.

The three dimensions are:

- **Depth (d)**: the number of layers of the network;

- **Width (w)**: the number of channels;
- **Resolution (r)**: the resolution of input images.

The network width, depth, and resolution are scaled in the following way:

$$\begin{aligned} &depth : d = \alpha^{\phi} \\ &width : w = \beta^{\phi} \\ &resolution : r = \gamma^{\phi} \\ &s.t.\ \alpha \cdot \beta^2 \cdot \gamma^2 \approx 2 \\ &\alpha \geq 1, \beta \geq 1, \gamma \geq 1 \end{aligned} \tag{1}$$

α, β, γ are constants that can be determined by a small grid search that also tries to maximize the model accuracy. And φ is the fixed coefficient.

3.2 PSPNet

PSPNet, or Pyramid Scene Parsing Network, is a semantic segmentation model that utilizes a pyramid parsing module to exploit global context information using different region-based context aggregation. The local and global clues together make the final prediction more reliable. Given an input image, PSPNet use a pretrained CNN with the dilated network strategy to extract the feature map. The final feature map size is $\frac{1}{8}$ of the input image. On top of the map, we use the pyramid pooling module to gather context information. Using our 4-level pyramid, the pooling kernels cover the whole, half of, and small portions of the image. They are fused as the global prior. Then we concatenate the prior with the original feature map in the final part of. It is followed by a convolution layer to generate the final prediction map [1].

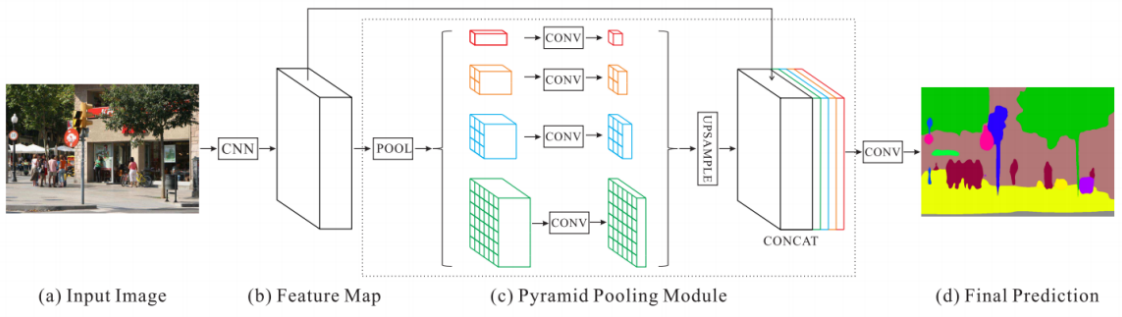

Figure 6: Overview of PSPNet

4 Results

The metrics used to evaluate the model are the following: IoU, precision and recall. The result of the final epoch for training and validation is collected into the following table:

Dataset	**IoU** (%)	**Precision** (%)	**Recall** (%)
Training	34.5	72.4	46.3
Validation	30.2	68.8	43.1

The predictions of the trained model on the validation set are also visualised below using both binary and multi class images.

4.1 Binary Class Images

In this subsection are presented some of the predicted segmentation masks of images that contain only one food class.

Figure 7

Figure 8

4.2 Multi Class Images

In this subsection are presented some of the predicted segmentation masks of images that contain more than one food class.

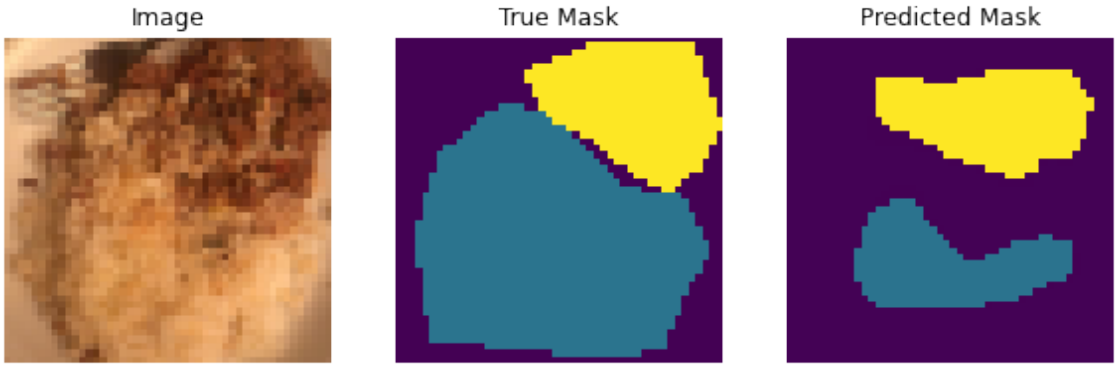

Figure 9

Figure 10

5 Conclusion

In this project, we demonstrated an example of how the semantic segmentation process is carried out on a food recognition task. The images, masks and corresponding augmentations have been crafted with a custom generator to ensure that the model is able to learn without running out of memory. The PSPNet architecture and an EfficientNet-B7 backbone was used. The trained model is shown to perform relatively well given the dataset and problem.

6 References

[1] Papers With Code. *PSPNet Explained*. URL: https://paperswithcode.com/method/pspnet.

[2] Mingxing Tan and Quoc V. Le. "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks". In: *CoRR* abs/1905.11946 (2019). arXiv: 1905.11946. URL: https://arxiv.org/abs/1905.11946.

[3] Irem Ülkü and Erdem Akagündüz. "A Survey on Deep Learning-based Architectures for Semantic Segmentation on 2D images". In: *CoRR* abs/1912.10230 (2019). arXiv: 1912.10230. URL: http://arxiv.org/abs/1912.10230.