

DenserNet: Weakly Supervised Visual Localization Using Multi-Scale Feature Aggregation

Dongfang Liu¹, Yiming Cui^{2*}, Liqi Yan³, Christos Mousas¹, Baijian Yang¹, Yingjie Chen¹

¹Purdue University

²University of Florida

³Fudan University

liu2538@purdue.edu, cuiyiming@ufl.edu, yanliqi@westlake.edu.cn,
cmousas@purdue.edu, byang@purdue.edu, victorchen@purdue.edu

Abstract

In this work, we introduce a Denser Feature Network (DenserNet) for visual localization. Our work provides three principal contributions. First, we develop a convolutional neural network (CNN) architecture which aggregates feature maps at different semantic levels for image representations. Using denser feature maps, our method can produce more keypoint features and increase image retrieval accuracy. Second, our model is trained end-to-end without pixel-level annotation other than positive and negative GPS-tagged image pairs. We use a weakly supervised triplet ranking loss to learn discriminative features and encourage keypoint feature repeatability for image representation. Finally, our method is computationally efficient as our architecture has shared features and parameters during forwarding propagation. Our method is flexible and can be crafted on a light-weighted backbone architecture to achieve appealing efficiency with a small penalty on accuracy. Extensive experiment results indicate that our method sets a new state-of-the-art on four challenging large-scale localization benchmarks and three image retrieval benchmarks with the same level of supervision. The code is available at <https://github.com/goodproj13/DenserNet>.

Introduction

The task of visual localization is to predict the geographic location of a query image, based on its comparisons to GPS-tagged images from a database (Salarian et al. 2018). Visual localization has drawn considerable attention recently due to its potential value to wide-ranging applications such as robot navigation or autonomous driving (Liu et al. 2020b,a). Under the region where GPS signal is partially or completely shadowed, visual localization is an effective addition to GPS to support the operation of these mobile agents.

In our work, we cast the visual localization problem as an image retrieval task (Arandjelovic et al. 2016). Image retrieval task relies on local features to search over a GPS-tagged image database to estimate the current location. The primary challenge is how to produce discriminative image representation so that images from nearby locations



Figure 1. DenserNet can correctly perform visual localization under challenging conditions. Despite environmental variations (such as pedestrian or vehicle occlusions, illumination changes, and seasonal changes), DenserNet can find the location (a) from the database based on the query (b).

would have similar representations in feature spaces while images from different locations would have dissimilar representations. Typically, a large-scale database contains images of similar man-made structures or landmarks that may cause severe ambiguities. Illumination variations or occlusions may also change object appearances which compromise the localization prediction (Tolias, Avrithis, and Jégou 2016). To address these problems, our method is designed to be robust for a large-scale dataset with different challenging conditions (Figure 1).

In the last decade, convolutional neural networks (CNNs) have emerged as a powerful technique to explore image representations including visual localization (Sandler et al. 2018; Simonyan and Zisserman 2014). CNN-based visual localization networks have a similar architecture (Dusmanu et al. 2019; Jin Kim, Dunn, and Frahm 2017). They generally have a convolutional backbone encoder to produce feature maps of the input image. Then, a detection-description decoder organizes the obtained feature maps to depict the image representation. The conventional approaches only use

*is the corresponding author

the feature maps from a single semantic level and fail to exploit multi-scale features from different semantic levels. This limitation motivates our approach to exploit features from multi-level semantics to improve localization behavior.

So far, many state-of-the-art methods are trained using pixel-level annotations to supervise the process of feature learning (Luo et al. 2020; Dusmanu et al. 2019). However, the ground truth correspondences between image pairs are expensive to obtain at the scale required to train a CNN-based method. Also, such supervised priors based on human annotations may not fully capture all relevant features for image representation to train a network (Jin Kim, Dunn, and Frahm 2017). Thus, training a CNN-based method with strong supervision is not efficient and effective enough for generalization. In contrast, an increasing number of studies (Liu, Li, and Dai 2019; Jin Kim, Dunn, and Frahm 2017) employ a weakly supervised manner for training using image-level labels. They compute the feature distance between the query-positive and the query-negative image pairs to discover imagery discriminativeness. These prior explorations impact our work.

Building on the lessons learned from the concurrent approaches (Arandjelovic et al. 2016; DeTone, Malisiewicz, and Rabinovich 2018), our work brings the following three contributions. First, we introduce a Denser Feature Network (DenserNet), a novel CNN-based method for visual localization tasks. We find inspiration from DenseNet (Huang et al. 2017) to aggregate feature maps at different semantic levels for image representations, as shown in Figure 2. Our method is an intuitive extension of NetVLAD (Arandjelovic et al. 2016) by adding feature extraction branches to obtain multi-scale features, in parallel with the existing backbone network. The feature extraction branches aggregate features from the lower-level, the mid-level, and the higher-level layer of the backbone network. Compared to conventional methods (Dusmanu et al. 2019; Jin Kim, Dunn, and Frahm 2017; DeTone, Malisiewicz, and Rabinovich 2018), our approach is able to produce denser keypoint features which increase the number of inlier matches between image pairs and, in turn, improve the matching accuracy under challenging conditions. Extensive experiment results indicate that our method improves the visual localization performance on several benchmarks by a large margin.

Second, we propose a weakly supervised approach for training. We design a modified triplet ranking loss to effectively organize obtained features and predict image representations. Our training requires no expensive pixel-wise ground truths other than GPS-tagged images, which are easy to obtain. In our design, task-relevant keypoint features are discovered in an unsupervised fashion, as the proposed method is able to learn in which context should be suppressed or emphasized to achieve better location recognition. Inspired by (Dusmanu et al. 2019), our training method performs a joint optimization for both detection and description tasks, which encourages the repeatability of discriminative detection and improves the description accuracy.

Finally, DenserNet is computationally efficient. Since the three feature extraction branches are all based on the same

CNN backbone, they have shared features and parameters during computation. Theoretical and empirical evaluations demonstrate that, although DenserNet uses extra sub-network branches to aggregate more feature maps for the location inference, it only requires a limited additional computation load. Thus, our method can achieve efficient inference and remain within the computational constraints.

Related Work

Recent advances in CNNs have made it possible to use local keypoint features of an image to predict location hypothesis (Arandjelovic et al. 2016). Although the CNN-based approaches are proved to be effective for visual localization, they struggle to perform well when the feature homogeneity occurs substantially (Liu, Li, and Dai 2019). Modern man-made structures frequently have architectural similarities so it is difficult for the conventional methods to handle this challenging situation (Salarian et al. 2018). In addition, the performance of conventional methods typically degrades under extreme environmental changes, such as illumination or landmark scale changes (Dusmanu et al. 2019).

A critical reason causing these problems for conventional methods (Dusmanu et al. 2019) is that they only use features from one semantic level for prediction. Thus, they fail to exploit features from different levels of semantics to capture more multi-scale details. This semantic gap introduces a critical problem in both feature learning and predicting (Sarlin et al. 2019). To address the above challenges, our method is designed to produce more keypoint features using feature extraction branches to organically extract features from different semantic levels. Compared to the latest similar work (Luo et al. 2020) that uses strong supervision for training and has a heavy working pipeline for inference, our method is simple yet effective. We can achieve competitive performance using a weak supervision fashion without any bells and whistles.

DenserNet

Overview

The architecture of DenserNet is shown in Figure 2. DenserNet includes a CNN backbone, three feature extraction branches, and a one-stage feature decoder. Our design leverages the intrinsic nature of modern CNN architecture which can produce rich hierarchical features from different convolutional layers from a single forward pass. Thus we can obtain multi-scale features with low additional costs to close the semantic gap in feature learning. Each feature extraction branch sticks out at different layers of the backbone network to extract features from different semantic levels. We aggregate the obtained features to capture strong image representation. Intuitively, our method is more robust to scale variances than methods learned from single-scale features and thus improves the localization performance. We elaborate on the detail of our method below.

Feature Extraction Branch

The design of DenserNet is flexible and allows various backbone options. For fast runtime, we use MobileNetV2 (San-

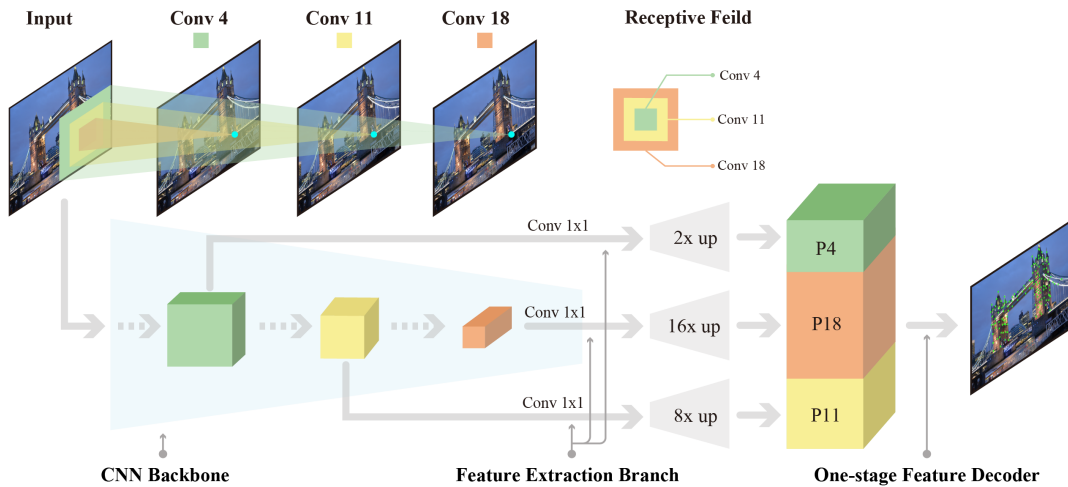


Figure 2. The demonstration of DenserNet using MobileNet backbone (Sandler et al. 2018). The proposed method can aggregate multi-level feature maps and thus produce significantly more local keypoint features than concurrent methods (Arandjelovic et al. 2016; Dusmanu et al. 2019). For simplicity, 3×3 convolution layers after upsampling are not shown.

andler et al. 2018), a light-weighted network, as our backbone network. DenserNet has the lower-level, the mid-level, and the higher-level branch based on different locations of the backbone. Specifically, the lower-level and mid-level branches stick out at the conv4 and conv11 respectively, and the higher-level branch is attached to the conv18.

The input features for each feature extraction branch are $\{C4, C11, C18\}$, having strides of $\{4, 16, 32\}$ pixels with respect to the input image. Each feature extraction branch employs a modified SuperPoint layer (DeTone, Malisiewicz, and Rabinovich 2018). Instead of having the keypoints and local descriptors from the original implementation, the modified SuperPoint only increases the channel dimensions and upsamples the feature maps. The SuperPoint layer has a shallow structure of a non-linear 1×1 convolutional layer and a upsampling layer. The non-linear 1×1 convolutional layer with ReLU6 activation is used to increase the channel dimensions of each branch. The upsampling layer increases the spatial resolution of the feature maps in a non-learned manner, which is much faster than using transposed convolutions (Sarlin et al. 2019). After SuperPoint layer, the output feature maps are $\{P4, P11, P18\}$ corresponding to $\{C4, C11, C18\}$. Compared to $\{C4, C11, C18\}$, the channel dimensions for $\{P4, P11, P18\}$ increase 3, 1.5, 1.1 times respectively, and their spatial resolutions is brought back to half of the input resolution, which retains more landmark details. Following the practice from (Lin et al. 2017), we also append a 3×3 convolution on each output in order to reduce the aliasing effect from upsampling. The obtained features from each feature extraction branch are aggregated by channel-wise concatenation. The aggregated features have channel dimensions of 520. Afterwards, the dense features are fed into the one-stage feature decoder to produce image representation, which is discussed in the next section.

The core concept for our design is to leverage the dense features to enhance image representations. Following the simple rule, our approach can use many design choices.

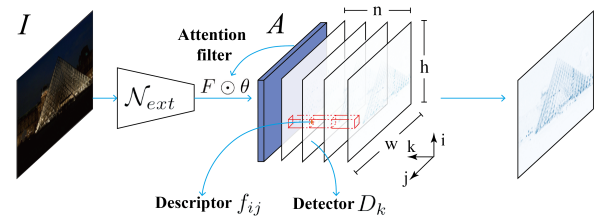


Figure 3. One-stage feature decoder. We use the attention filter to produce attention features and couple descriptor and detector together to delineate image representation.

Adding more feature extraction branches (more than three) can improve the prediction performance but demand more memory footprint. Empirically, our method has the best runtime and accuracy tradeoff. Our design is flexible to any CNN backbone. To achieve higher accuracy, we also implement a VGG16 (Simonyan and Zisserman 2014) version of DenserNet.

One-stage Feature Decoder

The feature extraction branches and the backbone network work together as the feature extraction encoder \mathcal{N}_{ext} that processes the input image I and produces the dense feature maps $F = \mathcal{N}_{ext}(I)$, $F \in \mathbb{R}^{h \times w \times n}$ ($h \times w$ is the feature map size and n is the number of channels). Similarly to (Dusmanu et al. 2019), we employ a one-stage feature decoder to delineate the image representation. Our one-stage feature decoder includes the feature attention filter, the feature descriptor, and the feature detector as shown in Figure 3.

Feature attention filter. Feature attention filter implicitly encodes spatial concept into the parameters of the attention features, which are flexible to represent the irregular landmarks. The attention features are computed by matrix multiplication of dense features F and attention filters θ :

$$A = \mathbb{R}(F \odot \theta), A \in \mathbb{R}^{h \times w \times n}, \quad (1)$$

where A is the attention features and \mathbb{R} is the ReLU activation. Based on CAM (Zhou et al. 2016), the attention filter θ screens out the spatial locations with pedestrians, vehicles, and vegetation on the feature maps.

Feature descriptor. The attention feature maps A can be expressed by a set of descriptor vectors f :

$$f_{ij} = A_{ij}, f_{ij} \in \mathbb{R}^n, \quad (2)$$

where $i \in \mathbb{R}^h$ and $j \in \mathbb{R}^w$. These descriptor vectors can be used to establish feature correspondences by calculating the Euclidean distance between images. We use $L2$ normalization to make the obtained descriptors in unit length. Then, we properly adjust these descriptors in training so that the same points for a scene produce similar descriptors, which robustly describe appearance variations.

Feature detector. In the same vein, the attention feature maps A can be expressed as a collection of detector D :

$$D_k = A_{:,k}, D_k \in \mathbb{R}^{h \times w} \quad (3)$$

where $k \in \mathbb{R}^n$. In this expression, the feature extraction network \mathcal{N}_{ext} produces n different detection response maps D_k . If pixel point (i, j) is detected, we denote a hard feature detector $D_{(ij)k}$ which is the most strong detection in all channels. We then perform a channel-wise softmax around its neighbours to obtain the local softmax score:

$$s_{ij} = \frac{\exp(D_{(ij)k})}{\sum_{i'=i-1}^{i+1} \sum_{j'=j-1}^{j+1} \exp(D_{(i'j')k})}. \quad (4)$$

Finally, we compute an image-level normalization for the softmax score to obtain the detection score at a pixel (i, j) :

$$\tilde{s}_{ij} = \frac{s_{ij}}{\sum_{i'=1}^h \sum_{j'=1}^w s_{i'j'}}. \quad (5)$$

Time Complexity Analysis

Similar to the conventional methods (Arandjelovic et al. 2016; DeTone, Malisiewicz, and Rabinovich 2018), our method has a backbone encoder for feature extraction and feature decoder for image representation. However, our approach also includes three feature extraction branches stick out from the lower, middle, higher level of the backbone network to efficiently produce denser features. Unlike patch-based networks MNV (Sarlin et al. 2018) and LF-Net (Ono et al. 2018) who adopt a Siamese sub-network to produce more features with a high computational cost, our approach has shared features across the three branches, which effectively avoids the computation overhead.

For runtime complexity, the ratio of our approach versus the conventional methods is:

$$r = \frac{\mathcal{O}(\mathcal{N}_{cnn}) + 3 \times \mathcal{O}(\mathcal{N}_{br}) + \mathcal{O}(\mathcal{N}_{de})}{\mathcal{O}(\mathcal{N}_{cnn}) + \mathcal{O}(\mathcal{N}_{de})}. \quad (6)$$

where $\mathcal{O}(\mathcal{N}_{cnn})$, $\mathcal{O}(\mathcal{N}_{br})$, and $\mathcal{O}(\mathcal{N}_{de})$ are the function complexity for the backbone encoder, the feature extraction branch, and the feature decoder for image representation. Since $\mathcal{O}(\mathcal{N}_{de}) \ll \mathcal{O}(\mathcal{N}_{cnn})$, the ratio can be approximated as: $r \approx 1 + \frac{3 \times \mathcal{O}(\mathcal{N}_{br})}{\mathcal{O}(\mathcal{N}_{cnn})}$. Compared to the conventional

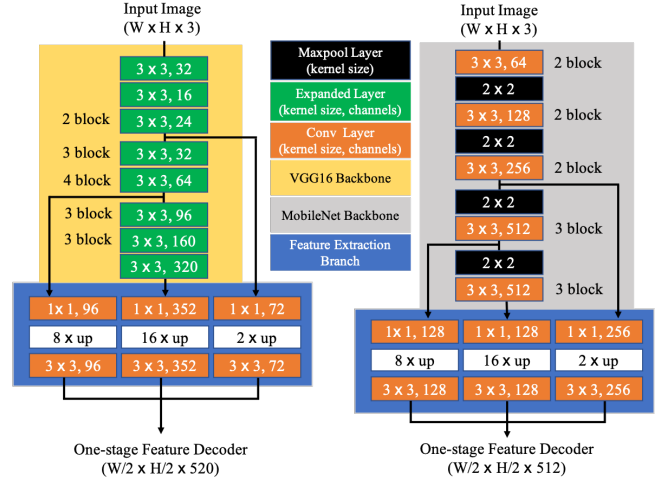


Figure 4. Architectures of the proposed method.

methods, the increased computational cost of the proposed method mainly comes from \mathcal{N}_{br} branches. This increased computational cost can be ignored because $\mathcal{O}(\mathcal{N}_{br}) \ll \mathcal{O}(\mathcal{N}_{cnn})$. Thus, our approach is appealing for its efficiency. We report empirical results in the later runtime evaluation section to resonate with our analysis here.

Model Architecture

In our experiments, we implement two models of variants which are based on MobileNetV2 (Sandler et al. 2018) and VGG-16 (Simonyan and Zisserman 2014). The two architectures are displayed in Figure 4. The MobileNet-based method is discussed in the section of Feature Extraction Branch. For VGG-based method, its feature extraction branches stick out at Con3_2, Con4_3, and Con5_3 respectively, and their corresponding feature maps are $\{P3_2, P4_3, P5_3\}$. In each feature extraction branch, the channel dimensions for $\{P3_2, P4_3, P5_3\}$ are multiplied 1, 0.25, 0.25 times respectively, and their spatial resolutions are brought back to one-fourth of the input resolution. The obtained features from each feature extraction branch are aggregated by channel-wise concatenation.

Training Objective

We propose a modified triplet ranking loss to jointly optimize our detectors and descriptors in a weakly supervised manner, which only requires cheap image triplets for training. In our setting, the location of a query image I_q is approximated by searching for the nearest neighbors in feature space among the reference images $\{I_r\}$. Thus, the objective for training our method is to match the positive references $\{I_r^+\}$ which are closer to the query image and vice-versa to the negative ones $\{I_r^-\}$.

For a pair of images (I^1, I^2) and potential corresponding feature points $P : p_1 \leftrightarrow p_2$ between them (where $p_1 \in I^1, p_2 \in I^2$), we want to minimize the distance $\sum_{p \in P} \|f_p^1 - f_p^2\|_2$ of the corresponding descriptors between the positive pairs while maximize the distance

between the negative ones. In order to increase the repeatability of detection (Dusmanu et al. 2019), we also add a detection term to encourage repeatability of effective detection between two images:

$$\mathcal{R}(I^1, I^2) = \sum_{p \in P} \frac{\tilde{s}_p^1 \tilde{s}_p^2}{\sum_{p' \in P} \tilde{s}_{p'}^1 \tilde{s}_{p'}^2} \|f_p^1 - f_p^2\|_2. \quad (7)$$

where P is the set of all corresponding feature points between I^1 and I^2 . \tilde{s}_p^1 and \tilde{s}_p^2 are the detection scores in (4) at each corresponding feature point of the paired images. Accordingly, our training objective can be defined as:

$$\mathcal{L}_{\mathcal{R}}(I_t, I_r^+, I_r^-) = \max(M + \mathcal{R}(I_t, I_r^+) - \mathcal{R}(I_t, I_r^-), 0). \quad (8)$$

where I_t , I_r^+ , and I_r^- are the training query image, the positive reference, and the negative reference respectively. In order to minimize the proposed loss, the distances of the discriminative descriptors between the training query and the positive reference are encouraged to be small and the associated detection scores are enforced to be large. Using the weakly supervised fashion for training, our method effectively learns feature representation pertaining to which features should be suppressed or emphasized.

Experiment and Results

In this section, we first describe the implementation details and the evaluation datasets. We then use an ablation study to investigate the improvements of our method from the baseline method. Next, we test the proposed method on several tasks in comparison with some of the state-of-the-art methods. Finally, we conclude with the runtime evaluation.

Implementation Details

Training data mining. We train the proposed method by using Pitts30k-training dataset (Arandjelovic et al. 2016). Following (Arandjelovic et al. 2016; Jin Kim, Dunn, and Frahm 2017), we group the positive $\{I_r^+\}$ and negative $\{I_r^-\}$ images for each training query image I_t . The positive images are the closest neighbors to each query image in the feature space at its nearby geo-locations, while the negative image is far away. Our training data mining purposefully selects positive images with fewer dynamic objects (i.e. pedestrians or bicycles). We will demonstrate that the stationary training data is beneficial to improve the localization behavior in the experiment. The 30K training query images generate four image triplets. Thus, we obtain a total of 120K image triplets with 112K for training and 8K for validation.

Training process. All experiments are performed on a workstation with an Intel Core i7-7820X CPU and four NVIDIA GeForce GTX 3080Ti GPU. Both VGG16 and MobileNetV2 based methods are pretrained on ImageNet (Deng et al. 2009). In training, we exploit standard data augmentation in training, such as motion blur, random Gaussian noise, brightness changes to improve the robustness of our methods to illumination variations and viewpoint changes. Specifically, the margin M is set at 0.1, 30 epochs are performed using batch size of 4 triplets, Adam (Kingma and Ba 2014) with the learning rates of 10^{-3} which is halved every

6 epochs, momentum of 0.9, and weight decay of 10^{-3} . We use the Precision-Recall curve to evaluate the training performance (Arandjelovic et al. 2016). A query is considered to be correct if at least one result from the top N retrieved database images is within $d = 25$ meters from the ground truth position of the query image. We use the best method (the highest *recall@5*) on the validation for testing.

Evaluation Datasets and Metrics

We assess our method on three different tasks with a number of publicly available benchmarks.

Feature matching task. HPatches dataset (Balntas et al. 2017) is adopted to evaluate the effectiveness of feature extraction and matching. Following (DeTone, Malisiewicz, and Rabinovich 2018), we use the detection repeatability and mean localization error (MLE) of the keypoint features to evaluate the detector; and we use the mean average precision (mAP) and the matching score (MS) to evaluate the descriptor. The mAP assesses a method’s ability to suppress spurious matches. MS reflects the overall performance of both the detector and the descriptor.

Large-scale visual localization task. Pitts250k-test (Torii et al. 2013), Tokyo 24/7 (Torii et al. 2015), TokyoTM-val (Arandjelovic et al. 2016), and and Sf-0 (Chen et al. 2011) are used to investigate the performance of visual localization. For the first three benchmarks, a query is considered to be correct if at least one result from the top N retrieved database images is within $d = 25$ meters from the ground truth position of the query image. For Sf-0, the query result is considered to be correct if at least one result from the top N retrieved database images has the same building ID.

Image retrieval task. Since we cast the visual localization problem as an image retrieval task, we also evaluate our method on image retrieval benchmarks. Oxford 5k (Philbin et al. 2007), Paris 6k (Philbin et al. 2008), and Holidays (Jegou, Douze, and Schmid 2008) is used to test the generalization of our method for image representations on image retrieval. We use mAP for evaluation.

Ablation Study

We conduct an ablation study to validate the improvements of DenserNet from a strong baseline and its variants. Method (a) is the baseline which only has the higher-level branch for feature extraction. It is identical to SuperPoint (DeTone, Malisiewicz, and Rabinovich 2018). Method (b) and (c) are two variants of the baseline method with the additional lower-level or mid-level branch respectively. Method (d) is the proposed method. The ablation study is conducted on feature matching and visual localization tasks.

The results for feature matching are demonstrated in Table 1. As expected, VGG-based methods outperform their MobileNet-based counterparts because VGG16 has a better feature extraction capacity. From method (a) to method (d), adding the feature branch can give a boost in the performance of both detector and descriptor. The repeatability of keypoint features increases with the increase of the feature extraction branch while errors only increase with a small scale. Meanwhile, mAP and matching scores also increase

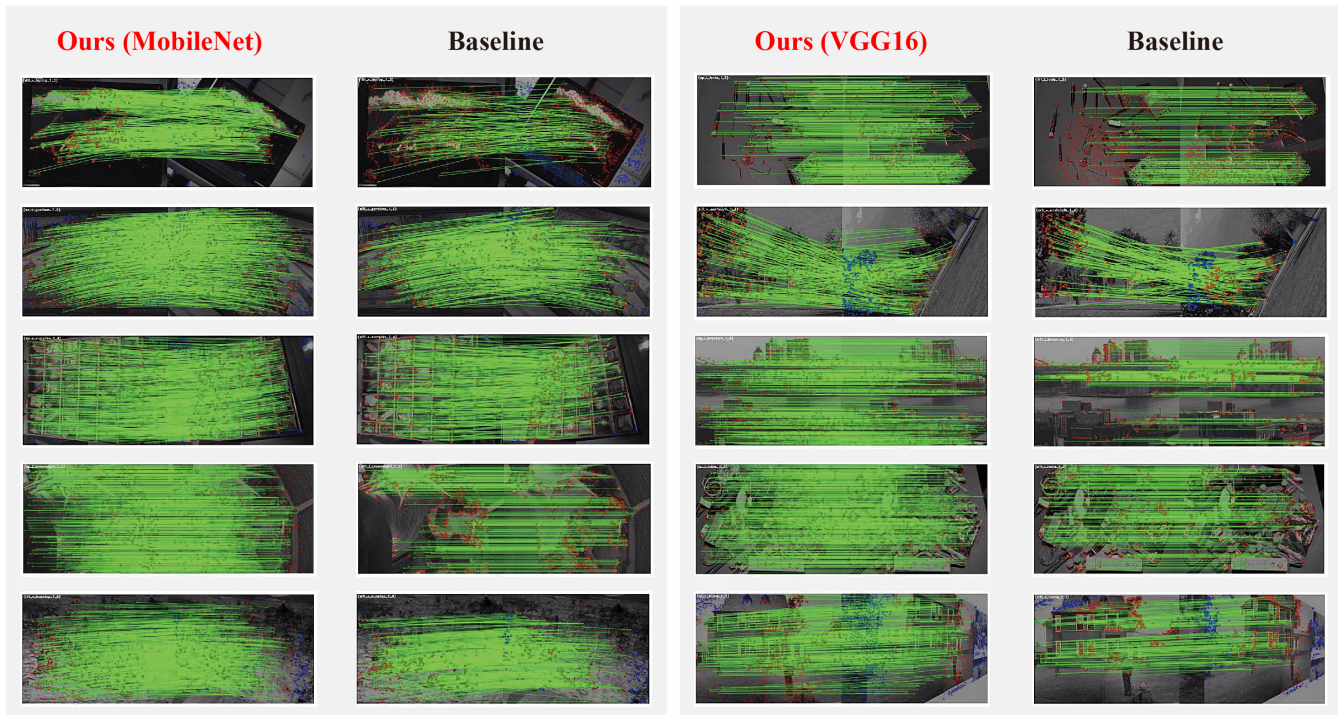


Figure 5. Qualitative examples on the HPatches dataset. The green lines indicate all correct correspondences which are repeatable. DenserNet can produce denser and more correct matches compared to the baseline method. Red dots are mis-matched points and blue dots are not visible to the shared viewpoint region of the image pairs. Best viewed in the digital format with zoom.

Meth.	HB	LB	MB	Det. Metric		Des. Metric	
				Rep.	MLE	mAP	MS
a (M)	✓			0.556	0.99	0.782	0.448
b (M)	✓	✓		0.568	1.02	0.799	0.456
c (M)	✓		✓	0.573	1.03	0.808	0.461
d (M)	✓	✓	✓	0.592	1.04	0.827	0.473
a (V)	✓			0.578	1.12	0.822	0.462
b (V)	✓	✓		0.588	1.14	0.847	0.471
c (V)	✓		✓	0.596	1.15	0.859	0.475
d (V)	✓	✓	✓	0.628	1.17	0.886	0.481

Table 1. Evaluation results from HPatches dataset. We use HPatches to examine the ability for feature extraction and matching. Method a is the baseline while method b and c are its variants. Method d is the proposed DenserNet. The results indicate the improvements of DenserNet from baseline in the all metrics. M and V in methods stand for MobileNet and VGG-based backbones respectively. HB, LB, MB stand for higher-level, lower-level, and mid-level branch respectively.

accordingly by adding the feature extraction branch. Figure 5 demonstrates the qualitative results of the VGG-based method. Our method can produce more keypoint features for effective matching than the baseline method.

The results for all VGG-based methods on large-scale vi-

sual localization tasks are demonstrated in Figure 6. Method (d) outperforms the baseline method and its variants on all benchmarks. We observe a steady increase of $recall@N$ from method (a) to method (d). Based on the Precision-Recall curve, using an extra feature extraction branch is proved to be effective to improve the performance of localization as methods having more feature extraction branches achieve better results on all benchmarks. The results for all MobileNet-based methods on large-scale visual localization tasks are demonstrated in Figure 7.

Comparison with State-of-the-Art Methods

To demonstrate the advancement of the proposed method, we compare our method with state-of-the-art methods on large-scale localization benchmarks and image retrieval tasks. We choose three leading methods NetVLAD, CRN, and SuperPoint for comparison. In order to have a fair comparison, we retrain all the methods with the same setup.

Large-scale localization benchmark results. We report the $recall@N$ results for different methods in Table 2. Both our VGG16 and MobileNet-based methods consistently outperform state-of-the-art methods by a significant margin on all benchmarks. For instance, on the Pitts250k dataset, our improvements over the next best method NetVLAD at $r@1$ is 3.45% for the VGG-based architecture and 1.87% for the MobileNet-based architecture. On the Sf-0 dataset, our VGG16 and MobileNet-based methods achieve $r@1$ of

Method	Pitts 250k-test			TokyoTM-val			Tokyo 24/7			Sf-0		
	r@1	r@5	r@10	r@1	r@5	r@10	r@1	r@5	r@10	r@1	r@5	r@10
Ours (VGG)	89.40	95.90	96.99	94.80	97.69	98.20	81.20	88.67	91.10	80.80	86.99	89.68
Ours (MobileNet)	87.82	94.89	96.26	94.62	97.37	98.12	79.17	87.99	90.17	79.12	85.68	88.56
CRN	85.50	93.49	95.50	93.07	95.97	97.61	75.39	83.81	87.31	77.62	84.31	86.80
NetVLAD	85.95	93.21	95.13	92.85	95.77	97.59	73.33	82.86	86.03	76.57	83.27	85.80
SuperPoint	85.78	93.36	95.26	92.81	96.04	97.53	75.37	83.44	86.73	75.52	84.01	86.60

Table 2. Comparison of Recalls at N top retrievals of different methods on the four large-scale visual localization benchmarks.

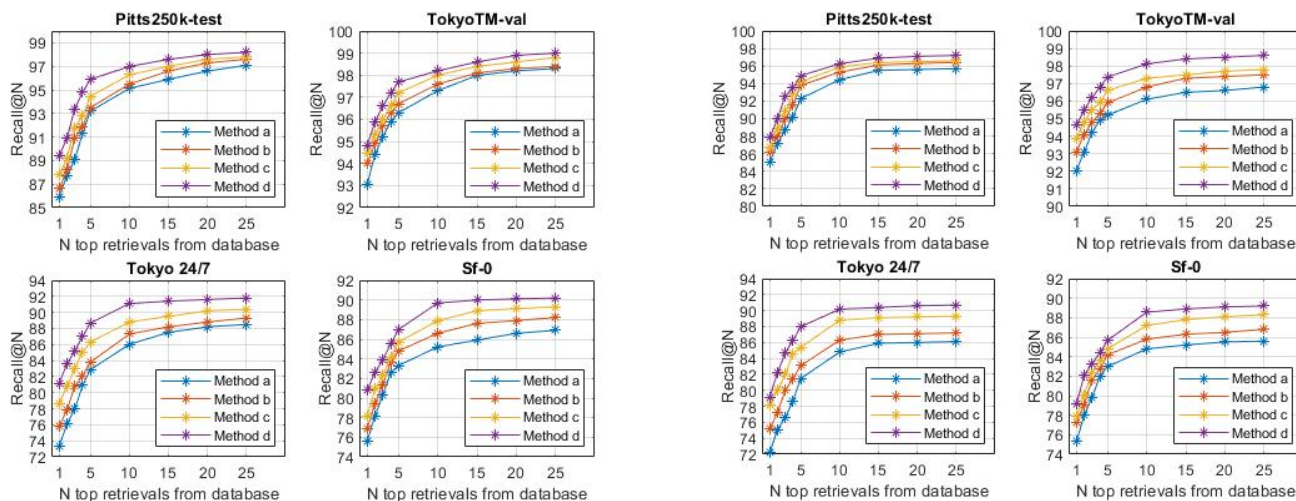


Figure 6. Ablation results of Recalls at N top retrievals with all VGG-based methods.

Method	Oxford 5k		Paris 6k		Holidays	
	full	crop	full	crop	orig	rot
Ours (V)	67.66	69.40	75.03	78.21	84.71	88.30
Ours (M)	66.88	68.26	75.00	78.30	84.01	88.28
CRN	63.95	65.52	72.88	75.85	83.19	87.30
NetVLAD	63.09	65.33	72.53	75.67	82.67	86.83
SuperPoint	63.14	65.50	72.83	75.10	82.92	86.90

Table 3. Comparison with state-of-the-art methods for compact image representations (256-D) on image retrieval tasks.

80.08% and 79.12% respectively compared to the next best method CRN by a margin of 3.18% and 1.5% respectively. Our methods also obtain similar improvements on Tokyo 24/7 and Tokyo TM. The results from all benchmarks confirm our assumption for this proposed work: leveraging denser features from multiple semantic levels and using the right supervision for training, the proposed method can effectively learn discriminative yet compact image representations for visual localization. More detailed comparisons are displayed in Figure 8.

For qualitative analysis, we visualize the regions of the input image which are emphasized for localization prediction (as shown in Figure 9). Particularly, we use the heatmaps

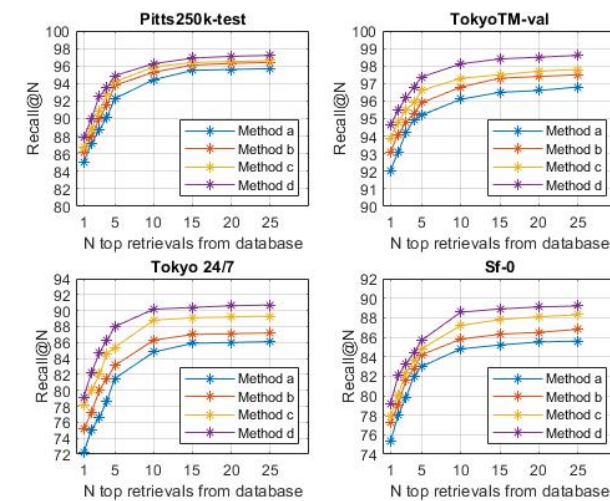


Figure 7. Ablation results of recalls at N top retrievals with all MobileNet-based methods.

(Grün et al. 2016) to highlight the feature emphasis of different regions on the input image. Based on the results, we observe that our method is superior to its counterparts in identifying more useful features for localization, despite the significant environmental variations in viewpoint or illumination. Our method focuses on the distinctive details of buildings that are identifiable for visual localization while avoiding confusing visual clues. In contrast, other methods generally focus on local features independently which are inherently limited. Many features are laid on confusing scenes such as pedestrians, vegetation, or vehicles which are hard for feature repeatability. We argue that our multi-scale feature aggregation, feature attention filter, training data mining, and training supervision technique collaboratively contribute to the improved localization behavior.

Image retrieval task results. To assess the generalizability of our approach, we evaluate our methods trained only on Pitts30k (Arandjelovic et al. 2016) without any fine-tuning on the standard image retrieval datasets. For Oxford 5k (Philbin et al. 2007) and Paris 6k (Philbin et al. 2008), we use both the full and cropped images; for Holidays (Jegou, Douze, and Schmid 2008), we use original and rotated images. The results are displayed in Table 3. Our results set the state-of-the-art for compact image representations (256-D) on all three datasets. On all

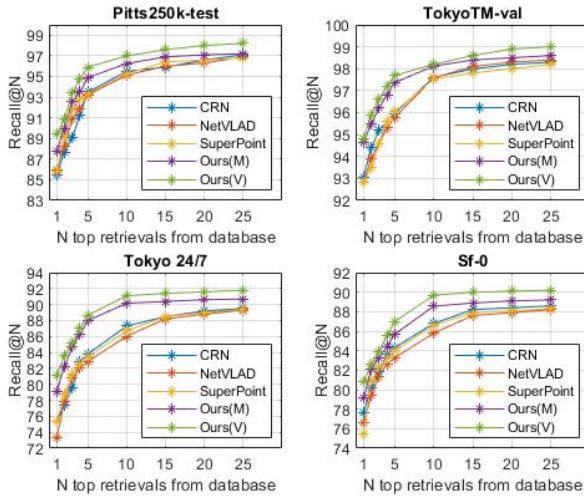


Figure 8. Comparison of recalls at N top retrievals with state-of-the-art methods. Ours (V) has VGG backbone and ours (M) has MobileNet backbone.

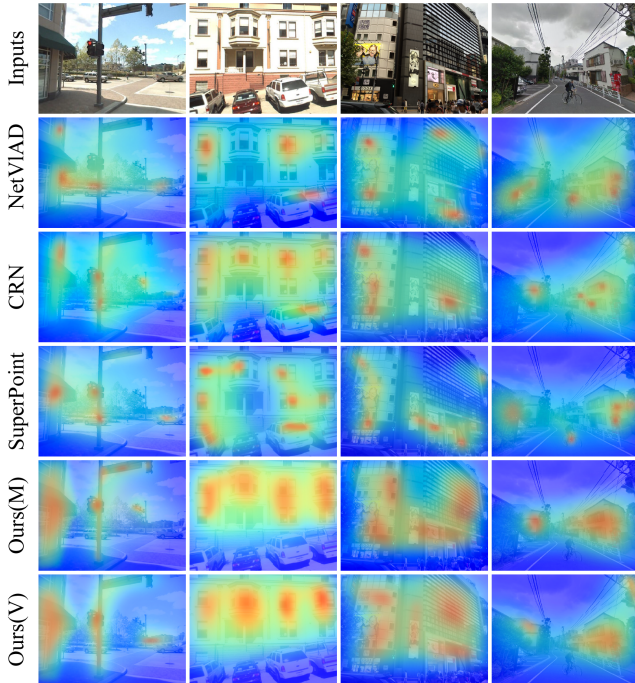


Figure 9. Comparison of feature emphasis. With the feature attention filter and training data mining, our methods focus on the distinctive details of buildings, while avoiding confusing visual clues such as pedestrians, vegetation, or vehicles which are hard for feature repeatability.

metrics, our margin consistently over the mAP of other methods is 1-4%. For example, there are a 3.71% (our VGG-based method) and 2.93% (our MobileNet-based method) improvements on Oxford 5k (full) than the next

Dataset	Method	Runtime(ms)				r@1
		BE	FEB	FD	Total	
Pitts250k-test	Ours (V)	92	6	2	100	89.40
	Ours (M)	15	6	2	23	87.82
	CRN	92	-	6	98	85.50
	NetVLAD	92	-	5	97	85.95
	SuperPoint	43	-	7	50	85.78
Tokyo 24/7	Ours (V)	96	7	3	106	81.20
	Ours (M)	18	7	3	28	79.17
	CRN	96	-	8	104	75.39
	NetVLAD	96	-	7	103	73.33
	SuperPoint	49	-	9	58	75.37

Table 4. Runtime report. BE, FEB, and FD means backbone encoder, feature extraction branch, and feature decoder respectively.

best method; there are a 3.88% (our VGG-based method) and 2.74% (our MobileNet-based method) improvements on Oxford 5k (crop) than the next best method. Since our methods only see building-oriented images, our results can be further improved by fine-tuning using the natural landmark images from the three image retrieval datasets.

Runtime Evaluation

In this work, we provide two versions of our method. VGG-based model is designed to achieve high performance in accuracy while our MobileNet-based model has a better efficiency with a small penalty on accuracy. We analyze its runtime and compare it with other state-of-the-art methods. All the measurements are conducted on the same workstation. As shown in Table 4, the increase of our computation cost stems from the feature extraction branch (FEB). One advantage of our method is that our FEB shares features and avoids runtime overhead. The increased computational cost is affordable as it is much smaller than that of the backbone encoder and has little impact on the total runtime cost. Specifically, our MobileNet-based models are around $4\times$ faster than CRN and NetVLAD which have the VGG-based architecture. For our VGG-based model, our speed is comparable to CRN and NetVLAD which have the same backbone network. Although our VGG-based models are slower than SuperPoint which uses a smaller backbone architecture, our methods outperform counterparts in localization $r@1$ by a large margin.

Conclusion

In this work, we propose DenserNet, a novel CNN-based architecture that aggregates denser features from multiple semantics to achieve strong image representation. Results from extensive experiments indicate that our method is competitive with the current state-of-the-art methods on large-scale localization tasks with the same level of supervision.

References

- Arandjelovic, R.; Gronat, P.; Torii, A.; Pajdla, T.; and Sivic, J. 2016. NetVLAD: CNN architecture for weakly supervised place recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 5297–5307.
- Balntas, V.; Lenc, K.; Vedaldi, A.; and Mikolajczyk, K. 2017. HPatches: A benchmark and evaluation of handcrafted and learned local descriptors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 5173–5182.
- Chen, D. M.; Baatz, G.; Köser, K.; Tsai, S. S.; Vedantham, R.; Pylvänäinen, T.; Roimela, K.; Chen, X.; Bach, J.; Pollefeys, M.; et al. 2011. City-scale landmark identification on mobile devices. In *CVPR 2011*, 737–744. IEEE.
- Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, 248–255. Ieee.
- DeTone, D.; Malisiewicz, T.; and Rabinovich, A. 2018. Superpoint: Self-supervised interest point detection and description. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 224–236.
- Dusmanu, M.; Rocco, I.; Pajdla, T.; Pollefeys, M.; Sivic, J.; Torii, A.; and Sattler, T. 2019. D2-Net: A Trainable CNN for Joint Detection and Description of Local Features. *arXiv preprint arXiv:1905.03561*.
- Grün, F.; Rupprecht, C.; Navab, N.; and Tombari, F. 2016. A taxonomy and library for visualizing learned features in convolutional neural networks. *arXiv preprint arXiv:1606.07757*.
- Huang, G.; Liu, Z.; Van Der Maaten, L.; and Weinberger, K. Q. 2017. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4700–4708.
- Jegou, H.; Douze, M.; and Schmid, C. 2008. Hamming embedding and weak geometric consistency for large scale image search. In *European conference on computer vision*, 304–317. Springer.
- Jin Kim, H.; Dunn, E.; and Frahm, J.-M. 2017. Learned contextual feature reweighting for image geo-localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2136–2145.
- Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- Lin, T.-Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; and Belongie, S. 2017. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2117–2125.
- Liu, D.; Cui, Y.; Chen, Y.; Zhang, J.; and Fan, B. 2020a. Video Object Detection For Autonomous Driving: Motion-aid Feature Calibration. *Neurocomputing*.
- Liu, D.; Cui, Y.; Guo, X.; Ding, W.; Yang, B.; and Chen, Y. 2020b. Visual Localization for Autonomous Driving: Mapping the Accurate Location in the City Maze.
- Liu, L.; Li, H.; and Dai, Y. 2019. Stochastic Attraction-Repulsion Embedding for Large Scale Image Localization. In *Proceedings of the IEEE International Conference on Computer Vision*, 2570–2579.
- Luo, Z.; Zhou, L.; Bai, X.; Chen, H.; Zhang, J.; Yao, Y.; Li, S.; Fang, T.; and Quan, L. 2020. Aslfeat: Learning local features of accurate shape and localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6589–6598.
- Ono, Y.; Trulls, E.; Fua, P.; and Yi, K. M. 2018. LF-Net: learning local features from images. In *Advances in neural information processing systems*, 6234–6244.
- Philbin, J.; Chum, O.; Isard, M.; Sivic, J.; and Zisserman, A. 2007. Object retrieval with large vocabularies and fast spatial matching. In *2007 IEEE conference on computer vision and pattern recognition*, 1–8. IEEE.
- Philbin, J.; Chum, O.; Isard, M.; Sivic, J.; and Zisserman, A. 2008. Lost in quantization: Improving particular object retrieval in large scale image databases. In *2008 IEEE conference on computer vision and pattern recognition*, 1–8. IEEE.
- Salarian, M.; Iliev, N.; Cetin, A. E.; and Ansari, R. 2018. Improved Image-Based Localization Using SFM and Modified Coordinate System Transfer. *IEEE Transactions on Multimedia* 20(12): 3298–3310.
- Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; and Chen, L.-C. 2018. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4510–4520.
- Sarlin, P.-E.; Cadena, C.; Siegwart, R.; and Dymczyk, M. 2019. From coarse to fine: Robust hierarchical localization at large scale. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 12716–12725.
- Sarlin, P.-E.; Debraine, F.; Dymczyk, M.; Siegwart, R.; and Cadena, C. 2018. Leveraging deep visual descriptors for hierarchical efficient localization. *arXiv preprint arXiv:1809.01019*.
- Simonyan, K.; and Zisserman, A. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.
- Tolias, G.; Avrithis, Y.; and Jégou, H. 2016. Image search with selective match kernels: aggregation across single and multiple images. *International Journal of Computer Vision* 116(3): 247–261.
- Torii, A.; Arandjelovic, R.; Sivic, J.; Okutomi, M.; and Pajdla, T. 2015. 24/7 place recognition by view synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1808–1817.
- Torii, A.; Sivic, J.; Pajdla, T.; and Okutomi, M. 2013. Visual place recognition with repetitive structures. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 883–890.
- Zhou, B.; Khosla, A.; Lapedriza, A.; Oliva, A.; and Torralba, A. 2016. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2921–2929.