



## **Fine-Grained Segmentation Networks: Self-Supervised Segmentation for Improved Long-Term Visual Localization**

Downloaded from: <https://research.chalmers.se>, 2024-03-13 08:56 UTC

Citation for the original published paper (version of record):

Larsson, M., Stenborg, E., Toft, C. et al (2019). Fine-Grained Segmentation Networks: Self-Supervised Segmentation for Improved Long-Term Visual Localization. Proceedings of the IEEE International Conference on Computer Vision(October): 31-41. <http://dx.doi.org/10.1109/ICCV.2019.00012>

N.B. When citing this work, cite the original published paper.

© 2019 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, or reuse of any copyrighted component of this work in other works.

# Fine-Grained Segmentation Networks: Self-Supervised Segmentation for Improved Long-Term Visual Localization

Måns Larsson Erik Stenborg Carl Toft Lars Hammarstrand Torsten Sattler Fredrik Kahl  
 Chalmers University of Technology

## Abstract

*Long-term visual localization is the problem of estimating the camera pose of a given query image in a scene whose appearance changes over time. It is an important problem in practice, for example, encountered in autonomous driving. In order to gain robustness to such changes, long-term localization approaches often use semantic segmentations as an invariant scene representation, as the semantic meaning of each scene part should not be affected by seasonal and other changes. However, these representations are typically not very discriminative due to the limited number of available classes. In this paper, we propose a new neural network, the Fine-Grained Segmentation Network (FGSN), that can be used to provide image segmentations with a larger number of labels and can be trained in a self-supervised fashion. In addition, we show how FGSNs can be trained to output consistent labels across seasonal changes. We demonstrate through extensive experiments that integrating the fine-grained segmentations produced by our FGSNs into existing localization algorithms leads to substantial improvements in localization performance.*

## 1. Introduction

Visual localization is the problem of estimating the camera pose of a given image relative to a visual representation of a known scene. It is a classical problem in computer vision and solving the visual localization problem is one key to advanced computer vision applications such as self-driving cars and other autonomous robots, as well as Augmented / Mixed / Virtual Reality.

The scene representation used by localization algorithms is typically recovered from images depicting a given scene. The type of representation can vary from a set of images with associated camera poses [8, 75, 98], over 3D models constructed from Structure-from-Motion [77, 81], to weights encoded in convolutional neural networks (CNNs) [8, 10, 12, 13, 35, 36, 52] or random forests [11, 16, 79]. In practice, capturing a scene from all possible view-

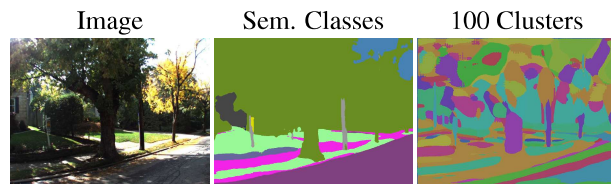


Figure 1. Rather than using a small set of human-defined semantic classes, we train a neural network that automatically discovers a large set of fine-grained clusters. We experimentally show that using a larger number of clusters improves localization performance.

points and under all potential conditions, *e.g.*, different illumination conditions, is prohibitively expensive [74]. Localization algorithms thus need to be robust to such changes.

In the context of long-term operation, *e.g.*, under seasonal changes, the scene appearance can vary drastically over time. However, the semantic meaning of scene parts remains the same, *e.g.*, a tree is a tree whether it carries leaves or not. Based on this insight, approaches for semantic long-term visual localization use semantic segmentations of images or object detections to obtain an invariant scene representation [4, 21, 64, 78, 80, 83, 85, 86, 93, 94, 94]. However, this invariance comes at the price of a lower discriminative power as often only few classes are available. For example, the Cityscapes dataset [22] uses 19 classes for evaluation, 8 of which cover dynamic objects such as cars or pedestrians that are not useful for localization. The Mapillary Vistas dataset [55] contains 66 classes, with 15 classes for dynamic objects. At the same time, annotating more classes comes at significant human labor cost and annotation time.

In this paper, we show that using significantly more class labels leads to better performance of semantic visual localization algorithms. In order to avoid heavy human annotation time, we use the following central insight: the image segmentations used by such methods need to be stable under viewpoint, illumination, seasonal, *etc.* changes. However, the classes of the segmentations do not need to map to human-understandable concepts to be useful, *i.e.*, they might not necessarily need to be semantic. Inspired by recent work on using *k*-means clustering to pre-train CNNs from unlabelled data [15], we thus propose a

self-supervised, data-driven approach to define fine-grained classes for image segmentation. More precisely, we use  $k$ -means clustering on pixel-level CNN features to define  $k$  classes. As shown in Fig. 1, this allows our approach, termed Fine-Grained Segmentation Networks (FGSNs), to create more fine-grained segmentations.

In detail, this paper makes the following contributions: 1) We present a novel type of segmentation network, the Fine-Grained Segmentation Network (FGSN), that outputs dense segmentation maps based on cluster indices. This removes the need for human-defined classes and allows us to define classes in a data-driven way through self-supervised learning. Using a 2D-2D correspondence dataset [42] for training, we ensure that our classes are stable under seasonal and viewpoint changes. The source code of our approach is publicly available<sup>1</sup>. 2) FGSNs allow us to create finer segmentations with more classes. We show that this has a positive impact on semantic visual localization algorithms and can lead to substantial improvements when used by existing localization approaches. 3) We perform detailed experiments to investigate the impact the number of clusters has on multiple visual localization algorithms. In addition, we compare two types of weight initializations, using networks pre-trained for semantic segmentation and image classification, respectively.

## 2. Related Work

The following reviews work related to our approach, most notably semantic segmentation and visual localization.

**Semantic Segmentation.** Semantic segmentation is the task of assigning a class label to each pixel in an input image. Modern approaches use fully convolutional networks [47], potentially pre-trained for classification [47], while incorporating higher level context [99], enlarging the receptive field [17, 19, 92], or fusing multi-scale features [18, 66]. Another line of work combines FCNs with probabilistic graphical models, *e.g.*, in the form of a post-processing step [17] or as a differentiable component in an end-to-end trainable network [41, 46, 100].

CNNs for semantic segmentation are usually trained in a fully supervised fashion. However, obtaining a large amount of densely labeled images is very time-consuming and expensive [22, 55]. As a result, approaches based on weaker forms of annotations have been developed. Some examples of weak labels used to train FCNs are bounding boxes [23, 37, 57], image level tags [57, 59, 62, 82], points [9], or 2D-2D point matches [42]. In this paper, we show that the classes used for “semantic” visual localization do not need to carry semantic meaning. This allows us to directly learn a large set of classes for image segmentation

from data in a self-supervised fashion. During training, we use 2D-2D point matches [42] to encourage consistency of the segmentations across seasonal changes and across different weather conditions.

**(Semantic) Visual Localization.** Traditionally, approaches for visual localization use a 3D scene model constructed from a set of database images via Structure-from-Motion [14, 20, 43–45, 73, 84, 95]. Associating each 3D model point with local image features such as SIFT [50], these approaches establish a set of 2D-3D correspondences between a query image and the model via descriptor matching. The resulting matches are then used for RANSAC-based camera pose estimation [26]. Machine learning-based approaches either replace the 2D-3D matching stage through scene coordinate regression [10, 12, 16, 52–54, 79], *i.e.*, they regress the 3D point coordinate in each 2D-3D match, or directly regress the camera pose from an image [8, 13, 35, 36, 89]. The former type of methods achieves state-of-the-art localization accuracy in small-scale scenes [12, 16, 53], but do not seem to easily scale to larger scenes [12]. The latter type of methods have recently been shown to not perform consistently better than image retrieval methods [76], *i.e.*, approaches that approximate the pose of the query image by the pose of the most similar database image [3, 38, 87]. As such, state-of-the-art methods for long-term visual localization at scale either rely on local features for matching [28, 71, 78, 83, 85, 86] or use image retrieval techniques [2–4, 63, 80, 87, 94].

One class of semantic visual localization approaches uses object detections as features [5, 6, 69]. In this paper, we focus on a second class of approaches based on semantic segmentations [4, 21, 28, 78, 80, 83, 85, 86, 94]. These methods use semantic image segmentations to obtain a scene representation that is invariant to appearance and (moderate) geometry changes. Due to the small number of classes typically available, the resulting representation is not very discriminative. Thus, semantic localization approaches use semantics as a second sensing modality next to 3D information [21, 78, 83, 85, 86]. In this paper, we show that the image segmentations used by such methods do not necessarily need to be semantic. Rather, we show that these approaches benefit from the more fine-grained segmentations with more classes produced by our FGSNs.

**Domain Adaption.** Semantic localization algorithms implicitly assume that semantic segmentations are robust to illumination, viewpoint, seasonal, and other changes. In practice, CNNs for semantic segmentation typically only perform well under varying conditions if these conditions are reflected in the training set. Yet, creating pixel-level annotations for large image sets is a time consuming task. Domain adaption approaches [27, 40, 48, 49, 68, 88, 101] thus consider the problem of applying algorithms trained on one

<sup>1</sup><https://github.com/maunzzz/fine-grained-segmentation-networks>

domain to new domains, where little to no labeled data is available. This makes training on synthetic datasets [65, 67] to improve the performance on real images [33, 70, 102] possible. In addition, the performance of the network on images taken during different weather and lighting conditions can be improved [90, 91]. In the context of (semantic) image segmentation, these approaches improve the robustness of the segmentations. However, they do not increase the number of available classes and are thus complimentary to our approach. We use a recently proposed correspondence dataset [42] for the same purpose, to ensure that our segmentations are robust to illumination and seasonal changes.

**Self-Supervised Learning.** Self-supervised learning approaches are a variant of unsupervised learning methods, where a model learns to predict a set of labels that can be automatically created from the input data. Several approaches train a CNN to perform a domain specific auxiliary task [25, 56, 61, 97]. Some examples of tasks include predicting missing image parts [60], ego motion [1], and the rotation of an image [30]. To solve these auxiliary tasks, the CNNs need to learn meaningful visual features that can then also be used for the actual task at hand. In [15], Caron *et al.* train a CNN for the task of image-level classification using labels acquired by  $k$ -means clustering of image features. We extend this approach to training an image segmentation network. We also use the actual clusters, or labels, explicitly for visual localization. This in contrast to [15], where the clusters are just a means for learning features for tasks such as classification.

### 3. Fine-Grained Segmentation Networks

The Fine-Grained Segmentation Network (FGSN) has the same structure as a standard CNN used for semantic segmentation. Given an input image, it produces a dense segmentation map. However, instead of being trained on a set of manually created annotations, labels are created in a self-supervised manner. During training, at certain intervals, features are extracted from the images in the training set and clustered using  $k$ -means clustering. The cluster assignments, one at each pixel, are then used as supervision during training, *i.e.* as labels. In this way, we can change the number of classes that the FGSN outputs without having to create annotations with the new set of classes. The FGSN is trained to output the correct label for each pixel.

We also use a set of 2D-2D point correspondences [42] during training to ensure that the predictions are stable under seasonal changes and viewpoint variations. Each sample of the correspondence dataset contains two images of the same scene taken from different traversals and thus in different seasonal or weather conditions. One of the images in each pair is always from the reference traversal, captured during favourable weather conditions. A set of 2D-

2D point correspondences between points in the images depicting the same 3D point is also available for each image pair. The network is encouraged to predict the same class for the two points in each correspondence to make the output robust to seasonal changes. Fig. 2 illustrates the training process. Note that creating the correspondence dataset is a significantly less laborious process than hand-labeling the same images with semantic labels, see details in [42].

**Label creation** For the creation of the labels we use the method developed by Caron *et al.* [15] based on  $k$ -means clustering. We, however, need to do some modifications to make it work well for dense output and training with 2D-2D correspondences. The main idea is to do  $k$ -means clustering on the output features of the CNN, then add a layer to the network and train using the cluster assignments as labels. After a fixed number of training iterations, the clustering is repeated and the final layer re-initialized.

For clustering we extract features from all images in the reference traversal of the correspondence dataset. This traversal contains images captured during favourable weather conditions, hence if we initialize the network with weights trained for semantic segmentation the features extracted will contain meaningful semantic information. For each image we get a dense map of image features, from which we randomly sample a set of features for clustering. Half of the features are extracted from pixel positions where we have 2D-2D correspondences and half are randomly sampled across the entire image. Given the set of extracted image features, clustering is done by solving

$$\min_{C \in \mathbb{R}^{d \times m}} \frac{1}{N} \sum_{n=1}^N \min_{\mathbf{y}_n \in \{0,1\}^m} \|\mathbf{d}_n - C\mathbf{y}_n\|_2^2 \quad (1)$$

$$\text{s. t. } \mathbf{y}_n^\top \mathbf{1}_m = 1,$$

where  $\mathbf{d}_n$  are feature vectors of length  $m$  sampled from the output feature maps produced by the CNN. Solving this problem provides a centroid matrix  $C^*$  and a set of optimal assignments ( $\mathbf{y}_n^*$ ). To avoid trivial solutions with empty clusters we do a reassignment of the centroids of empty clusters. For each empty cluster centroid, a centroid of a non-empty cluster is randomly chosen. The centroid of the empty cluster is then set to the same value as this centroid with a small perturbation [15, 34].

**Training Loss.** Our training loss consists of two parts, a correspondence part  $\mathcal{L}_{corr}$  and a cluster classification part  $\mathcal{L}_{class}$ . The latter encourages the model to output the correct label for each pixel in the reference images of dataset. We use a standard cross-entropy loss with the labels as targets. The final  $\mathcal{L}_{class}$  loss is an average of over all samples.

For  $\mathcal{L}_{corr}$ , we use the 2D-2D point correspondences. Denote the content of one sample from the correspondence dataset as  $(I^r, I^t, \mathbf{x}^r, \mathbf{x}^t)$ . Here  $I^r$  is an image from the



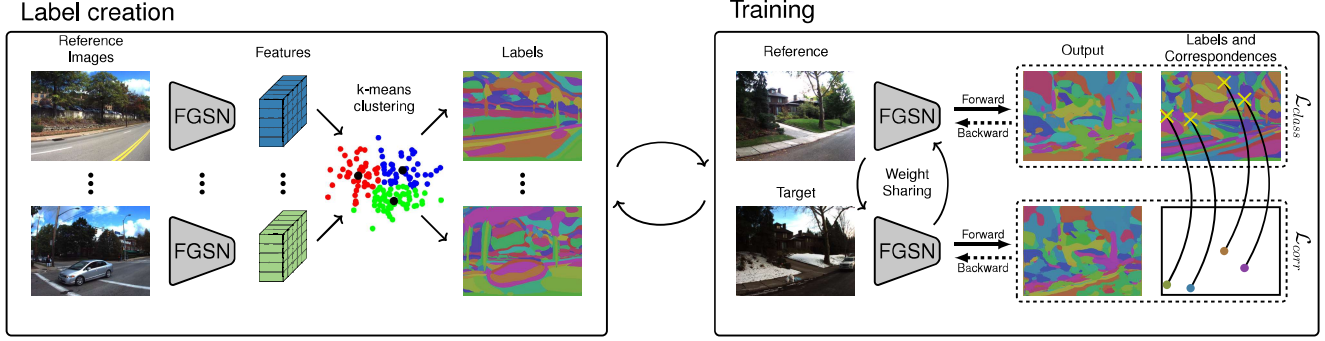


Figure 2. Illustration of the training procedure of an FGSN. To create the training data, features are extracted from all reference images from the correspondence dataset. The features are then clustered using  $k$ -means clustering and the assignments are used as labels for the images. In addition to having dense labels for the reference images, we also use the 2D-2D correspondences during training to encourage consistency across weather conditions and seasons as well as varying viewpoints.

reference traversal,  $I^t$  is an image from the target traversal<sup>2</sup>, and  $\mathbf{x}^r$  as well as  $\mathbf{x}^t$  are the pixel positions of the matched points in the reference and target images, respectively.

The correspondence loss function  $\mathcal{L}_{corr}$  is an average over all such samples

$$\mathcal{L}_{corr} = \frac{1}{M} \sum_{(r,t)} l_{CE}(I^r, I^t, \mathbf{x}^r, \mathbf{x}^t), \quad (2)$$

where  $M$  is the number of samples and  $l_{CE}$  is the cluster correspondence cross-entropy loss. Let  $\mathbf{d}_x \in \mathbb{R}^C$  denote the output feature vector of the network of length  $C$ , the number of clusters, at pixel position  $x$ . To calculate  $l_{CE}$  we begin by taking the cluster assignments, *i.e.* the labels, for the features in the reference image for all positions  $\mathbf{x}^r$ . By describing the label for a pixel at position  $x_i$  using the one-hot encoding vector  $\mathbf{c}_{x_i}$ , the loss can be written as

$$l_{CE} = -\frac{1}{N} \sum_{i=1}^N \mathbf{c}_{x_i}^T \left( \log(\mathbf{d}_{x_i^r}) + \log(\mathbf{d}_{x_i^t}) \right), \quad (3)$$

where  $\log(\cdot)$  is taken element-wise. The loss will encourage the pixels in the target image to have the same labels as the corresponding pixels in the reference image.

During training, we minimize  $\mathcal{L} = \mathcal{L}_{class} + \mathcal{L}_{corr}$ .

**Implementation Details.** During the training of the CNN, we minimize the loss  $\mathcal{L}$  using stochastic gradient descent with momentum and weight decay. During all experiments the learning rate was set to  $2.5 \cdot 10^{-5}$ , while the momentum and weight decay were set to 0.9 and  $10^{-4}$ , respectively. We used the PSPNet [99] network structure with a Resnet101 [32] base. Due to GPU memory limitations we train with a batch size of one. The networks are trained for 60000 iterations and use the weights that obtained the lowest correspondence loss  $\mathcal{L}_{corr}$  on the validation set. The training and evaluation are implemented in PyTorch [58].

<sup>2</sup>We refer to the second traversal as the target as we aim to ensure that its labeling is consistent with the reference traversal.

After every 10000 iterations, a new set of image features are extracted from the reference images and a new set of cluster centroids and labels are calculated. The features, of initial dimension 512, are PCA-reduced to 256 dimensions, whitened and  $l_2$ -normalized. The  $k$ -means clustering was done using the Faiss framework [34]. After clustering, the final layer of the network is randomly re-initialized using a normal distribution with mean 0 and standard deviation 0.01. The bias weights are all set to 0.

All evaluation and testing is done in patches of size  $713 \times 713$  pixels on the original image scale only. Patches are extracted from the image with a step size of 476 pixels in both directions. The network output is paired with an interpolation weight map that is 1 for the  $236 \times 236$  center pixels of the patch and drops off linearly to 0 at the edges. For each pixel the weighted mean, using the interpolation maps as weights, is used to produce the pixel's class scores. The motivation behind the interpolation is that the network generally performs better at the center of the patches, since there is more information about the surroundings available.

## 4. Semantic Visual Localization

This paper was motivated by the hypotheses that being able to obtain more fine-grained image segmentations will have a positive impact on semantic visual localization approaches. To test this hypothesis, we integrate the segmentations obtained with our FGSNs into multiple semantic visual localization algorithms. In the following, we briefly review these algorithms. All of them assume that a 3D point cloud of the scene, where each 3D point is associated with a class or cluster label, is available. Since the point clouds are linked to images, the labels are obtained by projecting the segmentations of the images onto the point cloud.

**Simple Semantic Match Consistency (SSMC) [86].** The first approach is a simple-to-implement match consistency filter used as a baseline method in [86]. Given a set of 2D-3D matches between features in a query image and

3D points in a Structure-from-Motion (SfM) point cloud, SSMC uses semantics to filter out inconsistent matches. A match between a feature  $f$  and a 3D point  $p$  is considered inconsistent if the label of  $f$  obtained by segmenting the query image and the label of  $p$  are not identical. All consistent matches are used to estimate the camera pose by applying a P3P solver [31, 39] inside a RANSAC [26] loop.

**Geometric-Semantic Match Consistency (GSMC) [86].** Assuming that the gravity direction and an estimate of the camera height above the ground is known, [86] proposes a more complicated match consistency filter. For each 2D-3D correspondences, again obtained by matching image features against a SfM model, a set of camera pose hypotheses is generated. For each such pose, the 3D points in the model (including points that are non-matching) are projected into the query image. The projections are used to measure a semantic consistency score for the pose by counting the number of points projecting into a query image region with the same label as the point. The highest score from all poses of a match is then the semantic consistency score of that correspondence. The scores are normalized and used to bias RANSAC’s sampling strategy to prefer selecting more semantically consistent matches. While performing significantly better than SSMC [86], GSMC makes additional assumptions and is computationally less efficient.

**Particle Filter-based Semantic Localization (PFSL) [83].** In this approach, localization is approached as a filtering problem where we, in addition to a sequence of camera images, also have access to noisy odometry information. Both these sources are combined in a particle filter to sequentially estimate the pose of the camera by letting each particle describe a possible camera pose. In the update step of the particle filter, the new weight of each particle is set proportional to how well the projection of the 3D point cloud matches the segmentation of the current image. A 3D point  $p$  is assumed to match well if the pixel which  $p$  is projected to has the same label as  $p$ . Note that this approach does not depend on forming direct 2D-3D correspondences using, e.g., SIFT-descriptors, and is therefore more reliant on discriminative segmentation labels.

## 5. Experiments

The main focus of our experiments is evaluating the impact of using FGSNs for “semantic” visual localization. In addition, we investigate whether the clusters learned by the FGSNs carry semantic information.

**Network variations.** For training, we use two cross-season correspondence datasets from [42], namely the CMU Seasons Correspondence Dataset and the Oxford RobotCar Correspondence Dataset. The available samples are split into a training set (70% of the samples) and a validation set (30% of the samples). The corresponding images are geo-

Init	Clusters	CMU		RobotCar	
		CS	WD	CS	WD
Seg	20	40.1	33.7	32.5	28.0
Seg	100	47.9	36.6	41.5	27.2
Seg	200	47.0	36.6	41.7	32.1
Seg	1000	45.7	35.8	35.6	26.1
Class	200	28.8	26.7	24.0	24.7
Class	1000	18.1	22.2	18.4	23.0

Table 1. Measuring the semantic information contained in our clusters. Using models trained on the CMU or RobotCar Correspondence data, we measure the normalized mutual information (in %) between our clusters and the 19 Cityscapes classes on the Cityscapes (CS) and the WildDash (WD) validation sets. “Seg” networks are pre-trained on semantic segmentation and “Class” networks on classification.

metrically separated from the query images in the Extended CMU Seasons and RobotCar Seasons benchmarks [74] used for evaluating the localization approaches.

In addition to comparing our results to several baselines, we investigate the impact of varying the number of output clusters as well as the impact of pretraining. For the latter, we evaluate a first variant that initializes the base of the network with weights from a network trained on ImageNet [24], while randomly initializing the rest of the network weights. A second variant uses a network pre-trained for semantic segmentation using the fine annotations of the Cityscapes dataset [22] and the training set of the Mapillary Vistas dataset [55]. To be able to combine these two datasets we mapped the Vistas semantic labels to the Cityscapes labels, hence 19 semantic classes were used during training.

Further, we train FGSNs with varying number of output clusters on Cityscapes and Vistas only. For these experiments  $\mathcal{L}_{corr}$  was not used since there are no available correspondences for these datasets.

### 5.1. Semantic Information in Clusters

Our FGSNs are inspired by the task of semantic segmentation and designed with the goal of creating more fine-grained segmentations. Our training procedure does not enforce that our segmentations convey semantic information. Still, an interesting question is whether our clusters can be related to standard semantic classes.

To investigate this, we calculate the normalized mutual information (NMI) to measure the shared information between cluster assignment and the semantic labels of the annotations in the Cityscapes [22] validation set. Denoting the cluster assignments as  $X$  and the semantic label assignments as  $Y$ , the normalized mutual information is given by

$$\text{NMI}(X; Y) = \frac{I(X, Y)}{\sqrt{H(X)H(Y)}}, \quad (4)$$

where  $I$  is the mutual information and  $H$  the entropy. If  $X$  and  $Y$  are independent,  $\text{NMI}(X; Y) = 0$ . If one of the assignments can be predicted from the other, then all information conveyed by  $X$  is shared with  $Y$  and  $\text{NMI}(X; Y) = 1$ .

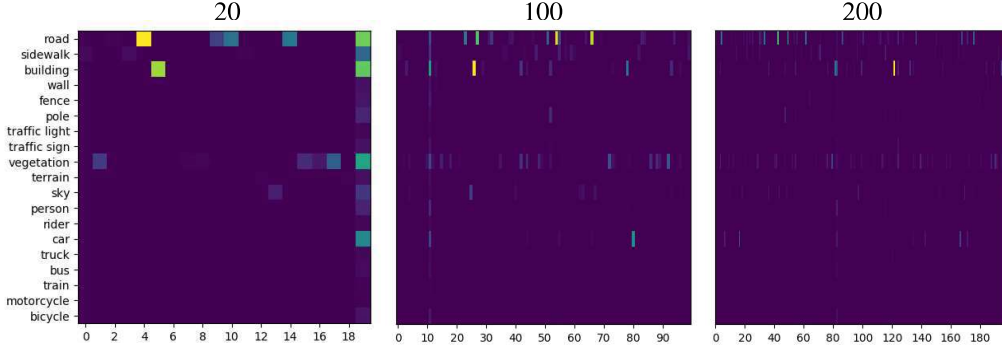


Figure 3. Visualization of contingency tables between Cityscapes classes and cluster indices for different number of clusters. The clusters were trained on the CMU Correspondence Dataset using a model pre-trained on semantic segmentation. The colormap goes from dark blue (lowest value) to yellow (highest value). The data used is the 500 images from the Cityscapes validation set. Many of the classes common in the test images such as road, building and vegetation are split into several clusters.

In addition to the Cityscapes dataset, we also compare the cluster assignments with the same 19 classes on the Wild-Dash dataset [96], which is designed to evaluate the robustness of segmentation methods under a wide range of conditions.

Tab. 1 shows the NMI for our networks. As expected, the networks pre-trained for semantic segmentation achieve a higher NMI compared to the networks pre-trained for classification. Intuitively, the clusters should thus contain semantic information that could be used for localization. However, a high NMI does not necessarily mean better localization performance. For example, a cluster containing pixels around the edges between house and sky would decrease the NMI between the cluster assignments and the semantic classes, but could be useful for localization.

Fig. 3 shows contingency tables between Cityscapes classes and our cluster indices for the networks trained on CMU with semantic segmentation initialization. Each contingency table displays the interrelation between two sets of assignments of the same data by forming a two-dimensional histogram, where each dimension corresponds to one of the assignments. In our case, the dimensions correspond to the semantic class labels and cluster indices, respectively. As can be seen, there are many cluster indices that are assigned to the same pixels as the semantic class vegetation. Since the CMU images contain a significant amount of vegetation, this is both expected and could lead to more information that can be used to localize the images. Looking at the contingency table for the network with 20 clusters, we can see that the cluster with index 19 overlaps with several of the semantic classes. This implies that many pixels are assigned to this cluster, indicating that semantic information is lost. This is also reflected in the NMI (*c.f.* Tab. 1), which is lower for 20-cluster networks compared to those trained with more clusters.

Fig. 3 also shows that many clusters do not directly correspond to semantic classes. This indicates that FGSNs deviate from the pre-trained networks used to initialize them.

## 5.2. Visual Localization

To verify that the learned clusters, even though they are not necessarily semantic in nature, contain useful information for visual localization, we perform experiments on two datasets for long-term visual localization: RobotCar Seasons [74] and the Extended CMU Seasons dataset [74].

**Datasets.** The RobotCar Seasons dataset consists of 32,792 images from the original RobotCar dataset [51]. Of these, 20,862 constitute a reference sequence with publicly known reference poses. A map triangulated from sparse features observed in these images is available as a reference 3D model as an aid for structure-based localization methods. The reference images are all captured under a single condition while the 11,934 test images are captured under a wide variety of different conditions, including seasonal, weather, and illumination changes. We use a slightly different version of the RobotCar Seasons dataset, also used in [42, 74], which consists of a test and training set. The RobotCar Correspondences Dataset that we use to train our FGSNs overlaps with the training set, but not the test set of this version of the RobotCar Seasons dataset.

The Extended CMU Seasons dataset<sup>3</sup> is a larger version of the CMU Seasons dataset from [74], based on the CMU Visual localization dataset [7]. Like the RobotCar Seasons dataset, the Extended CMU Seasons dataset consists of a reference sequence with publicly known camera poses, as well as a hidden test set whose camera poses are not publicly available. The reference sequence consists of 10,338 images captured during the same day in favourable conditions. The test set consists of 56,613 images captured during a wide variety of conditions (sunny, snowy, autumn, *etc.*). The dataset covers urban, suburban, and park-like areas dominated by vegetation on both sides of the road. The latter are the most challenging parts of this dataset [74].

Both datasets provide SIFT features for all test and training images. For SSMC and GSMC, we establish 2D-3D

<sup>3</sup>Available on [visuallocalization.net](http://visuallocalization.net).

Training Configuration / Dataset					Extended CMU Seasons			RobotCar Seasons	
FGSN	Clusters	Data	$\mathcal{L}_{corr}$	Init	Urban 0.25 / 0.5 / 5 [m] 2 / 5 / 10 [deg]	Suburban 0.25 / 0.5 / 5 [m] 2 / 5 / 10 [deg]	Park 0.25 / 0.5 / 5 [m] 2 / 5 / 10 [deg]	all day 0.25 / 0.5 / 5 [m] 2 / 5 / 10 [deg]	all night 0.25 / 0.5 / 5 [m] 2 / 5 / 10 [deg]
	19	CS+V			71.8 / 77.1 / 83.5	56.0 / 61.6 / 71.6	32.8 / 36.9 / 46.0	60.1 / 92.3 / 99.2	8.2 / 21.0 / 35.7
	19	CS+V+E	✓		75.4 / 80.7 / 87.1	56.3 / 62.1 / 72.0	35.0 / 39.4 / 49.0	60.3 / 92.2 / 98.9	8.2 / 21.2 / 35.7
	66	V			75.4 / 80.6 / 87.2	57.1 / 62.6 / 72.3	34.2 / 38.3 / 47.7	60.3 / 92.6 / 99.2	8.9 / 20.3 / 36.6
	66	V+E			65.8 / 70.4 / 77.6	47.7 / 52.7 / 63.7	29.6 / 33.1 / 41.9	59.4 / 92.4 / 99.0	6.1 / 16.3 / 31.5
	66	V+E	✓		66.5 / 71.2 / 78.2	48.1 / 53.3 / 64.1	29.2 / 32.7 / 42.1	59.7 / 91.2 / 98.3	7.2 / 19.6 / 36.1
✓	20	CS+V+E	✓	Seg	76.3 / 81.7 / 87.6	59.7 / 65.7 / 75.3	42.9 / 47.7 / 56.6	57.2 / 88.7 / 96.7	1.9 / 6.5 / 18.9
✓	100	CS+V+E	✓	Seg	81.8 / 87.4 / 91.3	68.9 / 75.6 / 83.5	51.3 / 57.5 / <b>65.7</b>	61.1 / 93.0 / <b>99.9</b>	8.9 / 25.4 / 40.6
✓	200	CS+V+E	✓	Seg	81.0 / 86.7 / 91.1	67.7 / 74.8 / 82.8	50.8 / 57.2 / 65.0	61.3 / 93.2 / 99.8	9.6 / 25.9 / 44.1
✓	1000	CS+V+E	✓	Seg	78.0 / 84.0 / 89.2	62.8 / 70.7 / 79.6	45.1 / 51.9 / 60.9	60.6 / 92.4 / 99.1	6.5 / 17.9 / 35.7
✓*	100	CS+V+E	✓	Seg	<b>85.3 / 91.0 / 94.6</b>	<b>69.5 / 76.4 / 83.7</b>	<b>51.4 / 57.6 / 65.5</b>	<b>61.6 / 93.5 / 99.7</b>	<b>11.0 / 28.4 / 45.2</b>
✓	200	CS+V		Seg	75.8 / 82.4 / 88.2	60.7 / 68.5 / 77.4	42.5 / 48.5 / 57.2	59.9 / 92.9 / 99.4	4.7 / 11.4 / 26.8
✓	1000	CS+V		Seg	69.8 / 77.0 / 84.0	54.6 / 63.2 / 73.0	37.3 / 43.4 / 52.1	54.7 / 86.6 / 94.1	1.4 / 7.7 / 19.3
✓	200	CS+V+E		Seg	78.7 / 84.9 / 89.9	64.9 / 72.4 / 81.1	47.5 / 54.0 / 62.1	61.3 / 93.1 / 99.5	7.0 / 17.9 / 34.0
✓	1000	CS+V+E		Seg	73.4 / 80.4 / 86.9	57.6 / 65.5 / 75.9	39.6 / 46.2 / 55.0	45.5 / 74.8 / 81.8	2.3 / 5.6 / 14.0
✓	200	CS+V+E	✓	Class	70.8 / 77.6 / 84.1	54.1 / 63.1 / 73.3	37.6 / 44.2 / 52.8	60.0 / 91.8 / 98.5	5.4 / 20.3 / 36.4
✓	1000	CS+V+E	✓	Class	47.4 / 55.7 / 64.8	35.1 / 44.4 / 57.4	22.3 / 27.7 / 35.9	48.0 / 73.0 / 79.9	1.9 / 4.0 / 7.5
✓	19	CS+V+O	✓		69.7 / 74.6 / 81.1	53.2 / 58.6 / 69.0	31.2 / 35.2 / 44.2	11.8 / 17.0 / 20.7	0.0 / 0.0 / 0.2
✓	200	CS+V+O	✓	Seg	75.2 / 81.4 / 86.7	60.0 / 67.6 / 76.6	40.9 / 46.8 / 55.4	61.1 / 93.2 / 99.8	3.5 / 10.7 / 27.0
✓	200	CS+V+O	✓	Seg	73.0 / 79.6 / 84.9	59.1 / 66.5 / 75.8	41.6 / 47.5 / 55.3	59.5 / 93.1 / 99.8	3.5 / 11.2 / 24.2
P3P RANSAC					65.3 / 70.1 / 77.6	44.5 / 49.7 / 61.5	27.3 / 30.6 / 39.6	58.4 / 88.6 / 97.1	3.7 / 10.7 / 23.3

Table 2. Localization performance for the SSMC method with different segmentation networks on the Extended CMU Seasons dataset and the RobotCar dataset. The first column marks entries from this paper, for the entry marked with \* clustering was not repeated during training. Column two indicates the number of clusters (or classes) output by the network. Note that for entries marked with 19 and 66 use the semantic classes of Cityscapes and Vistas respectively and were trained with the method presented in [42]. Column three details what datasets were used during training: CS (Cityscapes), V (Vistas), E (Extra *i.e.* CMU for CMU results and RobotCar for RobotCar results), O (Other extra *i.e.* RobotCar for CMU results and CMU for RobotCar Results). The fourth column indicates, with a ✓, if the correspondence loss was active during training while column five specifies the pretraining of the network (Seg for segmentation pretraining and Class for classification pretraining).

Method / Setting m deg	Urban 0.25 / 0.5 / 5 2 / 5 / 10	Suburban 0.25 / 0.5 / 5 2 / 5 / 10	Park 0.25 / 0.5 / 5 2 / 5 / 10
SSMC (FGSN, 100 clusters, trained on CMU)	85.3 / 91.0 / 94.6	69.5 / 76.4 / 83.7	51.4 / 57.6 / 65.5
GSMC (FGSN, 200 clusters, trained on CMU)	86.4 / 91.2 / 93.8	<b>77.0 / 82.9 / 88.7</b>	38.9 / 43.4 / 50.0
HF-Net [72]	<b>89.5 / 94.2 / 97.9</b>	76.5 / 82.7 / 92.7	<b>57.4 / 64.4 / 80.4</b>
Asymmetric Hypercolumn Matching [29]	65.7 / 82.7 / 91.0	66.5 / 82.6 / <b>92.9</b>	54.3 / <b>71.6 / 84.1</b>
GSMC [86]	84.3 / 89.4 / 93.2	69.9 / 75.9 / 83.0	37.8 / 42.0 / 49.3
City Scale Localization [84]	71.2 / 74.6 / 78.7	57.8 / 61.7 / 67.5	34.5 / 37.0 / 42.2
DenseVLAD [87]	14.7 / 36.3 / 83.9	5.3 / 18.7 / 73.9	5.2 / 19.1 / 62.0
NetVLAD [3]	12.2 / 31.5 / 89.8	3.7 / 13.9 / 74.7	2.6 / 10.4 / 55.9
PFSL (FGSN, 200 clusters, trained on CMU)	<b>95.3 / 99.5 / 100.0</b>	<b>87.6 / 98.3 / 99.9</b>	<b>64.8 / 81.5 / 89.3</b>
PFSL [83]	84.7 / 96.8 / 100.0	76.6 / 91.2 / <b>100.0</b>	39.0 / 61.2 / <b>95.6</b>

Table 3. Comparison to state-of-the-art methods on the Extended CMU Seasons dataset. Best results for single-shot image localization and sequential localization are marked separately.

matches via descriptor matching [86]. Following [86], the Lowe ratio test with a threshold of 0.9 is used to filter out outliers. P3P RANSAC is then run for 10,000 iterations to estimate the camera pose.

**Evaluation measures.** We follow the evaluation protocol from [74] and report the percentage of query images localized within  $X$  meters and  $Y$  degrees of the ground-truth poses, using the same thresholds as in [74].

**Impact of the number of clusters.** In a first experiment, we evaluate the impact of the number of clusters learned by FGSNs on localization performance. For this experiment, we focus on the Simple Semantic Match Consistency (SSMC) and compare the performance of SSMC using FGSNs with varying numbers of clusters to the performance obtained with semantic segmentation algorithms. For the latter, we use networks jointly trained on Cityscapes

and Vistas and on Cityscapes, Vistas, and the correspondence datasets [42], using the 19 Cityscapes classes and the 66 Vistas classes. Note that entries marked with [42] also uses a correspondence loss similar to ours but for semantic classes.

Table 2 show the results of the experiments for the RobotCar and CMU datasets. As can be seen, using FGSNs trained with more than 20 clusters improves the localization performance. Especially under challenging conditions, *i.e.*, night on RobotCar and Suburban and Park on CMU, the improvements obtained compared to semantic segmentations are substantial. Naturally, using too many clusters leads to an oversegmentation of the images and thus reduces the localization accuracy of SSMC. The experiments clearly show that SSMC benefits from using fine-grained segmentations, even though clusters might not necessarily



correspond to standard semantic concepts.

The reason why SSMC benefits from a larger number of clusters is that the corresponding segmentations provide a more discriminative representation of the query images and the 3D point cloud. This allows SSMC to filter out more wrong matches by enforcing label consistency. This in turn increases the inlier ratio and thus the probability that RANSAC finds the correct pose. Plots detailing the impact of FGSNs with different numbers of clusters on the number of inliers and the inlier ratio are provided in the supplementary material.

According to Table 2, adding the Extra dataset decreases performance, this is most likely explained by the fact that the network had to be re-implemented to produce the results.

**Impact of pretraining FGSNs.** Entries marked with Class in column for of Table 2 show results obtained when pre-training the base networks of our FGSNs on a classification rather than a semantic segmentations task. As can be seen, FGSNs pre-trained on a classification task result in a significantly lower performance compared to networks trained for semantic segmentation. This shows the importance of using segmentations that retain some semantic information, which is more the case for FGSNs pre-trained on semantic segmentation than for FGSNs pre-trained on classification (*c.f.* Sec. 5.1).

**Impact of using 2D-2D point correspondences** Results for networks trained without the additional dataset from [42] or with the correspondence loss disabled (where the clustering still is done on feature from the CMU/RobotCar images), are shown in Table 2 (row 11-14).

As can be seen from the results, using fine-grained segmentation yields better results than using semantic classes on the Extended CMU Seasons dataset (*c.f.* entries CS+V (19 classes) and V (66 classes)). These networks however, achieve lower results than their counterparts trained with the correspondence datasets. This indicates that the correspondence loss is important for localization performance.

**Generalization abilities.** Table 2 further show results obtained when training the FGSNs on a different dataset. We observe a substantial drop in performance compared to FGSNs trained on the same dataset. This behavior is not unexpected since the 2D-2D correspondences used to train our FGSNs encourage the network to learn dataset-specific clusters. While the performance of FGSNs trained on another dataset is comparable to using networks trained for semantic segmentation, our results indicate that there is still significant room for improving FGSNs.

**Repetition of clustering** Following the method developed by Caron *et al.* [15] the clustering is repeated after a set number of training iterations. Interestingly, we noticed that not resetting the network actually gives slightly better per-

formance, see entry marked with \* in Table 2. We attribute this to the network, pre-trained for semantic segmentation, retains semantic information more easily without resetting. Further investigation of this is left as future work.

**Comparison with state-of-the-art methods.** In a final experiment, we compare SSMC, GSMC and PFSL in combination with FGSNs to state-of-the-art on the Extended CMU Seasons dataset.

To this end, we compare against HF-Net [72], a CNN-based hierarchical localization approach, Asymmetric Hypercolumn Matching [29], an approach based on matching of hypercolumn features, DenseVLAD [87], a state-of-the-art image retrieval pipeline, and its trainable variant NetVLAD [3], City Scale Localization [84], a non-semantic approach based on 2D-3D matches, GSMC [86] using the semantic segmentation network from [86], and PFSL [83] using semantic segmentation network from [42].

As can be seen in Tab. 3, using segmentations with more labels, as afforded by our FGSNs, improves localization performance closing the performance gap to the current state-of-the-art. The results clearly validate the motivation behind FGSNs: using more segmentation labels to create more discriminative, yet still robust, representations for semantic visual localization.

## 6. Conclusion

In this paper, we have presented Fine-Grained Segmentation Networks (FGSN), a novel type of convolutional neural networks that output dense fine-grained segmentations. Using  $k$ -means clustering, we can train FGSNs in a self-supervised manner, using the cluster assignments of image features as labels. This enables us to use arbitrarily many output classes without having to create annotations manually. In addition, we have used a 2D-2D correspondence dataset [42] to ensure that the classes are stable under seasonal changes and viewpoint variations. Through extensive experiments, we have shown that using more fine-grained segmentations, as those of our FGSNs, is beneficial for the task of semantic visual localization.

Important future directions include further adapting visual localization methods to a larger number of clusters to ensure that the increased level of detail of the output segmentations is properly used. In addition, it would be interesting to further work on the generalization of FGSNs, *e.g.*, in combination with domain adaptation methods.

**Acknowledgements** This work has been funded by the Swedish Research Council (grant no. 2016-04445), the Swedish Foundation for Strategic Research (Semantic Mapping and Visual Navigation for Smart Robots) and Vinnova / FFI (Perception, grant no. 2017-01942).

## References

- [1] Pulkit Agrawal, Joao Carreira, and Jitendra Malik. Learning to see by moving. In *ICCV*, 2015. 3
- [2] Asha Anooosheh, Torsten Sattler, Radu Timofte, Marc Pollefeys, and Luc Van Gool. Night-to-Day Image Translation for Retrieval-based Localization. In *ICRA*, 2019. 2
- [3] Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. NetVLAD: CNN architecture for weakly supervised place recognition. In *CVPR*, 2016. 2, 7, 8
- [4] Relja Arandjelović and Andrew Zisserman. Visual vocabulary with a semantic twist. In *ACCV*, 2014. 1, 2
- [5] Shervin Ardeshtir, Amir Roshan Zamir, Alejandro Torroella, and Mubarak Shah. GIS-Assisted Object Detection and Geospatial Localization. In *ECCV*, 2014. 2
- [6] Nikolay Atanasov, Menglong Zhu, Kostas Daniilidis, and George J. Pappas. Localization from semantic observations via the matrix permanent. *IJRR*, 2016. 2
- [7] Hernán Badino, D Huber, and Takeo Kanade. Visual topometric localization. In *IV*, 2011. 6
- [8] Vassileios Balntas, Shuda Li, and Victor Prisacariu. ReLocNet: Continuous Metric Learning Relocalisation using Neural Nets. In *ECCV*, 2018. 1, 2
- [9] Amy Bearman, Olga Russakovsky, Vittorio Ferrari, and Li Fei-Fei. Whats the point: Semantic segmentation with point supervision. In *ECCV*, 2016. 2
- [10] Eric Brachmann, Alexander Krull, Sebastian Nowozin, Jamie Shotton, Frank Michel, Stefan Gumhold, and Carsten Rother. DSAC - Differentiable RANSAC for Camera Localization. In *CVPR*, 2017. 1, 2
- [11] Eric Brachmann, Frank Michel, Alexander Krull, Michael Ying Yang, Stefan Gumhold, and Carsten Rother. Uncertainty-driven 6d pose estimation of objects and scenes from a single rgb image. In *CVPR*, 2016. 1
- [12] Eric Brachmann and Carsten Rother. Learning Less is More - 6D Camera Localization via 3D Surface Regression. In *CVPR*, 2018. 1, 2
- [13] Samarth Brahmbhatt, Jinwei Gu, Kihwan Kim, James Hays, and Jan Kautz. Geometry-Aware Learning of Maps for Camera Localization. In *CVPR*, 2018. 1, 2
- [14] Song Cao and Noah Snavely. Minimal Scene Descriptions from Structure from Motion Models. In *CVPR*, 2014. 2
- [15] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *ECCV*, 2018. 1, 3, 8
- [16] Tommaso Cavallari, Stuart Golodetz, Nicholas A. Lord, Julien Valentin, Luigi Di Stefano, and Philip H. S. Torr. On-The-Fly Adaptation of Regression Forests for Online Camera Relocalisation. In *CVPR*, 2017. 1, 2
- [17] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *PAMI*, 2018. 2
- [18] Liang-Chieh Chen, Yi Yang, Jiang Wang, Wei Xu, and Alan L Yuille. Attention to scale: Scale-aware semantic image segmentation. In *CVPR*, 2016. 2
- [19] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV*, 2018. 2
- [20] Siddharth Choudhary and PJ Narayanan. Visibility probability structure from sfm datasets and applications. In *ECCV*, 2012. 2
- [21] Andrea Cohen, Johannes L. Schönberger, Pablo Speciale, Torsten Sattler, Jan-Michael Frahm, and Marc Pollefeys. Indoor-Outdoor 3D Reconstruction Alignment. In *ECCV*, 2016. 1, 2
- [22] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, 2016. 1, 2, 5
- [23] Jifeng Dai, Kaiming He, and Jian Sun. Boxesup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation. In *ICCV*, 2015. 2
- [24] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 5
- [25] Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *ICCV*, 2015. 3
- [26] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *CACM*, 1981. 2, 5
- [27] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *JMLR*, 2016. 2
- [28] Sourav Garg, Niko Suenderhauf, and Michael Milford. LoST? Appearance-Invariant Place Recognition for Opposite Viewpoints using Visual Semantics. In *RSS*, 2018. 2
- [29] Hugo Germain, Guillaume Bourmaud, and Vincent Lepetit. Sparse-to-dense hypercolumn matching for long-term visual localization. 2019. 7, 8
- [30] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. In *ICLR*, 2018. 3
- [31] Bert M Haralick, Chung-Nan Lee, Karsten Ottenberg, and Michael Nölle. Review and analysis of solutions of the three point perspective pose estimation problem. *IJCV*, 1994. 5
- [32] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 4
- [33] Judy Hoffman, Dequan Wang, Fisher Yu, and Trevor Darrell. Fcns in the wild: Pixel-level adversarial and constraint-based adaptation. *arXiv:1612.02649*, 2016. 3
- [34] Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with gpus. *arXiv:1702.08734*, 2017. 3, 4
- [35] Alex Kendall and Roberto Cipolla. Geometric Loss Functions for Camera Pose Regression With Deep Learning. In *CVPR*, 2017. 1, 2

- [36] Alex Kendall, Matthew Grimes, and Roberto Cipolla. PoseNet: A convolutional network for real-time 6-dof camera relocalization. In *ICCV*, 2015. 1, 2
- [37] Anna Khoreva, Rodrigo Benenson, Jan Hendrik Hosang, Matthias Hein, and Bernt Schiele. Simple does it: Weakly supervised instance and semantic segmentation. In *CVPR*, 2017. 2
- [38] Hyo Jin Kim, Enrique Dunn, and Jan-Michael Frahm. Learned Contextual Feature Reweighting for Image Geo-Localization. In *CVPR*, 2017. 2
- [39] Laurent Kneip, Davide Scaramuzza, and Roland Siegwart. A novel parametrization of the perspective-three-point problem for a direct computation of absolute camera position and orientation. In *CVPR*, 2011. 5
- [40] Brian Kulis, Kate Saenko, and Trevor Darrell. What you saw is not what you get: Domain adaptation using asymmetric kernel transforms. In *CVPR*, 2011. 2
- [41] Måns Larsson, Anurag Arnab, Shuai Zheng, Philip Torr, and Fredrik Kahl. Revisiting deep structured models for pixel-level labeling with gradient-based inference. *SIIMS*, 2018. 2
- [42] Måns Larsson, Erik Stenborg, Lars Hammarstrand, Torsten Sattler, Marc Pollefeys, and Fredrik Kahl. A Cross-Season Correspondence Dataset for Robust Semantic Segmentation. In *CVPR*, 2019. 2, 3, 5, 6, 7, 8
- [43] Viktor Larsson, Johan Fredriksson, Carl Toft, and Fredrik Kahl. Outlier rejection for absolute pose estimation with known orientation. In *BMVC*, 2016. 2
- [44] Yunpeng Li, Noah Snavely, Dan Huttenlocher, and Pascal Fua. Worldwide pose estimation using 3d point clouds. In *ECCV*, 2012. 2
- [45] Hyon Lim, Sudipta N. Sinha, Michael F. Cohen, Matt Uyttendaele, and H. Jin Kim. Real-time monocular image-based 6-DoF localization. *IJRR*, 34(4-5):476–492, 2015. 2
- [46] Ziwei Liu, Xiaoxiao Li, Ping Luo, Chen-Change Loy, and Xiaoou Tang. Semantic image segmentation via deep parsing network. In *ICCV*, 2015. 2
- [47] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015. 2
- [48] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael I Jordan. Learning transferable features with deep adaptation networks. In *ICML*, 2015. 2
- [49] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. Unsupervised domain adaptation with residual transfer networks. In *NIPS*, 2016. 2
- [50] David G Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 2004. 2
- [51] Will Maddern, Geoffrey Pascoe, Chris Linegar, and Paul Newman. 1 year, 1000 km: The Oxford RobotCar dataset. *IJRR*, 2017. 6
- [52] Daniela Massiceti, Alexander Krull, Eric Brachmann, Carsten Rother, and Philip HS Torr. Random Forests versus Neural Networks - What’s Best for Camera Relocalization? In *ICRA*, 2017. 1, 2
- [53] Lili Meng, Jianhui Chen, Frederick Tung, James J. Little, Julien Valentin, and Clarence W. de Silva. Backtracking Regression Forests for Accurate Camera Relocalization. In *IROS*, 2017. 2
- [54] Lili Meng, Frederick Tung, James J. Little, Julien Valentin, and Clarence W. de Silva. Exploiting Points and Lines in Regression Forests for RGB-D Camera Relocalization. In *IROS*, 2018. 2
- [55] Gerhard Neuhold, Tobias Ollmann, Samuel Rota Bulò, and Peter Kotschieder. The mapillary vistas dataset for semantic understanding of street scenes. In *ICCV*, 2017. 1, 2, 5
- [56] Mehdi Noroozi, Hamed Pirsiavash, and Paolo Favaro. Representation learning by learning to count. In *ICCV*, 2017. 3
- [57] George Papandreou, Liang-Chieh Chen, Kevin Murphy, and Alan L Yuille. Weakly-and semi-supervised learning of a dcnn for semantic image segmentation. In *ICCV*, 2015. 2
- [58] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. In *NIPS-W*, 2017. 4
- [59] Deepak Pathak, Philipp Krahenbuhl, and Trevor Darrell. Constrained convolutional neural networks for weakly supervised segmentation. In *ICCV*, 2015. 2
- [60] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A. Efros. Context encoders: Feature learning by inpainting. In *CVPR*, 2016. 3
- [61] Mattis Paulin, Matthijs Douze, Zaid Harchaoui, Julien Mairal, Florent Perronin, and Cordelia Schmid. Local convolutional features with unsupervised training for image retrieval. In *ICCV*, 2015. 3
- [62] Pedro O. Pinheiro and Ronan Collobert. From image-level to pixel-level labeling with convolutional networks. In *CVPR*, 2015. 2
- [63] Horia Porav, Will Maddern, and Paul Newman. Adversarial Training for Adverse Conditions: Robust Metric Localisation Using Appearance Transfer. In *ICRA*, 2018. 2
- [64] Noha Radwan, Abhinav Valada, and Wolfram Burgard. VLocNet++: Deep Multitask Learning For Semantic Visual Localization And Odometry. *RA-L*, 2018. 1
- [65] Stephan R Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. Playing for Data: Ground Truth from Computer Games. In *ECCV*, 2016. 3
- [66] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, 2015. 2
- [67] German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio M Lopez. The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *CVPR*, 2016. 3
- [68] Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. *Target*. 2
- [69] Renato F. Salas-Moreno, Richard A. Newcombe, Hauke Strasdat, Paul H. J. Kelly, and Andrew J. Davison. SLAM++: Simultaneous Localisation and Mapping at the Level of Objects. In *CVPR*, 2013. 2

- [70] Swami Sankaranarayanan, Yogesh Balaji, Arpit Jain, Ser Nam Lim, and Rama Chellappa. Learning from synthetic data: Addressing domain shift for semantic segmentation. In *CVPR*, 2018. 3
- [71] Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk. From Coarse to Fine: Robust Hierarchical Localization at Large Scale. In *CVPR*, 2018. 2
- [72] Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk. From coarse to fine: Robust hierarchical localization at large scale. In *CVPR*, 2019. 7, 8
- [73] Torsten Sattler, Bastian Leibe, and Leif Kobbelt. Efficient & Effective Prioritized Matching for Large-Scale Image-Based Localization. *PAMI*, 2017. 2
- [74] Torsten Sattler, Will Maddern, Carl Toft, Akihiko Torii, Lars Hammarstrand, Erik Stenborg, Daniel Safari, Masatoshi Okutomi, Marc Pollefeys, Josef Sivic, Fredrik Kahl, and Tomas Pajdla. Benchmarking 6DOF Outdoor Visual Localization in Changing Conditions. In *CVPR*, 2018. 1, 5, 6, 7
- [75] Torsten Sattler, Akihiko Torii, Josef Sivic, Marc Pollefeys, Hajime Taira, Masatoshi Okutomi, and Tomas Pajdla. Are Large-Scale 3D Models Really Necessary for Accurate Visual Localization? In *CVPR*, 2017. 1
- [76] Torsten Sattler, Qunjie Zhou, Marc Pollefeys, and Laura Leal-Taixé. Understanding the Limitations of CNN-based Absolute Camera Pose Regression. In *CVPR*, 2019. 2
- [77] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-Motion Revisited. In *CVPR*, 2016. 1
- [78] Johannes Lutz Schönberger, Marc Pollefeys, Andreas Geiger, and Torsten Sattler. Semantic Visual Localization. In *CVPR*, 2018. 1, 2
- [79] Jamie Shotton, Ben Glocker, Christopher Zach, Shahram Izadi, Antonio Criminisi, and Andrew Fitzgibbon. Scene coordinate regression forests for camera relocalization in rgb-d images. In *CVPR*, 2013. 1, 2
- [80] Gautam Singh and Jana Košecká. Semantically Guided Geo-location and Modeling in Urban Environments. In *Large-Scale Visual Geo-Localization*, 2016. 1, 2
- [81] Noah Snavely, Steven M. Seitz, and Richard Szeliski. Photo Tourism: Exploring Photo Collections in 3D. In *SIGGRAPH*, 2006. 1
- [82] Nasim Souly, Concetto Spampinato, and Mubarak Shah. Semi supervised semantic segmentation using generative adversarial network. In *ICCV*, 2017. 2
- [83] Erik Stenborg, Carl Toft, and Lars Hammarstrand. Long-term visual localization using semantically segmented images. *ICRA*, 2018. 1, 2, 5, 7, 8
- [84] Linus Svärm, Olof Enqvist, Fredrik Kahl, and Magnus Oscarsson. City-Scale Localization for Cameras with Known Vertical Direction. *PAMI*, 2017. 2, 7, 8
- [85] Carl Toft, Carl Olsson, and Fredrik Kahl. Long-Term 3D Localization and Pose from Semantic Labellings. In *ICCV Workshops*, 2017. 1, 2
- [86] Carl Toft, Erik Stenborg, Lars Hammarstrand, Lucas Brynte, Marc Pollefeys, Torsten Sattler, and Fredrik Kahl. Semantic Match Consistency for Long-Term Visual Localization. In *ECCV*, 2017. 1, 2, 4, 5, 7, 8
- [87] Akihiko Torii, Relja Arandjelovic, Josef Sivic, Masatoshi Okutomi, and Tomas Pajdla. 24/7 place recognition by view synthesis. In *CVPR*, 2015. 2, 7, 8
- [88] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *CVPR*, 2017. 2
- [89] Florian Walch, Caner Hazirbas, Laura Leal-Taixe, Torsten Sattler, Sebastian Hilsenbeck, and Daniel Cremers. Image-based localization using lstms for structured feature correlation. In *ICCV*, 2017. 2
- [90] Markus Wulfmeier, Alex Bewley, and Ingmar Posner. Addressing appearance change in outdoor robotics with adversarial domain adaptation. In *IROS*, 2017. 3
- [91] Markus Wulfmeier, Alex Bewley, and Ingmar Posner. Incremental adversarial domain adaptation for continually changing environments. In *ICRA*, 2018. 3
- [92] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. In *ICLR*, 2016. 2
- [93] Fisher Yu, Jianxiong Xiao, and Thomas A. Funkhouser. Semantic alignment of LiDAR data at city scale. In *CVPR*, 2015. 1
- [94] Xin Yu, Sagar Chaturvedi, Chen Feng, Yuichi Taguchi, Teng-Yok Lee, Clinton Fernandes, and Srikumar Ramalingam. VLASE: Vehicle Localization by Aggregating Semantic Edges. In *IROS*, 2018. 1, 2
- [95] Bernhard Zeisl, Torsten Sattler, and Marc Pollefeys. Camera pose voting for large-scale image-based localization. In *ICCV*, 2015. 2
- [96] Oliver Zendel, Katrin Honauer, Markus Murschitz, Daniel Steininger, and Gustavo Fernández Domínguez. Wilddash-creating hazard-aware benchmarks. In *ECCV*, 2018. 6
- [97] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *ECCV*, 2016. 3
- [98] Wei Zhang and Jana Kosecka. Image based Localization in Urban Environments. In *3DPVT*, 2006. 1
- [99] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *CVPR*, 2017. 2, 4
- [100] Shuai Zheng, Sadeep Jayasumana, Bernardino Romera-Paredes, Vibhav Vineet, Zhizhong Su, Dalong Du, Chang Huang, and Philip HS Torr. Conditional random fields as recurrent neural networks. In *ICCV*, 2015. 2
- [101] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*, 2017. 2
- [102] Yang Zou, Zhiding Yu, BVK Vijaya Kumar, and Jinsong Wang. Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In *ECCV*, 2018. 3