Resource Constrained Semantic Segmentation

1st Fabrizio Battiloro
Politecnico di Torino
Student id: s317786
s317786@studenti.polito.it

2nd Lorenzo Mossotto
Politecnico di Torino
Student id: s318971
s318971@studenti.polito.it

3rd Gabriele Sanmartino
Politecnico di Torino
Student id: s313353
s313353@studenti.polito.it

***Abstract*—In this project we address the problem of resource-constrained semantic segmentation for waste sorting. In the first task, object detection, we only want to binary segment objects and non-objects, while in the second task, instance segmentation, we also want to classify the material of the waste, from the set of possible labels. To achieve these, we use three popular tiny semantic segmentation models: ENet, BiSeNet, ICNet. We propose a data augmentation technique, which augments the dataset with different transformations and we explore alternative losses. Moreover, we change the sizes of the models and we try the strategy of self-supervised pre-training. Code available Here.**

I. Problem overview

As the global population continues to grow, the production of waste is also increasing, with estimates suggesting it will reach 2.6 tonnes per year by 2030. Consequently, there is an urgent need for efficient strategies in Material Recovery Facilities, which are centers responsible for sorting collected recyclable waste into separate bales. There is growing pressure for increased and more effective recycling practices. For instance, in the European Union, the European Commission Waste Directive 2018/850 mandates that by 2035, the amount of municipal waste sent to landfills should be reduced to 10% or less. Historically, the treatment of industrial waste has primarily relied on manual sorting to recover valuable and reusable materials. However, this method is labor-intensive and poses potential risks to human workers. In recent times, numerous automated systems have been developed to separate and recover materials, such as metal, paper, glass, and plastic, from waste streams. These systems utilize optical sorters, magnets, eddy currents, as well as inductive and near-infrared sensing technologies. The aim is to enhance the efficiency and safety of the waste sorting process, reducing the need for human involvement.

II. Proposed Approach

The goal of our project is, given a dataset of images, to be able to recognize if each image contains one or more objects, and then to classify the objects detected in their different categories, which are paper, bottle, aluminum and nylon. In particular, the first task we address is the *object detection*, whose aim is to classify individual objects and localize each. Then, we deal with the *instance segmentation* that is challenging because it needs to detect all objects in an image while also precisely segmenting each instance: it combines *object detection*, the same of the previous task, with the *semantic segmentation*, which aims to classify each pixel into a fixed set of categories without differentiating object instances [1]. At first, we try to tackle with these

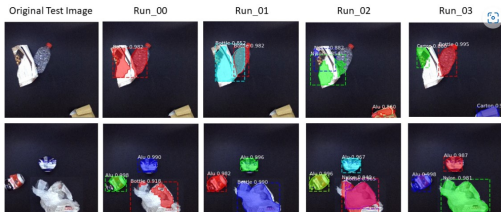

Fig. 1: instance segmentation process

tasks with the ENet model [2], whose main characteristics are lightweight design and fast inference speed. Then, we try to achieve the goal with two other models, ICNet [3] and BiSeNet [4]: these are more complex than ENet and therefore they provide higher accuracy. In order to increase the results, we explore different

loss functions, in particular the Cross Entropy, the Focal Loss and the Weighted Cross Entropy. We try to change the size of the models in order to have different time and accuracy. Moreover, we use Data Augmentation, a technique that creates a more varied dataset starting from the existing data samples and applying different transformations and masks. Another strategy that we test is to improve the model performance using self-supervised pre-training, in particular using rotations.

A. Dataset

The Resort-it dataset is a curated collection specifically designed for the task of instance segmentation. It focuses on the detection and segmentation of five different classes of waste objects in images, which include aluminum, bottle, paper, nylon, and a class denoting the absence of garbage. The dataset comprises

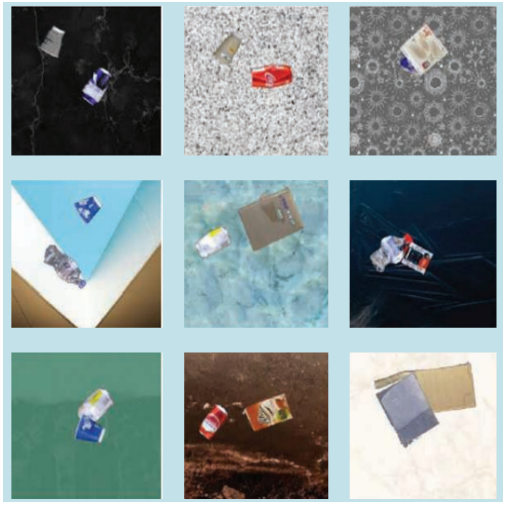

Fig. 2: images from Resort-it Dataset

a total of 5500 training images and 1460 test images, providing a substantial amount of data for model training and evaluation. Each image is composed of a random background with one or more waste objects placed within it. This variability in the number of objects per image allows for the development of models capable of handling multiple instances simultaneously. By utilizing the Resort-it dataset, we can develop and assess instance segmentation algorithms specifically tailored for waste object detection. This dataset offers a valuable resource for advancing the understanding and automation of waste management processes, ultimately contributing to environmental conservation efforts. In order to improve the performance and generalization of instance segmentation models trained on the Resort-it dataset, a data augmentation process is employed. Data augmentation involves applying various transformations to the original images, creating new samples with different variations. This augmentation helps the model learn to handle a wider range of scenarios and increases its ability to accurately detect and segment waste objects. [10] In the training phase, we employ three masks to augment the data: Scale, RandomCrop, and randomHorizontalFlip. The *Scale* technique is utilized to resize the input image while preserving the original aspect ratio. It allows for the exploration of different image dimensions during training. *RandomCrop* randomly selects a portion of the image and crops it to a specific size, introducing variations in object placement to enhance the model's ability to handle diverse scenarios. Additionally, the *randomHorizontalFlip* mask mirrors the image horizontally, facilitating the model's generalization to objects in varying orientations. In contrast, during the validation phase, we rely on two masks: Scale and CenterCrop. The *CenterCrop* mask crops the image from the center, concentrating the model's attention on the most relevant features during validation. By utilizing these specific masks

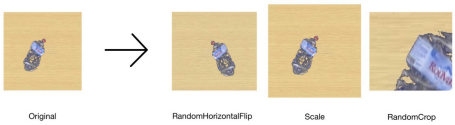

Fig. 3: example of Data Augmentation

for training and validation, we effectively augment the dataset, increasing its diversity and reducing the risk of overfitting.

B. Model

In this paper, we propose a method to apply an instance segmentation process to our dataset. Specifically, We have selected three models capable of performing classification: Enet, BiSeNet, and ICNet. Initially, we tested their ability to recognize objects within images. Once we confirmed their satisfactory performance, we further evaluated their ability to recognize and classify objects across five classes. The three models operate as follows:

- **ENet**: The ENet (Efficient Neural Network) model is a deep learning architecture designed for real-time semantic segmentation of images. It consists of an encoder-decoder structure with skip connections.
  The *Encoder* is responsible for extracting features from the input image. It starts with an initial block that consists of two branches: a convolution branch and a max-pool branch. The convolution branch applies multiple convolutional layers, followed by batch normalization and PReLU activation. The

max-pool branch performs max pooling on the input image. This pooling operation reduces the spatial dimensionality of the feature maps while retaining the most prominent features. Specifically, the max-pool branch performs max pooling with a 2x2 window and a stride of 2. This means that the feature maps are divided into 2x2 regions, and the maximum value within each region is selected. The stride of 2 indicates that the pooling regions move by 2 pixels horizontally and vertically. The outputs of both branches are concatenated to form an output with 16 channels. After the initial block, the encoder consists of several bottleneck modules. Each bottleneck module can have different variations: regular convolution, dilated convolution, asymmetric convolution, or downsampling/upsampling. These bottleneck modules help capture and refine the image features at different scales.

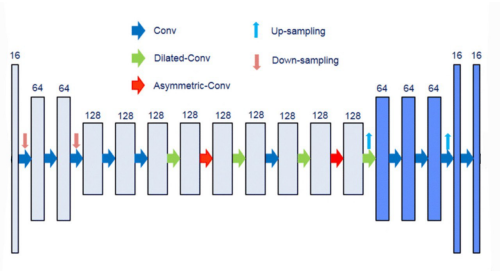

Fig. 4: ENet Architecture

The *Decoder* takes the output from the encoder, which consists of feature maps at a lower spatial resolution but with a higher level of semantic information. The goal of the decoder is to generate a final segmentation map that has the same spatial resolution as the original input image. To achieve this, the decoder in ENet uses a combination of upsampling and skip connections. During the upsampling process, each pixel in the low-resolution feature map is expanded into a larger receptive field in the output feature map. This expansion is achieved by inserting zeros (padding) between the pixels and then applying the convolution operation. The weights of the transposed convolutional layer are learned during the training process and help determine the mapping from the low-resolution to the high-resolution space. The upsampling process aims to increase the spatial resolution of the feature maps.

The ENet model can be trained in two modes: ”encoder” and ”all”. In the ”encoder” mode, only the encoder is trained, and the output of the encoder is upsampled to the desired segmentation map size. In the ”all” mode, both the encoder and decoder are trained jointly to produce the segmentation map directly. By combining the encoder and decoder, the ENet model can effectively perform pixel-level semantic segmentation, assigning a label to each pixel in the input image based on its semantic class.

- **ICNet**: The ICNet Model, *Image Cascade Network*, as ENet is a deep learning architecture designed for real-time semantic segmentation. It has been developed because didn't existed a model which could reach a decent prediction accuracy in the real-time field [3]. ICNet is a multi-resolution model that combines features from different levels of the image pyramid. It has three main branches: a *coarse branch*, a *middle branch*, and a *fine branch*. The *coarse branch* extracts coarse features from the image, the *middle branch* extracts middle-level features, and the *fine branch* extracts fine-grained features. The features from these three branches are then fused together by a Cascade Feature Fusion function in order to produce a final segmentation map.
  Our ICNet model is build on top the *ResNet50* Network which is responsible for extracting features from the input image.

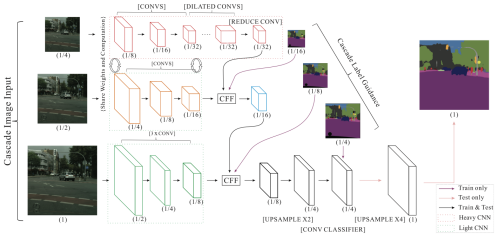

Fig. 5: ICNet Architecture

- **BiSeNet**: BiSeNet, developed in 2018, uses a two-pathway architecture that consists of a *spatial pathway* and a *context pathway*. This allows BiSeNet to extract both fine-grained details and global context information, which is essential for accurate semantic segmentation. In particular the *spatial pathway* is responsible for extracting fine-grained details from the input image by using a series of convolutional layers with small kernel sizes; while the *context pathway* is responsible for capturing global context information by using a series of convolutional layers with a larger kernel sizes.
  Our BiSeNet model is build on top the *ResNet* architecture Network which is responsible for extracting features from the input image. In par-

ticular we used *ResNet-18, ResNet-34, ResNet-50* and *ResNet-101* in order to find the more balance solution between Accuracy and Model Weight.

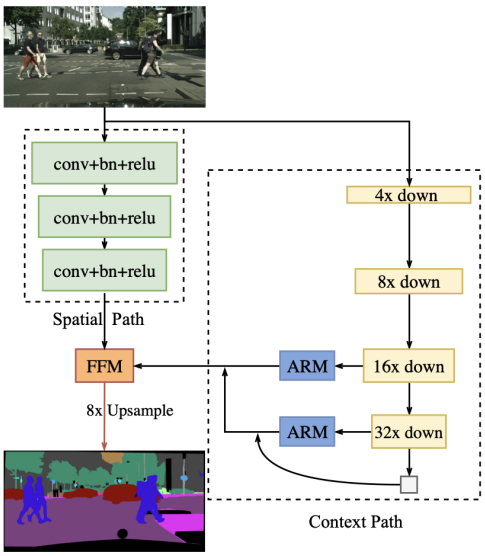

Fig. 6: BiSeNet Architecture

C. Methods to improve the performance

- **Self-supervised pre-training** The first technique that we used is the *Self-supervised rotation* technique. It is a type of self-supervised learning that uses rotation to learn representations of images. The basic idea is to rotate an image by a certain angle and then ask a model to predict the original orientation of the image. This can be done by using a variety of different neural network architectures and in this case we used the *ResNet-18* one in order to transfer this pretrained model as backbone of our BiSeNet model.
  This procedure aims to train the model on features that are robust to changes in rotation.
  In particular we previously trained the *Self-Supervised* model by creating a dataset of rotated unlabelled images from the original dataset by rotating the images by 0, 90, 180, and 270 degrees. Then we saved the model and imported it inside the project repository.
  In order to use it for the BiSeNet Architecture, we had to remove the fully connected layers of our pre-trained model and we used the remaining layers to extract features from the original dataset. Unfortunately, as it is possible to see in the results parapraph, this technique didn't produce the expected effect.
- **Data augmentation** In our code we add different image transforms which are used for *data augmentation* [5] in the context of semantic segmentation. *Data augmentation* is a technique which aims to increase the size and the variety of the training dataset by applying random transformations to the input images and their corresponding labels, in order to improve the generalization and robustness of the trained models and, as a consequence, reduce overfitting. In particular, the transforms we added later to increase performance are: *CenterCrop*, which crop the image and the mask in the center; *RandomVerticallyFlip*, that randomly flips the image and the mask vertically; *RandomRotation*, which randomly rotates the image and the mask within a range of degrees (we use 45°).
- **Model sizes** In order to improve performance in terms of processing time and result accuracy, we conduct experiments by altering the dimensions of the Enet model. Specifically, we attempt to reduce and increase the model size by adjusting the number of layers it comprises. Through this procedure, we anticipate achieving shorter processing times with smaller dimensions, while obtaining higher accuracy with larger dimensions.
  Moreover, we modify the model size by using different backbone networks for the BiSeNet model. We start with *ResNet-18*, then we gradually use deeper architectures which may capture more complex and abstract features at the cost of increased computational complexity: *ResNet-34*, *ResNet-50* and *ResNet-101* [8], where the number in the name corresponds to the number of layers in the network.
- **Loss functions** We explore and implement different loss functions to the BiSeNet model, because it provides better accuracy than the others. In multi-classification, at first we use the *Cross-entropy* as loss function, which is commonly used as the default function in multi-classification problems, presented in Eq 1 where $y_{i,c}$ represents the ground truth probability of pixel i belonging to class c and $\widetilde{y}_{i,c}$ represents the predicted probability.

$$L_{ce} = -\sum_{i=1}^{N}\sum_{c=1}^{C} y_{i,c}\log(\widetilde{y}_{i,c}) \tag{1}$$

Then, we try to use a modified version of the *Cross-entropy loss*, designed to address the issue of class imbalance: *Focal loss* [6], shown in Eq 2, whose strategy is to down-weight the well-classified examples focusing on hard and mis-classified class, thanks to the term $(1-\widetilde{y}_{i,c})^{\gamma}$.

$$L_{ce} = -\sum_{i=1}^{N}\sum_{c=1}^{C}(1-\widetilde{y}_{i,c})^{\gamma} y_{i,c}\log(\widetilde{y}_{i,c}) \quad (2)$$

Moreover, we implement and use the *Weighted loss* [7], whose functionality is similar to the *Focal loss* and address the problem of unbalanced class with pre-computed class weights based on the pixel frequency of the entire training data. It is presented in Eq 3, where the factor β_c is the weight to give more importance to less frequent classes.

$$L_{ce} = -\sum_{i=1}^{N}\sum_{c=1}^{C}\beta_c y_{i,c}\log(\widetilde{y}_{i,c}) \quad (3)$$

III. Results

In our evaluation process, we employed the average *Intersection over Union* (IoU) metric to assess the performance of our image models [9]. The IoU measures the overlap between the predicted and ground truth bounding boxes. In order to calculate the IoU for a specific class we use the following formula 4:

$$IoU = \frac{n_{ii}}{t_i + \sum_{n_{ji}} - n_{ii}} \quad (4)$$

where n_{ii} represents the number of pixels correctly predicted as class i, t_i represents the total number of pixels of class i in the ground truth, and $\sum_{n_{ji}}$ represents the total number of pixels predicted as class i. To determine the final performance of the models, we considered the highest IoU value achieved across multiple epochs. This approach allows us to identify the epoch in which the models produced the most accurate predictions for the given task. We test at first our three different models in the *object detection* task, in which we only want a binary classification between images with objects and images without. In this case we have *ResNet-18* as the backbone network for BiSeNet and the same loss is used for the three models: *Binary Cross-Entropy with Logits Loss*. The accuracies for this first task are displayed in Table I, where we can see that the results are very similar for all the models. In the table are also presented the *model size*, the *FLOPs* (Floating Point Operations), which is a measure of the computational complexity of a neural network often used as a proxy for inference time, and *parameters*, that is a measure of capacity or complexity and can affect both training time and generalization performance. As we expected, these values in ENet are lower than in the other models, because of its lower complexity.

As regards the second task, we use the same three models as the previous one, but adapting them to multi-classification problem. The new results are reported in table II. For the ENet model we reduce and increase the model size by adjusting the number of layers and, as a result, higher accuracy corresponds to higher number of layers. Moreover, also for the BiSeNet model we use different backbone networks with different model sizes. In particular, we implement *ResNet-18*, *ResNet-34*, *ResNet-50* and *ResNet-101*, which in this list are ordered from the smallest to the largest model size. Logically, a larger model size is a consequence of the larger number of layers and usually matches to a higher accuracy. Using different backbone networks for the BiSeNet model, we obtain the best accuracy with the *ResNet-50*, even if the *ResNet-101* has a larger model size.

Model	Mean iou	Model size	FLOPs	Params
ENet	0.8648	1.3862	25.43G	363.13K
ICNet	0.8668	107.9234	452.79G	26.24M
BiSeNet	0.8665	48.8113	178.40G	13.39M

TABLE I: Results in object detection.

Model	Mean iou	Model size
ENet	0.6574	0.7931
ENet(more layers)	0.6741	1.3862
ICNet	0.7064	107.9234
BiSeNet(ResNet-18)	0.7135	48.8113
BiSeNet(ResNet-34)	0.7346	89.6346
BiSeNet(ResNet-50)	0.7420	111.3055
BiSeNet(ResNet-101)	0.7354	183.7547

TABLE II: Results in multi-classification.

More detailed values are presented in table III, where are displayed the accuracies corresponding to the five class. As we can see from these results and without considering *no garbage* class, the *aluminum* class has the hardest classification, in fact its accuracy is the lowest for each model. Meanwhile, *bottle* class seams to be the most well-classified, because it has the best accuracy in ENet and BiSeNet, while in ICNet is lower than *nylon* class, even if only slightly. In each model, the *no garbage* class has a very high accuracy, very close to 1.

Class	ENet	ICNet	BiSeNet(ResNet-18)
No garbage	0.9887	0.9886	0.9888
Aluminum	0.4784	0.5745	0.5870
Paper	0.6113	0.6501	0.5993
Bottle	0.6842	0.6513	0.7055
Nylon	0.6080	0.6675	0.6871
Avg mIoU	0.6741	0.7064	0.7135

TABLE III: Results for each different class in multi-classification.

In table IV we present the accuracies of the multi-classification for the BiSeNet model, with different methods that we implement to improve the performance. In the first three lines ("Cross-entropy loss",

"Focal loss" and "Weighted loss") there are the results obtained by using the three different loss functions presented above. Comparing these, we obtain the best accuracy with the Focal loss. The "Data-augmentation" row presents the result obtained adding new image transforms, where we have the best accuracy for the BiSeNet model.

BiSeNet	Mean iou
Cross-entropy loss	0.7135
Focal loss	0.7260
Weighted loss	0.7200
Data-augmentation	0.7443
Self-supervised	0.6796

TABLE IV: Accuracies with different methods in BiSeNet model.

IV. Conclusion

In this work, we address the challenge of resource-constrained semantic segmentation for waste sorting using three models: ENet, ICNet, and BiSeNet. Our findings indicate that the BiSeNet model achieves the highest accuracy. Through various experiments presented in this paper, we demonstrate that employing data augmentation techniques or utilizing loss functions that specifically address class imbalance issues significantly improves performance. Additionally, although we observe that increasing the number of layers and the model size generally leads to higher accuracy, as evidenced by our exploration of different backbone networks for the BiSeNet model, we can notice that for small improvements the model weight increase consistently. It is worth noting that the accuracy is influenced by the material type, as each class behaves similarly across all models. The only method which didn't achieve the expected result is the pre-trained model which uses the rotation Self-supervised techniques.

Overall, we can affirm that to attain improved results, the recommended approach is to persist with Data Augmentation and carefully select a suitable Miou/weight backbone network.

References

[1] @inproceedingshe2017mask, title=Mask r-cnn, author=He, Kaiming and Gkioxari, Georgia and Dollár, Piotr and Girshick, Ross, booktitle=Proceedings of the IEEE international conference on computer vision, pages=2961–2969, year=2017

[2] @articlepaszke2016enet, title=Enet: A deep neural network architecture for real-time semantic segmentation, author=Paszke, Adam and Chaurasia, Abhishek and Kim, Sangpil and Culurciello, Eugenio, journal=arXiv preprint arXiv:1606.02147, year=2016

[3] @inproceedingszhao2018icnet, title=Icnet for real-time semantic segmentation on high-resolution images, author=Zhao, Hengshuang and Qi, Xiaojuan and Shen, Xiaoyong and Shi, Jianping and Jia, Jiaya, booktitle=Proceedings of the European conference on computer vision (ECCV), pages=405–420, year=2018

[4] @inproceedingsyu2018bisenet, title=Bisenet: Bilateral segmentation network for real-time semantic segmentation, author=Yu, Changqian and Wang, Jingbo and Peng, Chao and Gao, Changxin and Yu, Gang and Sang, Nong, booktitle=Proceedings of the European conference on computer vision (ECCV), pages=325–341, year=2018

[5] @articleperez2017effectiveness, title=The effectiveness of data augmentation in image classification using deep learning, author=Perez, Luis and Wang, Jason, journal=arXiv preprint arXiv:1712.04621, year=2017

[6] @INPROCEEDINGS8519409, author=Doi, Kento and Iwasaki, Akira, booktitle=IGARSS 2018 - 2018 IEEE International Geoscience and Remote Sensing Symposium, title=The Effect of Focal Loss in Semantic Segmentation of High Resolution Aerial Image, year=2018, volume=, number=, pages=6919-6922, doi=10.1109/IGARSS.2018.8519409

[7] @InProceedingsWeightedLoss, author = Liu, Qinghui and Kampffmeyer, Michael C. and Jenssen, Robert and Salberg, Arnt-Borre, title = Multi-View Self-Constructing Graph Convolutional Networks With Adaptive Class Weighting Loss for Semantic Segmentation, booktitle = Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, month = June, year = 2020

[8] @InProceedingsResNet, author = Mehta, Dushyant and Skliar, Andrii and Ben Yahia, Haitam and Borse, Shubhankar and Porikli, Fatih and Habibian, Amirhossein and Blankevoort, Tijmen, title = Simple and Efficient Architectures for Semantic Segmentation, booktitle = Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, month = June, year = 2022, pages = 2628-2636

[9] @articleeveringham2015pascal, title=The PASCAL Visual Object Classes Challenge: A Retrospective, author=Everingham, Mark and Eslami, SM Ali and Van Gool, Luc and Williams, Christopher KI and Winn, John and Zisserman, Andrew, journal=International Journal of Computer Vision, volume=111, number=1, pages=98–136, year=2015, publisher=Springer, doi=10.1007/s11263-014-0733-5,

[10] @article Simard, Patrice, David Steinkraus, and John C. Platt. "Best practices for convolutional neural networks applied to visual document analysis." In Proceedings of the Seventh International Conference on Document Analysis and Recognition, vol. 2, pp. 958-963. IEEE, 2003.