Leveraging semantic segmentation with resource constraints for better waste sorting

Emanuele De Leo
Polytechnic of Turin
Turin, Italy
s318658
s318658@studenti.polito.it

Giovanni Grossi
Polytechnic of Turin
Turin, Italy
s318690
s318690@studenti.polito.it

Luca Pesce
Polytechnic of Turin
Turin, Italy
s310211
s310211@studenti.polito.it

Abstract

The rapid increase in world population is contributing to an unprecedented surge in waste production, necessitating the need for more efficient and sustainable waste management systems. Our project focuses on the application of semantic segmentation models to automate the waste sorting process in Materials Recovery Facilities. However, the deployment of these models on edge devices presents a unique challenge due to their limited onboard processing capacity and memory constraints. Consequently, our project sets a critical objective to develop highly efficient semantic segmentation models that occupy no more than 10MB in memory, a realistic target reflecting the processing capability of a typical edge device. To achieve this goal, we will leverage techniques such as quantization to reduce the model size while preserving its performance.

1. Introduction

The purpose of this project is to explore the capability of small semantic segmentation models in waste sorting within industrial environments. These models are created to work within memory size restriction, which is a reasonable limit for edge applications such as smart cameras with limited onboard processing power.

The initial task of the project focuses on determining whether or not waste is present in the image doing binary segmentation, that essentially consists in detecting the presence of objects in the image. The following phase will further expand this segmentation to include different types of recyclable waste. The aim is to create a model that not only detects waste, but also identifies its type, thus providing a comprehensive solution for waste sorting.

We utilized the ReSORT-IT dataset to train and evaluate our models. This dataset consists of 5500 training images and 1460 test images and each of these can contains one or more types of objects with a very large variety of backgrounds, so it provides a comprehensive resource that reflects the intricacy and variety of real-world wastes. More precisely, each image of the dataset is characterized by the possible presence of one ore more waste materials from four distinct categories of : paper waste, plastic bottles, aluminum cans, and nylon fabric waste. These types of materials possess unique characteristics and they require specific sorting and recycling methods for effective waste management. As already said, one relevant element of the images is the background, since each one is distinctive and unrelated to the waste materials, moreover each one is unconnected with one another. This fact allows the algorithms to accurately classify and distinguish the waste without being

Figure 1. example of a dataset image

Table 1. Number of images for each class

Waste type	number of images
Paper	2440
Bottle	2484
Alluminium	1992
Nylon	2000

impacted by external factors.

Our project begun by relying on three tiny network models: Efficient Network (*ENet*), Image Cascade Network (*ICNet*) , and Bilateral Segementation Network (*BiSeNet*). We had to test and modify those networks, making them able to fit the data with our memory restriction having acceptable performances in terms of speed and accuracy. At the end we selected the networks that most satisfied the requirements trying to improve their performances by using different methods as data augmentation and different loss functions.

2. Related work

In the subsequent section of this paper, we delve into a systematic review of pertinent literature across two main domains of our project : Applications in Waste Sorting and Semantic Segmentation.

2.1. Semantic segmentation

Semantic segmentation plays a crucial role in our waste type detection project. This technique is used to depict and classify different categories of object in the images by their context such as those that compose our data set [1]. By classifying each pixel of an image into these predefined categories we can produce a useful feature map to achieve a detailed and comprehensive characterization of the present objects [9]. Thanks to this generation of features we are able to compute scores (such as the mIoU and the accuracy) which can be used to improve the learning of the models. This kind of nuanced understanding is critical to the efficient sorting of waste, which is critical to the goals of our project. Therefore, the use of state-of-the-art deep learning models for semantic segmentation significantly improves the accuracy of waste identification, moreover it allows to get better waste management solutions.

2.2. Applications in Waste Sorting

Machine learning is being used increasingly in classifying objects in images, because it has proven to be effective in achieving precise results. Autonomous recognition systems , precisely, trash sorting and recycling ones, can be examples of this task, consisting on understanding the context of an image. To streamline the sorting and recycling process, apps employ tools which can identify different types of waste such as plastic, paper, metal. This helps to ensure that these materials are efficiently recycled with minimal effort. Garbage sorting systems often use robots for automation [5, 7, 10]. AI-driven computer vision models and semantic segmentation are used to identify and classify various types of material [8], making the process simpler. Technology has come to the aid of humans to make garbage disposal more efficient [4]. Models such as ENet, developed for high-resolution visual recognition tasks with minimal computational resources, have made this even easier and faster [11]. Despite this, further research and analysis is needed for the effective application of these models in garbage sorting process. About deep learning technology, there is still much scope for improvement in this area. Further advances are required to make the identification and classification system more precise and reliable.

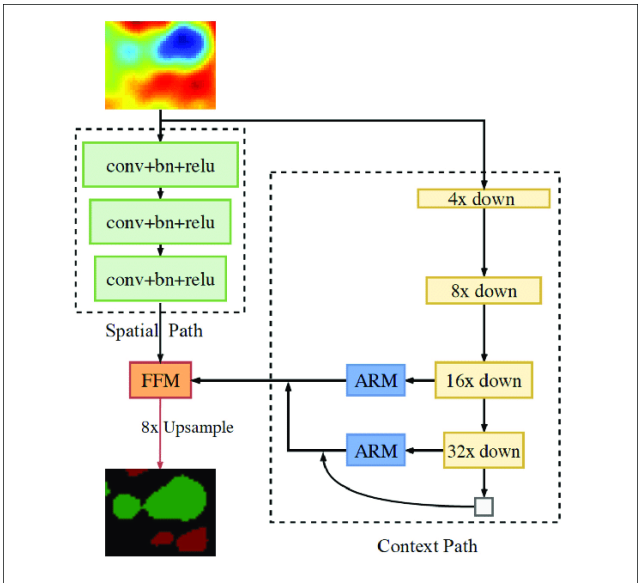

Figure 2. Schema of BiSeNet

3. Proposed Method

In this section we explain our approach to complete the task. We present the chosen networks and we discuss our proposed procedures to obtain from them the configuration that gives the best results.

3.1. Models overview

In developing our waste type detection system, we exploit the power of compact, yet efficient, Convolutional Neural Networks used for semantic segmentation, in particular we focus on ENet, BiSeNet and ICNet. These networks are designed for real-time segmentation tasks and they have proven to be very suitable for resource-constrained platforms, due to their balance of performance and computa-

tional efficiency [11–13, 15]. The synergy of these models improves the accuracy and efficiency of our waste type identification system, thereby making a significant contribution to related efforts in automated waste detection and management. Here there is a closer examination of each network:

- *ENet* is a neural network that has been designed with a focus on real-time semantic segmentation for devices with constrained computational power. It is a small and effective network. [11] It follows the encoder-decoder structure and reduces the amount of computation necessary by a series of early down sampling and late up sampling to produce quick output. Its main benefit is a lean architecture that lowers computational load and parameter count, enabling faster processing speeds.

- The Bilateral Segmentation Network, also known as *BiSeNet*, aims to strike a balance between accuracy and speed. Two paths make up its distinctive architecture: a Spatial Path, that collects high-resolution spatial data. and a Context Path, that extracts a broader context [12, 13]. To provide precise semantic understanding, these two paths operate concurrently and they converge to a final prediction. ResNet-18 or ResNet-101 can serve as the foundation network for BiSeNet, specifically its context paths; depending on the task at hand, we have the option of starting from scratch or using a pre-trained model.

- *ICNet* stands for Image Cascade Network. This network quickly processes high-resolution images while maintaining accurate segmentation results thanks to a multi-scale input processing strategy [15]. This network makes use of an image cascade structure and a cascade fusion unit to combine multi-resolution features coming from various stages. ResNet-50 or ResNet-101 can be used as the backbone, with the option of using a pre-trained model.

3.2. Model selection

The presented networks were initially trained for binary segmentation with the objective of detecting the objects in the images. We kept an eye on important characteristics of each network, such as the size, FLOPs, FPS and the number of parameters. As we developed a solid binary segmentation foundation, we advanced to instance segmentation, enhancing the networks' capacity to categorize different materials in the images. We did some experiments with various configurations, which gave us the opportunity to strike a balance between model's performance, computational efficiency and size, in order to accomplish the goal of the project. In the first phase, we trained the three networks to select the one that would have worked well while maintaining a small size and a high speed (by analyzing the values of the FPS); these characteristics are critical for the goal of developing a system that works in real time on devices with small memory capacity. We obtained satisfactory results with all three networks. ICNet turned out to be a satisfactory network, however we thought its characteristics far exceeded the limit for our goal. ENet, despite maintaining a small size, obtained good results in binary segmentation, but lost a lot by switching to segmentation by instances. We though about how to improve ENet for the convenience of starting by a very small network in our project, so we tried to modify its structure, adding one additional layer and increasing the number of channels in the hidden ones. The unsatisfactory results on ENet led us to choice to work on BiSeNet, tested with two different backbones, Resnet18 and Resnet101. We decided to start with these backbones already trained, this means that their weights were already initialized thanks to the train on a large dataset of images; this fact could make the model up to better generalization and domain adaptation.

3.3. Loss Functions

We relied on the Cross Entropy loss function both for the binary and instance segmentation. This is a commonly used loss function in machine learning and deep learning models, particularly for classification tasks where we want to minimize the distances between the predicted probabilities of our model and the ground truths . [2, 14]

- *Binary Cross-entropy Loss* This is employed when the model's output is binary, that is, when the task is to predict one of two potential classes. The computation for a binary cross-entropy loss for an individual sample is expressed as:

$$L = -y \cdot \log(p) - (1 - y) \cdot \log(1 - p) \qquad (1)$$

  In this formula, y stands for the actual label either positive or negative(1 or 0), and p is the model's predicted probability that the sample would be labeled as 1. Conceptually, this equation, before taking an average across the samples, computes the logarithm of the anticipated probability if the true label is 1, and the log of one less the expected probability if the true label is 0.

- *Categorical Cross-Entropy Loss* This loss is suitable for situations where the model's output is categorical with more than two distinct labels. The Categorical Cross-Entropy loss for a single sample, for a classification task with i classes, may be calculated as follows:

$$L = -\sum_{i} (Y_i \cdot \log(p_i)) \qquad (2)$$

Table 2. Binary segmentation

Models name	Input size (pixels)	Binary (mIoU)	FLOPS (Giga)	N Parameters	Model Size (Mb)
ENet	224x448	0.81	1.3	363260	1.63
ICNet (resnet 50)	250x250	0.85	9.64	28168072	101
BiSeNet (resnet 18)	224x448	0.82	4.65	12410754	47.4
BiSeNet (resnet101)	224x448	0.85	16.90	50337750	202

Table 3. classwise mIoU for each model

Models Name	Background (mIoU)	Aluminuim (mIoU)	Paper (mIoU)	Plastic bottle (mIoU)	Nylon (mIoU)	average mIoU
ENet	0.94	0.46	0.51	0.60	0.50	0.61
ICNet (resnet 50)	0.97	0.60	0.63	0.71	0.69	0.72
BiSeNet (resnet 18)	0.96	0.65	0.69	0.71	0.70	0.75
BiSeNet (resnet 101)	0.98	0.70	0.76	0.79	0.76	0.79

In this equation, the summation is over the five classes i, y_i denotes the true label for class i (which takes the value 1 for the correct class and 0 for all others), and p_i represents the predicted probability of the sample belonging to class i.Essentially, this equation computes the product of the true label and the logarithm of the predicted probability for each class, and then sums these for all the classes.

- *Weighted Categorical Cross-Entropy Loss* His is a special case of the previous categorical cross-entropy loss. This is especially helpful when working with data sets that are unbalanced and may have underrepresented classes. The Weighted Categorical Cross-Entropy loss for a single sample in a problem with five classes may be calculated as follows:

$$L = -\sum_{i}(w_i \cdot y_i \cdot \log(p_i)) \tag{3}$$

In this case W_i stands for the weight related to class i. The weights can be chosen using any appropriate criterion, such as the inverse frequency of the class's occurrence in the data set. For example, in our case involving images, where the background class is present in each image while the other classes are more less present in a quarter of the images, the weight for the background class can be set lower while the weights for the other classes can be set higher to account for their under-representation. Basically, the first step in this equation is to calculate the product of the weight given to the class, the actual label, and the logarithm of the anticipated likelihood for each class, and then to add these products for all classes. This aids in adjusting the model's learning process to accommodate for the dataset's unbalanced class representation.

3.4. Improvement criteria and hyperparameters tuning

With the goal of improving the learning for the models, we augmented the Resortit dataset applying random crop and random flip transformations to the images. These transformations helped us to induct variations on the training data, allowing the model to generalize better without the focus on, for example, the objects orientation. In order to have a better generalized model we relied on cross validation.. As mentioned, our choice of the final model for this task fell on BiSeNet (with Resnet-18), but we continued to do analysis also on ENet. BiSeNet had on average good performance, but we saw that its prediction performance in terms of mIoU where not uniform between all the classes, in particular it didn't recognize the aluminium as well ass the others 3 classes. This fact is extremely clear with ENet and it could perturb the predictions on images outside the Resortit dataset, not permitting to the model to generalize. We decided to rely on the weighted cross entropy loss function based on the frequencies distribution of the different classes in the training set. We noticed that there were a lot of images where more than one class was represented. To overcome the issue we sensibly updated the weights of the cross entropy, also halving the weight corresponding to the the image's background class making the model able to focus better on the waste classes. With this new loss function we achieved better generalized results on each class mIoU on BiSeNet (Table 5), moreover it leaded to improve overall accuracy. Focusing more on BiSeNet, we looked for the best hyperparameters values doing fine tuning basing on the learning rate, the weight decay and momentum. We based the tuning on maximizing the mean accuracy and, at the same time, on further reducing the loss coming out from the validation on the test set each training. We set the object

Table 4. Hyperparameters Tuning

Model	Parameters	Values Range	Step
BiSeNet Resnet18	Learning Rate	0.001→0.035	0.003
	Weight Decay	1e-5, 1e-4	None
	Momentum	0.8 → 0.9	0.01

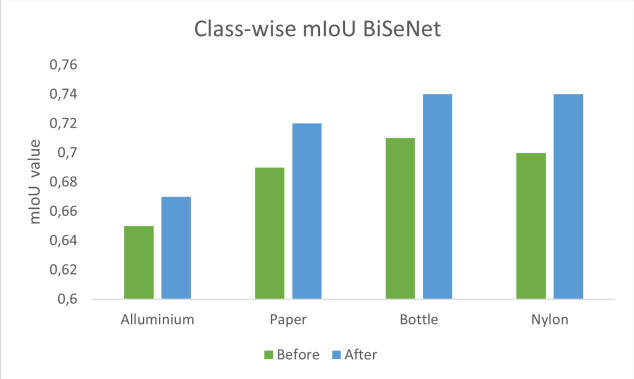

Figure 3. mIoU for each class using BiSeNet

of our minimization problem to the difference between the loss and the accuracy founding the best parameters.

3.5. Static quantization

The last issue we had was the necessity of reducing the size of BiSeNet. We adopted the quantization technique, specifically the post training static quantization. This particular method is used after the training to compact the model without sacrificing its performance too much. The quantization is based on storing the input tensors with lower precision, changing the type of some or all the layers of the network from floating point with 32bit to integer with 8 bit. This technique allows us to have a more compact model with reduced size and memory bandwidth requirements, approaching to a size of 10 MB starting from one of 49 MB. We applied the quantization on all the layers of the context path in BiSeNet, including those used for batch normalization and ReLu activation; moreover some activation layers could be fused into the preceding layer where it is possible. We obtained a model with a size that was the half of the initial one (23 MB) getting very similar accuracy.

4. Experimental Results

This section delves into the specific findings from the BiSeNet network selection and upgrade phases. This contains in-depth explanations of the attributes of the suggested models.

The networks were trained for 20 epochs for instance segmentation and 10 epochs for binary segmentation due to computing capacity restrictions. We trained ENet with a learning rate of 0.0005 and a train weight decay of 2e-4, Adam optimizer applied to the Categorical Cross Entropy loss function. It displayed a remarkable mIoU of 0.81 with 32 frames per second. The model's efficiency was highlighted by the fact that there were a low number of parameters (363,260 parameters) and a low FLOP value (1.3 G); as we said, ENet was the best network in terms of model size (1.63 MB). We carried over the same model parameters to the instance-level segmentation. The mIoU results largely varied across different types of waste as aluminium (0.46) and bottle (0.64), therefore the average mIoU was low (0.61). Despite this low performance, the model performed effectively especially in light of its small size and high computational speed.

ICNet returned a mIoU of 0.85 doing the binary segmentation task and an average mIoU of 0.72 doing instance segmentation. Despite these results, the number of parameters, the FLOPS (9.64 G) and, therefore, the high model size (101 MB), made it uncomfortable for our initial purpose.

The initial tests on BiSeNet were carried out with a ResNet-101 backbone. Although this configuration gave better results both in the binary task (0.85) and in the second task (avg mIoU=0.79), the 202MB model size was much larger than our ideal target size. Consequently, we shifted our focus to BiSeNet with a ResNet-18 as context path. This backbone was chosen due for the smaller size (44,7MB) and for the low value of FLOPs (4.65G), with an high value of FPS (92), that made it very fast on processing data. In order to perform binary segmentation, BiSeNet was trained with a learning rate of 0.005 on SGD (Stochastic Gradient Descent) optimizer and the CrossEntropy loss function. This network pratically matched ENet in the first task, with a mIoU of 0.82, and it had approximately better generalized performance as we expected in the second task: Aluminium: 0.65, Paper: 0.69, Plastic: 0.71, Nylon: 0.70. Regarding instance segmentation, BiSeNet outperformed ENet scoring an average mIoU of 0.75. We tuned BiSeNet trying to find the best learning rate (from 0.001 to 0.035), train weight decay (1e-5 and 1e-4) and momentum (0.8 and 0.9) for the SGD optimizer launching different trials in which we trained it for 15 epochs with 10 images for batch size doing cross validation, so we tested it at each iteration on the validation set, at the end on the test set. BiSeNet had very better results in accuracy (0.83) and class-wise mIoU than in the first configuration (Table 5), holding as generalization as possible on the identification of the dif-

Table 5. classwise mIoU for BiSeNet with weighted cross-entropy loss

Model	Background (mIoU)	Aluminuim (mIoU)	Paper (mIoU)	Plastic bottle (mIoU)	Nylon (mIoU)	Average (mIoU)	Model Size (Mb)
BiSeNet resnet 18	0.99	0.67	0.72	0.74	0.74	0.77	23

ferent materials with an average mIoU of 0.77. The best configuration of BiSeNet (Resnet-18) achieved by the addition of the weights to the Cross Entropy loss, successively by the hyperparameters tuning, leaded to an high accuracy score during the training(0.86) and a better classwise mIoU (Figure 3) with 20 epochs using a learning rate of 0.025, a weight decay of 1e-4 and momentum 0.9.

5. Discussion

Despite we did not reach the ambitious memory limit, our studies have shown, that combining lightweight qualities with high accuracy in garbage picture categorization on hardware with limited resources has a lot of promise. BiSeNet has been quantized to bring it near to the desired memory barrier, while ENet had lightweight qualities that made quantization unnecessary. Weighted loss functions and data augmentation methods have been used to improve the models' capacity in fitting different classes, which led to increased accuracy and system robustness. Also the hyperparameter fine tuning ,using cross validation technique, allowed us to optimize the models' performances. There are some procedures that could further improve our results:

- *Self supervised pre-training* provides effective representations for downstream tasks without requiring labels that could imporve models' robustness and uncertainty [3]

- Other *backbones* smarter then those presented could be used for ICNet and BiSeNet, such as MobileNetV2, observing the size and the performance. Moreover we though about ICNet with Resnet-18 but due to time limitations we coudn't explore [7]

- *Pruning* is the process of carefully deleting pointless connections or filters from a model to lower the amount of parameters and execution-related memory needed. We could have created a more compact and lightweight version of our model by using pruning in order to speed up our model without considerably sacrificing accuracy. [6]

- *Another dataset* such as TACO may have offered a wider viewpoint and greater metric of generalization for our models, allowing for a more thorough examination of our solutions in a larger context and other domains.

The investigation performed by the underlying study will provide the readers with valid notions to understand the approaches adopted by deep learning in waste picture categorization and to gain insights on how these processes are able to overcome the core issues that mainly burden the restricted hardware resources while providing very accurate final results.

Source code here GitHub for further details

References

[1] Anurag Arnab, Ondrej Miksik, and Philip H. S. Torr. On the robustness of semantic segmentation models to adversarial attacks. 2018.

[2] Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-balanced loss based on effective number of samples, 2019.

[3] Dan Hendrycks, Mantas Mazeika, Saurav Kadavath, and Dawn Song. Using self-supervised learning can improve model robustness and uncertainty. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[4] Liu Jingyi, Balatti Pietro, Ellis Kirsty, Hadjivelichkov Denis, Stoyanov Danail, Ajoudani Arash, and Kanoulas Dimitrios. Garbage collection and sorting with a mobile manipulator using deep learning and whole-body control. In *2020 IEEE-RAS 20th International Conference on Humanoid Robots (Humanoids)*, 2021.

[5] Maria Koskinopoulou, Fredy Raptopoulos, George Papadopoulos, Nikitas Mavrakis, and Michail Maniadakis. Robotic waste sorting technology: Toward a vision-based categorization system for the industrial robotic separation of recyclable waste. *IEEE Robotics & Automation Magazine*, 2021.

[6] Hao Li, Asim Kadav, Igor Durdanovic, Hanan Samet, and Hans Peter Graf. Pruning filters for efficient convnets, 2017.

[7] Xueying Li and Ryan Grammenos. A smart recycling bin using waste image classification at the edge, 2022.

[8] Jingyi Liu, Pietro Balatti, Kirsty Ellis, Denis Hadjivelichkov, Danail Stoyanov, Arash Ajoudani, and Dimitrios Kanoulas. Garbage collection and sorting with a mobile manipulator using deep learning and whole-body control. In *2020 IEEE-RAS 20th International Conference on Humanoid Robots (Humanoids)*, pages 408–414, 2021.

[9] Tianjian Meng, Golnaz Ghiasi, Reza Mahjourian, Quoc V. Le, and Mingxing Tan. Revisiting multi-scale feature fusion for semantic segmentation, 2022.

[10] Md. Shahariar Nafiz, Shuvra Smaran Das, Md. Kishor Morol, Abdullah Al Juabir, and Dip Nandi. Convowaste: An automatic waste segregation machine using deep learning. 2023.

[11] Adam Paszke, Abhishek Chaurasia, Sangpil Kim, and Eugenio Culurciello. Enet: A deep neural network architecture for real-time semantic segmentation.

[12] Changqian Yu, Changxin Gao, Jingbo Wang, Gang Yu, Chunhua Shen, and Nong Sang. Bisenet v2: Bilateral network with guided aggregation for real-time semantic segmentation, 2020.

[13] Changqian Yu, Jingbo Wang, Chao Peng, Changxin Gao, Gang Yu, and Nong Sang. Bisenet: Bilateral segmentation network for real-time semantic segmentation, 2018.

[14] Zhilu Zhang and Mert Sabuncu. Generalized cross entropy loss for training deep neural networks with noisy labels. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

[15] Hengshuang Zhao, Xiaojuan Qi, Xiaoyong Shen, Jianping Shi, and Jiaya Jia. Icnet for real-time semantic segmentation on high-resolution images, 2018.