Incremental Learning In Semantic Segmentation

Vito Palmisano
s288859@studenti.polito.it
Politecnico di Torino

Valerio Zingarelli
s281586@studenti.polito.it
Politecnico di Torino

Daniele Falcetta
s289319@studenti.polito.it
Politecnico di Torino

Abstract

In this paper, we will present an approach for incremental semantic segmentation task consisting of implementing BiSeNet into a MiB architecture. The former is a Deep Network Architecture made to extract context and spatial features and makes a right balance between the speed and segmentation performance. The latter is a solution to the catastrophic forgetting problem of deep architectures. In the last part of this work, the approach we are presenting is a solution to the problem of weakly supervised semantic segmentation in incremental learning. In particular, we use an approach proposed in SEAM to generate ground truths from class labels and then we train incremental steps using such ground truths. This is an interesting topic since we'll use a model, which has been pre-trained with original VOC12 segmentation labels, to learn incrementally new segmentation classes that have been generated starting from the class labels. Our code is readable online. [1]

1. Introduction

Semantic segmentation is one of the most important problems in computer vision. It requires spatial information, but modern approaches, usually compromise this kind of information to obtain high-speed performances. In such a context, BiSeNet[1] manages to achieve a good balance between accuracy and speed performances using the combination of a context feature extractor and a spatial one that are merged by a Feature Fusion Module. Incremental Learning, on the other side, is a problem usually related to classification and object detection but, in MiB[2], an Incremental Class Learning for semantic segmentation approach is proposed. In this paper, we are using a MiB architecture but with the insertion of BiSeNet, in replacement of DeepLabV3[3], to create a faster incremental model in the semantic segmentation field. Finally, we will analyze the problem of weak supervision in incremental semantic segmentation. We will use an approach based on SEAM (Self-supervised Equivariant Attention Mechanism) [4].

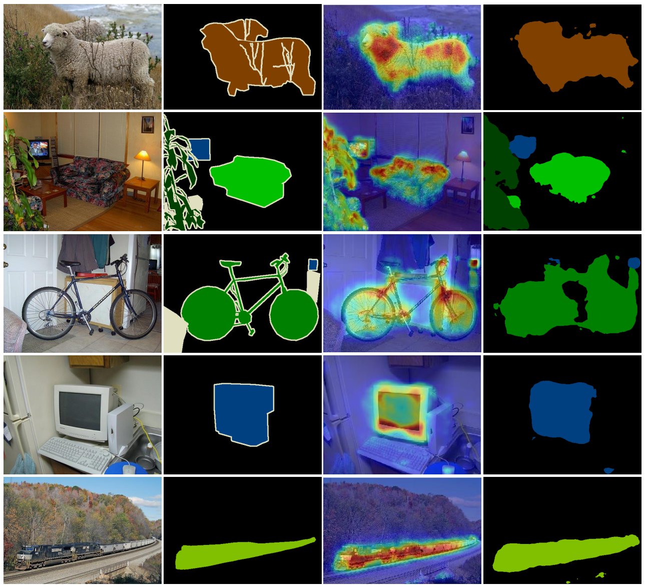
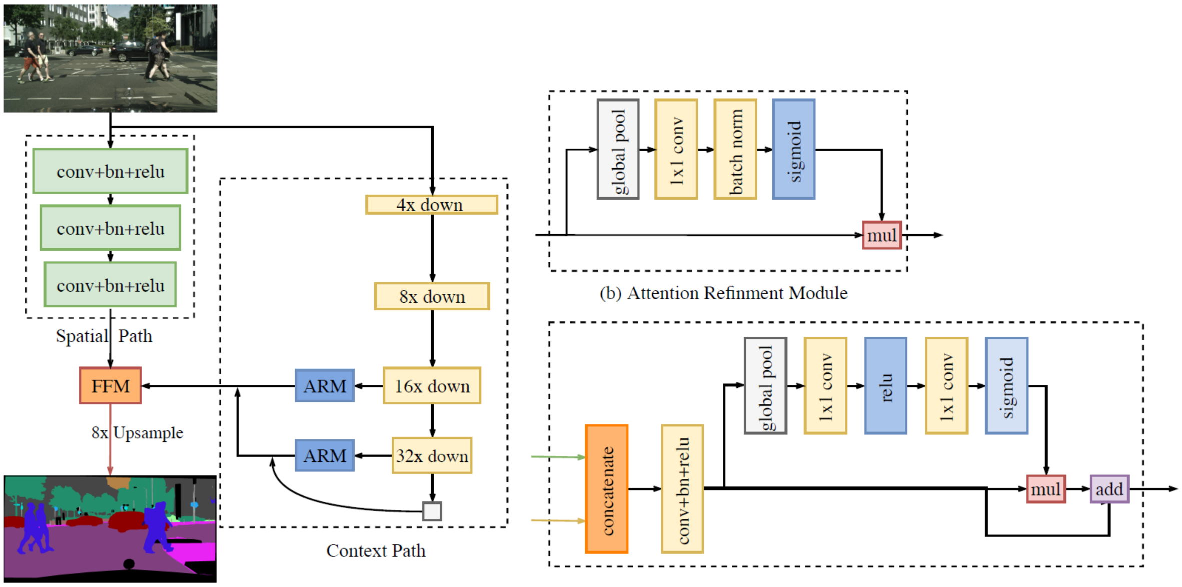


Figure 1. Comparison between original segmentation (left one) and SEAM segmentation (right one). In addition to that, the probability map generated by SEAM and applied on the image is provided too.

SEAM can generate pseudo labels based on an improved version of CAMs (Fig.1) using a specific module called PCM; these pseudo-labels will be used to train the incremental steps of MiB. In this way, we will discover if it's possible to incrementally train a model using images or labels generated by another model.
To summarize, in this paper, we will present three points:

- The study of BiSeNet's performances to find the best possible parameters configuration;
- The implementation of BiSeNet into MiB and its performances in two different scenarios: 15-5 and 15-1;
- Techniques for improving performances in incremental steps when only weak supervision is available.

[1] https://github.com/VitoPalmisano/MiB_BiSeNet_SEAM



(a) Network Architecture

(b) Attention Refinment Module

(c) Feature Fusion Module

Figure 2. Overview of BiSeNet network (Fig.a) with a deep illustration of the Attention Refinement Module (Fig.b) and Feature Fusion Module (Fig.c).

2. Related Works

In this section, we will introduce some works concerning the Semantic Segmentation task, Incremental Semantic Segmentation and generation of pseudo labels for segmentation, starting from class labels.

2.1. BiSeNet

BiSeNet (Bilateral Segmentation Network) is a model for semantic segmentation which can achieve a good trade-off between speed and performance. It's composed of two different paths called Spatial Path and Context Path, which are devised to confront the loss of spatial information and shrinkage of receptive field respectively. Finally, they are merged by a module called Features Fusion Module.

Spatial Path. In semantic segmentation, different works [3] suggested the importance of spatial information in trying to achieve high performances. Based on this observations, BiSeNet proposes a Spatial Path to preserve the spatial size of the original input image and encode reach spatial information. This path consists in three layers including a convolution with stride = 2, followed by batch-normalization. [5] and ReLU [6].

Context Path. Another important component of semantic segmentation is represented by the receptive field which is of great significance for the performance. BiSeNet's Context Path utilizes a lightweight model and global average pooling to provide a large receptive field that encodes high-level semantic context information. A global average pooling is added on the tail of the lightweight model and, in the end, the up-sampled output feature of global pooling and the features of the lightweight model are combined.

An **Attention Refinement Module (ARM)** guides the feature learning in the Context Path, to refine features of each stage and a **Features Fusion Module (FFM)** merges Context Path and Spatial Path, scaling the different features of the two paths and re-weighting them, computing a feature selection and combination.

The **loss function** which guides BiSeNet Eq. (1) is the combination between the principal loss l_p, which supervises the output of the whole BiSeNet, and the two auxiliary losses l_i which supervise the output of the Context Path. In addition to this, it's important to precise that all the loss functions are softmax functions.

$$L = l_p + \alpha \sum_{i=2}^{k} l_i \tag{1}$$

In BiSeNet, $\alpha = 1$ and $k = 3$.
In Fig.2 it is possible to have a look at the general architecture of BiSeNet with a further explanation of all its composing modules .

2.2. MiB

The aim of MiB (**M**odel**i**ng the **B**ackground for Incremental Learning in Semantic Segmentation) [2] is to extend the incremental setting to the Semantic Segmentation task.

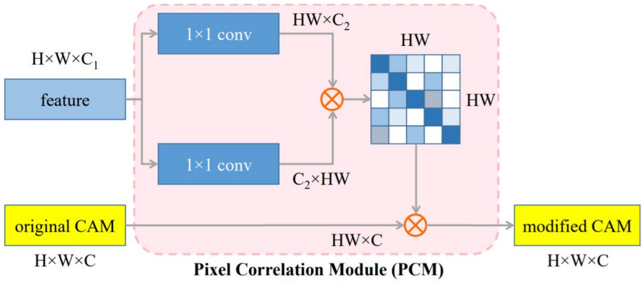


Figure 3. Structure of PCM, where H,W,C/C_1/C_2 denote height, width and channel numbers of feature maps respectively.

MiB handles the catastrophic forgetting problem in such a scenario. The most important problem to consider in this setting, when we try to incrementally learn new classes, is the background. When we learn new classes, there is a high probability we are learning classes previously learned as background. In other words, the semantics associated with the background class change over time.
In the following, we will use MiB notations.

To deal with the semantic shift of the background class, MiB revisits the classical distillation-based framework for incremental learning by introducing two novel loss terms.

$$L(\theta^t) = \frac{1}{|T^t|} \sum_{(x,y)\in T^t} \left(l_{ce}^{\theta^t}(x,y) + \lambda l_{kd}^{\theta^t}(x) \right) \qquad (2)$$

In the overall loss function, showed in (2), MiB revisits the cross entropy loss l_{ce} and the distillation loss l_{kd}[7].

Revisiting Cross-Entropy Loss. MiB takes into account the fact that the training set that we use to update the model at time t, contains only information about novel classes C^t. To deal with this problem, it takes the standard cross-entropy loss and substitute the probability $q_x^t(i, y_\text{i})$ of pixel i to belong to the ground truth y_i, with the (3). In this way, MiB considers that the background class in T^t might include also pixels associated with the previously seen classes in Y^{t-1}. This allows the model to predict the new classes and, at the same time, account for the uncertainty over the actual content of the background class.

$$\tilde{q}_x^t(i,c) = \begin{cases} q_x^t(i,c) & \text{if } c \neq b \\ \sum_{k\in Y^{t-1}} q_x^t(i,k) & \text{if } c = b \end{cases} \qquad (3)$$

Revisiting Distillation Loss. In the standard distillation loss the probability $\hat{q}_x^t(i, y_\text{i})$ of class c for pixel i given by f_{θ^t} is re-normalized across all the classes in Y^{t-1}. It completely ignores the fact that annotations for background in T^s, with $s < t$, might include pixels of classes in C^t. To take it into account, MiB changes this setting with the one proposed in (4).

$$\hat{q}_x^t(i,c) = \begin{cases} q_x^t(i,c) & \text{if } c \neq b \\ \sum_{k\in C^t} q_x^t(i,k) & \text{if } c = b \end{cases} \qquad (4)$$

In this way, the probabilities obtained with the current model are kept unaltered and, more importantly, the background class probability is directly compared with the probability of having either a new class or the background. This allows MiB, first, to still use the full output space of the old model to distil knowledge in the current one and second, to propagate the uncertainty we have on the semantic content of the background in $f_{\theta^{t-1}}$ without penalizing the probabilities of new classes we are learning in the current step t.

MiB doesn't consider only the importance the background has on the loss, but also the impact on the parameters for the new classes. We can reasonably assume that $f_{\theta^{t-1}}$ will likely assign pixels of C^t to b. So a random initialization of the classifiers for the novel classes could lead to possible training instabilities while learning novel classes since the network could initially assign high probabilities to pixels in C^t to b.

To address this issue, MiB proposes to initialize the classifier's parameters for the novel classes in the following way. Given an image x and a pixel i, the probability of the background $q_x^{t-1}(i,b)$ is uniformly spread among the classes in C^t. For this purpose, MiB initializes the weights ω_c^t and the bias β_c^t as in (5) and (6).

$$\omega_c^t = \begin{cases} \omega_b^{t-1} & \text{if } c \in C^t \\ \omega_c^{t-1} & \text{otherwise} \end{cases} \qquad (5)$$

$$\beta_c^t = \begin{cases} \beta_b^{t-1} - \log(|C^t|) & \text{if } c \in C^t \\ \beta_c^{t-1} & \text{otherwise} \end{cases} \qquad (6)$$

2.3. SEAM

Image-level weakly supervised semantic segmentation is a challenging problem that has been deeply studied in recent years. The most adopted approach is represented by CAMs [8] which are an effective way to localize objects from labels. The main problem of this kind of solution is represented by the fact that such an architecture over-activates in the zones of the image corresponding to the most significant part of the object. That's an issue in semantic segmentation where we want to track the contours of the objects. SEAM's most relevant contributions in this scenario, can be summarized in the following two points:

- A Self-supervised Equivariant Attention Mechanism (SEAM), incorporating equivariant regularization with Pixel Correlation Module (PCM) (3), to narrow the supervision gap between fully and weakly supervised semantic segmentation.

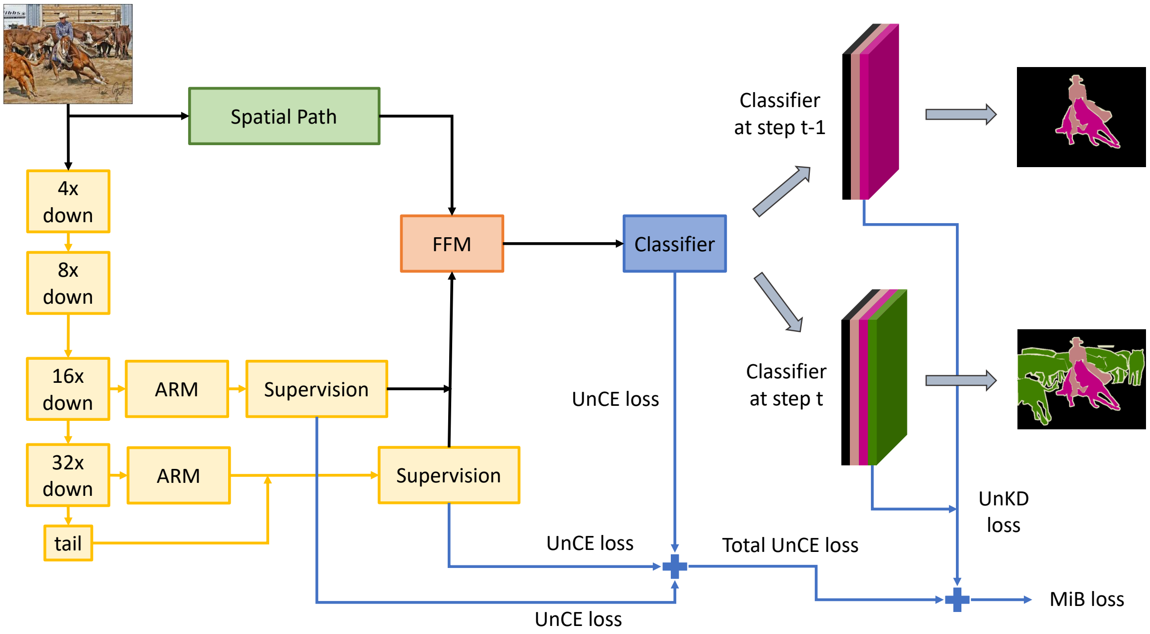


Figure 4. In this picture a schema of the implementation of BiSeNet inside MiB is provided. Notice that in step t, the instantiation of a new classifier occurs.

- The design of siamese network architecture with Equivariant Cross Regularization (ECR) loss, which efficiently couples the PCM and self-supervision, producing CAMs with both fewer over-activated and underactivated regions.

In Fig.1 a little comparison between simple CAMs and SEAM is provided.

Equivariant Regularization. During the data augmentation phase in semantic segmentation, the same affine transformation (e.g. rotations, flips, etc.) must be applied to both the image and the labels. But in our scenario the label is only the presence of the classes. To remedy this issue, Equivariant Regularization is introduced (7):

$$\mathcal{R}_{ER} = ||F(A(I)) - A(F(I))||_1 \tag{7}$$

where F(·) is the network and A(·) is an affine transformation.

Pixel Correlation Module. In Fig. 3, a schema of PCM is provided. This module takes features from an extractor and, after two 1x1 convolutions, merges the results with the ones produced by original CAM to obtain a modified one.

Loss SEAM's loss is the results of the sum of 3 different losses (8), called L_{ECR}, L_{cls} and L_{ER}

$$L = L_{ECR} + L_{cls} + L_{ER} \tag{8}$$

The classification loss is used to roughly localize objects, while the ER loss is used to narrow the gaps between pixel and image-level supervisions. The ECR loss is used to integrate PCM with the trunk of the network, to make consistent predictions over various affine transformations.

3. Method

In this paragraph, we will analyze our contribution to the project. During the implementation of BiSeNet inside MiB, a revisitation of BiSeNet's Loss has been proposed to avoid compatibility issues. For what concerns the last point of the project, we have chosen to study the problem of weak supervision in semantic segmentation but in an incremental setting. In such a scenario, we decided to deepen the issue of not having each pixel of the images classified but only the list of the classes in the single image.

Starting from a model, which has been pre-trained over the first 15 classes, we use SEAM's solution to generate the ground truths we need for incremental steps. In particular, SEAM generates pseudo-labels using class labels as a starting point; in this way, the problem of not having classified pixels could be solved. In the end, we train the incremental part of MiB with the pseudo labels we have generated.

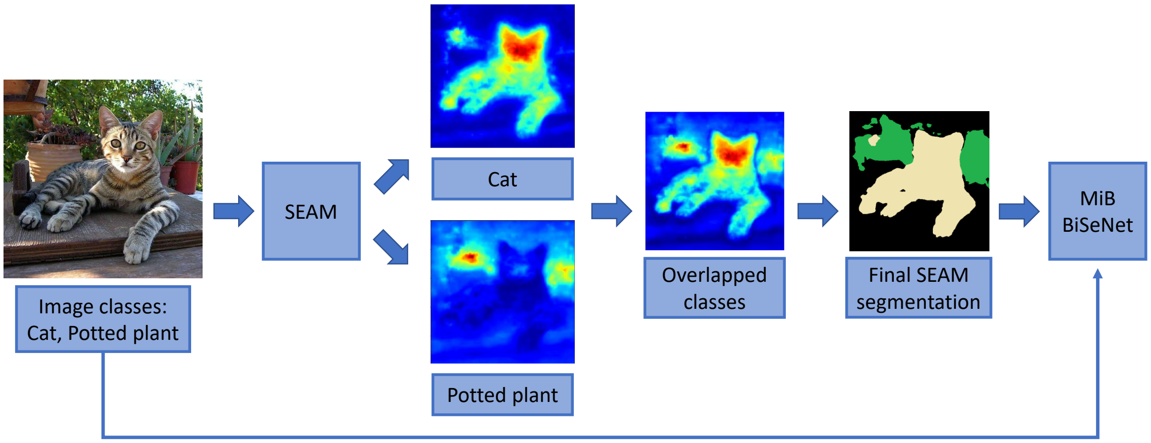


Figure 5. An overview of our SEAM implementation. In a weakly supervised scenario we need to create segmentation labels for images. For each class in the image, SEAM produces a pixel-wise probability map. Then all these maps are joined together and the final segmentation label is produced. In the last step we pass the input image and the pseudo-label to MiB-BiSeNet model.

Revisiting BiSeNet and MiB's Losses. Starting from (1), we decided to use a similar loss inside MiB but, in this case, all the activation functions are not softmax anymore. We use the revisited Cross-Entropy Loss developed into MiB (3) instead. Following BiSeNet's approach, we combine the principal loss with two auxiliary losses. In particular, different cross-entropy losses have been computed for the output of FFM and the outputs of Context Path. In the end, we sum these losses to obtain the final Cross-Entropy Loss (9).

$$L_{CE} = l_{FFM} + \sum_{i=2}^{k} l_{cx_i} \tag{9}$$

Regarding the Distillation Loss used by MiB (4), we compute it on the output of the FFM and we sum it to the previous Cross Entropy Loss. In this way we obtain the final loss which is the combination between the BiSeNet and the MiB's losses (10).

$$L(\theta^t) = \frac{1}{|T^t|} \sum_{(x,y)\in T^t} \left(L_{ce}^{\theta^t}(x,y) + \lambda L_{kd}^{\theta^t}(x) \right) \tag{10}$$

Producing Pseudo-Labels. For each incremental step, we use SEAM algorithm to produce new segmentation labels to train that step. For all the images in that specific step, SEAM predicts the corresponding segmentation and, after that, the incremental step of MiB (with BiSeNet into it) is executed. In Fig. 5 a schema of the pipeline we execute is provided.

4. Experiments

We conduct three kinds of experiments. On the PASCAL-VOC dataset, we train BiSeNet alone first, then MiB with two kinds of dataset divisions (15-5 and 15-1) and, last but not the least, we use pseudo-labels, generated using SEAM's architecture, to learn new classes incrementally. In the first part, we try to figure out the best parameters for BiSeNet. In the second one, we remove DeepLabV3 from MiB and substitute it with BiSeNet, since it manages to obtain similar results in a faster way. In the end, we try to discover what happens using a SEAM architecture to generate pseudo-labels and use them on a pre-trained incremental model.

Table 1: BiSeNet's results for different learning rate's values and batch size's values.

Resnet	**LR**	**Batch Size**	**mIoU**	**Precision**
18	0.001	16	53.3	88.0
18	0.002	32	54.2	88.9
50	0.001	16	64.0	90.6
101	0.001	16	62	90.1
101	**0.002**	**16**	**66.9**	**91.3**

Table 2: mIoU for 15-5 and 15-1 settings for different baselines.

	15-5			**15-1**		
	1-15	16-20	All	1-15	16-20	All
FT	5.2	32.5	12.0	1.3	4.1	2.0
LWF	51.6	36.2	47.7	5.3	7.8	5.9
ILT	59.1	35.8	53.3	3.4	6.3	4.1
MiB	63.4	40.9	58.2	16.1	7.9	14.2
Joint	76.6	69.9	74.9	76.6	69.9	74.9

BiSeNet hyperparameters tuning. The first point of the project required to try different configurations for BiSeNet to find the best one. In Tab.1 it's possible to find a comparison between different training settings for

Table 3: mIoU for 15-5 and 15-1 for different MiB and FT using the outputs produced by SEAM. Notice that a comparison between the performances obtained by FT and MiB in this scenario and in the previous one is provided.

	15-5			**15-1**		
	1-15	16-20	All	1-15	16-20	All
FT-S	4.4	29.5	10.4	4.3	2.6	3.9
FT-VOC	5.2	32.5	12.0	1.3	4.1	2.0
MiB-S	62.5	34.4	56.1	15.5	1.3	11.9
MiB-VOC	63.4	40.9	58.2	16.1	7.9	14.2
Joint-S	55.4	52.0	54.6	55.4	52.0	54.6

BiSeNet. In each setting, we change learning rate, batch size and ResNet. In the end, we decide to use ResNet101 with learning rate = 0.002 and batch size = 16 as the best configuration since it provides good performances and not a very significant time variation with respect to ResNet50. All the configurations are trained for 30 epochs. In all the settings, data augmentation is applied to the images; in particular, Rotation, Vertical and Horizontal Flips, Crop and Color Jittering are applied.

BiSeNet into MiB. The second requirement of the project demands to move BiSeNet into MiB environment to replace DeepLab on two different database splitting: 15-5 and 15-1. The former consists of having 15 classes for the first training step and the remaining 5 classes for the second one. In the latter, after the first 15 classes, one single class is learnt in each training step. In this kind of setting, we run 30 epochs for each required standard ICL baseline. We show the obtained results in Tab. 2.

Introduction of Weak Supervision The last project point demands to implement something new into Mib-BiSeNet scheme. Our choice is to study the problem of weak supervision in semantic segmentation, but in an incremental learning scenario. In incremental steps, we run SEAM over the images belonging to the new classes we want our model to learn. After that, we take pseudo-labels to train the incremental steps in MiB. We validate such steps using the original segmentation labels. In Tab. 3 all the obtained results are presented.

5. Conclusions

In this work, we took into consideration some of the most studied semantic segmentation's problems. We used BiSeNet because it provides an architecture able to extract spatial information and to provide large receptive fields, encoding high-level semantic context information. Then, we implemented it into an incremental scenario. We merged it with MiB which handles the catastrophic forgetting and the shift of background class problems. In the end, we had to deal with the problem of weak supervision. To handle this, we decided to develop a pipeline using SEAM, which generates pseudo-labels for segmentation, starting from class labels.

To sum-up, our obtained results can be explained in the following points:

- Inside MiB architecture, it's possible to substitute DeepLab with BiSeNet without losing in terms of performances and reducing the time needed to train the model.
- It is possible to incrementally train a pre-trained model on new classes using pseudo-labels generated from class labels. In particular, this kind of training allows us to reach performances not much lower than the one obtained using VOC's original segmentations.

References

[1] C. Yu, J. Wang, C. Peng, C. Gao, G. Yu, and N. Sang, "Bisenet: Bilateral segmentation network for real-time semantic segmentation," *CoRR*, vol. abs/1808.00897, 2018.

[2] F. Cermelli, M. Mancini, S. R. Bulò, E. Ricci, and B. Caputo, "Modeling the background for incremental learning in semantic segmentation," *CoRR*, vol. abs/2002.00718, 2020.

[3] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," 2017.

[4] Y. Wang, J. Zhang, M. Kan, S. Shan, and X. Chen, "Dblp:self-supervised equivariant attention mechanism for weakly supervised semantic segmentation," 2020.

[5] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *CoRR*, vol. abs/1502.03167, 2015.

[6] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks.," in *AISTATS* (G. J. Gordon, D. B. Dunson, and M. Dudík, eds.), vol. 15 of *JMLR Proceedings*, pp. 315–323, JMLR.org, 2011.

[7] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," 2015.

[8] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," 2015.