Delving Deeper into Convolutional Neural Networks for Camera Relocalization

Jian Wu[1], Liwei Ma[2] and Xiaolin Hu[1]

***Abstract*— Convolutional Neural Networks (CNNs) have been applied to camera relocalization, which is to infer the pose of the camera given a single monocular image. However, there are still many open problems for camera relocalization with CNNs. We delve into the CNNs for camera relocalization. First, a variant of Euler angles named Euler6 is proposed to represent orientation. Then a data augmentation method named pose synthesis is designed to reduce sparsity of poses in the whole pose space to cope with overfitting in training. Third, a multi-task CNN named BranchNet is proposed to deal with the complex coupling of orientation and translation. The network consists of several shared convolutional layers and splits into two branches which predict orientation and translation, respectively. Experiments on the 7Scenes dataset show that incorporating these techniques one by one into an existing model PoseNet always leads to better results. Together these techniques reduce the orientation error by 15.9% and the translation error by 38.3% compared to the state-of-the-art model Bayesian PoseNet. We implement BranchNet on an Intel NUC mobile platform and reach a speed of 43 fps, which meets the real-time requirement of many robotic applications.**

I. Introduction

The problem of camera relocalization is to infer the orientation and translation of a camera given only a single picture, which is often encountered in many robotic applications, such as navigation and Simultaneously Localization and Mapping (SLAM). In SLAM, if the tracking is lost, global relocalization is started to initialize camera's pose estimation. In the past several decades, many approaches are developed for solving this problem [1], [2], [3], [4] [5], [6]. Recently, convolutional neural networks (CNNs) have been used to perform camera relocalization because of their robustness to real-life scenarios and scalability to training data size. A CNN-based relocalization framework named PoseNet is introduced to regress camera poses [7]. A Bayesian PoseNet is trained end-to-end with dropout and obtains relocalization uncertainty by averaging Monte Carlo dropout samples from posterior distribution of the Bayesian CNN's weights [8]. The two models have exhibited good performance on both indoor and outdoor relocalization datasets.

However, there are still many open problems for camera relocalization with CNNs. First, there are many orientation representations that can be utilized by CNNs for pose regression, such as Euler angles, quaternion and rotation matrix.

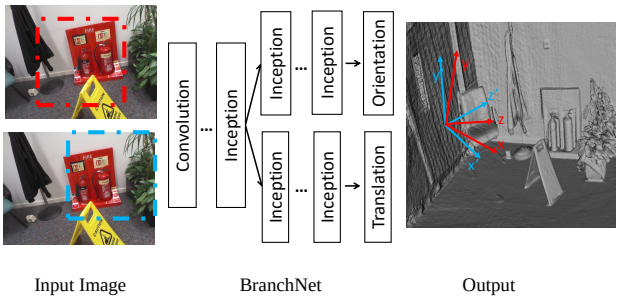


Fig. 1. Multi-task CNN for camera relocalization. Given an input image, the CNN predicts the 6-DOF of the camera. A new orientation representation Euler6 is employed. Both data and label are augmented by a method named pose synthesis, which gives different label to different patches extracted from the input image.

But which representation is most suitable for this problem is unknown. PoseNet employs quaternion as orientation representation but quaternion faces the problem that it represents the same orientation operation with its additive inverse. So similar images may be assigned very different quaternions in pose regression.

Second, due to the high cost for collecting data, camera poses in training set are always very sparse in the whole pose space. The camera poses in training set always belong to several trajectories used to capture key frames. So only a sparse sampling of the whole pose space is given. Because of the sparsity of sampled poses, the effective range of camera relocalization is limited to the nearby regions to the training trajectories. PoseNet augments data by random cropping to cope with overfitting but is unable to reduce the sparsity of pose because it does not change the distribution of poses in training set.

Third, the relationship between orientation and translation is complex which entails specific network architecture to achieve high accuracy. The orientation and translation are usually treated as a whole and are optimized together with a single network in PoseNet. Regressing them together may not be the optimal strategy.

To tackle the problems stated above, we present three techniques for CNN-based camera relocalization methods. First, we propose a variant of Euler angles named Euler6 to represent orientation. Second, we propose a new data augmentation method named pose synthesis to augment data and label simultaneously, which gives different poses to different image patches and effectively reduce sparsity of sampled

[1] Jian Wu and Xiaolin Hu are with the Tsinghua National Laboratory for Information Science and Technology (TNList), Department of Computer Science and Technology, Tsinghua University, 100084, Beijing, China wuj16@mails.tsinghua.edu.cn, xlhu@tsinghua.edu.cn

[2] Liwei Ma is with the Intel Labs China, Intel Corporation, 100090, Beijing, China liwei.ma@intel.com

poses compared to ordinary random cropping. Third, we present a generic multi-task network named BranchNet to match the relationship between the orientation and translation. Experimental results showed that all of these techniques could improve the performance of PoseNet.

II. Related Work

In this section, we first review camera relocalization methods and then multi-task CNNs because we will design a multi-task CNN for camera relocalization.

A. Camera Relocalization Method

There are mainly two kinds of approaches for vision-based camera relocalization: keypoints-based approaches and keyframes-based approaches.

The keypoints-based approaches detect interest points in the image, extract their local features and match them against a database of features. Local features such as SIFT [9] and ORB [10] are exploited to register points. A 3D geometric test is employed to retrieve a set of 2D-3D points and rule out false matches [1]. SCoRe Forests use random forests to regress scene coordinate labels to relocalization [2]. A hybrid discriminative-generative learning architecture uses a set of multiple predictors to reduce relocalization error [3]. SCoRe Forests approach is extended with a probabilistic approach which exploits uncertainty from regression forests for pose estimation [4].

The keyframes-based approaches get the camera pose by computing image similarity scores between a query image and a set of key frames with known key poses. The final camera pose is computed as a weighted average of the key poses or set to the pose of the keyframe with the highest image similarity score [5]. The synthetic RGB-D views are used as key frames in [6]. However, these methods only provide a coarse estimation to the camera pose because of the sparsity of poses in training set.

The past few years have witnessed the success of convolutional neural networks on a wide variety of computer vision tasks, such as classification, object detection and image parsing. The hierarchical structure has been shown good at extracting high level feature representations and robust to many real-life scenarios. PoseNet [8] leverages high level feature representations of CNN to regress camera pose. It can be considered as a keyframes-based approach which encodes the key frames in training set into the parameters of models. PoseNet is extended to a Bayesian model trained end-to-end with dropout [8]. At inference, averaging Monte Carlo dropout samples from posterior distribution of Bayesian CNN's weights significantly improves relocalization accuracy. SE3-Net regresses rigid body motion of moving objects from raw point cloud data and action. However, using point cloud data limits this algorithm to RGB-D data and the number of predicted objects must be specified in training [11].

B. Multi-task CNN

Multi-task learning is a way of utilizing shared information to solve multiple problems at the same time [12]. Multi-task CNNs have been applied to many tasks. TCDCN is proposed to jointly optimize facial landmark detection with a set of related tasks such as appearance attribute and expression [13]. HyperFace employs a separate CNN followed by a multi-task learning algorithm for simultaneously detecting faces, localizing landmarks, estimating head pose and identifying gender [14]. An R-CNN detector with multiple loss functions is trained for the tasks of human pose estimation and action detection [15]. Attributes and object classes are learned jointly to improve overall classification performance [16]. MCNNs take advantage of attribute relationships to improve accuracy of attribute classifiers [17]. In these network architectures, the lower layers are shared to extract low level common knowledges and higher layers are separated for related aims to generate specific predictions.

In fact, these multi-task CNN architectures can be considered as hierarchical network architectures with only one splitting node. A general hierarchical network has a tree structure which has multiple splitting nodes. A vast literature has explored hierarchical network structures for classification and achieved significant improvement on accuracy. HD-CNNs are presented which consists of both coarse component trained over all classes and fine components trained over subsets of classes [18]. Network of experts replaces expensive training of the base CNN model over all classes with learning a generalist that discriminates a much smaller number of specialities [19].

III. Methods

Our task is to infer a camera's 6-DOF pose consisting of orientation R and translation t where $R \in \mathbf{SO(3)}$ and $t \in \mathbb{R}^3$. The outputs are two vectors respectively representing orientation and translation. In this section, we present the novel orientation representation Euler6, pose synthesis and the BranchNet to predict camera pose.

A. Orientation Representation

Several orientation representations can be used to describe the orientation of a rigid body, such as Euler angles, quaternion and rotation matrix. Quaternion is employed by PoseNet [7] because arbitrary 4-D values are easily mapped to legitimate rotations by normalizing them to unit length, which is simpler than the orthonormalization required by rotation matrix. PoseNet optimizes pose vector with Euclidean Loss function:

$$loss(I) = \|\hat{t} - t\|_2 + \beta \left\| \hat{q} - \frac{q}{\|q\|} \right\|_2 \tag{1}$$

where $\hat{t}$ is the groundtruth of translation, q is the orientation in quaternion representation, $\hat{q}$ is the groundtruth of orientation and β is a scale factor.

However, a quaternion q represents the same orientation operation with $-q$. In practical situations, one element of q is fixed to be non-negative to avoid this ambiguity. For instance, without loss of generality we choose the first element to be non-negative. If a CNN predicts an image's orientation as $[0,1,0,0]$ while the groundtruth of the orientation is

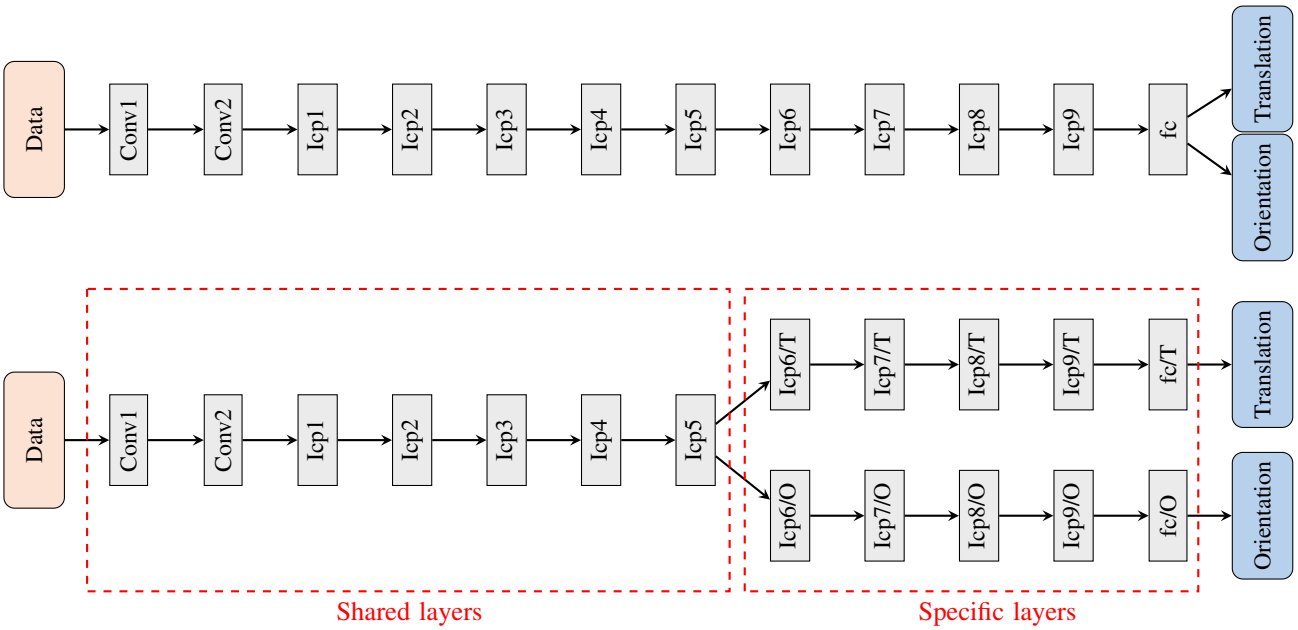


Fig. 2. PoseNet and BranchNet. **Top:** Architecture of PoseNet in which orientation and translation vectors are predicted by the same fully connected layer. **Bottom:** Architecture of BranchNet. The orientation and translation vectors are predicted by different branches. The "Icp" means "Inception" module of GoogLeNet.

$[0,-1,0,0]$, the orientation term in the loss function (1) reaches its maximum but in fact the prediction of the CNN is correct.

The Euler angles are three angles to describe the orientation of a rigid body. They represent a sequence of rotations about the axes of a coordinate system. For instance, a first rotation ϕ about z axis, a second rotation θ about x axis, and a third rotation ψ about y axis. Original Euler angles face a similar problem with quaternion: the periodicity leads to quite different angle values for similar images. To avoid this problem, we employ a variant of Euler angles named Euler6 to represent orientation.

The Euler6 is a 6D vector:

$$e = [\sin\phi, \cos\phi, \sin\theta, \cos\theta, \sin\psi, \cos\psi] \tag{2}$$

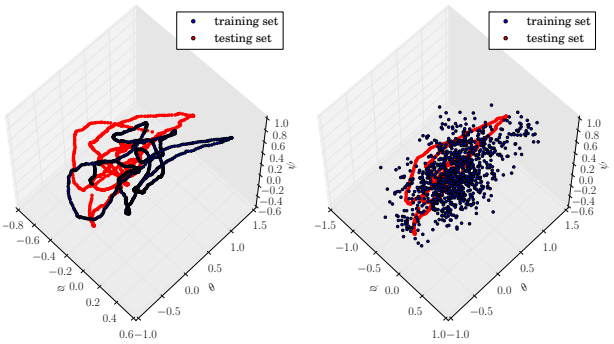
A Euclidean loss function is defined to optimize orientation and translation:

$$loss(I) = \|t - \hat{t}\|^2 + \beta \|e - \hat{e}\|^2 \tag{3}$$

where $\hat{t}$ is the translation groundtruth, $\hat{e}$ is the orientation groundtruth and β is a scale factor to balance the penalties of the orientation and translation. The loss function (3) reaches its minimum when the prediction of CNN is completely accurate.

B. Pose Synthesis

A common problem for camera relocalization is that the camera poses in training set are always very sparse in the whole pose space. Fig. 3a shows the training and testing trajectories on the Heads scene from 7Scenes dataset [2], which are clearly non overlaying. The camera poses in the training set belong to a limited number of trajectories. We



(a) Before pose synthesis (b) After pose synthesis

Fig. 3. Distribution of Euler angles on the Heads scene from 7Scenes dataset. **Left:** distribution before pose synthesis. **Right:** distribution after pose synthesis.

are only given a sparse sampling of the whole pose space. Relocalization error becomes large when the pose of the query image differs significantly from the closest pose in the training set. So overfitting is a serious problem in training of supervised learning methods for accomplishing this task.

PoseNet utilizes random cropping to cope with overfitting. Random cropping is a common method of data augmentation. With random cropping, PoseNet learns to associate a broader range of spatial activation statistics with a certain class label which improves its robustness. However, this method is unable to change the distribution of poses in training set and reduce the sparsity of poses.

We propose a variant of random cropping named pose synthesis to augment data which can significantly reduce sparsity of training pose and alleviate overfitting. Translations of

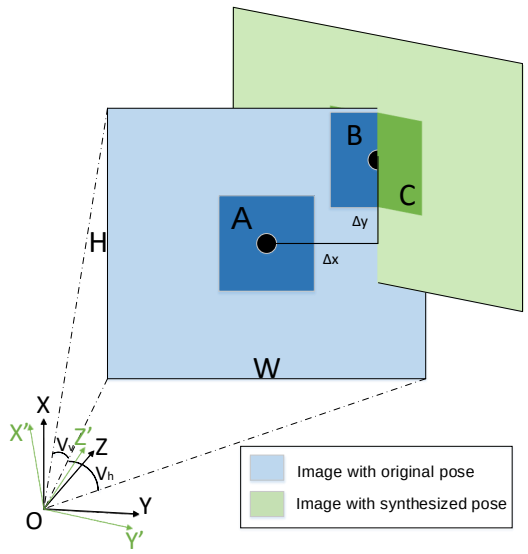


Fig. 4. Pose synthesis. OXYZ and OX'Y'Z' are coordinate systems of original camera c and rotated camera c'. Patch A and patch C are central patches of images captured by c and c'. Patch B has tiny difference with C so that the new pose OX'Y'Z' can be synthesized.

patches in the image plane can be interpreted as rotations of the camera about X axis and Y axis as illustrated in Fig. 4. Rotating the camera along X axis and then along Y axis, we synthesize a new pose. And the central patch C of the image with synthesized pose is similar to patch B in the original image. By ignoring tiny difference between patches B and C, we can assume that patches with different translations in the image plane is the central patch of another image with a new pose. In this way, we extend the limited training dataset by extracting patches and synthesizing new poses. Fig. 3b shows the distribution of synthesized orientation, which is much more uniform than the original distribution.

The effect of dataset expansion can improve relocalization performance while the label error generated by approximating the pose of C as the pose of B will degrade relocalization performance. These two factors are both proportional to the distance r between centers of the extracted patch and the original image. The farther away patch B in Fig. 4 is to patch A, the greater label error is. When (x,y) in equation 4 is $(0,0)$, patch B and synthesized patch C are just the same with patch A so that there is no label error between patch B and patch C. When patch B is sampled at a corner of original image, then the difference between B and C in appearance reaches maximum. At the same time, the farther sampled patches are away to the center of the original image, the more various synthesized poses are, which would alleviate overfitting more in training CNNs.

An ideal pose synthesis approach is to synthesize each of the 6 DOFs for a cropped patch. However, translation of patch can not be synthesized because we assume the depth information is unavailable in this study. Three orientation angles can be synthesized according to geometry information, but only θ and ψ are synthesized in this study as described above because we found that synthesis of ϕ resulted in little improvement but great data preprocessing overhead with patch rotation in the image plane.

The Euler angles $\hat{\phi},\hat{\theta},\hat{\psi}$ to convert camera from OXYZ system to OX'Y'Z' system in Fig. 4 are:

$$\hat{\phi} = 0 \tag{4}$$

$$\hat{\theta} = -\frac{\Delta x}{W}\cdot V_h \tag{5}$$

$$\hat{\psi} = \frac{\Delta y}{H}\cdot V_v \tag{6}$$

where V_h is the horizontal angle of view, V_v is the vertical angle of view, W is the width of image and H is the height of image. The synthesized pose is the combination of the original pose and $[\hat{\phi},\hat{\theta},\hat{\psi}]$.

C. Mutli-task CNN for Camera Relocalization

The relationship between orientation and translation is complex. To quantitatively understand relationship between orientation and translation, we calculated the correlations between the 6 DOFs. We first arranged 6 DOF pose vector $([X,Y,Z,\phi,\theta,\psi])$ of all images in a scene as a $6\times N$ matrix (N is the number of images in this scene), then computed the correlations between the columns which resulted in a 6×6 correlation matrix. Fig. 5 visualizes the correlation matrices of seven scenes and their average. The 3×3 matrix in the upper left of correlation matrix M is the correlation matrix of translation and the 3×3 matrix in the lower right of correlation matrix M is the correlation of Euler angles. On average, intra group correlations (0.391 for orientation and 0.293 for translation, self-correlations are not involved) are greater than inter group correlations (0.256).

In the extreme case, regressing orientation and translation separately by two individual networks may also give better results. This was verified in experiments (see section IV-E). But regressing orientation and translation individually significantly increases the computing cost. To achieve the trade-off between computing and relocalization performance, we branch the network to reduce the disturbance between regression of orientation and translation, as illustrated in Fig. 2.

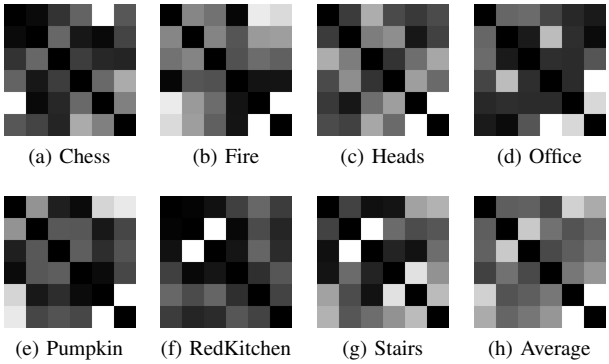
Our baseline PoseNet-Euler6 has the same architecture with PoseNet [7] except that the orientation representation is Euler6 as illustrated in Fig. 2a. The "Icp" means "Inception" module of GoogLeNet. The chunk of BranchNet-Euler6 splits into two branches which are respectively used to predict orientation and translation at "Icp6". As shown in Fig. 2b, BranchNet-Euler6 comprises two parts, namely shared layers and specific layers. The shared layers on the left side of Fig. 2b process pictures and extract low-level features shared by orientation branch and translation branch. The specific layers on the right side of Fig. 2b respectively predict orientation and translation.

In order to keep the number of parameters of BranchNet-Euler6 to be approximately equal to that of PoseNet-Euler6, we decrease the channels of specific layers. The detailed settings for PoseNet-Euler6 and BranchNet-Euler6 are described in Table I.

TABLE I

INCARNATION OF THE POSENET-EULER6 AND BRANCHNET-EULER6 ARCHITECTURE.

Type	PoseNet-Euler6 / BranchNet-Euler6							
	# channel	# 1x1	# 3x3 reduce	# 3x3	# 5x5 reduce	# 5x5	pool proj	params / k
Conv1	64 / 64							9.25 / 9.25
max pool	64 / 64							
Conv2	192 / 192		64 / 64	192 / 192				112.25 / 112.25
max pool	192 / 192							
Icp1	256 / 256	64 / 64	96 / 96	128 / 128	16 / 16	32 / 32	32 / 32	159.86 / 159.86
Icp2	256 / 256	128 / 128	128 / 128	192 / 192	32 / 32	96 / 96	64 / 64	379.63 / 379.63
max pool	480 / 480							
Icp3	512 / 512	192 / 192	96 / 96	208 / 208	16 / 16	48 / 48	64 / 64	367.36 / 367.36
Icp4	512 / 512	160 / 160	112 / 112	224 / 224	24 / 24	64 / 64	64 / 64	438.63 / 438.63
Icp5	512 / 512	128 / 128	128 / 128	256 / 256	24 / 24	64 / 64	64 / 64	498.15 /498.15
Icp6	528 / 528	112 / 112	144 / 144	288 / 288	32 / 32	64 / 64	64 / 64	591.19 / 591.19
Icp7	832 / 582	256 / 180	160 / 112	320 / 224	32 / 22	128 / 90	128 / 90	848 / 955.71
max pool	832 / 582							
Icp8	832 / 582	256 / 180	160 / 112	320 / 224	32 / 22	128 / 90	128 / 90	1019 / 998.32
Icp9	1024 / 716	384/ 269	192 / 135	384 / 269	48 / 34	128 / 90	128 / 90	1410.23 / 1389.68
avg pool	1024 / 1024							
fc	2048 / 1024							2048 / 2048



(a) Chess (b) Fire (c) Heads (d) Office

(e) Pumpkin (f) RedKitchen (g) Stairs (h) Average

Fig. 5. Visualizations of 6-DOF pose vector's correlation matrices. The brightness is proportional to correlation value. The diagonal of correlation matrices is set to 0.

IV. EXPERIMENTS

The proposed methods were evaluated on an indoor relocalization dataset 7Scenes [2]. All experiments were based on Caffe [20].

A. Dataset

The 7Scenes dataset is an indoor RGB-D relocalization dataset which is obtained from a handheld Kinect RGB-D camera at 640×480 resolution. This dataset contains significant ambiguities, motion-blur, flat surfaces and lighting conditions so it is extremely challenging for purely visual relocalization.

In all experiments, we assume the depth information is unavailable. To make the image size match with network input size, we rescaled the input image to 343×256 pixel from 640×480 pixel. Image mean for each scene was subtracted.

B. Overall Settings

The models were trained using stochastic gradient descent. The scale factor β in the loss function (3) was set to 20 in all experiments The momentum was 0.9, weight decay was 0.0002 and minibatch size was 60. The initial learning rate was 10^{-5} and dropped by 90% every 10000 iterations. Training was ended at 45000 iterations. With two NVIDIA Titan X GPUs, training took about three hours. During training, crops of 224×224 pixel were randomly extracted from the images in the training set. During inference, only the central crop of a test image was inputted to a model unless otherwise specified.

Four models were evaluated in experiments:

- PoseNet-Euler6: This network is the same as PoseNet except the orientation representation is Euler6.
- BranchNet-Euler6: Details can be found in III-C.
- PoseNet-Euler6-Aug: PoseNet-Euler6 augmented by pose synthesis.
- BranchNet-Euler6-Aug: BranchNet-Euler6 augmented by pose synthesis.

Test results are shown in Table II. As in [7] and [8], we report median error for each scene. The angular error is defined as the angle that is needed to convert CNN's orientation prediction to the orientation groundtruth. To calculate angular error of Euler6, convert e and $\hat{e}$ in (3) into rotation matrices R_e and $R_{\hat{e}}$ [21] and calculate rotation angle of $R_e^{-1}R_{\hat{e}}$. The translation error is defined as the Euclidean distance between CNN's translation prediction and the translation groundtruth.

C. Euler6

Experimental results showed that PoseNet-Euler6 significantly outperformed PoseNet (see Table II). PoseNet-Euler6's translation error had a 13.6% reduction over PoseNet after replacing quaternion with our Euler6 as orientation representation, while the orientation error was also reduced 5.4%. This demonstrates that our novel orientation representation Euler6 is more suitable for camera pose regression compared to quaternion.

In what follows, we only report the results using this new orientation representation.

TABLE II

MEDIAN ERRORS OF DIFFERENT MODELS ON THE 7 SCENES [2] DATASETS.

Scene	PoseNet	Bayesian PoseNet	PoseNet-Euler6	PoseNet-Euler6-Aug	BranchNet-Euler6	BranchNet-Euler6-Aug
Chess	8.12°, 0.32m	7.24°, 0.37m	6.51°, 0.24m	5.34°, 0.22m	6.55°, 0.20m	**5.17°, 0.18m**
Fire	14.4°, 0.47m	13.7°, 0.43m	12.27°, 0.46m	11.04°, 0.45m	11.73°, 0.35m	**8.99°, 0.34m**
Heads	12.0°, 0.29m	**12.0°**, 0.31m	14.83°, 0.22m	12.49°, 0.20m	15.50°, 0.21m	14.15°, **0.20m**
Office	7.68°, 0.48m	8.04°, 0.48m	8.15°, 0.41m	7.64°, 0.36m	8.43°, 0.31m	**7.05°, 0.30m**
Pumpkin	8.42°, 0.47m	7.08°, 0.61m	6.51°, 0.48m	5.41°, 0.40m	6.03°, **0.24m**	**5.10°**, 0.27m
RedKitchen	8.64°, 0.59m	7.54°, 0.58m	8.69°, 0.50m	**7.12°**, 0.42m	9.50°, 0.35m	7.40°, **0.33m**
Stairs	13.8°, 0.47m	13.1°, 0.48m	11.9°, 0.35m	10.99°, **0.32m**	10.99°, 0.45m	**10.26°**, 0.38m
Average	10.4°, 0.44m	9.81°, 0.47m	9.83°, 0.38m	8.58°, 0.34m	9.82°, 0.30m	**8.30°, 0.29m**

D. Pose Synthesis

Both PoseNet-Euler6-Aug and BranchNet-Euler6-Aug augmented by pose synthesis outperformed their baselines (see Table II). With pose synthesis, PoseNet-Euler6-Aug had a 12.7% error reduction on orientation and a 10.5% error reduction on translation while BranchNet-Euler6-Aug had a 15.4% error reduction on orientation and a 3.3% error reduction on translation.

Pose synthesis is able to alleviate overfitting. The average training errors of PoseNet-Euler6 and PoseNet-Euler6-Aug on 7Scenes were (2.80°, 0.15m) and (3.03°, 0.15m). With pose synthesis, PoseNet-Euler6 got greater test error but even smaller training error than the corresponding PoseNet-Euler6-Aug, which indicates that overfitting occurred in training PoseNet-Euler6.

The effect of dataset expansion can improve relocalization performance while the label error will degrade relocalization performance. These two factors are both proportional to the distance r between centers of extracted patch and original image. An experiment was designed to explore the influence of r on relocalization performance. For convenience, we constrained the sampled crops in a rectangle centered at the origin and the size of this rectangle is $[c+(H-c)\cdot s]\times[c+(W-c)\cdot s]$, where s is the range factor between 0 and 1, c is the size of crop, H and W are the height and width of the image. When s is 0, sampled crops are all located at the center of training images. With s increasing from 0 to 1, crops can be sampled from position farther from the center of the image. When s is 1, crops can be sampled at anywhere of the image.

Fig. 6 shows relocalization errors with different range factor s on the Chess scene. When s was set to 0, relocalization performance were worse than PoseNet-Euler6. This demonstrates that random cropping could improve performance of CNNs for camera relocalization. With s increasing, both angular and translational errors decreased and reached their minima when s was set to 1. The negative impact of label error is suppressed by the positive impact of dataset expansion. Similar conclusions can be drawn on other scenes in this dataset.

E. BranchNet

We employed two individual PoseNet-Euler6-Aug to regress orientation and translation separately and the average error over 7Scenes was (8.38°, 0.25m). This method outperformed a single PoseNet-Euler6-Aug, which indicates

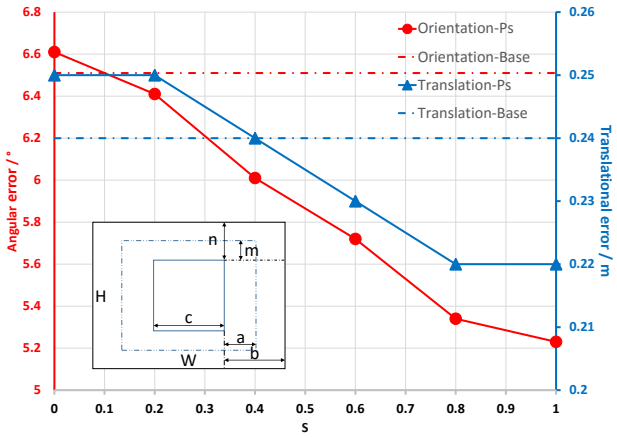


Fig. 6. Median results of PoseNet-Euler6 with different range factor s on the Chess scene. The inset shows the location of extracted patch in the image, specified by s, which is defined as $s=\frac{a}{b}=\frac{m}{n}$. The black rectangle is the image. The solid blue square is the central patch of the image. The dashed rectangle devotes the area in which the patch can be extracted.

that regressing orientation and translation together brings disturbance to each one's training. But predicting poses with two models is expensive.

BranchNet is a balance between a single PoseNet and two separate PoseNets. But where to split into two branches is a problem. We evaluate the influence of location for splitting on the Office scene. The result is shown in Fig. 7. The network without branch had the worst performance. The split of final fully connected layer led to the greatest reduction of both orientation and translation errors. With the increasement of the number of shared layers, the errors are reduced firstly and then converge. Similar conclusions can be drawn on other scenes.

Compared with PoseNet-Euler6, the translational error of BranchNet-Euler6 was reduced 21.1%. Compared with PoseNet-Euler6-Aug, the translational error of BranchNet-Euler6-Aug was reduced 14.7%. Orientation errors also had certain reduction but the improvement was negligible.

Note that we simply choose "Icp6" as the splitting node for BranchNet-Euler6 and it may not be the optimal splitting depth.

Compared to PoseNet, BranchNet-Euler6-Aug had a 19.3% reduction on orientation error and a 34.1% reduction on translation error. Compared to Bayesian PoseNet, BranchNet-Euler6-Aug had a 15.9% reduction on orientation

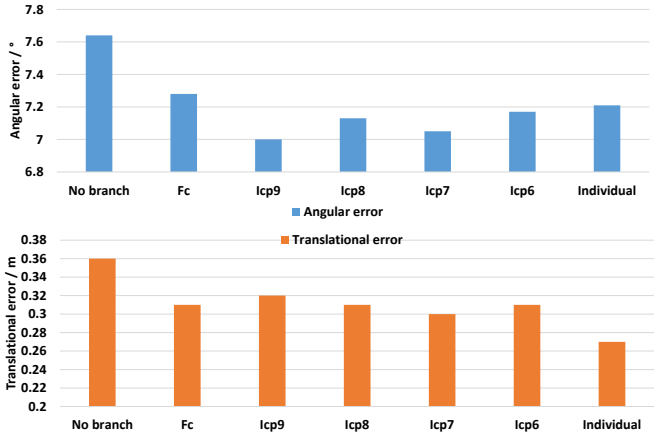


Fig. 7. Median results of BranchNet-Euler6-Aug with different branch splitting node on the Office scene.

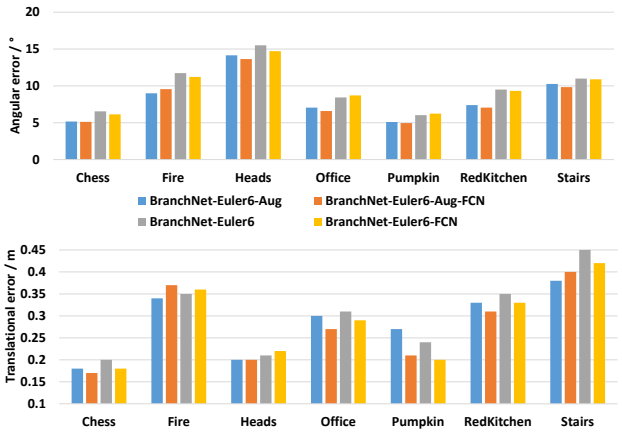


Fig. 9. Comparison of the models with and without fully convolution inference scheme

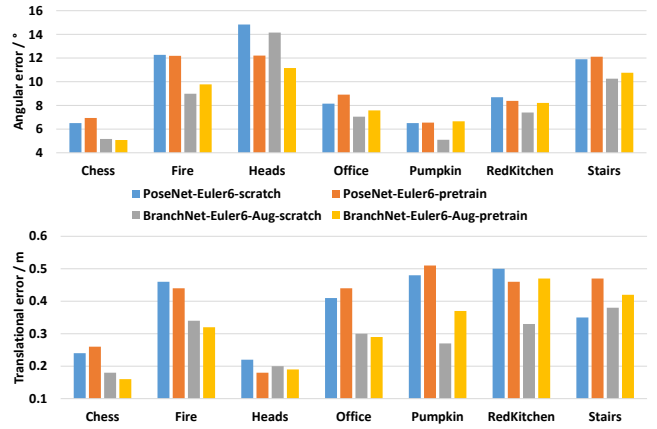


Fig. 8. Median results of PoseNet-Euler6 and BranchNet-Euler6-Aug trained from scratch and fine-tuned from pretrained model.

error and a 38.3% reduction on translation error.

F. Fine-tuning

Fine-tuning generally leads to a faster and more accurate training procedure. For fair comparison, we attempted to train models from pretrained GoogLeNet model. In order to adapt BranchNet-Euler6-Aug to pretrained GoogleNet model, the numbers of channels of BranchNet-Euler6-Aug's specific layers were changed to the numbers of corresponding layer's channels in PoseNet-Euler6.

To our surprise, pretrained model from ImageNet [22] did not lead to much better performance and even harmed performance on some scenes compared to learning from scratch. This suggests that the features learned from the large-scale classification ImageNet benchmark [22] do not generalize well on the indoor relocalization dataset.

G. Fully Convolutional Network

Evaluation with multiple crops of input image is a common method to improve performance. However, this approach is inefficient as the network needs to re-compute each crop. For example, PoseNet was evaluated with 128 uniformly spaced crops of the input image, which resulted in a 5% reduction of orientation error on 7Scenes and the computational time increased from 5ms to 95ms with parallel GPU processing.

We adopted a more efficient approach by using the full convolution trick [23]. At inference time, we evaluated an input image in the following way. First, the fully connected layers in models were converted to convolutional layers with 1×1 kernel. Then, fully convolutional networks were applied to the input image rescaled to 256×342 pixel and generated a regression map. Finally, the regression map was spatially averaged to obtain the final 9D pose vector.

Results are shown in Fig. 9. With fully convolutional networks applied over the whole image, small improvements were achieved. This demonstrates that spatial average could improve the relocalization performance. It is not surprising that the improvement is only about 2% since this testing process is equivalent to averaging predictions over only eight crops uniformly distributed in the image due to the limited image size.

H. Efficiency of the BranchNet

We kept the number of parameters of BranchNet similar to that of PoseNet [7]. Storing weights took 46 MB for BranchNet-Euler6. Branching networks slowed down the forward speed from 5ms to 6ms per frame on a NVIDIA Titan X GPU. We then tested BranchNet-Euler6 in the GPU of an Intel NUC mobile platform (Intel CoreTM i5-6260U) with clCaffe [24], and reached a speed of 43 fps, which meets the real-time requirement of many robotic applications.

V. Conclusion and Discussion

In this paper, we present three techniques for CNN-based camera relocalization. The first one is a new orientation representation Euler6. The second one is the pose synthesis for data augmentation. And the third one is the BranchNet for multi-task regression. Experiments showed that all of the above techniques improved the relocalization accuracy, and

they together reduced the error of previous methods by a significant margin.

One limitation of our approach, as well as PoseNet approaches [7], [8], is that it is only suitable for scenarios while depth information is unavailable because when depth information is available, there exist much more accurate approaches, for example, SCoRe Forests [2]. We attempted to utilize the depth information by simply add the depth image as the fourth channel to the original input which has RGB channels but did not obtain much better results than our current results. How to utilize the depth information to improve the performance of CNN remains to be an open problem.

ACKNOWLEDGEMENT

This work was supported in part by the National Basic Research Program (973 Program) of China under Grant 2013CB329403, in part by the National Natural Science Foundation of China under Grant 61273023, Grant 91420201, Grant 61332007, and Grant 61621136008, and in part by the German Research Foundation (DFG) under Grant TRR-169.

REFERENCES

[1] J. Martinez-Carranza, A. Calway, and W. Mayol-Cuevas, "Enhancing 6d visual relocalisation with depth cameras," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2013, pp. 899–906.

[2] J. Shotton, B. Glocker, C. Zach, S. Izadi, A. Criminisi, and A. Fitzgibbon, "Scene coordinate regression forests for camera relocalization in rgb-d images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 2930–2937.

[3] A. Guzman-Rivera, P. Kohli, B. Glocker, J. Shotton, T. Sharp, A. Fitzgibbon, and S. Izadi, "Multi-output learning for camera relocalization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1114–1121.

[4] J. Valentin, M. Nießner, J. Shotton, A. Fitzgibbon, S. Izadi, and P. H. Torr, "Exploiting uncertainty in regression forests for accurate camera relocalization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 4400–4408.

[5] G. Klein and D. Murray, "Improving the agility of keyframe-based slam," in *European Conference on Computer Vision*. Springer, 2008, pp. 802–815.

[6] A. P. Gee and W. W. Mayol-Cuevas, "6d relocalisation for rgbd cameras using synthetic view regression." in *BMVC*, 2012, pp. 1–11.

[7] A. Kendall, M. Grimes, and R. Cipolla, "Posenet: A convolutional network for real-time 6-dof camera relocalization," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 2938–2946.

[8] A. Kendall and R. Cipolla, "Modelling uncertainty in deep learning for camera relocalization," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 4762–4769.

[9] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

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

[11] A. Byravan and D. Fox, "Se3-nets: Learning rigid body motion using deep neural networks," *arXiv preprint arXiv:1606.02378*, 2016.

[12] A. Evgeniou and M. Pontil, "Multi-task feature learning," *Advances in Neural Information Processing Systems*, vol. 19, p. 41, 2007.

[13] Z. Zhang, P. Luo, C. C. Loy, and X. Tang, "Facial landmark detection by deep multi-task learning," in *European Conference on Computer Vision*. Springer, 2014, pp. 94–108.

[14] R. Ranjan, V. M. Patel, and R. Chellappa, "Hyperface: A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition," *arXiv preprint arXiv:1603.01249*, 2016.

[15] G. Gkioxari, B. Hariharan, R. Girshick, and J. Malik, "R-cnns for pose estimation and action detection," *arXiv preprint arXiv:1406.5212*, 2014.

[16] G. Wang and D. Forsyth, "Joint learning of visual attributes, object classes and visual saliency," in *2009 IEEE 12th International Conference on Computer Vision*. IEEE, 2009, pp. 537–544.

[17] E. M. Hand and R. Chellappa, "Attributes for improved attributes: A multi-task network for attribute classification," *arXiv preprint arXiv:1604.07360*, 2016.

[18] Z. Yan, H. Zhang, R. Piramuthu, V. Jagadeesh, D. DeCoste, W. Di, and Y. Yu, "Hd-cnn: hierarchical deep convolutional neural networks for large scale visual recognition," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 2740–2748.

[19] K. Ahmed, M. H. Baig, and L. Torresani, "Network of experts for large-scale image categorization," *arXiv preprint arXiv:1604.06119*, 2016.

[20] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *Proceedings of the ACM International Conference on Multimedia*. ACM, 2014, pp. 675–678.

[21] K. Shoemake, "Euler angle conversion," *Graphics gems IV*, pp. 222–229, 1994.

[22] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009, pp. 248–255.

[23] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[24] J. Bottleson, S. Kim, J. Andrews, P. Bindu, D. N. Murthy, and J. Jin, "clcaffe: Opencl accelerated caffe for convolutional neural networks," in *2016 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*. IEEE, 2016, pp. 50–57.