Discriminative Features Matter: Multi-layer Bilinear Pooling for Camera Localization

Xin Wang*[1]
wangx@buaa.edu.cn
Xiang Wang*[1]
vixwang@buaa.edu.cn
Chen Wang[1]
wangchenbuaa@buaa.edu.cn
Xiao Bai[1]
baixiao@buaa.edu.cn
Jing Wu[2]
wuj11@cardiff.ac.uk
Edwin Robert Hancock[1,3]
edwin.hancock@york.ac.uk

[1] School of Computer Science and Engineering, Beijing Advanced Innovation Center for Big Data and Brain Computing, Jiangxi Research Institute
Beihang University
Beijing, China

[2] School of Computer Science and Informatics
Cardiff University
Cardiff, U.K

[3] Department of Computer Science
University of York
York, U.K

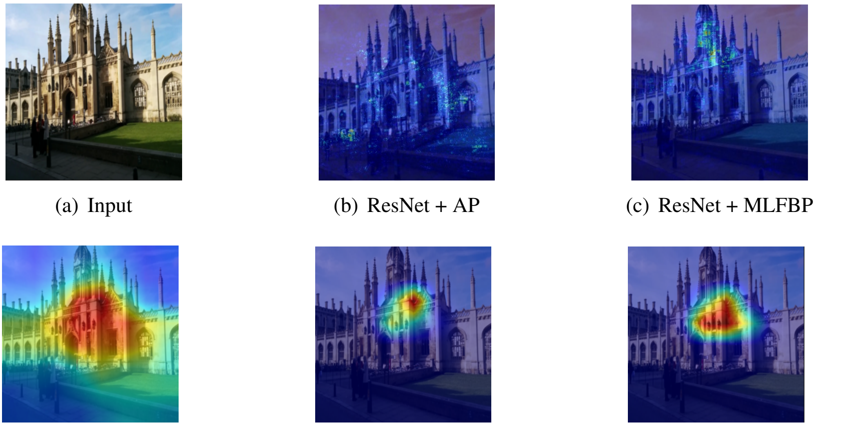
Abstract

Deep learning based camera localization from a single image has been explored recently since these methods are computationally efficient. However, existing methods only provide general global representations, from which an accurate pose estimation can not be reliably derived. We claim that effective feature representations for accurate pose estimation shall be both "informative" (focusing on geometrically meaningful regions) and "discriminative" (accounting for different poses of similar images). Therefore, we propose a novel multi-layer factorized bilinear pooling module for feature aggregation. Specifically, informative features are selected via bilinear pooling, and discriminative features are highlighted via multi-layer fusion. We develop a new network for camera localization using the proposed feature pooling module. The effectiveness of our approach is demonstrated by experiments on an outdoor *Cambridge Landmarks* dataset and an indoor *7 Scenes* dataset. The results show that focusing on discriminative features significantly improves the network performance of camera localization in most cases. Codes will be available soon.

1 Introduction

Camera localization is a task to determine the absolute pose (position and orientation) of the camera in the scene given an observed image. It is a vital task of many computer vision applications such as SLAM, augmented reality, autonomous driving and visual navigation. Early methods estimate the camera pose based on feature matching between a given 2D image and the whole scene information provided in the form of either a 3D model or an

* These authors contributed equally.



(a) Input (b) ResNet + AP (c) ResNet + MLFBP

(d) Original features (e) Bilinear features (f) Multi-layer bilinear features

Figure 1: Comparison of the saliency map (b)(c) and activation map (d)(e)(f) between using average pooling (AP) and multi-layer factorized bilinear pooling (MLFBP). It is obvious that bilinear pooling can drive the network to focalize more informative regions such as building parts, and multi-layer fusion helps enlarge discriminative parts for more accurate results.

image database. While these methods work for many scenes, erroneous or no pose estimation might be given in cases where hand-crafted features fail to be correctly matched, such as in textureless or repetitive scenes. Also, searching for correct matches in a large-scale 3D model or retrieving the most similar image in a large dataset is time-consuming, which requires efficient retrieval techniques [1, 2, 21]. Recently, deep learning approaches have drawn much attention due to their ability to extract more representative features. Some attempts [11, 12, 13, 24] have been made to directly regress the camera pose from an input image with the powerful feature learning capability of CNN. These methods are computationally efficient and work when feature-based methods fail.

Nonetheless, previous methods like PoseNet [13] use average pooling to aggregate feature maps into a holistic feature vector for pose estimation. Such feature representation is not optimal since a larger area than required is activated. Some uninformative features might produce unreliable pose results and should be discarded. Thus, camera localization requires more discriminative details that account for precise camera pose estimation. These details should satisfy two conditions. First, they should be "informative", i.e., the activated features should lie in geometrically meaningful regions. For example, in the outdoor scenes in Figure 1(b-e), building parts should be more focused, but sky or roads are trivial since they are common in many images. Second, the details should be "discriminative", i.e., distinct parts in informative regions can be located when facing similar images captured for different locations. Figure 1(f) highlights more essential parts of the building, leading to more reasonable result.

To highlight informative details, inspired by the recent works from fine-grained image recognition [18], we propose to employ bilinear pooling techniques to enhance the features for camera localization. Bilinear pooling forms a global image descriptor by computing the outer product of feature maps from CNN. Specifically, it calculates the correlation between different channels of feature maps, and amplifies the activation of informative areas implicitly. Different from spatial pooling methods like average pooling and max pooling that introduce invariance to image deformation, bilinear pooling obtains statistics that maintain feature selectivity. In camera localization, bilinear pooling helps the network to focus more on those geometrically meaningful parts, and suppresses the activations in trivial regions that produce uncertain pose estimation.

Although trivial regions for localization can be suppressed, single-layer bilinear pooling at the last layer may overemphasize some parts but underrate other informative regions, like those accounting for different locations with similar appearances. Combining multi-layer features is an option to complement some missing details in features from the last layer. We adopt the same bilinear pooling model to form cross-layer features between the bilinear feature from the last layer and original features from the preceding two layers, respectively. Multi-layer features is formed as a rich feature representation for discriminative details by concatenating both bilinear features and cross-layer features. As shown in Figure 1(e)(f), the activation map of using multi-layer bilinear pooling captures more discriminative parts than those from single-layer bilinear pooling.

Our work makes the following novel contributions. (1) We analyze the camera localization problem from the perspective of feature aggregation, and propose that both informative and discriminative features are important for pose regression. (2) We propose a multi-layer factorized bilinear pooling module to the feature pooling layer of the pose regression network. We utilize the factorized bilinear pooling approach to the last conv layer of the CNN to focus on geometrically meaningful regions, and adopt a multi-layer feature fusion strategy to address discriminative features that account for precise pose estimation. (3) Our method achieves superior performance on two camera localization datasets, *Cambridge Landmarks* and *7 Scenes*, using only a single image as input. Visualization results show that our method consistently activates informative and discriminative regions.

2 Related Works

Absolute camera pose regression tries to get camera poses directly from a given image by training a specific CNN, treating the weights of the network as a map representation for the task. PoseNet [13] is the first attempt towards end-to-end learning of 6DoF poses by appending a pose regression module to the pretrained GoogLeNet. Acting as the feature extractor, GoogLeNet pretrained on classification datasets like ImageNet or Places produces features that are not informative enough to pose regression. Subsequent works use Bayesian methods to estimate uncertainty of pose results [4, 11], and learn weights between the camera position and orientation loss as well as incorporating the reprojection loss given the scene model [12]. Both works don't address the impact of the features from pretrained CNN. Walch *et al.* [24] introduced LSTM units to the network for structured dimensionality reduction on the feature vector from CNN and improving localization results. Melekhov *et al.* [19] used an hourglass network to promote features by recovering fine-grained details. These approaches improve features from CNN in a global view without emphasizing informative details. Some other methods attempt to involve more information than a single input image in pose regression, such as sequences of images[6], other sensory perception (visual odometry, GPS, etc.) [3] or a multi-task framework (with visual odometry and semantic segmentation) [20, 23]. Aforementioned approaches involve more input than a single image, and thus are beyond the scope of this paper.

Bilinear pooling is a common technique of emphasizing the most informative part in the feature map from a holistic perspective by aggregating the pairwise feature interactions. This method is widely used in fine-grained image recognition [17, 18] whose goal is to distinguish subordinate categories that have similar appearances. By calculating the second-order statistics, feature selectivity is maintained and bilinear features gain more representational power. Recent works try to reduce computational burdens of bilinear pooling due to very high-dimensional feature representation via compact kernel design [7, 9] or low-rank approximation [5, 14, 15, 16, 25]. While most works apply bilinear pooling only after the last convolution layer, inter-layer part interactions are neglected. Cai *et al.* [5] models the interactions between layers by concatenating the activation maps from multiple convolution layers. The most recent work [25] employs bilinear pooling in a cross-layer manner, capturing inter-layer feature relations and archiving the best performance in fine-grained recognition task. While these methods mostly focus on fine-grained image recognition task, our proposed approach is designed for camera localization problem from a feature learning and fusion perspective.

3 Methodology

In this section, we develop our framework for regressing the camera pose directly from an input image of the scene. Our goal is to train a network to learn a mapping f from an image I to its absolute pose $\mathbf{p}$, $I \xrightarrow{f} \mathbf{p}$. The mapping f is done via a neural network, composed of a CNN feature extractor, a feature aggregator (commonly using a pooling layer) and a fully connected pose regressor. In this paper, we focus on feature aggregator and reckon that it should play two roles: selecting the most informative features for accurate pose regression and fusing discriminative features that account for different poses of similar images. From this perspective, we propose a multi-layer factorized bilinear pooling module for feature selection and fusion. Based on this module, we design a network for camera pose regression, whose architecture is illustrated in Figure 2.

3.1 Factorized Bilinear Pooling for Feature Correlation

Previous deep learning methods, such as PoseNet, use an average pooling layer after the last convolution layer to gather the information of each feature channel. Although spatial pooling methods like average pooling provide adequate information for image recognition tasks, such feature aggregation neglects details that account for different poses in camera localization, hence leads to improper activation. As illustrated in Figure 1(b), networks with average pooling will activate some uninformative areas like sky or roads.

Bilinear pooling models interactions of features by computing the outer product of two feature vectors. Compared to common first-order pooling methods, bilinear pooling brings more powerful representations by capturing feature correlations. Thus it can encourage network to suppress the activation from unrelated regions to the task. Therefore, we use bilinear pooling to replace average pooling for feature aggregation. Figure 1(c) plots the saliency map

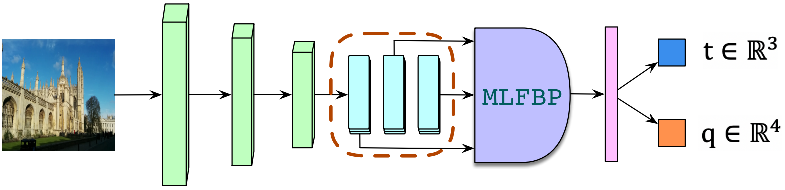


Figure 2: Network architecture of our proposed method.

of *Cambridge Landmarks* dataset. Notice that bilinear pooling focuses more on distinctive building elements, compared with the result of average pooling.

Denoting the feature maps by $\chi \in \mathbb{R}^{c\times hw}$ and each feature vector by $\mathbf{x}_i \in \{\mathbf{x}_i | \mathbf{x}_i \in \mathbb{R}^c, i \in S\}$, where h, w and c are the height, width, and number of channels, respectively, bilinear pooled features can be computed as:

$$\mathbf{B}(\chi) = \chi\chi^T = \sum_{i\in S} \mathbf{x}_i\mathbf{x}_i^T \tag{1}$$

However, bilinear pooling generally has a large dimensional output, e.g. $c \times c = 262,144$ when $c = 512$, leading to high computational cost and a risk of overfitting. Recently, many researchers [9, 14, 16] present factorized bilinear model to reduce the output dimensionality of bilinear pooling. When appending a fully connected layer after bilinear pooling as a classification layer or a projection matrix for feature embedding [16], the bilinear pooling can be rewritten as:

$$\mathbf{z} = \mathbf{b} + \mathbf{W}vec(\mathbf{B}(\chi)) = \mathbf{b} + \mathbf{W}vec(\sum_{i\in S} \mathbf{x}_i\mathbf{x}_i^T) \tag{2}$$

$$z_j = b_j + \mathbf{W}_j^T vec(\sum_{i\in S} \mathbf{x}_i\mathbf{x}_i^T) = b_j + \sum_{i\in S} \mathbf{x}_i^T \mathbf{W}_j^R \mathbf{x}_i \tag{3}$$

where $\mathbf{W} \in \mathbb{R}^{c^2\times d}$ is the projection matrix and $\mathbf{W}_j^R \in \mathbb{R}^{c\times c}$ is a matrix reshaped from $\mathbf{W}_j$ which is the j-th row of $\mathbf{W}$. A low-rank bilinear method is suggested to reduce the rank of the weight matrix $\mathbf{W}_j^R$ to have less parameters for regularization [14]. Specifically, $\mathbf{W}_j^R$ is decomposed as $\mathbf{W}_j^R = \mathbf{U}_j\mathbf{V}_j^T$ where $\mathbf{U}_j$ and $\mathbf{V}_j$ are one-rank vectors. So equation (3) can be rewritten as:

$$z_j = b_j + \sum_{i\in S} \mathbf{x}_i^T \mathbf{U}_j \mathbf{V}_i^T \mathbf{x}_i = b_j + \sum_{i\in S} \mathbf{U}_j^T \mathbf{x}_i \circ \mathbf{V}_j^T \mathbf{x}_i = SumPooling(\mathbf{U}_j^T \mathbf{x} \circ \mathbf{V}_j^T \mathbf{x}) \tag{4}$$

where *SumPooling* is a pooling operation which sums the value of all spatial locations in each feature map and $\circ$ is the Hadamard product. Redefining $\mathbf{U}, \mathbf{V} \in \mathbb{R}^{c\times d}$ as low rank projection matrices, equation (2) becomes:

$$\mathbf{z} = SumPooling(\mathbf{U}^T \mathbf{x} \circ \mathbf{V}^T \mathbf{x}) \tag{5}$$

To further increase the model capacity and avoid overfitting, nonlinear activation, like *tanh* or *ReLU*, and dropout can be added after the projection operation. We replace the traditional average pooling by the factorized bilinear pooling to enhance the correlation of features, encouraging the network to focus on the meaningful areas of the input image.

3.2 Multi-layer Bilinear Pooling for Feature Fusion

Some recent work show that deeper convolutional filters can work as weak part detectors [22, 27] and activations from different convolutional layers can be treated as represen-

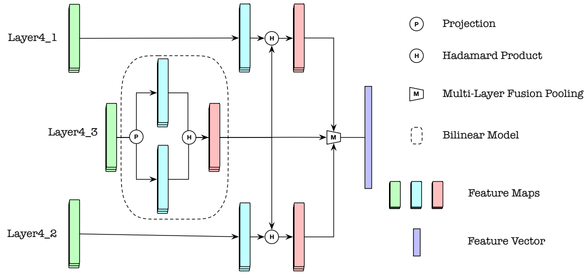


Figure 3: Multi-layer Bilinear Pooling Module.

tations of different part properties [6, 25]. Therefore, modeling inter-layer part interactions can help the network to extract more discriminative features. Motivated by this observation, we propose to integrate the features from multiple convolutional layers to capture the interaction of multiple discriminative part attributes. In the recent methods of Visual Question Answering (VQA), bilinear model has been regarded as a multi-modal fusion approach for combining representations from different modalities into a single representation [8, 26]. In addition, Yu *et al*. developed a cross-layer bilinear pooling to model the interactions of different convolution layers [25] to improve the capability of fine-grained feature learning. Inspired by these works, we develop a multi-layer bilinear model by combining the bilinear feature of the last block with features of the preceding two blocks in ResNet.

Using factorized bilinear pooling as a feature fusion approach, equation (5) can be further applied between two sets of feature maps χ and γ, similar to the cross-layer bilinear pooling operation [25]:

$$F(\chi,\gamma) = SumPooling(\mathbf{U}^T\chi \circ \mathbf{S}^T\gamma) \tag{6}$$

Since the feature maps from deeper layers have semantic information which is more related to the target task, and the bilinear feature map of the last conv layer is more informative, we employ the bilinear features of the last layer as one of the features in equation (6). Such operation can be seen as using features from preceding layers to complement the final bilinear features. Therefore, our complete multi-layer bilinear model can be written as:

$$\begin{aligned} F(\chi,\gamma,\zeta) = \mathbf{P}^T Concat(&SumPooling((\mathbf{U}^T\chi \circ \mathbf{V}^T\chi)\circ(\mathbf{S}^T\gamma)), \\ SumPooling((\mathbf{U}^T\chi \circ \mathbf{V}^T\chi))&\circ(\mathbf{W}^T\zeta), SumPooling(\mathbf{U}^T\chi \circ \mathbf{V}^T\chi)) \end{aligned} \tag{7}$$

where $\mathbf{P} \in \mathbb{R}^{d\times n}$ is a projection matrix for feature embedding, *concat* indicates concatenation operation and $\mathbf{U},\mathbf{V},\mathbf{S},\mathbf{W}$ are the projection matrices of the feature maps respectively. Different from [25], we first calculate bilinear features of the last conv layer and then fuse the bilinear feature maps with preceding feature maps for more discriminative feature representations. The overall multi-layer bilinear pooling module is shown in Figure 3.

3.3 Network Architecture and Loss Function

Network Architecture. Our work is built upon previous works in DNN-based pose estimation methods [3, 6, 11, 12, 13, 19, 24]. ResNet-34 pretrained on *Places* dataset is adopted as feature extractor backbone [3]. We replace the global average pooling layer after the last conv layer in other camera pose regression network by our proposed multi-layer factorized bilinear pooling module, with feature maps from the last three ResBlocks as the input of the module. In the module we set the hyperparameter $d = 8192$ and $n = 2048$. This module produces a 2048-dimensional feature vector, followed by *ReLU* and dropout with rate $p = 0.2$. A final fully connected layer is followed that outputs a 6DoF camera pose.

Loss Function and Parameterization. The camera pose $\mathbf{p} = [\mathbf{t},\mathbf{q}]$ is represented by the position $\mathbf{t} \in \mathbb{R}^3$ and a quaternion $\mathbf{q} \in \mathbb{R}^4$ for the orientation. We use the same loss function as that in PoseNet [12]:

$$L_i = ||\mathbf{t}_i - \hat{\mathbf{t}}_i||_\gamma + \beta||\mathbf{q}_i - \frac{\hat{\mathbf{q}}_i}{||\hat{\mathbf{q}}_i||}||_\gamma \tag{8}$$

where γ is a distance norm and we use $\gamma = 1$ in this paper, $[\mathbf{t},\mathbf{q}]$ and $[\hat{\mathbf{t}},\hat{\mathbf{q}}]$ are ground truth and estimated positions and orientations, respectively. Since a quaternion $\mathbf{q}$ is identical to $-\mathbf{q}$, we constrain all quaternions to one hemisphere to make each rotation be a unique value. The parameter β is the scale factor that balances the position and orientation losses. We tune the scale factor β to optimally learn both position and orientation simultaneously, and set $\beta = 500$ for outdoor scenes and $\beta = 10$ for indoor scenes.

4 Experiments and Results

In this section, we present the results of our method on two well-known public datasets, prove its efficacy and give visualized analysis to show that our bilinear model can extract more discriminative features to improve camera localization accuracy.

Dataset. We evaluate our model on two well-known public datasets — *Cambridge Landmarks* [13] for large-scale outdoor scenes and *7 Scenes* [10] for small-scale indoor scenes. *Cambridge Landmarks* is an outdoor dataset which was collected using a smart phone and provides labeled video data to train and test pose regression algorithms. *7 Scenes* is an indoor dataset which contains RGB-D image sequences of seven indoor environments captured with a Kinect sensor. We follow the same training/test split of these two dataset as in PoseNet [13].

Implementation Details. We use ResNet34 as the network backbone which is initialized with the pretrained weight of Places dataset [28]. We implement our algorithm with PyTorch, using the SGD optimizer with learning rate $5e-4$ and a weight decay of $5e-4$, and employ the *Plateau* learning rate policy to reduce the learning rate. All experiments are performed on an 11GB NVIDIA RTX 2080Ti with batch size 64 for *Cambridge Landmarks* and batch size 16 for *7 Scenes*. For *Cambridge Landmarks* dataset, the input images are rescaled to 256 × 256 pixels before cropping to the 224 × 224 pixels and normalized with the mean and standard deviation computed from the ImageNet dataset. For *7 Scenes* dataset, only rescaling and cropping are implemented. Then we use the pixel mean and deviation same as in MapNet [3] to normalize the cropped input images. For both datasets, we use random crops during training and central crops during testing.

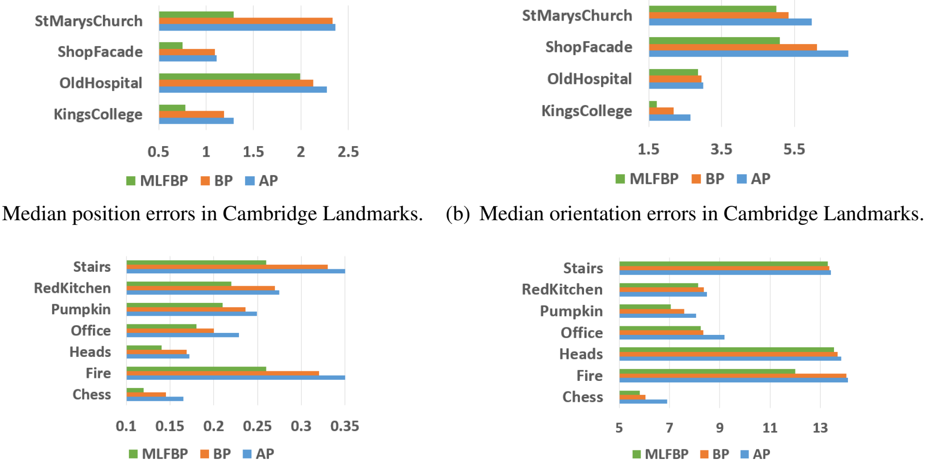
Comparison with Previous Methods. We compare our method with six state-of-the-art CNN-based approaches: PoseNet [13], Bayesian PoseNet [11], PoseLSTM [24], PoseNet (learn weight) [12], PoseNet (Geometric Reprojection) and MapNet [3]. MapNet only published the results on *7 Scenes*, so we only compare it on the same dataset. We compare the median localization errors in different scenes with the previous methods as shown in Table 1. On *Cambridge Landmarks* dataset, the position regression results of our model outperform all the methods except PoseLSTM in "Old Hospital" scene. On *7 Scenes* dataset, the position results of our model outperform all the methods expect MapNet in "chess" and "office" scene. Notice that PoseNet with Geometric Reprojection introduced depth information to form the geometric reprojection error and MapNet minimized both the loss of the per-image absolute pose and the loss of the relative pose between image pairs. Our methods only requires a single RGB image as input, without the need of depth information or image pairs to regress the camera pose. Nevertheless, our position results outperformed both MapNet and PoseNet with Geometric Reprojection. The orientation results of our model are not always at the first place, but are better than those from PoseNet and Bayesian PoseNet, and are comparable with other methods.

Ablation Study of Pooling Options. We provide an ablation study among different pooling options for feature aggregation in camera localization. Previous methods use global average pooling as the feature aggregator which computes the average values of each channel of the final feature map to generate a feature vector. To extract more informative and discriminative features, we propose to use bilinear pooling and multi-layer factorized bilinear

Scene	Area of Volume	Singe Image Input					With Scene models provided
		PoseNet	Beyasian PoseNet	PoseLSTM	PoseNet (learn Weight)	ours	PoseNet (Geometric Reprojection)
King's College	5600 m^2	1.66*m*, 4.86°	1.74*m*, 4.06°	0.99*m*, 3.65°	0.99*m*, **1.06**°	**0.76***m*, 1.72°	0.88*m*, **1.04**°
Old Hospital	2000 m^2	2.62*m*, 4.90°	2.57*m*, 5.14°	**1.51***m*, 4.29°	2.17*m*, 2.94°	1.99*m*, **2.85**°	3.20*m*, 3.29°
Shop Facade	875 m^2	1.41*m*, 7.18°	1.25*m*, 7.54°	1.18*m*, 7.44°	1.05*m*, **3.97**°	**0.75***m*, 5.10°	0.88*m*, **3.78**°
St Mary's Church	4800 m^2	2.45*m*, 7.96°	2.11*m*, 8.38°	1.52*m*, 6.68°	1.49*m*, **3.43**°	**1.29***m*, 5.01°	1.57*m*, **3.32**°

Scene	Area of Volume	Single Image Input					With Scene models provided	Image Pairs Input
		PoseNet	Beyasian PoseNet	PoseLSTM	PoseNet (learn Weight)	ours	PoseNet (Geometric Reprojection)	MapNet
Chess	6 m^2	0.32*m*, 6.6°	0.37*m*, 7.24°	0.24*m*, 5.77°	0.14*m*, **4.50**°	**0.12***m*, 5.82°	0.13*m*, 4.48°	**0.08***m*, **3.25**°
Fire	2.5 m^2	0.47*m*, 14.0°	0.43*m*, 13.7°	0.34*m*, 11.9°	0.27*m*, **11.8**°	**0.26***m*, 11.99°	0.27*m*, **11.3**°	0.27*m*, 11.69°
Heads	1 m^2	0.30*m*, 12.2°	0.31*m*, **12.0**°	0.21*m*, 13.7°	0.18*m*, 12.1°	**0.14***m*, 13.54°	0.17*m*, 13.0°	0.18*m*, 13.25°
Office	7.5 m^2	0.48*m*, 7.24°	0.48*m*, 8.04°	0.30*m*, 8.08°	0.20*m*, **5.77**°	**0.18***m*, 8.24°	0.19*m*, **5.55**°	0.17*m*, 5.15°
Pumpkin	5 m^2	0.49*m*, 8.12°	0.61*m*, 7.08°	0.33*m*, 7.00°	0.25*m*, **4.82**°	**0.21***m*, 7.05°	0.26*m*, **4.75**°	0.22*m*, 4.02°
Red Kitchen	18 m^2	0.58*m*, 8.34°	0.58*m*, 8.34°	0.37*m*, 8.83°	0.24*m*, **5.52**°	**0.22***m*, 8.14°	0.23*m*, **5.35**°	0.23*m*, 4.93°
Stairs	7.5 m^2	0.48*m*, 13.1°	0.48*m*, 13.1°	0.40*m*, 13.7°	0.37*m*, **10.6**°	**0.26***m*, 13.55°	0.35*m*, 12.4°	0.30*m*, 12.08°

Table 1: Median localization results for *Cambridge Landmarks* and *7 Scenes* datasets. The best results, as well as the best results of methods with a single image as input, are shown in bold.



(a) Median position errors in Cambridge Landmarks. (b) Median orientation errors in Cambridge Landmarks.

(c) Median position errors in 7Scenes. (d) Median orientation errors in 7Scenes.

Figure 4: Performance comparison of average pooling (AP), bilinear pooling (BP) and multi-layer factorized bilinear pooling (MLFBP).

pooling. In Figure 4, it is obvious that bilinear pooling performs better than commonly used average pooling, while multi-layer factorized bilinear pooling achieves the best performance.

Visualization Analysis. We use saliency maps to demonstrate that our proposed pooling module can extract more informative features to improve localization accuracy and use activation map to further demonstrate why our multi-layer bilinear pooling can extract meaningful and discriminative features. Here saliency map is the magnitude of the gradient of the mean of the network output [3] and the value of each position is the importance of each pixel of the input image, so it reveals the informative parts. And the activation map is the magnitude of feature activations across channels and it can reflect the effects of the bilinear model and multi-layer fusion on feature aggregation.

We visualize the saliency maps of PoseNet, PoseLSTM ands our model in *Cambridge Landmark*. Limited by the paper space, we sample a typical example as shown in Figure 5. PoseNet uses average pooling for feature aggregation traditionally and PoseLSTM uses LSTM module for feature correlation. Compared with them, our proposed model fo-

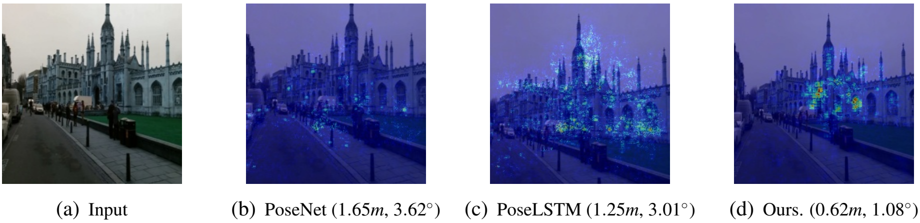

(a) Input (b) PoseNet (1.65m, 3.62°) (c) PoseLSTM (1.25m, 3.01°) (d) Ours. (0.62m, 1.08°)

Figure 5: Comparison of the saliency map and the errors of different models. Notice that strong responses of our proposed model lie in geometrically meaningful regions but others in uninformative sky or roads, so that our model produces higher accuracy. Localization error of each image is shown inside the brackets.

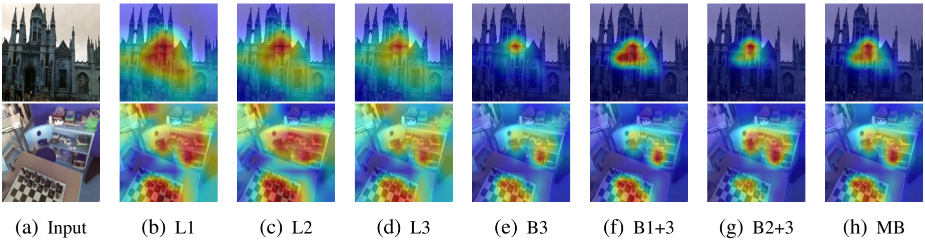

(a) Input (b) L1 (c) L2 (d) L3 (e) B3 (f) B1+3 (g) B2+3 (h) MB

Figure 6: Visualization of activation maps of different layers on sample images from the Cambridge Landmarks and 7 Scenes datasets. Li: original features from Layer4_i. Bi: bilinear features from Layer4_i. Bi+j: features from bilinear fusion of Layer4_i and Layer4_j. MB: multi-layer bilinear features. Note that bilinear features focus on essential parts of objects and multi-layer bilinear features cover larger discriminative regions.

calizes main parts of the building, but relatively strong responses of PoseNet erroneously lie in roads and that of PoseLSTM is located in parts of sky and roads. This phenomenon indicates our modal extracts more informative features than PoseNet and PoseLSTM. From the localization results, our method performs better than PoseLSTM, the improved variant of PoseNet. It suggests that extracting informative features is important to improve localization accuracy.

Compared with the original ResNet outputs in Figure 6(b)(c)(d), the bilinear feature maps in Figure 6(e)(f)(g) have more accurate and stronger activations at highly specific semantic parts, such as the cabinet and the chessboard, and show reduced feature activations in the background. This suggests that the original network output only provides rough localization of the important objects, while the bilinear model further draws attention to more essential parts of the objects. Among bilinear feature activations in Figure 6(e)(f)(g), there are also diversity of strongly activated regions, while all of them are more concentrated than the original features. This indicates that different levels of features can serve as complements to the bilinear feature of the last layer. Thus, combining multi-layer features can provide more discriminative features than only using the bilinear features from the last conv layer. In conclusion, our proposed multi-layer bilinear model not only focus on the essential parts for camera localization but also finds more discriminative features from multiple layers.

5 Conclusion

We present a novel approach for camera localization problem from the perspective of feature aggregation. To make the features more informative and discriminative, we propose a multi-layer factorized bilinear pooling module for feature selection and fusion. Bilinear pooling method is employed to select features that lie in geometrically meaningful regions, and multi-layer feature fusion helps the network to focus on the discriminative features that account for precise locations. Through the experiments on outdoor *Cambridge Landmarks* dataset and indoor *7 Scenes* dataset, we show that our method improves the performance of PoseNet and its variants using only a single image as the input for position estimation.

Acknowledgement

This work was supported by the National Natural Science Foundation of China project no. 61772057 and the support funding from Jiangxi and Qingdao Research Institute of Beihang University.

References

[1] Xiao Bai, Haichuan Yang, Jun Zhou, Peng Ren, and Jian Cheng. Data-dependent hashing based on p-stable distribution. *IEEE Transactions on Image Processing*, 23 (12):5033–5046, 2014.

[2] Xiao Bai, Cheng Yan, Haichuan Yang, Lu Bai, Jun Zhou, and Edwin Robert Hancock. Adaptive hash retrieval with kernel based similarity. *Pattern Recognition*, 75:136–148, 2018.

[3] Samarth Brahmbhatt, Jinwei Gu, Kihwan Kim, James Hays, and Jan Kautz. Geometry-aware learning of maps for camera localization. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2616–2625, 2018.

[4] Ming Cai, Chunhua Shen, and Ian Reid. A hybrid probabilistic model for camera relocalization. In *British Machine Vision Conference*, 2018.

[5] Sijia Cai, Wangmeng Zuo, and Lei Zhang. Higher-order integration of hierarchical convolutional activations for fine-grained visual categorization. In *IEEE International Conference on Computer Vision*, pages 511–520, 2017.

[6] Ronald Clark, Sen Wang, Andrew Markham, Niki Trigoni, and Hongkai Wen. VidLoc: A deep spatio-temporal model for 6-DoF video-clip relocalization. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2652–2660, 2017.

[7] Yin Cui, Feng Zhou, Jiang Wang, Xiao Liu, Yuanqing Lin, and Serge Belongie. Kernel pooling for convolutional neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3049–3058, 2017.

[8] Akira Fukui, Dong Huk Park, Daylen Yang, Anna Rohrbach, Trevor Darrell, and Marcus Rohrbach. Multimodal compact bilinear pooling for visual question answering and visual grounding. In *Conference on Empirical Methods in Natural Language Processing*, 2016.

[9] Yang Gao, Oscar Beijbom, Ning Zhang, and Trevor Darrell. Compact bilinear pooling. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 317–326, 2016.

[10] Ben Glocker, Shahram Izadi, Jamie Shotton, and Antonio Criminisi. Real-time RGB-D camera relocalization. pages 173–179, 2013.

[11] Alex Kendall and Roberto Cipolla. Modelling uncertainty in deep learning for camera relocalization. In *IEEE International Conference on Robotics and Automation*, pages 4762–4769, 2016.

[12] Alex Kendall and Roberto Cipolla. Geometric loss functions for camera pose regression with deep learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 6555–6564, 2017.

[13] Alex Kendall, Matthew Grimes, and Roberto Cipolla. Posenet: A convolutional network for real-time 6-DOF camera relocalization. In *IEEE International Conference on Computer Vision*, pages 2938–2946, 2015.

[14] Jin-Hwa Kim, Kyoung-Woon On, Woosang Lim, Jeonghee Kim, Jung-Woo Ha, and Byoung-Tak Zhang. Hadamard product for low-rank bilinear pooling. In *International Conference on Learning Representations*, 2017.

[15] Shu Kong and Charless Fowlkes. Low-rank bilinear pooling for fine-grained classification. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 7025–7034, 2017.

[16] Yanghao Li, Naiyan Wang, Jiaying Liu, and Xiaodi Hou. Factorized bilinear models for image recognition. In *IEEE International Conference on Computer Vision*, pages 2098–2106, 2017.

[17] Tsung-Yu Lin and Subhransu Maji. Improved bilinear pooling with cnns. In *British Machine Vision Conference*, 2017.

[18] Tsung-Yu Lin, Aruni RoyChowdhury, and Subhransu Maji. Bilinear CNN models for fine-grained visual recognition. In *IEEE International Conference on Computer Vision*, pages 1449–1457, 2015.

[19] Iaroslav Melekhov, Juha Ylioinas, Juho Kannala, and Esa Rahtu. Image-based localization using Hourglass networks. In *IEEE International Conference on Computer Vision Workshops*, pages 870–877, 2017.

[20] Noha Radwan, Abhinav Valada, and Wolfram Burgard. VLocNet++: Deep multitask learning for semantic visual localization and odometry. *IEEE Robotics and Automation Letters*, 3(4):4407–4414, 2018.

[21] Torsten Sattler, Bastian Leibe, and Leif Kobbelt. Efficient & effective prioritized matching for large-scale image-based localization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(9):1744–1756, 2017.

[22] Marcel Simon and Erik Rodner. Neural activation constellations: Unsupervised part model discovery with convolutional networks. In *IEEE International Conference on Computer Vision*, pages 1143–1151, 2015.

[23] Abhinav Valada, Noha Radwan, and Wolfram Burgard. Deep auxiliary learning for visual localization and odometry. In *IEEE International Conference on Robotics and Automation*, pages 6939–6946, 2018.

[24] Florian Walch, Caner Hazirbas, Laura Leal-Taixe, Torsten Sattler, Sebastian Hilsenbeck, and Daniel Cremers. Image-based localization using LSTMs for structured feature correlation. In *IEEE International Conference on Computer Vision*, pages 627–637, 2017.

[25] Chaojian Yu, Xinyi Zhao, Qi Zheng, Peng Zhang, and Xinge You. Hierarchical bilinear pooling for fine-grained visual recognition. In *The European Conference on Computer Vision*, pages 595–610, 2018.

[26] Zhou Yu, Jun Yu, Jianping Fan, and Dacheng Tao. Multi-modal factorized bilinear pooling with co-attention learning for visual question answering. In *IEEE International Conference on Computer Vision*, pages 1839–1848, 2017.

[27] Xiaopeng Zhang, Hongkai Xiong, Wengang Zhou, Weiyao Lin, and Qi Tian. Picking deep filter responses for fine-grained image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1134–1142, 2016.

[28] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(6):1452–1464, 2018.