Deep Learning Models on Resource-Constrained Hardware

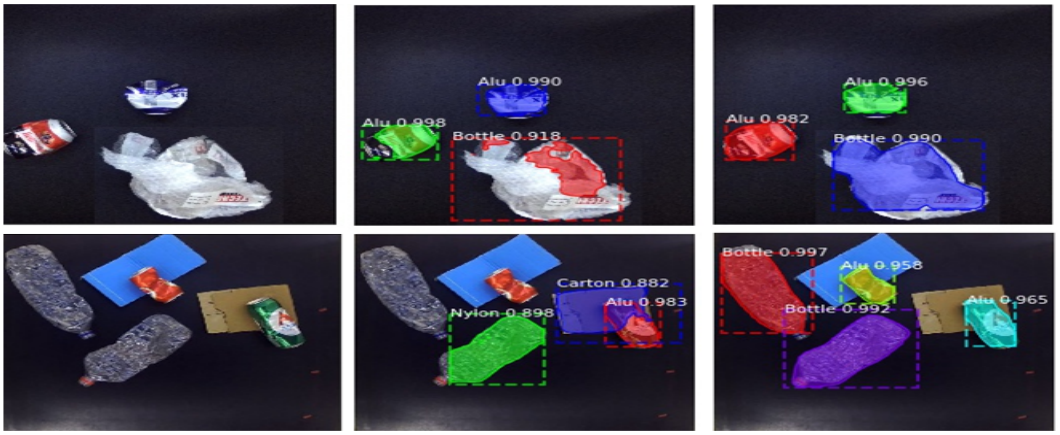
Teaching Assistant

Shyam Nandan Rai (email: shyam.rai@polito.it)

Project 4B: Resource Constrained Semantic Segmentation for Waste Sorting

Overview

As the world population grows, the estimated waste production is also becoming large. Hence, there is a need for efficient strategies for Materials Recovery Facilities (the center of the recycling process) to maximise the **detection of recyclable waste** in order to minimise the adverse effect of rising waste in the environment. In this project, your task is to use **tiny semantic segmentation** models to segment recyclable waste that could help deploy in industrial settings. The goal is for the models to be able to fit in **10MB**, which represents a realistic memory constraint for an edge application using a smart camera with some small onboard processing capacity.



Dataset

You will use the ReSORT-IT dataset for all the experiments. It consists of 5500 training images and 1460 test images.

Starting Code Repository [link]

Goals

1. Get acquainted with the task of Waste Sorting and understand the similarities and differences of a few notable tiny semantic segmentation models [**1,3**].
2. Run some experiments with a few popular baseline models for tiny semantic segmentation [**1,3,4**], with the constraint that the model must occupy at most **10 MB of memory**.
3. Propose your own extensions to improve the system or to overcome some of the challenges for the task. You are free to propose your own ideas and choose what you want to focus on.

Project Steps

1. Study the literature and get familiar with the task

As a preliminary step to get familiar with the task, start by studying the problem and dataset [**10**], and read about some popular lightweight models for semantic segmentation [**1,2,3,4**]. These models will serve as the basis for the experiments you will carry out in the project, so make sure you understand how they work.

1. Experiments

a. Binary segmentation experiments

Train a segmentation network that gives pixel-wise prediction for the waste area. At this step, we only want to segment the waste without classifying which type (class) of waste it is. In other words, we only want to binary segment objects and non-objects. Perform the experiments on ICNet, BiSeNet (Xception39 backbone), and ENet reporting mIOU, FLOPS, and # of trainable model parameters.

The table of results should look something like this

	Object (mIoU)	FLOPS	#parameters	Model size (MB)
ICNet				
BiSeNet				
ENet				

a.

b. Instance segmentation experiments

Train a segmentation network that gives pixel-wise predictions for the waste

area and its class. Differently from the previous step, we now also want to classify the material of the waste, from the set of four possible labels (paper, bottle, aluminum, nylon). Perform the experiments on ICNet, BiSeNet (Xception39 backbone), and ENet reporting mIOU, FLOPS, and # of trainable model parameters.

With respect to the previous case, the table of results should now report both the per class and average mIoU

	Paper (mIoU)	Bottle (mIoU)	Aluminum (mIoU)	Nylon (mIoU)	avg . mIoU
ICNet					
BiSeNet					
ENet					

a.

c. Improvements/further analyses

Now that all the baseline results are ready, you can move further with the analysis, trying a couple of different things:

i. Changing the model encoder and decoder sizes.
ii. Effect of using data augmentation

Don't forget to report the results and try to understand why they change the way they do.

3. Personal Contribution

It is now time for you to put into practice what you have learned so far and propose additional analyses and extensions. The extensions may have different goals, e.g., add a new analysis that investigates some problem, improve the performance (mIoU) of the model, or reduce the size of the model while trying to keep the same accuracy. Here are a few examples of possible extensions, but feel free to propose your own:

a) Rather than training a small semantic segmentation model from scratch, another approach could be to distill knowledge from a larger model (e.g. BiSeNet [ResNet-18] -> BiSeNet [Xception39]).
b) The segmentation performance may be improved by addressing the class imbalance that is implicit in the task. In this sense, one idea could be exploring alternative losses such as the weighted cross-entropy loss, the focal loss, and the class balanced loss.

c) A strategy to improve the model performance could be to use self-supervised pre-training (e.g. using rotations for self-supervision)
d) Another direction could be to try and reduce the model size and latency, by employing quantisation and pruning methods to the trained models. [Link] [Link]
e) How well does the model trained on ReSort-IT work on different data distributions? You may explore this problem using other datasets (e.g. TACO) and try to improve results using domain adaptation or domain generalisation techniques.
f) Be creative, and come up with your own ideas and proposals. It is not crucial that these extensions produce very good results as long as you provide good motivations for why you are trying something.

References

1. BiSeNet: Bilateral Segmentation Network for Real-time Semantic Segmentation
2. BiSeNet V2: Bilateral Network with Guided Aggregation for Real-time Semantic Segmentation
3. ICNet for Real-Time Semantic Segmentation on High-Resolution Images
4. Enet: A deep neural network architecture for real-time semantic segmentation
5. Unsupervised representation learning by predicting image rotation
6. Context Encoders: Feature Learning by Inpainting
7. Focal Loss for Dense Object Detection
8. Class-Balanced Loss Based on Effective Number of Samples
9. Garbage Collection and Sorting with a Mobile Manipulator using Deep Learning and Whole-Body Control
10. Vision-based Material Classification for Robotic Urban Waste Sorting