2021 IEEE International Conference on Robotics and Automation (ICRA 2021)
May 31 - June 4, 2021, Xi'an, China

SoftMP: Attentive feature pooling for joint local feature detection and description for place recognition in changing environments

Fangming Yuan, Peer Neubert, Stefan Schubert, and Peter Protzel

Abstract— **Visual place recognition is the task of finding matchings of images that show the same place in the world. Combinations of appearance changes (e.g. changing illumination or weather) and geometric changes (e.g. viewpoint changes or occlusions) challenge existing approaches. Learning-based local image feature pipelines are a promising approach to this type of problem. We present a novel attentive feature pooling method that can be used to train a CNN to jointly detect and describe local image features. It can be trained on small or moderately sized datasets with weak supervision in a classification training setup (e.g. we use a set of 24k images of publicly available web-camera images in our experiments). We propose to use a joint loss function that combines the cross-entropy loss for the classification task with a mean squared error in order to increase the repeatability of feature detections. We show how the approach can be integrated in a place recognition pipeline and run experiments on several standard place recognition datasets. Despite the small training dataset, we demonstrate a 15% improvement in the average performance compared to the best of a number of compared state-of-the-art approaches, and, probably more importantly, a 3x improvement in the worst-case performance. Open source code is available.**

I. INTRODUCTION

Visual place recognition (VPR) is an important capability of mobile robots. It can be the basis for robot localization and loop-closure detection in SLAM. Major challenges for VPR are image appearance changes (e.g., caused by object shadows, varying illumination, weather conditions, and seasonal changes) and viewpoint changes or occlusions (e.g. caused by dynamic objects). While holistic feature approaches (that use a single descriptor per image) can be robust towards appearance changes, they are prone to viewpoint changes and occlusions [1]. In contrast, using sets of local features (e.g. keypoints) can significantly increase the robustness towards image displacement, viewpoint changes, and occlusions. Further, the relative pose of local features can be used in graph embeddings to increase the performance of image place recognition and pose estimation [2] [3]. The development of (deep learning based) local feature pipelines is an active field of research where many approaches have been proposed. However, the repeated extraction of discriminative local features, in particular in changing environments, remains to be challenging for existing approaches.

Some local feature pipelines use salient activations in feature maps of a pre-trained CNN to localize and extract local features [4] [5]. However, the CNNs they use are not particularly designed and trained for local feature extraction. Most local feature pipelines follow the detect-then-describe policy that first localizes the keypoint location and then extracts the corresponding feature. They utilize either an external interesting point proposal algorithm [6] [7] [8] or use a dedicated neural network [9] [10] [11] [12]. For example, DELF [12] is trained with two steps of training to optimize the feature network and attention network sequentially. Whether and how the two steps of training can be combined and performed jointly remains open. In another direction, D2-Net [13] is a joint detection and description pipeline. However, since it uses large-scale SfM reconstructions for training, the requirements on the training dataset are rather high.

In this paper, we propose the local feature approach LocalSPED-SoftMP that combines a novel attentive feature pooling method based on softmax-pooling (SoftMP) with our previously proposed LocalSPED [14] pipeline for visual place recognition. LocalSPED-SoftMP takes inspiration from the joint detection and description approach of D2-Net but has lower demands on the training dataset. Our pipeline is implicitly trained by a place image classification task similar to DELF (e.g. using simple webcam images). However, in contrast to DELF, we merge the feature and attention network to a single network to achieve a one step detect-and-describe local feature extraction policy that simultaneously detect a local feature and extract its feature. We also train both the feature detection and extraction jointly in a single step of training. Sec. III will explain how this is achieved using the SoftMP feature pooling method together with a joint loss function that combines cross-entropy loss and mean squared error. The intuition behind additionally using a mean squared error term in the classification-like training is to reduce unnecessary activations in the generated attention map (in order to increase the repeatability of local features). In contrast to existing deep learning based local feature pipelines (e.g. [12] [13]), LocalSPED-SoftMP can be trained using small or moderately sized training sets. This opens potential for domain or environment specific training. For example, the experiments presented in Sec. IV use a dataset of 24,000 webcam images with illumination, weather, and seasonal appearance changes for training. These images are easy to acquire and only need week annotation. We evaluate the performance of LocalSPED-SoftMP on a series of standard place recognition datasets (Oxford RobotCar, Nordland, StLucia, Gardens Point Walking, CMU) and compare against the original LocalSPED and four state-of-the-art local

This work is funded by the German Federal Ministry for Economic Affairs and Energy (Bundesministerium für Wirtschaft und Energie) in the project ViPRICE. All authors are with Faculty of Electrical Engineering and Automation Technology, Chemnitz University of Technology, Chemnitz, Germany {firstname.lastname}@etit.tu-chemnitz.de

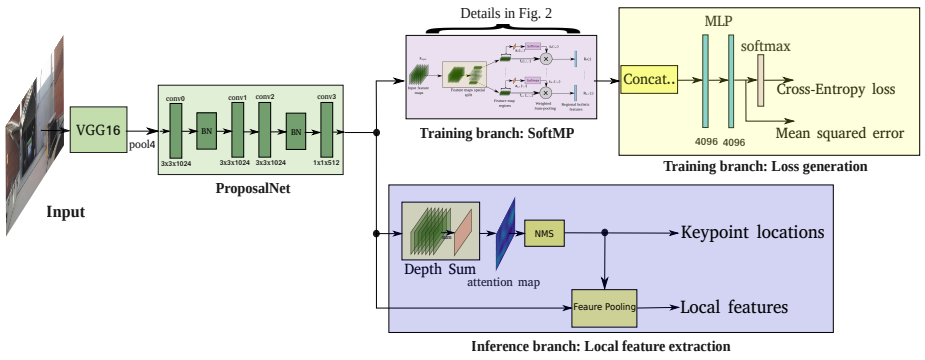

Fig. 1. Block diagram for LocalSPED-SoftMP.

and holistic feature approaches (DELF, D2-Net, NetVLAD, AlexNet). Despite the low training data demands, the proposed approach surpasses the state-of-the-art performance and significantly improves the worst-case performance across all datasets. The experiments will also demonstrate the effectiveness of the joint error function. The source code, additional information, and a pre-trained model can be found at `https://www.tu-chemnitz.de/etit/proaut/softMP`

II. RELATED WORK

This paper builds upon LocalSPED from our short paper [14], which in turn uses a training scheme similar to SPED [15]. LocalSPED is a deep learning based local feature extraction pipeline that can be trained with small amounts of data. Here, we integrate the proposed SoftMP attention mechanism in LocalSPED and also provide a more detailed description of the overall pipeline. The experiments in Sec. IV will show the significant benefit for image feature extraction.

Image features are important means for tasks like image retrieval [16] [12], visual place recognition [17] [18], and object detection [19] [20]. Early local feature and visual place recognition pipelines used the pixel intensity information for keypoint detection and extract handcrafted features [21] [22] [6] [23] [7] [17] [24]. In particular since 2012, convolutional neural networks (CNN) demonstrated promising abilities for image feature extraction. E.g., [1] use the output of early and intermediate layers of general purpose CNNs (pre-trained for image classification) to create holistic features for place recognition. Later, this type of descriptor has also been combined with hand-crafted local region detectors [8] [25]. The intermediate feature maps of CNNs can also be used to propose local feature locations [4] [5] [26]. In contrast to the here proposed method, these methods do not particularly train the CNN for the task of local feature extraction but rely on pre-trained networks. The attention mechanism in the field of computer vision [27] [28] [29] helps the CNN to recognize salient and distinctive features, which is also a promising way to extract local feature for place recognition. DELF [12] proposed an attentive local feature descriptor that learns salient features by adopting the network to an image classification task. The pipeline has a dedicated attention neural network placed after the dense feature extraction network to calculate attention scores for each of the dense features. Our work also incorporates the idea of attention, but combines the attention score proposal and feature extraction into one network.

The typically used approach for local feature extraction is a two-step detect-then-describe policy that first detects the locations of interesting point and then extracts the interesting point feature [6] [7] [8] [9] [10] [11]. D2-Net [13] is a local feature approach that combines the two-steps of local feature extraction to an one-step detect-and-describe policy. It uses the combined channel-wise and patch-wise softened activations in the CNN intermediate feature map to generate scores for each dense feature. The feature scores are integrated into a pair-wise triplet margin ranking loss [30] of the correspondences. Similar to D2-Net, we also propose a method that detect and describe the local feature simultaneously in a single network. The D2-Net was trained using a dataset with image patch-level labels, which are obtained from image depth information. The depth information is generated by multi-view stereo datasets that require the camera intrinsic and extrinsic parameters. Our pipeline is trained by classifying images captured from web-cameras to the class of their corresponding ground-truth camera which requires significantly less supervision.

III. ALGORITHM DETAILS

Fig. 1 provides an overview of the LocalSPED-SoftMP pipeline. We upgrade our previously proposed local feature pipeline LocalSPED by integrating the novel SoftMP layer into the training branch. Same as in LocalSPED, the VGG16 and ProposalNet are responsible for extracting dense image features. The subsequent processing differs between training and inference. During training, SoftMP is used to alter the weights in the ProposalNet in order to learn to extract salient and condition invariant features in an image classification training setup. Details of the SoftMP layer will be given in Sec. III-A. For inference, the training branch is discarded and the local features are extracted directly from the dense feature output of the ProposalNet by a simple channel-wise accumulation in an attention map (or indication map). Keypoint locations are obtained from local maxima in the attention

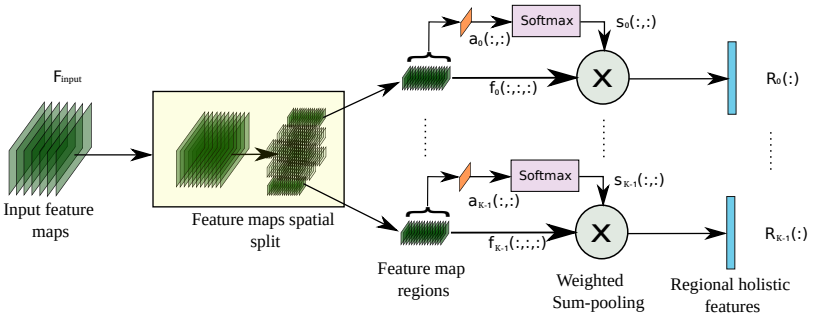

Fig. 2. Block diagram for the proposed SoftMP layer.

map (after non-maximum suppression). Simultaneously, the associated features (descriptors) are obtained by aggregating vectors from the same intermediate feature map that was used to generate the attention map. This will be explained more in detail in Sec. III-B.

A. SoftMP (Soft max-pool)

In our previous work LocalSPED [14], the attention map was not learned implicitly during the training. Instead, it was learned in an indirect way by max-pooling. This section provides details of the SoftMP layer that soften the max-pooling operation and implicitly embed the attention map into the training through attention score weighted sum-pooling to pool image-level features for image classification. The proposed method results in a joint optimization method for both feature detection and extraction.

1) Intuition: The purpose of attention score weighted sum-pooling is to force the ProposalNet to recognize and highlight the score for the key features, which are salient and conditional invariant. The intuition is visualized in Fig. 3. The arrows in the 2-D feature space represent the dense features extracted by the ProposalNet from several images of the same place under different image conditions. The green and red vectors represent the key features of these images before and after the training. The black vectors represent the other unimportant features, which show high variation under image displacements or changing environmental conditions. The green and red dots are the results of attention score weighted sum-pooling that is applied for each extracted features. Before the training, all the feature vectors have random scores, which result in a wide distribution of green dots in the green circle. This leads to inconsistent representations for these input images. During the training, the CNN tries to reduce the distribution range of the green dots by recognizing the key features and assign high activations for the key features, while minimize the activations of the non-key features. The result of the score weighted sum-pooling will gradually concentrate in the red circle and provides a more consistent representation of these input images. Fig. 4 shows attention maps and their corresponding input image in the test set after the training. It can be seen that despite the image displacement and conditional change, the highlighted key features in the attention map correspond to the same area in the images.

2) The SoftMP layer: Fig. 2 illustrates the structure of the SoftMP layer. Input is a CNN feature map F_{input}, a 3-D tensor of shape $H \times W \times C$. Where $H \times W$ denotes the spatial resolution of the feature map and C denotes the number of channels. In SoftMP, the input feature maps are spatially split into K rectangular regions in space $H \times W$. Each feature map region has dimension $h \times w \times C$. E.g., for $H = 3h$ and $W = 3w$, the feature map is split into $K = 9$ regions. Given a specific value $0 \leq k < K$, the 3-D tensor f_k is used to refer to the corresponding feature map region. For each of the feature map regions, a C-dimensional regional holistic

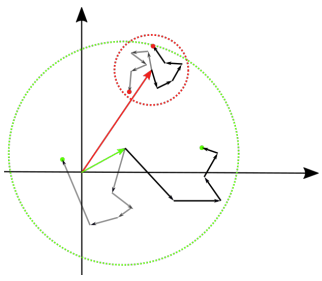

Fig. 3. The attention score weighted sum of the features in a feature map region. The goal is that regional holistic features that distributed in the green circle before the optimization will distribute in the red circle after the optimization, i.e. repeated observations will be mutually much more similar.

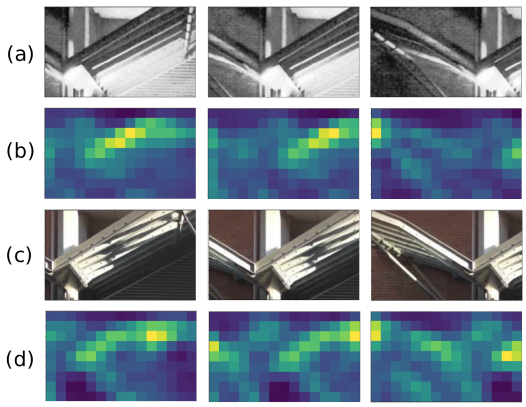

Fig. 4. Randomly shifted input image patches taken from the testing dataset Gardens Point Walking (rows a, c) and their associated attention maps (rows b, d) defined by equation 2 after optimization. In row a, the patches are taken from the night image. In row c, the patches are from the corresponding day image. The high activation in all attention map follows the image shifts.

TABLE I

NETWORK STRUCTURE OF THE PROPOSALNET. BN ARE BATCH NORMALIZATION LAYERS. THE a_i VALUE FOR LEAKYRELU IS 0.2.

layer name	input channel	output channel	kernel size	Leaky-ReLU	resolution	stride
BN					$\times 1$	
conv0	512	1024	3×3	✓	$\times 1$	1
conv1	1024	1024	3×3	✓	$\times 1$	1
conv2	1024	1024	3×3	✓	$\times 1$	1
BN					$\times 1$	
conv3	1024	512	1×1	✓	$\times 1$	1

feature R_k is calculated by the attention score weighted sum-pooling, which is described by equation 1:

$$R_k(c) = \sum_{y=0}^{h-1} \sum_{x=0}^{w-1} s_k(y,x) \cdot f_k(y,x,c) \tag{1}$$

x and y are used to address the C-dimensional feature vectors in the feature map region. The region indicator k indicates the feature map region, in which the operation performs. c is the feature map channel index. The feature attention scores $s_k(y,x)$ are obtained from the regularized channel-wise accumulated feature activations $a_k(y,x)$:

$$a_k(y,x) = \sum_{c=0}^{C-1} f_k(y,x,c) \tag{2}$$

Similar to other attention-based algorithms, $a_k(y,x)$ is regularized by the softmax function:

$$s_k(y,x) = \frac{e^{a_k(y,x)}}{\sum_{y=0}^{h-1} \sum_{x=0}^{w-1} e^{a_k(y,x)}} \tag{3}$$

The equations 1, 2, and 3 define the overall concept of SoftMP layer. The only hyper-parameter for the SoftmaxPool layer is the number of feature map regions. The output of the SoftMP layer is a set of regional holistic features $R(c)$. Each represents a feature map region.

B. LocalSPED-SoftMP: The SoftMP layer based joint detection and description local feature extractor

A structural overview of LocalSPED-SoftMP is given in Fig. 1. It is composed of five parts: VGG16 block, ProposalNet, local feature extraction block, SoftMP layer, and training loss block. The VGG16 [31] provides the dense backbone features for the entire pipeline from its $pool4$ output. Four layers of trainable convolutional blocks named $ProposalNet$ follow after the VGG16 $pool4$ output. The detailed network structure is provided in Table I. The purpose of $ProposalNet$ is to translate the backbone features to salient and conditional invariant features for both training branch and inference branch. The pipeline is trained by a classification task which classifies each scene image in the training dataset to its ground-truth class similar to [15]. The training branch consists of the SoftMP layer and two multi-layer perceptron (MLP) layers. The SoftMP layer splits the dense feature from $ProposalNet$ into several sub-regions of dense features as described in the previous section. Each generates a 512-dimensional regional holistic feature by the attention score weighted sum-pooling. All the regional holistic features are then concatenated to a long global holistic feature which is input for the successive MLP layer. The MLP layer outputs a N-dimensional vector, where N is the number of classes. This vector is further processed in two ways: First, it is directly compared with the N-dimensional ground-truth one-hot vector to generate a mean squared error term. Second, it is fed to a softmax layer to generate a cross-entropy loss. The weights of VGG16 are fixed. Only the $ProposalNet$ and MLPs are trained to minimize the joint loss function defined below:

$$Loss = L_{CrossEntropy} + \alpha \cdot L_{MeanSquared} \tag{4}$$

Where α is the weighting factor. $L_{CrossEntropy}$ and $L_{MeanSquared}$ are the cross-entropy loss and mean squared error loss calculated by the following two equations:

$$L_{CrossEntropy} = -\sum_{i=0}^{N-1} c_i \cdot log(softmax(v_i)) \tag{5}$$

$$L_{MeanSquared} = \sum_{i=0}^{N-1} (c_i - v_i)^2 \tag{6}$$

In the above equations, c_i represents the ground truth label and v_i represents the vector output by MLP.

For the local feature extraction block in the inference branch, the attention map M is calculated by equation 7 that channel-wisely accumulates the ProposalNet output feature maps.

$$M(h,w) = \sum_{c=0}^{C-1} F_{input}(h,w,c) \tag{7}$$

A 3×3 windowed non-maximum-suppression (NMS) operation is applied on the attention map to localize the keypoints. The feature for each keypoint is extracted by pooling a $d \times d$ patch around the keypoint location in F_{input}. Finally, the pooled patches are unrolled and compressed by PCA.

IV. EXPERIMENTAL RESULTS

This section characterizes the training dataset in Sec. IV-A, provides details of the model training in Sec. IV-B, and finally evaluates the proposed LocalSPED-SoftMP together with four state of the art pipelines in a series of place recognition experiments. The experiments also include an evaluation of the joint loss function.

A. Training Dataset

We train on the same dataset that was used for training LocalSPED [14]. It was created by capturing images from open-access web-cameras around the world from June to October 2019. We collected a set of 1000 cameras with good image quality and with small camera viewpoint displacement and rotation. To sample images with slow seasonal changes, the instantaneous images of each valid web-camera have been captured for one day every two weeks. For each capture-day, the images have been sampled continuously every three hours. In total, 24,000 images have been captured. All images

TABLE II

COMPARISON OF AREA UNDER CURVE (AUC) VALUES OF PRECISION-RECALL CURVES OF THE ORIGINAL LOCALSPED APPROACH, THE PROPOSED LOCALSPED-SOFTMP TRAINED BY MEAN SQUARED ERR(MSE), CROSS-ENTROPY AND JOINT-LOSS, THE LOCAL FEATURE PIPELINES DELF, AND D2-NET, AS WELL AS THE TWO HOLISTIC FEATURE PIPELINE NETVLAD AND ALEXNET. VALUES IN PARENTHESES ARE THE AVERAGE NUMBERS OF EXTRACTED KEYPOINTS (AVGKPNUM). FOR D2-NET, WE EXTRACTED 100 AND 200 LOCAL FEATURES PER IMAGE RESPECTIVELY (D2-NET-100 AND D2-NET-200). *GPW* IS THE GARDENS POINT WALKING DATASET.

Dataset	Variations	LocalSPED (AvgKp-Num)	LocalSPED-SoftMP (AvgKpNum) JointLoss [Proposed]	LocalSPED-SoftMP (AvgKpNum) CrossEntropy	DELF (AvgKpNum)	D2-Net-100	D2-Net-200	NetVLAD	AlexNet
Nordland	fall-spring	0.60 (33)	**0.86 (61)**	0.38(118)	0.85 (70)	0.48	0.72	0.39	0.82
	spring-winter	0.45 (34)	0.83 (61)	0.24(114)	**0.85 (71)**	0.45	0.69	0.11	0.59
	summer-spring	0.58 (33)	**0.79 (61)**	0.33(118)	0.73 (70)	0.44	0.63	0.32	0.77
	summer-fall	0.85 (32)	**0.94 (61)**	0.64(118)	0.92 (66)	0.80	0.88	0.63	**0.94**
	fall-winter	0.22 (34)	0.60 (60)	0.11(115)	0.57 (68)	0.28	0.43	0.06	**0.63**
StLucia	110909-1545 - 180809-1545	0.41 (27)	0.60 (47)	0.18(85)	0.44 (51)	0.55	**0.67**	0.27	0.60
	100909-0845 - 190809-0845	0.51 (26)	0.66 (46)	0.22(85)	0.49 (52)	0.61	**0.70**	0.41	0.59
	100909-1210 - 210809-1210	0.47 (27)	0.64 (46)	0.24(84)	0.45 (48)	0.67	**0.77**	0.51	0.55
	100909-1410 - 190809-1410	0.45 (28)	**0.67 (48)**	0.15(87)	0.43 (50)	0.50	0.64	0.38	0.61
	100909-1000 - 210809-1000	0.51 (27)	0.65 (46)	0.24(84)	0.48 (51)	0.66	**0.73**	0.47	0.57
Oxford	14-12-09 - 14-12-16	0.77 (79)	0.80 (141)	0.69(281)	**0.88 (152)**	0.19	0.51	0.85	0.45
	14-12-09 - 15-02-03	0.95 (79)	0.99 (149)	0.90(289)	**1.0 (157)**	0.22	0.59	0.98	0.80
	14-12-09 - 15-05-19	0.71 (79)	0.82 (143)	0.64(286)	**0.97 (156)**	0.62	0.77	0.90	0.27
	15-05-19 - 15-02-03	0.48 (77)	0.80 (149)	0.76(293)	**0.96 (149)**	0.36	0.85	0.84	0.51
CMU	20110421 - 20110202	0.58 (65)	**0.80 (117)**	0.33(234)	0.67 (104)	0.30	0.37	0.61	0.32
	20110421 - 20101221	0.48 (65)	**0.65 (117)**	0.19(237)	0.62 (113)	0.22	0.32	0.56	0.33
	20110421 - 20100915	0.72 (65)	**0.77 (119)**	0.60(233)	0.74 (127)	0.52	0.56	**0.77**	0.58
GPW	day-left - day-right	0.84 (44)	**0.95 (81)**	0.47(151)	0.84 (85)	0.80	0.87	0.97	0.58
	day-right - night-right	0.54 (45)	**0.72 (81)**	0.42(151)	0.17 (76)	0.17	0.30	0.51	0.51
	day-left - night-right	0.20 (45)	**0.58 (79)**	0.06(150)	0.07 (76)	0.1	0.14	0.40	0.10
Max	(best-case)	0.95	0.99	0.90	**1.0**	0.80	0.88	0.98	0.94
Min	(worst-case)	0.20	**0.58**	0.06	0.07	0.1	0.14	0.06	0.10
Mean	(average-case)	0.57	**0.76**	0.39	0.66	0.45	0.61	0.55	0.53

were resized to 480×640 pixels. The dataset scale is only 16 % of the dataset that was used to train DELF, which contains 175,754 images. And 4.5 % of the 308,887 pairs of annotated images used to train D2-Net.

B. Model training

LocalSPED-SoftMP is trained using the training procedure in Algorithm 1. The model is trained in an image classification task that classifies images into one of 1000 possible classes. Each class consists of all images from the same web-camera. At the beginning of the training, the model graph is built and all weights are randomly initialized (lines 2 and 3). For the VGG16 backbone block, pre-trained weights are loaded and fixed throughout the entire training process. The training is divided into 250 epochs (line 6). At the beginning of each epoch, images from 800 randomly selected cameras are chosen to create a training batch. The training batch is intensively augmented by image translation and rescaling (line 8). Therefore, each image is first scaled with a random scale factor ranging from 0.9 to 1.2. Then, for the image translation, the rescaled images are randomly clipped by a 432×432 dimensional rectangle. With this augmented image batch and its associated ground-truth label, 3000 iterations of gradient descent are performed to minimize the loss function defined in equation 4, with weighting parameter α set to 10. The LocalSPED-SoftMP is implemented in $Tensorflow$. The Adam [32] algorithm is used to minimize the loss function with a learning rate of 0.00001. For the two MLP layers, a drop-out layer is inserted with a drop-out rate of 0.5. No weight decay nor Momentum are used.

Algorithm 1 LocalSPED-SoftMP training procedure

```
1:  procedure TRAINING(dataset)
2:      Graph=BuildGraph()
3:      Init(Graph)
4:      LoadVGG16Weights()
5:      SceneNum = 800
6:      for i = 1 : 250 do
7:          ImgBatch=RandSelect(dataset, SceneNum)
8:          ImgBatch=Transformation(ImgBatch)
9:          for j = 1 : 3000 do
10:             MinimizeLoss(Graph, ImgBatch)
11:     return Graph.weights
```

C. Experimental setup for place recognition

Place recognition is the task of finding matchings between sets of database and query images. We follow [8] for calculating similarities between pairs of query and database images based on their local features. Given a set of local features D_{DB} for a database image and a set D_Q for a query image, we compute the set of all reciprocal matching pairs $M = \{(i \in D_{DB}, j \in D_Q) : i \text{ and } j \text{ are mutually nearest neighbors}\}$. The overall image similarity is computed by:

$$S_{DB,Q} = \frac{1}{\sqrt{|D_{DB}| \cdot |D_Q|}} \sum_{(i,j) \in M} sim(D^i_{DB}, D^j_Q) \quad (8)$$

The function $sim()$ computes the similarity of two descriptors; we use cosine similarity.

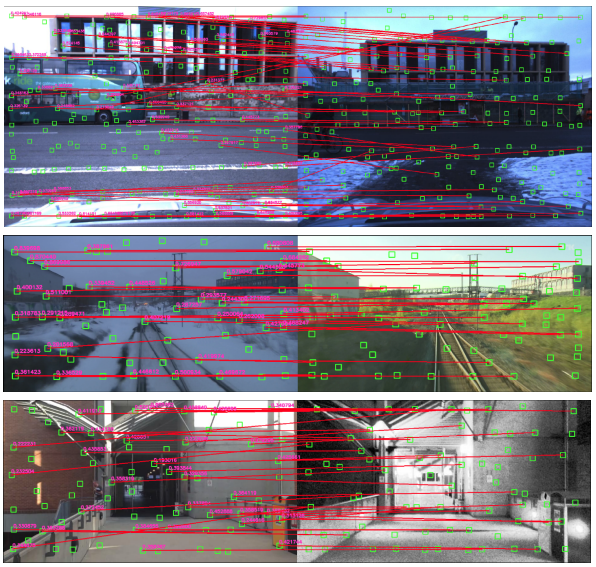

Fig. 5. Local feature matching examples for joint loss trained LocalSPED-SoftMP with different scene appearances and viewpoint changes.

To evaluate the place recognition performance on a dataset in our experiments, we exhaustively compute the pairwise similarities between all database and query images using equation 8. We follow the procedure from [25] to compare the similarities to ground-truth information about place matchings and to create precision-recall curves. Finally, we report the AUC (area under curve) value of the precision-recall curve for evaluation. We use the same datasets as [18]: Nordland, StLucia, Oxford RobotCar, CMU and Gardens Point Walking (GPW). Only for Oxford, we sampled each 160th image and cropped the images to remove the hood of the car. We demonstrate the pair-wise matching of local features extracted by the proposed pipeline in testing dataset images in figure 5.

D. Performance evaluation

Table II provides experimental results of the proposed LocalSPED-SoftMP in comparison to the original LocalSPED [14], DELF [12], D2-Net [13], as well as the two holistic descriptors NetVLAD [33] and AlexNet-conv3 [34]. We trained LocalSPED-SoftMP with MSE loss, the cross-entropy loss and the joint-loss respectively. For a fair comparison, the average number of features of the local feature approaches should be similar. For LocalSPED-SoftMP, the maximum number of features for each dataset is extracted. Each local feature is extracted from the 7×7 patch in the feature map and compressed to 1024 dimensions by PCA. For DELF, the feature score threshold is set to 110 to extract a comparable number of features, each with a 1024-dimensional descriptor. For D2-Net, 100, and 200 features for each image are extracted respectively. Each feature provides a 512-dimensional descriptor.

The results in Table II show a significant improvement over the original LocalSPED.

The comparison to the other evaluated algorithms shows that despite its rather small training dataset, the proposed

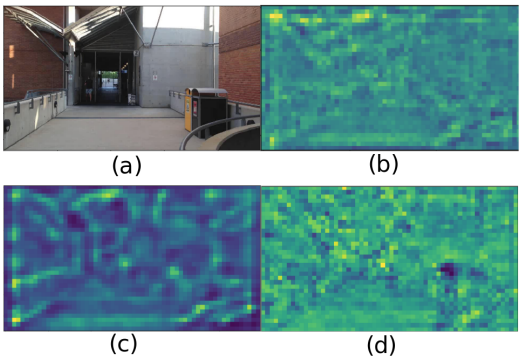

Fig. 6. The attention map extracted from image (a) by LocalSPED-SoftMP that was trained by (c) the joint loss function or (d) cross-entropy loss and (b) the mean squared error. In the attention map, the warmer the color, the higher the attention score value.

LocalSPED-SoftMP can considerably increase the average performance compared to the state of the art. For many datasets, it provides the best results. A notable exception are some of the Oxford sequences, where DELF performs better by some margin. However, a very important property of the proposed approach is the significantly better worst-case performance compared to all other approaches. In particular the night sequences of Gardens Point Walking are challenging for all other approaches but significantly better handled by LocalSPED-SoftMP.

E. Comparison of the three loss functions

Finally, we want to evaluate the joint loss function. Table II shows the performance of LocalSPED-SoftMP trained with the joint loss function and when only using the standard cross-entropy loss function. We have not been able to achieve considerable place recognition performance with a pipeline that only uses mean squared error in the loss function. When trained with the joint loss function, LocalSPED-SoftMP extracts less local features but performs better than when trained only with cross-entropy loss or only mean squared error. We speculate that for joint loss function, the mean squared error term condensed the network, which provides a cleaner and more sparse feature attention map, while the cross-entropy loss keeps the distinctive features still active. The sparse and clean attention map reduces noise and increases the repeatability of feature detection. Example feature attention maps extracted by LocalSPED-SoftMP with the three different loss functions are shown in Fig. 6.

V. CONCLUSIONS

The paper presented a novel attentive feature pooling method for joint local feature detection and description. Key components are an attention score weighted sum pooling and a joint error function for training. We showed how the approach can be embedded in a place recognition pipeline, that it can be trained using a small to moderately sized dataset, and demonstrated its performance on a variety of standard datasets. A moderate improvement in average performance compared to state-of-the-art methods and in particular a significant improvement in the worst-case performance make the algorithm promising for practical application.

REFERENCES

[1] N. Sünderhauf, F. Dayoub, S. Shirazi, B. Upcroft, and M. Milford, "On the performance of convnet features for place recognition," *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4297–4304, 2015.

[2] P.-E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich, "SuperGlue: Learning feature matching with graph neural networks," in *CVPR*, 2020.

[3] Z. Luo, T. Shen, L. Zhou, J. Zhang, Y. Yao, S. Li, T. Fang, and L. Quan, "Contextdesc: Local descriptor augmentation with cross-modality context," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2527–2536.

[4] Z. Chen *et al.*, "Only look once, mining distinctive landmarks from convnet for visual place recognition," in *International Conference on Intelligent Robots and Systems (IROS)*, 09 2017, pp. 9–16.

[5] A. Khaliq, S. Ehsan, Z. Chen, M. Milford, and K. McDonald-Maier, "A holistic visual place recognition approach using lightweight cnns for significant viewpoint and appearance changes," *IEEE Transactions on Robotics*, vol. 36, no. 2, pp. 561–569, 2020.

[6] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[7] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool, "Speeded-up robust features (surf)," *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346 – 359, 2008.

[8] N. Sünderhauf *et al.*, "Place recognition with convnet landmarks: Viewpoint-robust, condition-robust, training-free," in *Proc. of Robotics: Science and Systems*, 2015.

[9] D. DeTone, T. Malisiewicz, and A. Rabinovich, "Superpoint: Self-supervised interest point detection and description," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 337–33 712, 2018.

[10] J. Revaud, P. Weinzaepfel, C. D. Souza, N. Pion, G. Csurka, Y. Cabon, and M. Humenberger, "R2d2: Repeatable and reliable detector and descriptor," *ArXiv*, vol. abs/1906.06195, 2019.

[11] K. M. Yi, E. Trulls, V. Lepetit, and P. Fua, "Lift: Learned invariant feature transform." *CoRR*, vol. abs/1603.09114, 2016.

[12] H. Noh *et al.*, "Large-scale image retrieval with attentive deep local features," in *Proc. of International Conference on Computer Vision*, 2017.

[13] M. Dusmanu, I. Rocco, T. Pajdla, M. Pollefeys, J. Sivic, A. Torii, and T. Sattler, "D2-net: A trainable cnn for joint description and detection of local features," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 8084–8093.

[14] F. Yuan, P. Neubert, and P. Protzel, "Localsped: A classification pipeline that can learn local features for place recognition using a small training set," in *Proc. of Towards Autonomous Robotic Systems Conference (TAROS)*, 2020.

[15] Z. Chen *et al.*, "Deep learning features at scale for visual place recognition," in *Proc. of International Conference on Robotics and Automation*, 2017.

[16] B. Kong, J. Supancic, D. Ramanan, and C. C. Fowlkes, "Cross-domain image matching with deep feature maps," *International Journal of Computer Vision*, vol. 127, no. 11-12, pp. 1738–1750, 2019.

[17] M. Cummins and P. Newman, "Highly Scalable Appearance-Only SLAM – FAB-MAP 2.0," in *Robotics Science and Systems*, 2009.

[18] S. Schubert, P. Neubert, and P. Protzel, "Unsupervised learning methods for visual place recognition in discretely and continuously changing environments," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 4372–4378.

[19] R. Girshick, "Fast r-cnn," in *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 1440–1448.

[20] L. Cuimei, Q. Zhiliang, J. Nan, and W. Jianhua, "Human face detection algorithm via haar cascade classifier combined with three additional classifiers," in *2017 13th IEEE International Conference on Electronic Measurement Instruments (ICEMI)*, 2017, pp. 483–487.

[21] C. G. Harris, M. Stephens, *et al.*, "A combined corner and edge detector." in *Alvey vision conference*, vol. 15, no. 50. Citeseer, 1988, pp. 10–5244.

[22] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf," in *2011 International Conference on Computer Vision*, 2011, pp. 2564–2571.

[23] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *European conference on computer vision*. Springer, 2006, pp. 430–443.

[24] R. Paul and P. Newman, "Fab-map 3d: Topological mapping with spatial and visual appearance," in *2010 IEEE International Conference on Robotics and Automation*, 2010, pp. 2649–2656.

[25] P. Neubert, "Superpixels and their Application for Visual Place Recognition in Changing Environments." PhD Thesis, Chemnitz University of Technology, 2015. [Online]. Available: http://nbn-resolving.de/urn:nbn:de:bsz:ch1-qucosa-190241

[26] L. Liu, C. Shen, and A. v. d. Hengel, "Cross-convolutional-layer pooling for image recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 11, pp. 2305–2313, 2017.

[27] S. Woo, J. Park, J.-Y. Lee, and I.-S. Kweon, "Cbam: Convolutional block attention module," in *ECCV*, 2018.

[28] X. Wang, Z. Cai, D. Gao, and N. Vasconcelos, "Towards universal object detection by domain attention," *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7281–7290, 2019.

[29] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7132–7141.

[30] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 815–823.

[31] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. of Intern. Conf. on Learning Representations*, 2015.

[32] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[33] R. Arandjelovi, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "Netvlad: Cnn architecture for weakly supervised place recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 6, pp. 1437–1451, 2018.

[34] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 1097–1105.