Deep Learning

Food Recognition

Gee Jun Hui Leonidas Yunani
Marini Luca

University of Bologna
Artificial Intelligence Master Degree course

Contents

1 Introduction

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 [5].

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 [4].

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 food dataset is divided into 3 parts: training, validation, and test set:

- The training set consists of 24120 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 test set is provided as a debug dataset for Round-3. Its images are the same as those of the validation set.

All the 273 food classes are visible in the horizontal bar plot in Figures 1 and 2. It can be noticed that the classes are not uniformly distributed. The two most frequent food classes are *water* and *bread-white*.

Due to the dataset being a collection of user-submitted photos, certain segmentation errors exist in them. For example, some images may have incorrectly drawn segmentations mask or even not having one at all.

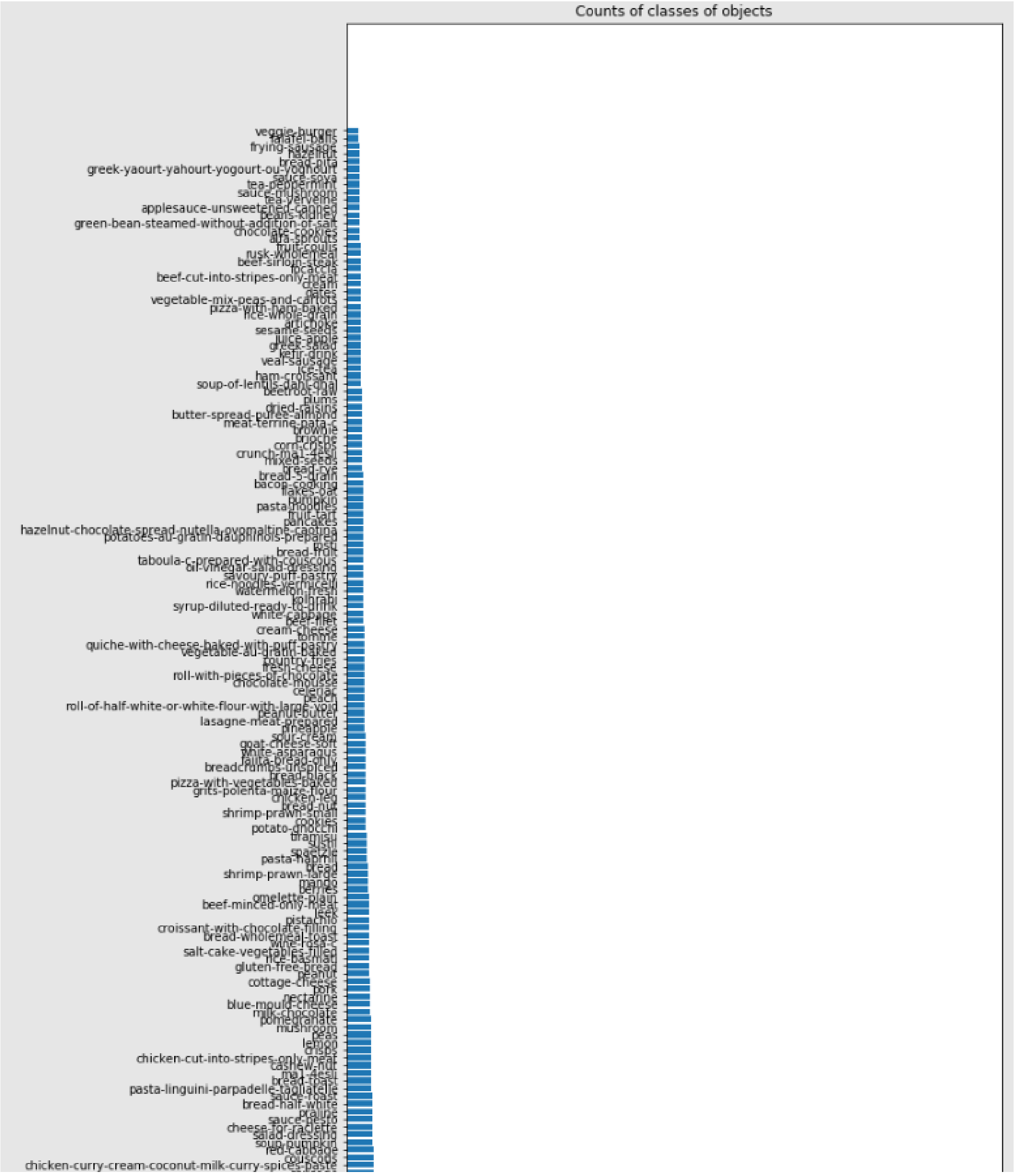

Figure 1: First part of the food classes

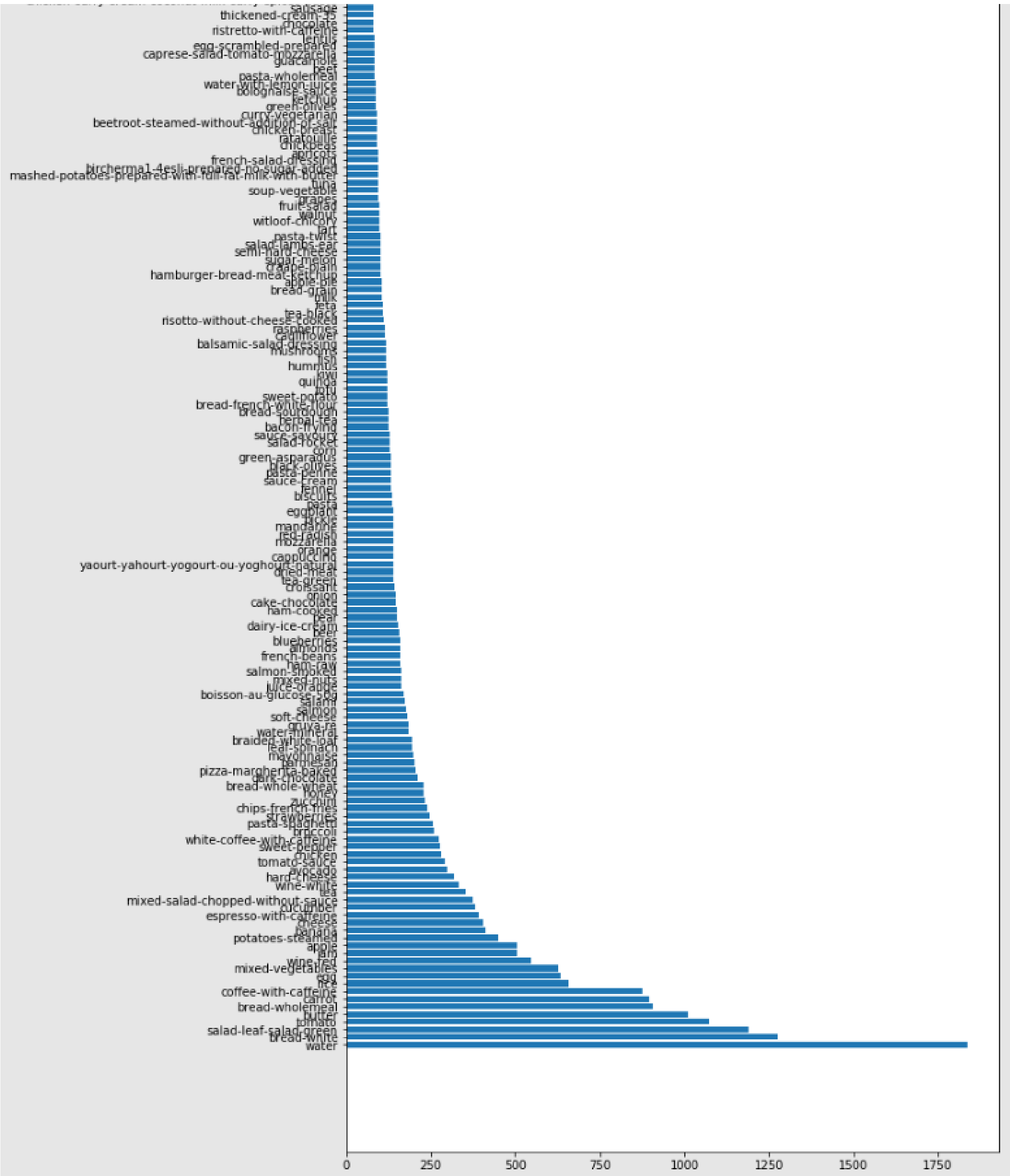

Figure 2: Second part of the food classes

2 Techniques

2.1 Multi-channel masks

For multi-class semantic segmentation, a multi-channel mask must be created for each image. Each channel represents a binary mask of the object 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.

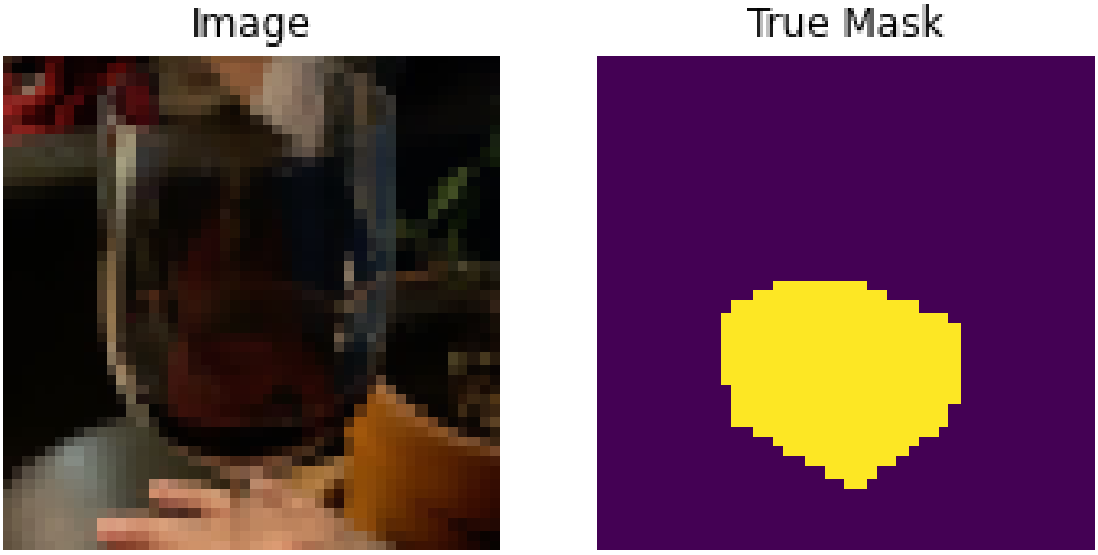

Figure 3: Visualization of an image and its ground truth mask

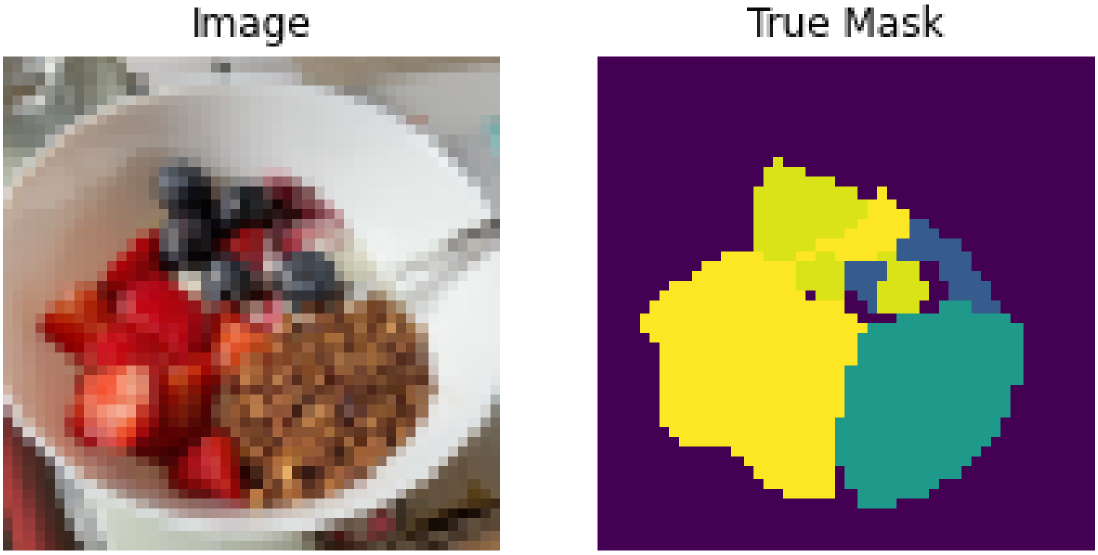

Figure 4: Visualization of an image and its ground truth mask. Different colors in the mask represent different food classes

The height and width of the images and masks are standardised to *48x48* pixels (Figure 3 and 4). The primary reason for this is due to the computational limitations of Kaggle's environment. In Kaggle, a maximum of 16 GB of RAM is allotted with a maximum execution time of 9 hours. For these reasons, using too large images will result in the kernel crashing due to a lack of memory space or the training time taking too long and exceeding Kaggle's execution time limit.

2.2 Data Generator

To reduce further 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.

2.3 Image Augmentation

To 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 augmentation methods, such as applying a Gaussian filter, were examined. However, it results that those types of data augmentations reduced the model's performance.

3 Model

3.1 PSPNet

PSPNet, or Pyramid Scene Parsing Network, is a semantic segmentation model that utilizes a pyramid parsing module that exploits global context information by different region-based context aggregation. The local and global clues together make the final prediction more reliable [2].

Therefore, the PSPNet architecture considers the global context of the image to make the local level predictions.

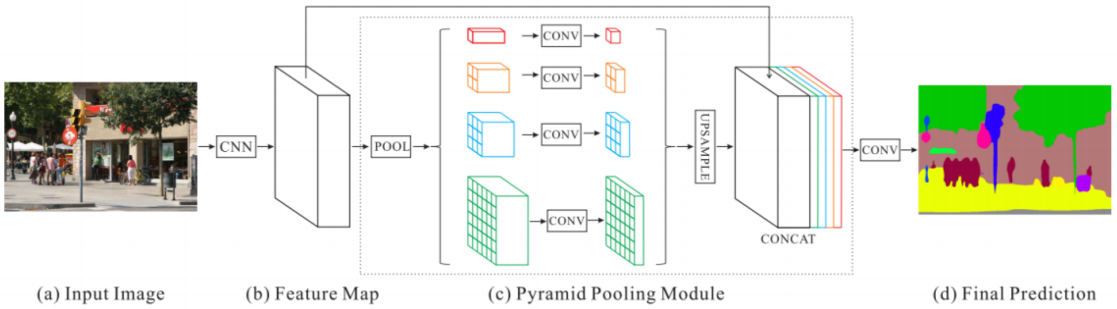

Figure 5: Overview of PSPNet

3.2 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 [3].

EfficientNet-B7 is obtained by applying a compound scaling method that uniformly scales all three dimensions of the baseline with a fixed ratio (Figure 6).

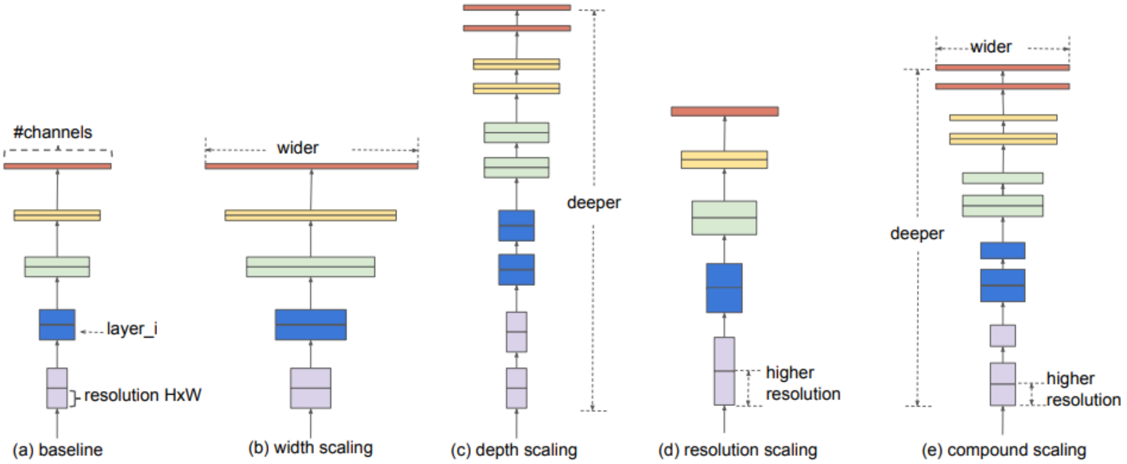

Figure 6: Model Scaling. (a) is a baseline network example; (b)-(d) are conventional scaling that only increases one dimension of network width, depth, or resolution. (e) is compound scaling method that uniformly scales all three dimensions with a fixed ratio.

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:

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

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.

3.3 Pyramid Pooling Module

The Pyramid Pooling Module (also called Pyramid Pooling Layer) is applied to harvest different sub-region representations [5] and to detect and segment the input objects at multiple scales. This Module helps the model to capture the global context in the image. Therefore, it improves the classification of the pixels by exploiting the global information present in the image.

After the application of the backbone (Figure 5 (b)), the features are pooled in 4 levels. The pooling kernels in the 4-level pyramid cover the whole, half of, and small portions of the image. As stated in [5], average pooling works better than max pooling in all tested settings.

The kernels are then fused as the global prior. After that, the prior is concatenated with the original feature map in the final part. It is followed by a convolution layer that generates the final prediction map (Figure 5 (d)).

3.4 Loss function

[1] suggests to use Focal loss with highly-imbalanced dataset because it downweights the contribution of easy examples, enabling model to learn hard examples. Nonetheless, the loss function we used is the sum of the categorical cross-entropy loss with the dice loss because this type of loss improved the final precision and mIoU results.

4 Metrics

The metrics used to evaluate the model's performance are discussed in this Section.

4.1 mIoU

The mean Intersection over Union ($mIoU$) is the average of the IoU over classes.

$$mIoU = \frac{1}{c}\sum_{c=1}^{C} IoU_c$$

where the Intersection over Union of a single class c is defined as:

$$IoU_c = \sum_{images} \frac{area\ of\ intersection}{area\ of\ union}$$

IoU_c is the sum across all images of the ratio between:

- the **area of intersection**, which is the intersection of the number of pixels predicted as class c and the number of pixels that are of class c (ground truth pixels);
- the **area** (# pixels) **of union**, which is the sum of:
  - the number of pixels predicted as class c;
  - the number of ground truth pixels;
  - all minus the **area of intersection**.

5 Tests

5.1 Backbone

Two different backbones were tested using the same hyper-parameters. The results are visible in the following Table:

Backbone	**mIoU**(%)	**AP**(%)	**AR**(%)
ResNet-152	24.2	52.9	41.6
EfficientNet-B7	**25.9**	**54.6**	**43.4**

Initially, the first used backbone was ResNet-50, but then we tested ResNet-152 because increasing the depth of ResNet can improve the score of Mean IoU [5].

Table 5.1 shows that Efficient-Net-B7 approximately improves all the metrics by 2% compared to ResNet-152. The second row contains our **final results**.

6 Visualization of the results

In this section are shown some of the results (final predictions) obtained with the previously described PSPNet.

6.1 Binary class images

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

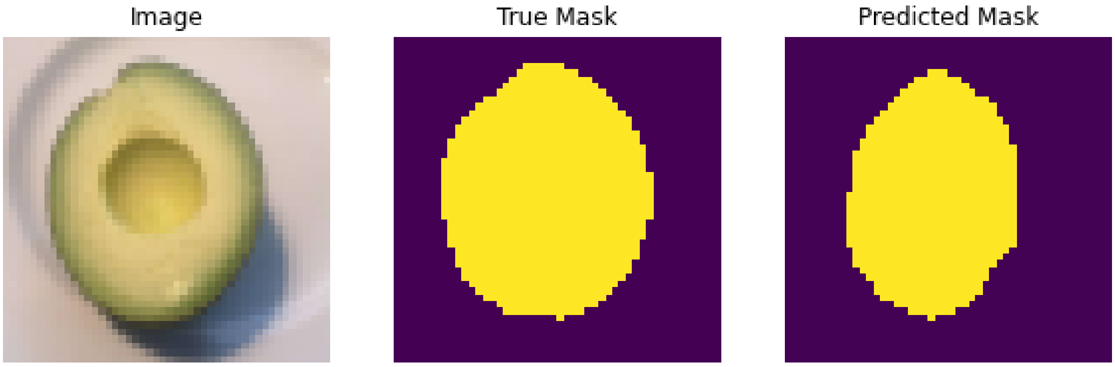

Figure 7

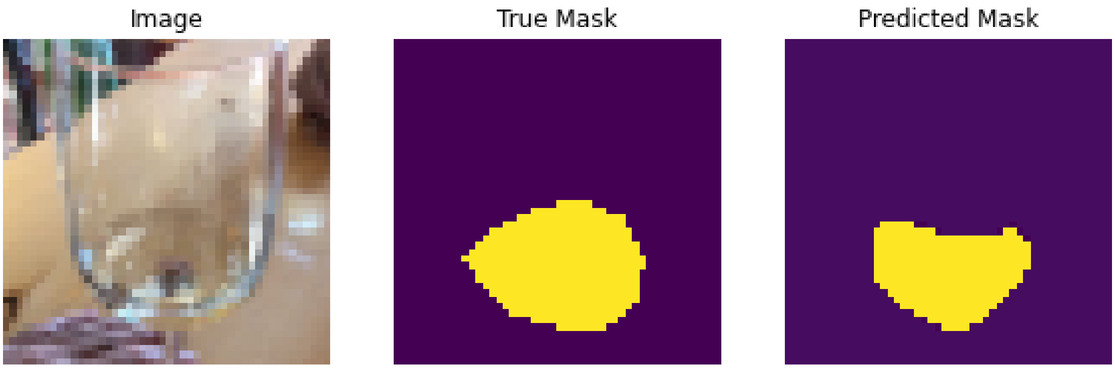

Figure 8

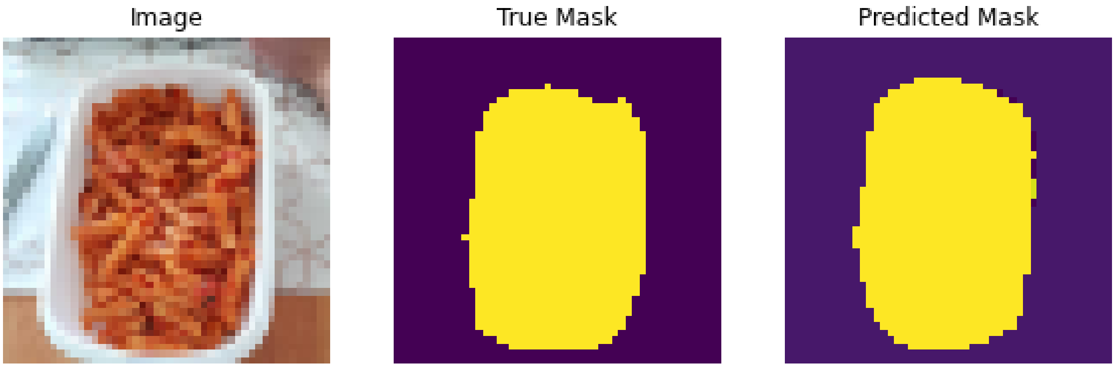

Figure 9

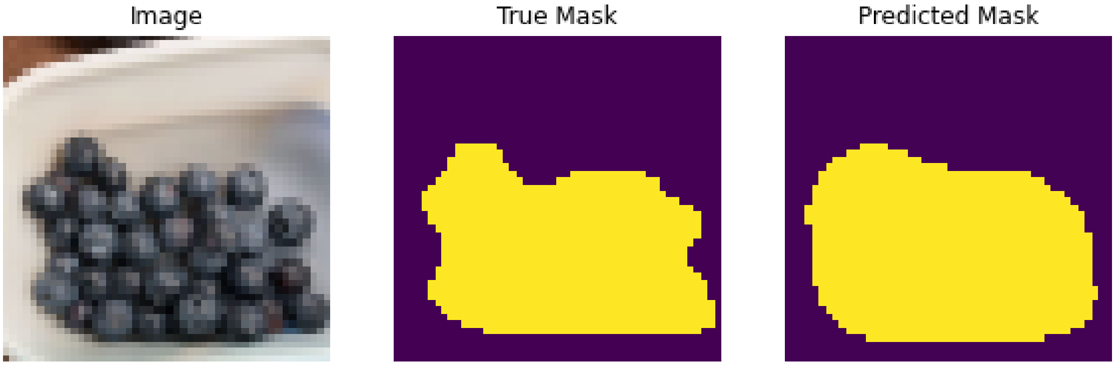

Figure 10

6.2 Multi-class images

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

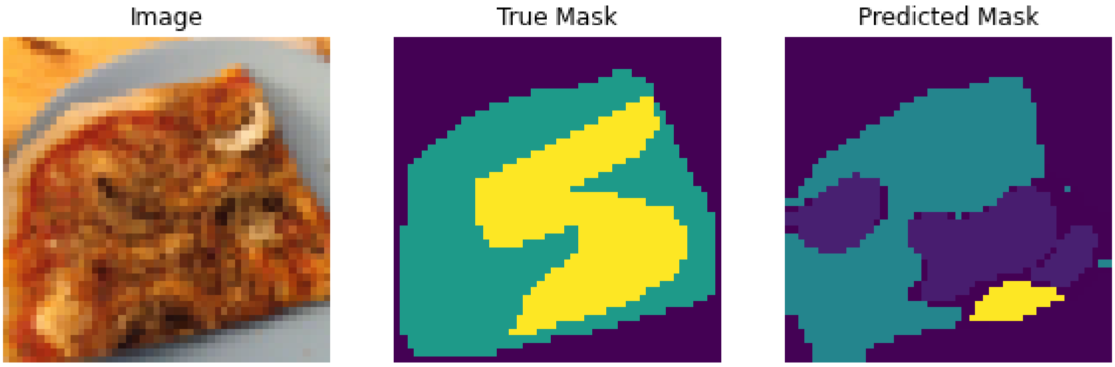

Figure 11

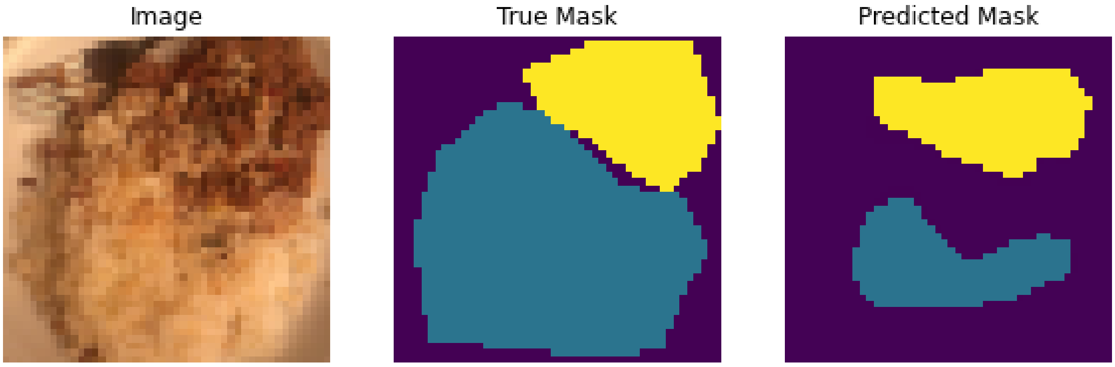

Figure 12

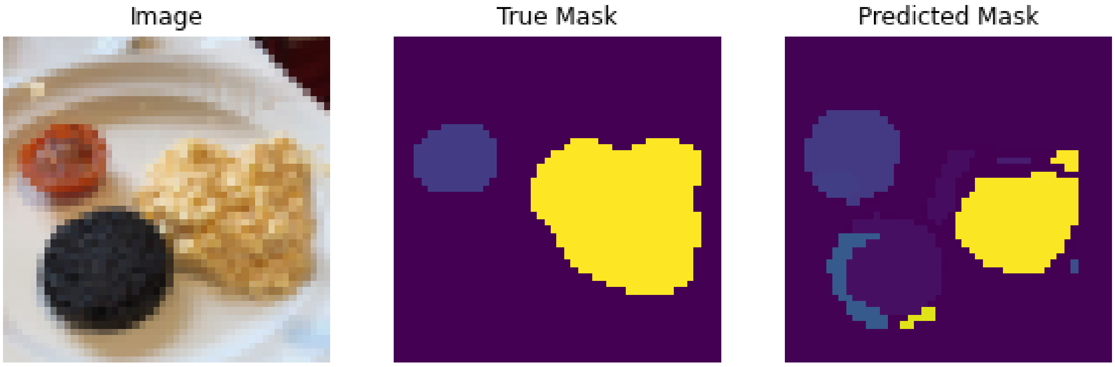

Figure 13

7 Conclusions

The model predicts the segmentation masks of binary class images almost perfectly (Subsection 6.1).

It estimates accurately the segmentation masks of multi-class images (Subsection 6.2), but it also makes some errors in the assignment of the food class. The principal cause of the mistakes is the high number of food classes (273).

References

[1] Shruti Jadon. "A survey of loss functions for semantic segmentation". In: *2020 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)* (Oct. 2020). DOI: 10.1109/cibcb48159.2020.9277638. URL: http://dx.doi.org/10.1109/CIBCB48159.2020.9277638.

[2] *PSPNet*. URL: https://paperswithcode.com/method/pspnet.

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

[4] 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.

[5] Hengshuang Zhao et al. *Pyramid Scene Parsing Network*. 2017. arXiv: 1612.01105 [cs.CV].