Scaling Semantic Segmentation Beyond 1K Classes on a Single GPU

Shipra Jain[1,2], Danda Pani Paudel[2], Martin Danelljan[2], Luc Van Gool[2,3]
KTH Royal Institute of Technology, Stockholm, Sweden[1]
Computer Vision Lab, ETH Zurich, Switzerland[2]
KU Leuven, Belgium[3]
shipra@kth.se {paudel,martin.danelljan,vangool}@vision.ee.ethz.ch

In the supplementary material, we first report more implementation details followed by details on COCO+LVIS dataset and its experiments, discuss the training and inference time of our approach, followed by visualization of semantic class embeddings and more qualitative results. Code is available at https://github.com/shipra25jain/ESSNet.

A . More implementation details

In DeepLabV3+ architecture, we use output stride as 16 and dilation rate for ASPP = [6, 12, 18]. All models are trained using the polynomial learning rate scheduler : $lr = baselr*(1 - \frac{iter}{total\_iter})^{power}$, the SGD optimizer with momentum, and the weight decay of 1e-4. For baseline and our segmentation network, both power and momentum are set to 0.9. These two parameters for our embedding matrix are set to 0.95. The base learning rate is set to 1e-2 for ADE20k, COCO-Stuff10k, and COCO+LVIS dataset and 1e-1 for Cityscapes and Pascal VOC dataset. The learning rate for the backbone is 0.1 times that of the main network and the momentum of its BN layers as 1e-2.

B . More details about COCO+LVIS

The vocabulary of our bootstrapped COCO+LVIS dataset is build from 181 stuff and 1203 thing classes from COCO and LVIS annotations, respectively. As stuff classes can sometimes be things and vice-versa depending upon the scene and context, there is an overlap of 10 classes between COCO and LVIS vocabulary. The common classes are pillow, curtain, table, cabinet, banner, towel, salad, napkin, blanket, and cupboard. Figure 2 shows examples of classes considered as stuff and thing depending upon the context. COCO+LVIS has 1284 classes. Figure 1 shows the class names and their font size determined by pixel ratio. Figure 3 shows the number of images every class occur in and the long-tail distribution of classes.

We also evaluate our and baseline model with GN for n-most frequent classes in Figure 5. It shows that our model clearly outperforms the baseline, even when Group Normalization is used. We perform ablations on COCO+LVIS dataset. In our approach, replacing nearest neighbour sampling by random sampling gives mIoU of 0.5. Keeping all the components of our approach in place and only removing normalization layer gives 4.15 mIoU. Similarly, removing only regularization loss leads to 3.9 mIoU. These results confirm the significance of sampling technique, normalization and regularization loss in our approach.

C . Training and Inference time

Table 1 shows inference and training times for datasets with a different number of semantic classes. In comparison to the baseline model, our model takes slightly higher inference time for datasets with a lower number of semantic categories and lower inference time for datasets with higher number of classes. During inference, we use index functionality of FAISS library, which first builds an index using class embeddings and then another function call is used to perform the nearest neighbour search. The inference time in our computation includes the duration of the forward pass and segmentation prediction and does not include time for model initialization. We compute inference time for models with ResNet50 backbone and use maximum validation batch size that can fit in GPU. Images with 1024×2048 resolution for Cityscapes dataset and 512×512 for ADE20k and COCO+LVIS dataset are used.

We train models for 200, 80 and 40 epochs for Cityscapes, ADE20k and COCO+LVIS datasets respec-

dataset	model	inference time	training time
Cityscapes	baseline	0.195	4.94
	ours	0.233	5.34
ADE20k	baseline	0.023	3.01
	ours	0.026	4.80
COCO+LVIS	baseline	0.049	4.20
	ours	0.036	5.06

Table 1. Analysis of inference and training time. Inference time is given in seconds per image and training time is given in seconds per iteration. Lower training and inference time is better.

Figure 1. Word cloud of semantic classes of COCO+LVIS dataset. Bigger font size means higher pixel ratio.

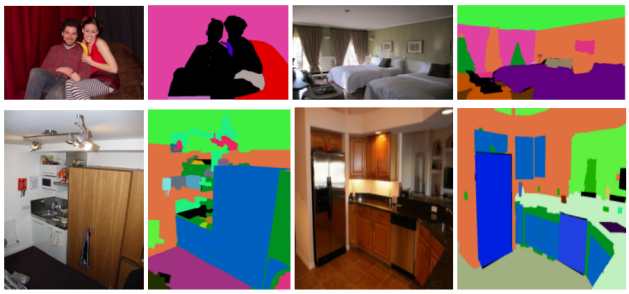

Figure 2. Top: Curtain as stuff class (in left) and as thing class (in right). Bottom: Cabinet as stuff class (in left) and as thing class (in right).

Figure 3. Number of images per semantic class.

tively. In terms of training time, our model takes higher time per iteration for all datasets. It is one of the significant weakness of our approach. k-nearest search computation performed for every pixel is the major bottleneck in our computation time. This computation can be optimized using non-exhaustive search methods like clustering the class embeddings and searching the neighbours for the query only in the cluster in which it lies. We notice that output pixel embeddings for adjacent pixels are very close in embedding space, and this property can be used to compute nearest neighbour search for only 0.25 or 0.125 fractions of total pixels and use same negative samples for 4 or 8 neighbouring pixels. While training both the models for the same number of epochs, we also notice that the baseline converges in 2-6 fewer epochs than our model. This depends upon the number of nearest neighbours k used during the training.

D . Semantic Embeddings and Visualizations

In this section, we investigate the relation between embeddings of different pixels in an image and the class embeddings learned by our model. Figure 4 shows the correlation between the frequency of classes and length of class embedding when normalization layers are not used. Therefore, normalization is essential to suppress the bias caused due to the class imbalance. Figure 6 shows an example of ground truth segmentation mask from ADE20k dataset and corresponding pixel embeddings from our model projected in 2D space. As desired, the pixels belonging to the same

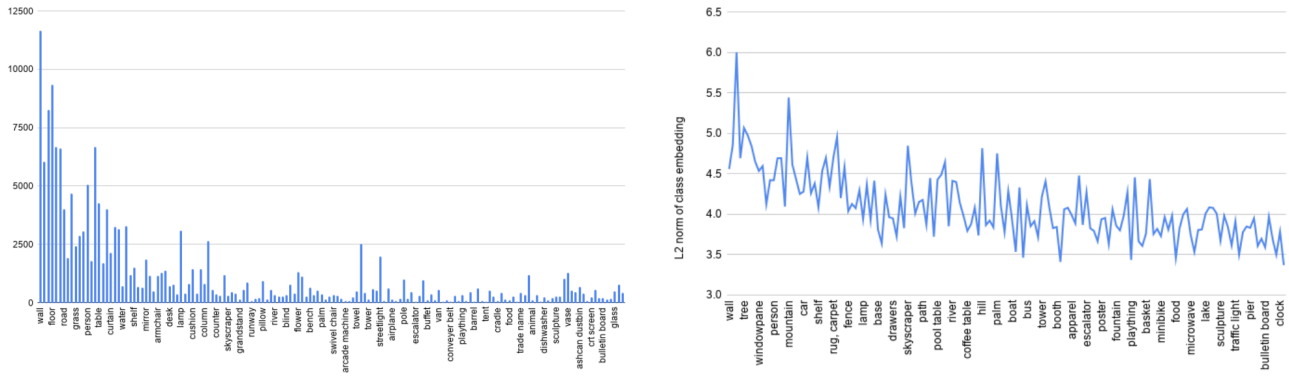

Figure 4. **Class Frequency and Embedding length** Left: Frequency of classes in training dataset for ADE20k dataset. Right: Length of class embeddings when trained the model without normalization layers. There is a correlation between frequency of class and distance of its class embedding from origin.

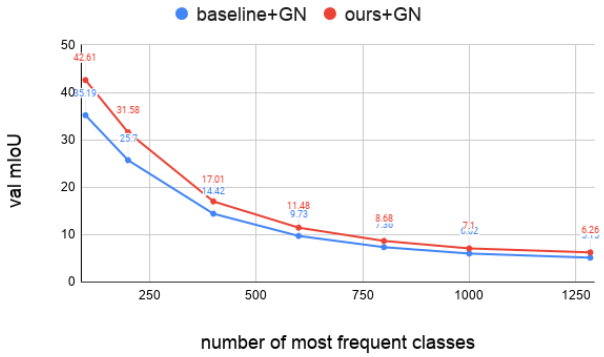

Figure 5. mIoU for COCO+LVIS dataset with increasing number of classes, with most frequent first.

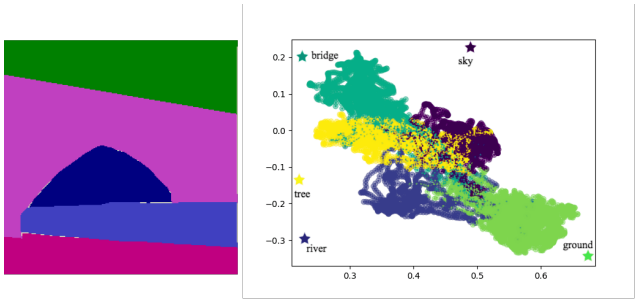

Figure 6. **Pixel Embeddings in 2D space** Left: Example of groundtruth segmentation mask from ADE20k dataset. Right: Circle-shaped markers - pixel embeddings, output from our model is projected into 2D. Star-shaped markers- class embeddings. The color of circular marker denotes the target class of pixel.

class (with the same color) are clustered together. We also notice that the transition of embedding from one class pixel to an adjacent pixel of another class is smooth. This nature of our pixel embeddings might lead to misclassification of pixels at the boundary of the object. Figure 7 shows examples of predicted masks and projection of their pixel embeddings to RGB space. The same colour of pixels in the projection image suggests that their pixel embeddings are closer in feature space, but their nearest class embedding can be different (can be seen from predicted masks).

We perform agglomerative clustering of classes based on the class embeddings learned by our model in Figure 8 and 9. We notice in Figure 8 that classes which occur in a similar context or are semantically similar are closer in feature space. There are several small sub-trees for different contexts like kitchen, scenery, bedroom, interior and many more. For example, pillow, cushion, bed, couch, stool, chair and hassock are clustered together. Also, kitchen equipment like microwave, refrigerator, cabinet, dishwashing machine, cooking stove, sink, kitchen island and countertop fall in same sub-tree. Semantically similar classes like monitoring device and CRT screen are adjacent. The light source and lamp is another pair of adjacent classes with the same semantics. In Figure 9, we perform agglomerative clustering for the hundred most frequent classes from COCO+LVIS dataset. We observe similar clusters for COCO+LVIS dataset also. We also performed k-means clustering on embeddings from ours+GN model and we have attached the list of 70 clusters in supplementary. Clusters such as (bear, grizzly, polar_bear) and (cup, mug, teacup) suggests that embeddings are semantically meaningful.

E . Detailed Qualitative Results

We report more qualitative results for two most challenging datasets *i.e*. COCO-Stuff10k and COCO+LVIS in Figure 10 and 11 respectively.

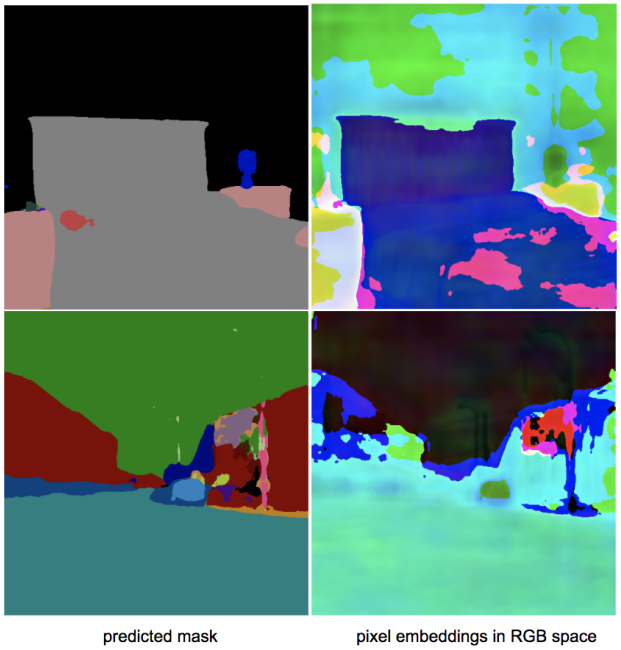

Figure 7. **Pixel Embeddings in RGB Space** Examples of predicted segmentation mask for ADE20k dataset (left). Pixel embeddings are projected into 3D space and transformed to RGB space (right).

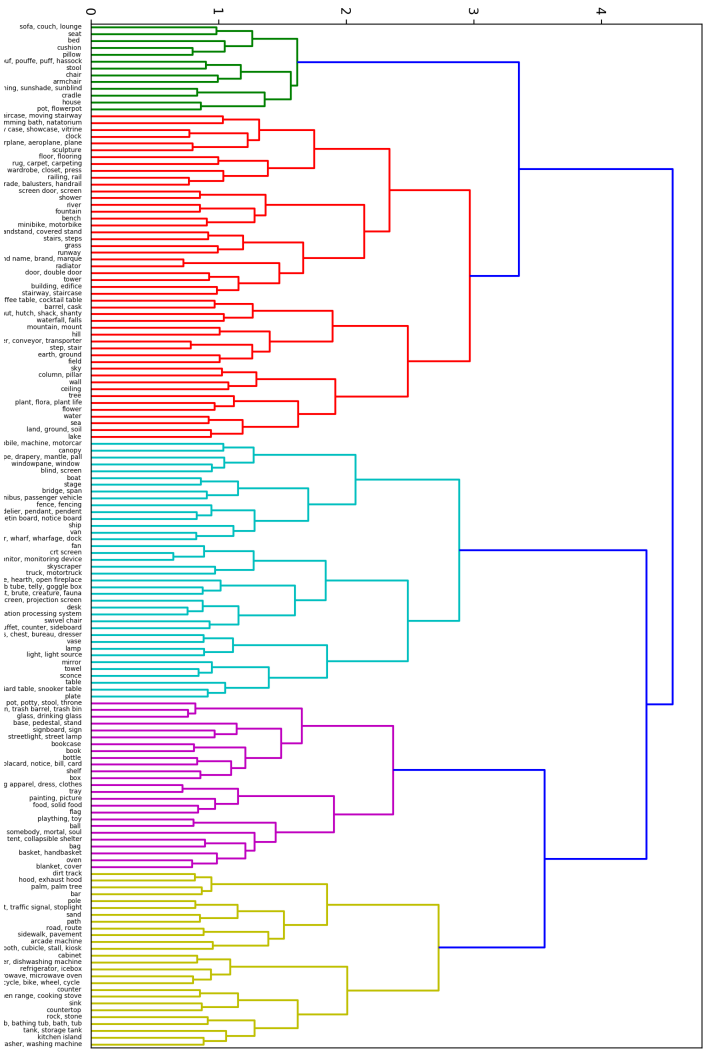

Figure 8. **Similarities in Class Embeddings**: Agglomerative clustering for ADE20k classes based on class embeddings learned by our ESS approach. We observe some of the semantically similar classes clustered together. For example, green sub-tree has couch, seat, bed, pillow, hassock, cushion, chair, stool and cradle classes clustered together. These classes often occur together in a bedroom or drawing room scene and are used for sitting or sleeping. In the yellow sub-tree towards bottom, we notice kitchen appliances clustered together.

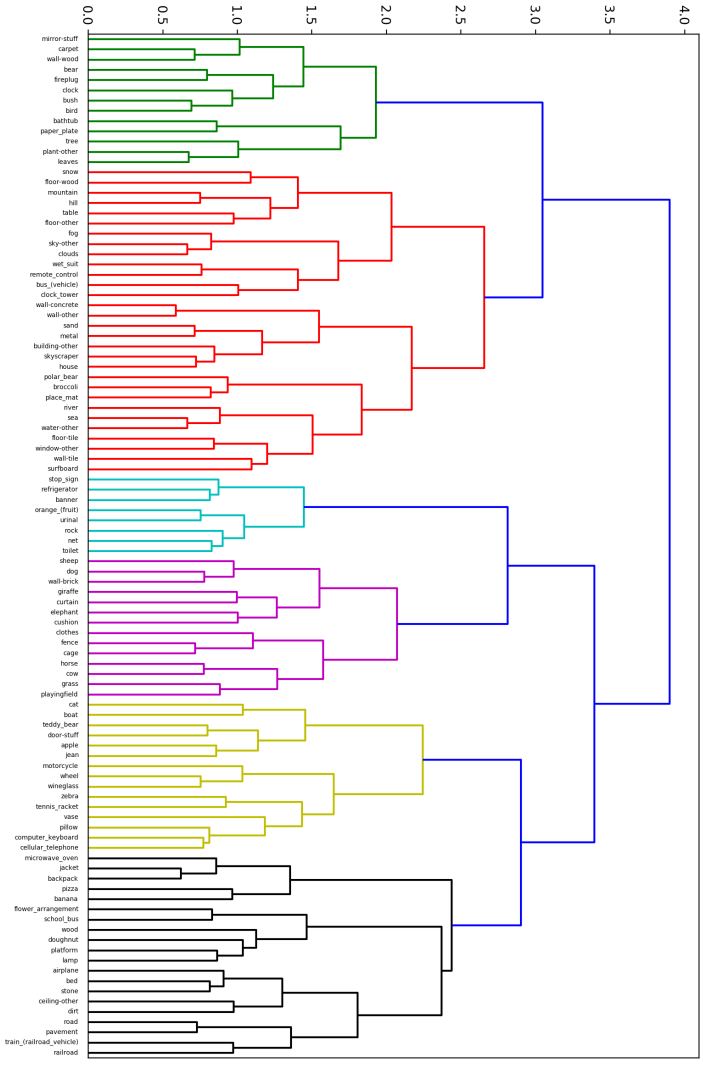

Figure 9. **Similarities in Class Embeddings**: Agglomerative clustering for COCO+LVIS classes based on class embeddings learned by our ESS approach

Figure 10. Qualitative results for COCO-Stuff10k dataset.

Figure 11. Qualitative results for COCO+LVIS dataset. Black color denotes the unlabelled pixels.