Resource Constrained Semantic Segmentation for Waste Sorting

1st Snigdha Das
s316597

2nd Fatemeh Zahiri Koupaei
s295942

3rd Rahul Kumar Sahoo
s316411

***Abstract*—We focused on overcoming the limitations of resources to accomplish precise and efficient semantic segmentation for waste sorting. One of the most necessary problems in refining waste management systems is resource-constrained semantic segmentation for waste sorting. To address this issue of resource constraints, we developed model architectures, model compression, distillation, and hardware optimization. All these approaches can be used to create resource-constrained semantic segmentation models, permitting automated waste sorting with lessened processing needs. We have worked on the stimulating contest of real-time semantic segmentation, which has practical applications but requires expensive pixel-wise labels. To tackle this issue, we looked into three types of models which are an image cascade network (ICNet), a bilateral segmentation network (BiSeNet), and an ENet with multi-resolution branches focused on suitable labels. This study delivers a depth analysis of the system and presents the cascade feature arrangement unit to rapidly achieve high-quality segmentation. One of the importances of these projection methods is demonstrated by the assessment of interesting datasets like implementation and cityscapes.**

Fig. 1. Detecting and localizing waste objects

I. Introduction

In this paper, we explore the complexity of trash management, where typical artificial intelligence approaches fall short and computational resources are infrequent. By generating a cutting-edge framework that merges the art of garbage sorting with effective data processing, our goal is to get around these restrictions. Our method, which is influenced by the natural world, holds both minimalism and modification. We use lightweight network designs that are precisely designed for sorting, which permits us to achieve accurate semantic segmentation with little computational effort. We push the limits of efficiency in garbage sorting algorithms by utilizing the fundamental qualities of waste items. Furthermore, we examine the ideas of few-shot learning and transfer learning to maximize the potential of our scant annotated training data. Using this novel method enables our system to learn from a little dataset, minimizing the need for broad interpretations and speeding up the waste sorting procedure. We adopt the upper-hand computing and hardware optimization principles to close the gap between theory and practice. We offer real-time waste sorting with unique effectiveness by logically distributing computation across edge devices and utilizing hardware accelerators. Our system helps to push the limits of resource usage and becomes a model of sustainable computing. At last, our research pursues to transform garbage sorting procedures by providing a durable resolution that incorporates technology and environmental stewardship. Our resource-reserved semantic segmentation approach sets the way for a greener future where waste management becomes a unified and sustainable process by incorporating the principles of straightforwardness, efficiency, and flexibility influenced by the natural world, embraces. The computerized sorting of waste items is necessary for the development of sustainable waste management practices. Nevertheless, the garbage sorting methods now in use mostly depend on computationally semantic segmentation algorithms. We propose a novel approach for resource-constrained semantic segmentation for waste sorting to overcome this problem.

II. Related Work

A. High-Quality Semantic Segmentation

The precise and almost accurate explanation of objects inside an image at a certain pixel level is referred to as excellent semantic segregation. Deep learning intricacy to expand the receptive field for compressed labeling. In which each pixel is assigned a class whose semantic class fits. The objective is to deliver segmentation results t are nearby true as practical, with as rare inaccuracies or conflicts as possible.

B. High-Efficiency Semantic Segmentation

High-Efficiency is the process of properly arranging and segmenting objects within an image, with a focus on receiving highly efficient performance in terms of both accuracy and efficiency. Semantic segmentation includes a specific class

label for each pixel in an image, therefore distinctive objects or regions. It is broadly used in plenty of applications such as medical analysis, object recognition, and autonomous driving.

III. Image Cascade Network

While high-quaSub-network segmentation has evolved significantly, research along the line to make semantic segmentation run fast while not sacrificing too much quality has been left behind. These methods can inspire or enable many practical tasks in automatic driving, robotic interaction, online video processing, and even mobile computing. We make comprehensive consideration of the two factors of speed and accuracy. According to the time budget analysis, we adopt intuitive speedup strategies to reduce running time, including downsampling input, shrinking feature maps, and conducting model compression. The proposed system image cascade network (ICNet) [1] does not simply choose either way but takes cascade image inputs. Even though the top branch usage, the input resolution is short, resulting in limited computation. The bottom branch uses even fewer layers but still achieves good-quality segmentation. We propose a cascade feature fusion (CFF) unit, which combines cascade features from different-resolution inputs by up-sampling and dilated convolution. This unit uses small kernels, compared to deconvolution, which causes more computation.

A. Network Architecture

- **Image Cascade**: ICNet [1]functions on various scales instantaneously. It takes an input image and produces three scales: (i) full resolution (ii) half-resolution (iii) quarter-resolution. Respective scale captures diverse levels of detail and serves as input to the subsequent sub-networks.
- **Pylon Sub-network**: The pylon sub-network operates on the full-resolution scale image. It consists of a standard convolutional neural network (CNN) with a large receptive field. This subnetwork captures high-level semantic information and generates an initial segmentation map.
- **Pyramid Subnetwork**: This sub-network operates one of the pylon sub-network i.e. half resolution scale image. [2] [3]It joins expanded convolutions, which increase the accessible field without down-sampling the spatial resolution. Here, the sub-network captures context information at various scales and produces a refined segmentation map. The pyramid sub-network operates on the half-resolution scale image. It incorporates dilated convolutions, which increase the receptive field without downsampling the spatial resolution. This sub-network captures context information at multiple scales and produces a refined segmentation map.
- **Cascade Feature Fusion:** This segmentation operates segmentation by generating using both pylon and pyramid sub-networks. In which the sub-networks fuse to create an intermediate segmentation map. This fusion is achieved at the corresponding spatial resolution.
- **Refinement Sub network**: The modified sub-network takes the intermediate segmentation map as an input. Then, further improves the segmentation boundaries. It consumes a lightweight CNN [4]with a smaller receptive field, making it efficient.
- **Final Segmentation Map**: The result of the refinement sub-network is the final segmentation map. In this, each pixel is assigned [1] a class label based on the object or region it belongs to.

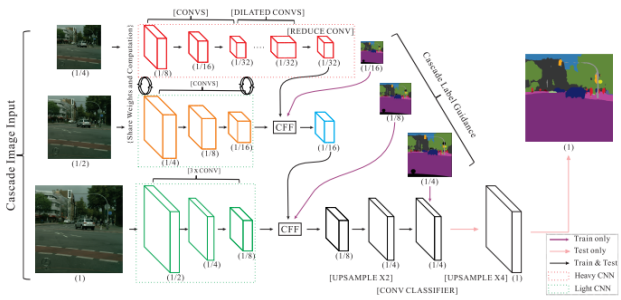

Fig. 2. Network architecture of ICNet. 'CFF' stands for cascade feature fusion detailed in Sec. 3.3. Numbers in parentheses are feature map size ratios to the full-resolution input. Operations are highlighted in brackets. The final ×4 upsampling in the bottom branch is only used during testing.

In ICNet class we initialized the model by defining its component it is a sequence of three convolutional layers used for processing the input at a lower resolution. self.ppm is an instance of the PyramidPoolingModule class, which performs pyramid pooling on the feature maps. self.head is an instance of the ICHead class, which represents the head of the ICNet model. Then in pyramidpooling class It takes the input input and performs adaptive average pooling at different scales specified by self.pyramids. Each pooled feature map is then interpolated back to the original size and added to the original feature map. The final feature map is return then in ichead class here it ombines features from different scales. The method initializes the head by creating two instances of the CascadeFeatureFusion class and a convolutional layer. The forward method performs the forward pass of the head.

It takes the feature maps xsub1, xsub2, and xsub4 as inputs. It passes xsub4 and xsub2 through the first CascadeFeature-Fusion module to obtain xcff24 and x24cls. It passes through the second CascadeFeatureFusion module It upsamples and applies a convolution It upsamples again then The outputs are returned. then in convrelu class it initializes the module with the specified parameters and performs the forward pass of the module by applying convolution, batch normalization, and ReLU activation to the input x. then in cascadefeaturefusion class, the most important class takes the low-resolution feature map low and the high-resolution feature map high as inputs. It upsamples low to the size of high using bilinear interpolation. It applies convolution and batch normalization to low and high. It combines the resulting feature maps and applies ReLU activation. It applies a convolution The combined feature map are returned.

IV. Bilateral Segmentation Network

In this we propose a new architecture based on Bilateral Segmentation Network (BiSeNet) [8] with Spatial Path and Context Path in detail after that, we elaborate on the effectiveness of these two paths correspondingly. After that, we explained the effectiveness of the two paths respectively. At last, we showed how to combine the features of these two paths with the Feature Fusion Module and the whole architecture of our BiSeNet. [6] Semantic segmentation is a fundamental task in computer vision and is used in augmented reality devices, autonomous driving, and video surveillance. There are mainly three approaches to accelerating the model:

- Restrict the input size to reduce the computation complexity.
- Prune the channels of the network to boost the inference speed.
- Drop the last stage of the model.

Researchers widely utilize the U-shape structure to remedy the loss of spatial details mentioned above, but this technique has two weaknesses:

- It reduces the speed of the model
- Most spatial information lost cannot be easily recovered.

Bilateral Segmentation Network (BiSeNet) and elaborate on the effectiveness of Spatial Path and Context Path:

- Spatial Path (SP)
- Context Path (CP)

Bilateral Segmentation Network (BiSeNet) and elaborate on the effectiveness of Spatial Path and Context Path.

- **Spatial path:** Existing approaches to semantic segmentation attempt to preserve the resolution of the input image to encode enough spatial information, but the small size of the input image or lightweight base model damages spatial information. Based on this observation, we propose a Spatial Path that contains three layers [10], including a convolution with stride = 2, followed by batch normalization and ReLU. This path encodes rich spatial information. So that is the major reason the path extracts the output feature maps that are 1/8 of that of the original image.
- **Context path:** The Spatial Path encodes spatial information, and the Context Path provides sufficient receptive field. The receptive field is of great significance for the performance of semantic segmentation. We propose the Context Path, which utilizes a lightweight model and global average pooling to provide a large receptive field and efficient computation simultaneously. The Context Path uses a U-shape structure to fuse the features of the last two stages. We propose a specific Attention Refinement Module (ARM) [6]to refine the features of each stage in the Context Path. ARM participates in the global context information effortlessly without any upsampling operation

In the BiSeNet architecture, the ConvBlock represents a basic convolutional block that consists of a two-dimensional

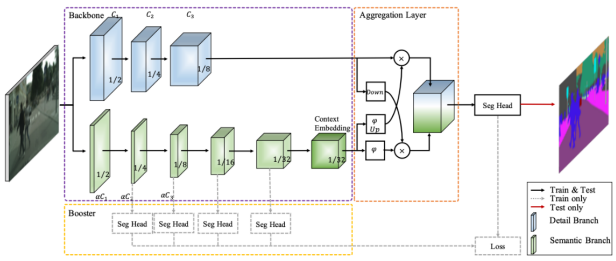

Fig. 3. Overview of the Bilateral Segmentation Network

convolutional layer, batch normalization layer, and rectified linear activation function (ReLU). It is a fundamental building block for the network.

The Spatial path class applies a series of ConvBlock operations to the input tensor. These convolutional operations help extract spatial information from the input image and produce feature maps with reduced spatial dimensions.

The Attention Refinement Module takes an input tensor and applies global average pooling [7], which reduces the spatial dimensions to one and one while preserving channel information. It then performs one convolution followed by a sigmoid activation function. This produces attention weights that are element-wise multiplied with the input tensor, resulting in refined attention-weighted feature maps.

Overall, these modules play crucial roles in capturing spatial information, refining attention, and producing meaningful feature maps in the BiSeNet architecture. we used 'resnet 18' and 'resnet 101' as context paths in our BiSeNet model for that we made a separate python file and a build context path function that takes the name 'resnet 18' or 'resnet 101' and returns an instance of the corresponding model than a random input tensor 'x' of size (1,3,256,256) is generated and is passed through the forward method of each model ('resnet18' and 'resnet 101') to obtain the output feature maps and the tail tensor.

V. Efficient neural network

a) Recent interest in augmented reality wearables, home-automation devices [11], and self-driving vehicles has created a strong need for semantic segmentation algorithms that can operate in real-time on low-power mobile devices. Deep convolutional neural networks have surpassed many conventional computer vision algorithms. Several neural network architectures have been proposed to both spatially classify and finely segment images, but these models have huge numbers of parameters and long inference times. Semantic segmentation is important in applications such as driving aids and augmented reality. Deep neural networks are now one of the most widely used techniques for many different tasks. These techniques are often inefficient and fail to label the classes that occupy fewer pixels in a frame.:

b) We divide our network into several stages, which are highlighted by horizontal lines in the table and the first digit after each block name. The initial stage contains a single

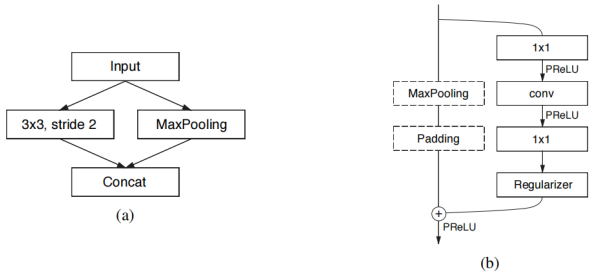

Fig. 4. (a) ENet initial block. MaxPooling is performed with non-overlapping 2×2 windows, and the convolution has 13 filters, which sums up to 16 feature maps after concatenation. (b) ENet bottleneck module. conv is either a regular, dilated, or full convolution (also known as deconvolution) with 3×3 filters, or a 5×5 convolution decomposed into two asymmetric ones.

block, and the following stages are the encoder and decoder. We did not use bias terms in any of the projections and did not use pooling indices in the last upsampling module. The final output has C feature maps, and the last module takes up a sizeable portion of the decoder processing time.:

c) To achieve good performance and real-time operation, it is crucial to realize that processing large input frames is very expensive. We propose a large encoder and a small decoder, which work similarly to the original classification architectures. We tried using ReLU and Batch Normalization layers before convolutions but found that removing most ReLUs in the initial layers of the network improved the results. We replaced all ReLUs with PReLUs, which use an additional parameter per feature map.:

d) Factorizing filters allows decomposing of a large convolutional layer into several smaller ones, and greatly reduces the number of parameters, making them less redundant. Dilated convolutions were used to improve the model by taking a wider context into account, and gave a significant accuracy boost, raising IoU on Cityscapes at no additional cost. Regularization is important for pixel-wise segmentation, as neural networks quickly begin to overfit small datasets. We tried L2 weight decay and stochastic depth, but it turned out that dropping whole branches is a special case of applying Spatial Dropout.: [14]

The provided code defines a neural network model called ENet (Efficient Neural Network) for image segmentation tasks. It consists of an encoder-decoder architecture with skip connections.

1) Brief discussion::

- **InitialBlock**: This block consists of two branches. The convolutional branch contains 13 convolutional layers applied to the input RGB image. These layers are responsible for extracting features from the input. The max pool branch performs max pooling on the input RGB channels, resulting in three output layers. The outputs of both branches are concatenated to produce an output tensor with 16 channels. This block serves as the initial feature extraction step in the ENet model.
- **BottleNeck**: This block represents a bottleneck module in the ENet architecture [12]. It has different variants based on the configuration parameters. The main purpose of this block is to refine and transform the input feature maps. Here are the variants:
- **Regular Convolution:** This variant applies a regular convolutional operation to the input feature maps. It can optionally downsample the feature maps if specified.
- **Dilated Convolution:** This variant applies a dilated (atrous) convolution operation, which introduces holes in the convolutional kernel. The dilation factor determines the spacing between the kernel elements and affects the receptive field of the convolution operation.
- **Asymmetric Convolution:** This variant performs an asymmetric convolution using decomposed filter sizes of 5x1 and 1x5 separately. This operation allows the network to capture different patterns along different spatial dimensions.
- **Upsampling:** This variant is used for upsampling the feature maps. It includes a spatial convolution and a transpose convolution operation to increase the spatial dimensions of the feature maps.
- **Spatial Dropout Regularization:** This block also supports spatial dropout regularization, which randomly drops out individual channels in the feature maps during training, helping to prevent overfitting.

Each bottleneck module consists of a combination of convolutional layers, batch normalization, and activation functions (PReLU or ReLU). These modules play a crucial role in learning and refining the feature representations throughout the network. [12] [13]

VI. Binary Segmentation

a) Binary segmentation is a task that includes an image into two separate expenses : i)foreground (ii)background. The main goal of binary segmentation is to categorize each pixel in an image as either belonging to the foreground or the background. In binary segmentation, the output is either a binary mask or a binary image, where every pixel is allocated a value of 0 or 1, which represents the background and foreground, respectively. :

b) Machine learning techniques are usually engaged to perform binary segmentation, such as deep learning models as in convolutional neural networks (CNNs). These models are trained on a huge dataset of labeled images, where every image is marked with pixel-level ground truth indicating the foreground and background regions. :

c) Once skilled, the binary segmentation model can be functional to new unnoticed images to spontaneously segment the foreground objects from the background. This information can then be used for additional analysis or tasks, such as object detective, image editing, or extracting features for higher-level computer vision tasks.:

d) Binary segmentation is used in waste sorting systems to distinguish different types of waste material based on their characteristics. Waste sorting is an important process in

	OBJECT(mIoU)	FLOPS	parameter	Model size
Enet	0.7993	1555.8	363.13k	1.418 mb
BiSeNet	0.7904	64.84	50.21m	191.94 mb
ICNet	0.8255	28.26	26.24m	100.3mb

TABLE I
BINARY SEGMENTATION

	Paper	Bottle(MIOU)	Aluminium (MIOU)	Nylon (MIOU)	Avg.(MIOU)
ICNet	0.562761	0.528545	0.7215306	0.6398218	0.68502
BiSeNet	0.543737	0.472335	0.5861002	0.1840494	0.552853
ENet	0.5001462	0.6099185	0.513045	0.574	0.636491

TABLE II
INSTANCE SEGMENTATION

recycling facilities and waste management centers to recover valuable resources and reduce the amount of waste sent to landfills.: Here's an overall view of how binary segmentation can be applied to waste sorting:

- **Prepossessing:** the images undergo a prepossessing step to help to enhance image quality, and remove noise, and lighting conditions. These steps provide a lot of help to improve the accuracy of the segmentation process.
- **Segmentation:** To pre-process the images we apply the binary algorithms. These algorithms fully analyze the image step by step, into the segments, pixel by pixel with similar properties into the segments. In this case, the goal is to detach waste objects from the background and each other for the waste sorting process.
- **Feature Extraction:** After the waste objects are segmented, significant features are removed from each segment. These features can be color, texture, shape, and size characteristics.
- **Sorting and Processing:** Centred on the allocated labels, the waste sorting system can through each segment to the suitable processing unit or conveyor belt. This enables additional sorting, recycling, or disposal of the waste materials according to their respective categories.

VII. INSTANCE SEGMENTATION

The speed analysis of the ICNet can differ reliant on various factors. It can be on the basis of hardware setup, implementation optimizations, and size and complexity of the input images.

- **Image Acquisition**: It is similar to binary segmentation; waste items are captured using cameras or sensors in the waste sorting system. Images are attained as waste objects move through the sorting process.
- **Pre-processing:** The attained images might go through pre-processing steps. Such as resizing, normalization, and noise reduction for quality improvement and organizing them for the instance.
- **Object Detection:** This is the algorithm applied to identify and localize individual objects within the image. Popularly this technique includes faster R0CNN, YOLO (You Only Look Once), and SSD (Single Shot multi box Detector). These algorithms are usually an arrangement of deep learning models and anchor-based or anchor-free methods, which detect objects and generate bounding the boxes across them
- **Instance Segmentation:** As soon as objects are identified and confined, instance segmentation algorithms are engaged to assign a unique label and create pixel-level masks for each detected object. One most common use in the instance segmentation approach is Mask R-CNN, which ranges object detection models which have a mask prediction. Here, Mask R-CNN predicts masks for every object instance, enabling pixel-level segmentation.
- **Classification and Sorting:** Once the instance segmentation step, each individual is assigned a label based on its class (e.g., glass, paper, plastic) and is given a unique identifier. Then this information is used to sort the waste objects into appropriate categories for specific processing units.

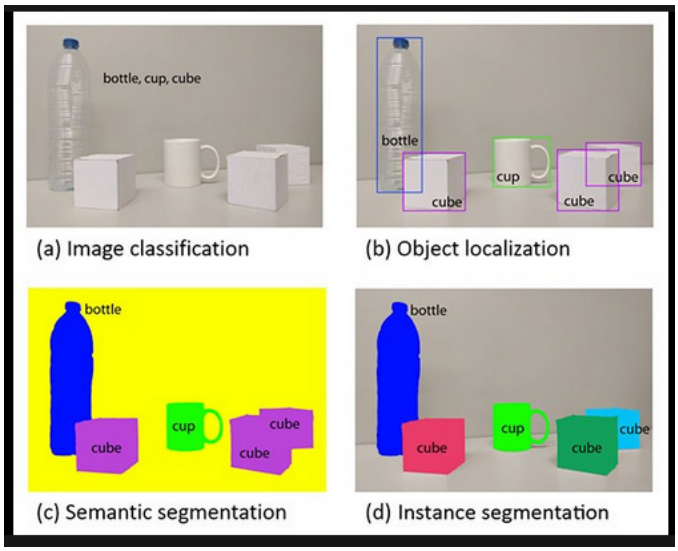

Fig. 5. Quick Understanding: Instance segmentation vs. Semantic segmentation in Image Analysis

VIII. RESULT:

We trained all three models using different values of epochs and we found 15 epochs to be better efficient and had less computational time than others we used only an encoder by keeping in mind the time we have results were quite satisfactory for us according to our model hence here is the table of our observations for binary and semantic segmentation for all three model and four classes. When conducting instance segmentation for waste sorting under resource constraints, several considerations come into play. The first is the selection of an instance segmentation model that balances accuracy and efficiency. While models like Mask R-CNN offer high accuracy, they may be computationally intensive, whereas lighter models like EfficientDet or YOLO prioritize faster inference at the expense of some accuracy. Evaluating different models and choosing one that suits the available resources is crucial.

Additionally, optimizing the chosen model can help reduce computational demands. Techniques such as model quantization, network pruning, and knowledge distillation can reduce the model size and resource requirements without sacrificing performance. Hardware acceleration using GPUs or TPUs can further speed up inference. Data augmentation and preprocessing techniques can enhance model generalization and reduce memory requirements. Leveraging pre-trained models or transfer learning can save training time and resources. Cloud computing or distributed processing can provide access to additional computational power if local resources are limited. By carefully considering these factors and tailoring the approach to the specific resource constraints, effective waste sorting instance segmentation can still be achieved.

IX. Conclusion

To put it briefly, we have projected a resource constraint semantic segmentation. This plays a dynamic role in waste-sorting applications. The aim was to develop segmentation models which can be functioned successfully within the limitation of resources, time, and available hardware. For the real-time problem, we needed real-time solutions, due to which efficiency became a critical factor in waste sorting scenes to become real-time or nearly real-time processing, mainly in automated waste sorting systems. By accruing various techniques, we can achieve high-efficiency semantic segmentation for waste sorting precisely. We believe the optimal balance of speed and accuracy makes our system important since it can benefit many other tasks that require fast scene and object segmentation. It significantly improves the applicability of semantic segmentation in various areas.

Acknowledgment

We would like to express our deepest gratitude to our thesis advisor, Shyam Nandan Rai, for their guidance, support, and invaluable insights throughout the entire research process. Their expertise, patience, and encouragement were instrumental in shaping this thesis and enhancing my understanding of the subject matter. Also immensely grateful to Prof. Barbara Caputo for providing us with the resources, facilities, and funding necessary to conduct my research. In conclusion, this thesis would not have been possible without the support and contributions of the aforementioned individuals and institutions. we truly are grateful for their assistance and proud to have had the opportunity to work with such exceptional individuals.

References

[1] Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: CVPR. (2015)

[2] J. Zhao, H., Shi, J., Qi, X., Wang, X., Jia, J.: Pyramid scene parsing network. In: CVPR. (2017)

[3] Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The cityscapes dataset for semantic urban scene understanding. In: CVPR. (2016)

[4] Treml, M., Arjona-Medina, J., Unterthiner, T., Durgesh, R., Friedmann, F., Schu- berth, P., Mayr, A., Heusel, M., Hofmarcher, M., Widrich, M., Nessler1, B., Hochre- iter, S.: Speeding up semantic segmentation for autonomous driving. NIPS Work- shop (2016)

[5] R. Nicole, "Title of paper with only first word capitalized," J. Name Stand. Abbrev., in press.

[6] Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Semantic image segmentation with deep convolutional nets and fully connected crfs. ICLR (2015)

[7] adrinarayanan, V., Kendall, A., Cipolla, R.: SegNet: A deep convolutional encoder-decoder architecture for image segmentation. IEEE Transactions on Pat- tern Analysis and Machine Intelligence 39(12), 2481–2495 (2017)

[8] Li, X., Liu, Z., Luo, P., Loy, C.C., Tang, X.: Not all pixels are equal: difficulty-aware semantic segmentation via deep layer cascade. IEEE Conference on Computer Vi- sion and Pattern Recognition (2017)

[9] Xie, S., Tu, Z.: Holistically-nested edge detection. In: IEEE International Confer- ence on Computer Vision (2015

[10] Zhang, R., Tang, S., Zhang, Y., Li, J., Yan, S.: Scale-adaptive convolutions for scene parsing. In: IEEE International Conference on Computer Vision. pp. 2031– 2039 (2017)

[11] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," arXiv preprint arXiv:1502.03167, 2015

[12] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," arXiv preprint arXiv:1502.03167, 2015

[13] D. Eigen and R. Fergus, "Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture," in Proceedings of the IEEE International Conference on Computer Vision, 2015

[14] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," arXiv preprint arXiv:1603.05027, 2016.

[15] S Kovács · 2023 ·