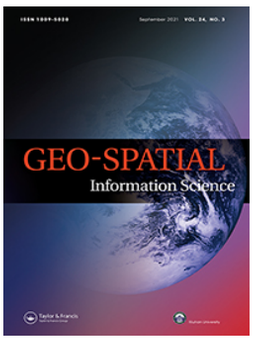


Geo-spatial Information Science

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/tgsi20

# VNLSTM-PoseNet: A novel deep ConvNet for real-time 6-DOF camera relocalization in urban streets

Ming Li, Jiangying Qin, Deren Li, Ruizhi Chen, Xuan Liao & Bingxuan Guo

**To cite this article:** Ming Li, Jiangying Qin, Deren Li, Ruizhi Chen, Xuan Liao & Bingxuan Guo (2021) VNLSTM-PoseNet: A novel deep ConvNet for real-time 6-DOF camera relocalization in urban streets, Geo-spatial Information Science, 24:3, 422-437, DOI: 10.1080/10095020.2021.1960779

**To link to this article:** https://doi.org/10.1080/10095020.2021.1960779



Published online: 20 Aug 2021.

GEO-SPATIAL INFORMATION SCIENCE
2021, VOL. 24, NO. 3, 422–437
https://doi.org/10.1080/10095020.2021.1960779



# VNLSTM-PoseNet: A novel deep ConvNet for real-time 6-DOF camera relocalization in urban streets

Ming Li[a,b], Jiangying Qin[a], Deren Li[a], Ruizhi Chen[a], Xuan Liao[c] and Bingxuan Guo[a]

[a]State Key Laboratory of Information Engineering in Surveying Mapping and Remote Sensing, Wuhan University, Wuhan, China; [b]Department of Physics, ETH Zurich, Zurich, Switzerland; [c]Department of Land Surveying and Geo-informatics, The Hong Kong Polytechnic University, Hong Kong, China

**ABSTRACT**

Image-based relocalization is a renewed interest in outdoor environments, because it is an important problem with many applications. PoseNet introduces Convolutional Neural Network (CNN) for the first time to realize the real-time camera pose solution based on a single image. In order to solve the problem of precision and robustness of PoseNet and its improved algorithms in complex environment, this paper proposes and implements a new visual relocation method based on deep convolutional neural networks (VNLSTM-PoseNet). Firstly, this method directly resizes the input image without cropping to increase the receptive field of the training image. Then, the image and the corresponding pose labels are put into the improved Long Short-Term Memory based (LSTM-based) PoseNet network for training and the network is optimized by the Nadam optimizer. Finally, the trained network is used for image localization to obtain the camera pose. Experimental results on outdoor public datasets show our VNLSTM-PoseNet can lead to drastic improvements in relocalization performance compared to existing state-of-the-art CNN-based methods.

**ARTICLE HISTORY**
Received 5 October 2020
Accepted 23 July 2021

**KEYWORDS**
Camera relocalization; pose regression; deep convnet; RGB image; camera pose

## 1. Introduction

Image-based camera relocalization is a basic problem in many computer vision applications, such as autonomous vehicle driving, mobile robots, Augmented Reality (AR), pedestrian visual positioning, Structure from Motion (SfM) (Li et al. 2020b; Tateno et al. 2017; Asadi et al. 2019; Liu et al. 2020; Acharya et al. 2019a; Niu et al. 2019), and so on. It refers to estimating the camera's pose, that is position, and orientation, according to the image. Traditional geometric-based image positioning is mainly realized by local feature matching. Its main idea is to extract local features from the image and establish 2D-3D matching with corresponding 3D points, and then determine the camera pose according to the matching relationship. Geometry-based visual positioning methods rely on correct local feature matching, however, not enough accurate matching points can be found in all scenarios (Li et al. 2020a; Jin et al. 2021; Miao et al. 2021). Various complex situations that may exist in the real environments, such as object occlusion, viewpoint changes, motion blur, illumination changes, and lack of texture, may affect feature matching and make it difficult to obtain accurate camera poses or successful positioning.

In recent years, deep learning, an important branch of machine learning, has been widely used in many computer vision fields, such as object recognition, image retrieval, image classification, and so on. (Li et al. 2020b; Husain and Bober 2019; Hu et al. 2018; Zhang, Li, and Du 2018; Singh et al. 2020). In 2015, Kendall, Grimes, and Cipolla (2015) innovatively introduced Convolutional Neural Networks (CNN) into the field of image-based camera positioning and proposed PoseNet method. This method uses transfer learning from large-scale classification data to directly obtain 6-DOF camera pose from a single image in an end-to-end manner. It significantly improves the robustness and efficiency of geometric positioning based on local features and positioning using bag of word vectors and random forests image retrieval technology in traditional machine learning. Although PoseNet overcomes many limitations of existing methods, especially reduces the dependence on rich textures, and improves the robustness and efficiency of localization, its localization accuracy is still far behind the geometric-based visual relocalization method when the local features perform well. Therefore, how to improve the accuracy of image positioning based on convolutional neural networks is still an important problem to be solved in many precise positioning applications.

Based on this, in order to further improve the accuracy and robustness of image positioning method based on convolution neural network, this paper proposes a high-precision image relocalization method

**CONTACT** Jiangying Qin jy_qin@whu.edu.cn

(VNLSTM-PoseNet) based on Long Short-Term Memory network (LSTM) (Shi et al. 2015). This method mainly improves PoseNet from the following three aspects. (1) By improving the clipping method of input image, the image can obtain a larger receptive field, thereby obtaining more characteristic information for image positioning. (2) Based on the Pytorch framework, the Nadam optimizer is used to optimize the network to obtain more suitable network parameters. (3) The LSTM structure is introduced into the PoseNet network to perform structural dimensionality reduction on the Fully Connected (FC) layer and select the most useful relevant features for camera relocalization tasks. Experiments show that the method proposed in this paper has better accuracy and stronger robustness than PoseNet.

## 2. Relate work

At present, there are three main methods for image-based camera positioning, namely, image positioning method based on geometry, image positioning method based on deep learning, and image positioning method based on fusion of geometry and deep learning.

Geometry-based image positioning method is based on the known three-dimensional environment. It takes a set of photos of a certain scene and creates a three-dimensional model of the scene through the Structure from Motion (SfM) (Sattler, Leibe, and Kobbelt 2016; Han, Laga, and Bennamoun 2019) or Simultaneous Localization and Mapping (SLAM) (Mur-Artal, Montiel, and Tardos 2015; Mur-Artal and Tardos 2017). It matches the local 2D feature points extracted from the query image with the corresponding 3D feature points in the model to establish the corresponding relationship (e.g. SIFT (Lowe 2004), SURF (Bay et al. 2008), ORB (Rublee et al. 2011) algorithms), and solves the camera pose of six degrees of freedom through Perspective-n-Point (PnP) and other algorithms (e.g. EPnP, UPnP (Hesch and Roumeliotis 2011; Lepetit, Moreno-Noguer, and Fua 2009) algorithms). For the mismatched points in the matching process, the Random Sampling Consensus algorithm (RANSAC (Fischler and Bolles 1981), Progressive Sampling Consensus (PROSAC) (Chum and Matas 2005)) is used to eliminate the mismatching points and accelerate the camera pose calculation (Qin et al. 2019). Among them, when performing 2D-3D matching, it is necessary to search for the 3D feature points corresponding to the 2D feature points in the 3D point cloud feature library. This process is usually implemented by the nearest neighbor search method. Common nearest neighbor search algorithms are KD tree (Silpa-Anan and Hartley 2008) and K-means (Nister and Stewenius 2006). However, the cost of matching in a large and dense feature space is very large. In order to speed up this feature matching process, Sattler, Leibe, and Kobbelt (2011) used visual vocabulary for effective 2D to 3D matching, and Sattler, Leibe, and Kobbelt (2016) proposed an active search mechanism based on feature-to-point and point-to-feature. Glocker et al. (2015) used the Bag of Words model (BoW) to find key frames with the same visual words as the current frame. This type of method can reduce the matching cost to a certain extent. However, because the matching cost increases exponentially with respect to the number of key points, this type of method is not suitable for complex large-scale 3D scenes. On the other hand, the accuracy of the camera pose in the geometry-based image positioning method directly depends on the accuracy of feature matching, and it is difficult to obtain accurate matching points in some complex scenes, which will seriously affect the accuracy of the camera pose calculation. These are also the important factors that restrict the image positioning method based on geometry.

In recent years, with the development of deep learning, scholars have also begun to try to introduce deep learning into the field of image localization and have made a lot of progress (Brachmann et al. 2016; Melekhov et al. 2017; Behera et al. 2020; Shukla et al. 2018; Prins and Van Niekerk 2021; Lock and Pettit 2020; Kosowski et al. 2020). (Shotton et al. 2013) trained a random forest on RGB-D images and transformed the positioning problem into a problem of minimizing the energy function on the possible camera position assumptions. This method eliminates the need for traditional pipeline of feature extraction, feature description, and feature matching. Valentin et al. (2015) used the uncertainty in the model to further improve this method. It starts from the unique point estimation, and then predicts its uncertainty thus achieving more reliable continuous pose optimization. However, these two methods require input of depth information during training, which is detrimental to the adaptability and generalization of the model. In 2015, PoseNet proposed by Kendall et al. was the first attempt to apply CNN to the task of camera pose regression. This method modifies the GoogleNet (Szegedy et al. 2015) architecture and uses the transfer learning in the ImageNet (Deng et al. 2009) classification task to regress the 6-DOF camera pose from RGB images in an end-to-end manner. However, the accuracy of this method is still far behind the traditional geometry-based positioning methods. Therefore, many scholars have devoted themselves to modifying the PoseNet method to improve its accuracy and have proposed many algorithms. Kendall and Cipolla (2016) used Bayesian CNN to estimate the uncertainty of positioning, thus improving the positioning

accuracy of the system. Kendall and Cipolla (2017) proposed a loss function based on geometry and reprojection errors aiming to solve the problems of hyperparameter training caused by the use of L2 distance in the PoseNet loss function. Valada, Radwan, and Burgard (2018) combined the geometric knowledge and semantic knowledge of the world to locate and proposed a novel geometric consistency loss function. These two methods improve the simple loss function of PoseNet and add other constraints to improve the positioning accuracy. Wu, Ma, and Hu (2017) proposed to use Euler angle variant to represent orientation, and designed BranchNet multi-task CNN to deal with the complex coupling of position and direction. It solves the problem that the weighting factor between the position error and the orientation error in the loss function is not robust to different scenarios caused by PoseNet learning position and orientation at the same time. Khoshelham and Winter (2019) and Acharya et al. (2019b) proposed to obtain a 3D indoor model from the existing Building Information Model (BIM) to generate synthetic images of known camera poses, and then fine-tune the Bayesian convolutional neural network to perform camera positioning. This is a new attempt to locate by synthetic images, but the accuracy of the method is not high. Nguyen et al. (2017) proposed an SP-LSTM framework based on CNN and LSTM. CNN and LSTM were used to learn the depth features and spatial dependence of images, respectively. It uses time information to enhance camera pose estimation. Ming, Chunhua, and Reid (2018) established a hybrid model of two main machine learning frameworks, CNN, and Gaussian Process Regression (GPR), and designed a unified objective function (minimizing KL divergence function) to drive the model to be trained in an end-to-end manner. The introduction of probability model is conducive to further improving the positioning accuracy.

Image positioning method based on fusion of geometry and deep learning is to combine geometry methods and deep learning methods to estimate the camera pose. Deep learning part is used to learn and predict the 3D position of a pixel in world coordinates while geometry part infers the camera pose from these correspondences. Guzman-Rivera, Kohli, and Glocker et al. (2014) tried to use hybrid methods for image localization, but their main limitation was that they require the use of RGB-D images for training and testing. Cavallari et al. (2017) optimized this limitation and proposed to use only automatic context random forest from RGB images for positioning. L. Meng et al. (2016) performed RGB image localization by using regression forest to estimate the initial camera pose, then queried the nearest neighbor key frame image, and optimized the initial pose by sparse feature matching between the camera input image and the nearest key frame. Brachmann et al. (2017) used VGG style architecture to predict scene coordinates and proposed a distinguishable RANSAC, so that it could learn a matching function, which optimizes pose quality. Although these methods improve positioning accuracy, they require thousands of predictions about scene coordinates, which cause RANSAC to spend more and more time to estimate the best camera pose.

In order to further optimize the positioning accuracy and robustness of PoseNet in challenging scenarios, this paper proposes a novel visual positioning method: VNLSTM-PoseNet. Aiming at the problem of image information loss caused by cropping the image, which is resized according to the aspect ratio used by the traditional PoseNet, this paper intends to directly resize the input image to obtain a larger receptive field, and trains the proposed deep learning network based on LSTM. In addition, this method uses the Nadam optimizer to optimize the deep learning network. Experimental results show that the method proposed in this paper is significantly better than PoseNet method in terms of both position accuracy and orientation accuracy.

## 3. Methodology

The experimental images used in this paper are automatically generated sample labels (i.e. camera pose) by SfM in advance. During image preprocessing, in order to obtain the image of fixed size, this paper proposes to directly resize the training image to the corresponding size without cropping. Then, the images and corresponding labels are trained in the high-precision positioning network based on LSTM units. On the basis of PoseNet network structure, the network introduces LSTM to perform structural dimensionality reduction on the full connection layer and select the most useful features for camera pose estimation task. In addition, this paper adopts the Nadam optimizer to optimize the network to train the most suitable parameters. Figure 1 is the architecture of the proposed pose regression ConvNet.

### 3.1. PoseNet image positioning network

a. Training image resize. PoseNet is based on GoogleNet network. Its biggest innovation is to propose transfer learning, which uses a classifier and a small amount of training samples to obtain a regression for positioning. In this way, the problem of insufficient training samples can be effectively solved. However, one disadvantage of using transfer learning and pre-trained networks is that it has strict limitations on the network structure. Specifically, the size of the RGB image input to the network must be

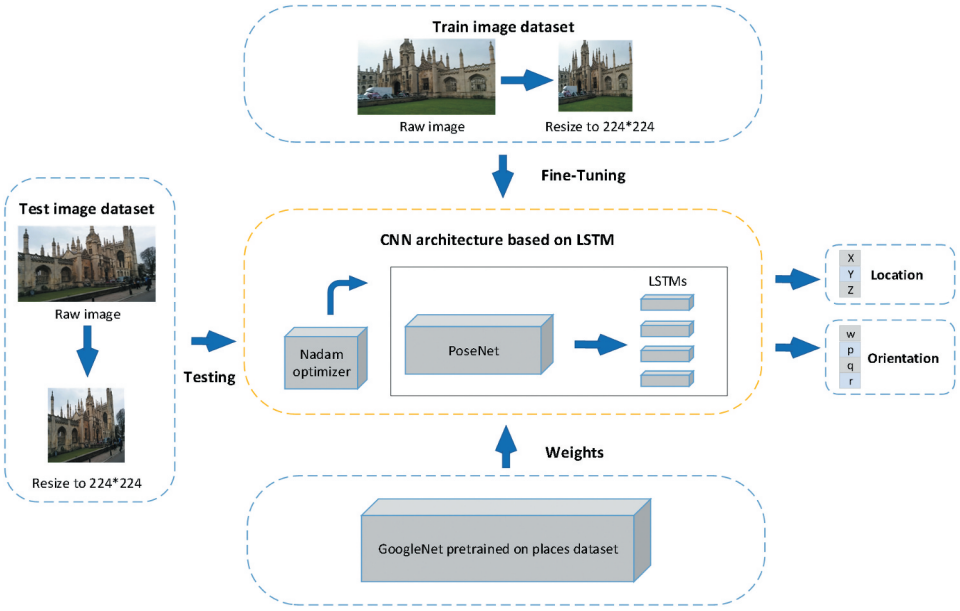



**Figure 1.** Architecture of the proposed pose regression ConvNet.

224×224 pixels specified by GoogleNet (Seifi and Tuytelaars 2019). However, the actual RGB image participating in the training may not be the specified size. To solve this problem, PoseNet's processing method is to resize the minimum side size of the image to 256 pixels according to the aspect ratio of the original image, and then crop the 224×224 window in the middle of the scaled image as the training image. As shown in Figure 2, Figure 2(a) is the original image, Figure 2(b) is the image whose height is resized to 256 pixels according to the aspect ratio, and Figure 2(c) is the image after cropping 224×224 pixels in the center of Figure 2(b). One disadvantage of this process is that the image information outside the cropping window will be lost and cannot be added to the network for training. However, the missing part may also contain key information to assist positioning, which may affect the accuracy of positioning.

b. Network structure. GoogleNet (Szegedy et al. 2015) is a new deep learning framework proposed by Christian Szegedy in 2014. It is originally designed for object classification and detection. GoogleNet innovatively uses the Inception module to make the existing dense components close to and cover the best local sparse structure in the convolutional visual network. GoogleNet neural network has 22 layers. It uses a 224×224 pixels image as input, uses a Rectified Linear unit (ReLu) as an activation function, and propagates it through nine stacked Inception modules. Each layer in the network learns a further abstraction of the input data. The highest level of abstraction (located in the last layer of the network) is fed to the fully connected layer and the softmax layer along with the two intermediate abstractions to predict the class of the object. The network structure of GoogleNet is shown in Figure 3.

Based on GoogleNet, PoseNet changes the three softmax classifiers to three regressors. The softmax layer is removed, and the output of the final fully connected layer is the camera pose vector. In addition, it inserts another fully connected layer whose feature size is 2048-demensional before the final regression, which constitutes a positioning feature vector and can be used to achieve feature generalization. The network structure of PoseNet is shown in Figure 4. The blue part represents the pre-training module inherited

(a) Original image (b) Image after resizing (c) Image after cropping

**Figure 2.** The example of PoseNet image preprocessing. (a) Original image (b) Image after resizing (c) Image after cropping.

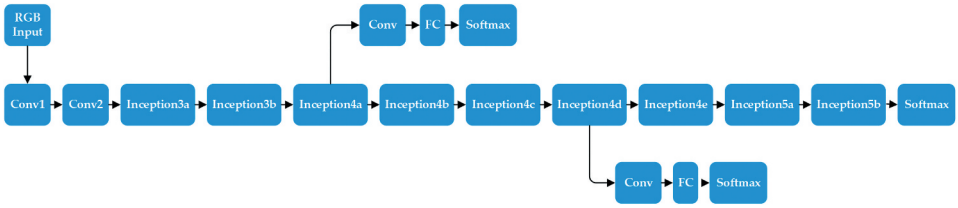



**Figure 3.** The structure of GoogleNet.

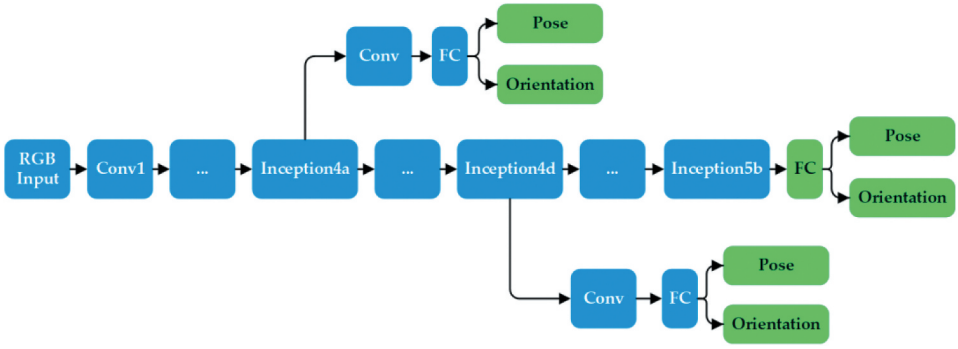



**Figure 4.** The structure of PoseNet.

from GoogleNet while the green part represents the improved PoseNet module.

c. Network optimizer. PoseNet uses Stochastic Gradient Descent algorithm (SGD) (Zinkevich et al. 2011) to optimize the network. SGD is a very common optimization algorithm in neural network model training, which is based on the gradient descent algorithm. The basic idea of the gradient descent algorithm is to obtain the partial derivative of each hyperparameter, and then the current gradient can be obtained. The hyperparameter is updated in the opposite direction of the gradient. Through iteratively updating the loss function in this way, the global optimal solution of the hyperparameter can be obtained and the loss function can be minimized. However, each step of the gradient descent algorithm needs to calculate the gradient of all hyperparameters so the iteration speed is bound to be very slow, which brings challenges to solving large-scale data optimization problems. In order to quickly achieve gradient descent, the stochastic gradient descent algorithm proposes to randomly use a sample to represent all samples for gradient descent during each update, and then adjust the hyperparameters. The SGD gradient is shown in Equation (1), where $\alpha$ is the learning rate and $g_t$ is the gradient of the current batch.

$$\eta_t = \alpha \cdot g_t \tag{1}$$

There are two problems with the SGD algorithm. The first is that it is difficult to choose an appropriate learning rate, so SGD uses the same learning rate for all parameters. But in practical applications, for sparse data or features, we may want to update faster, and for features that do not appear frequently, we hope that it can be updated slower to reduce training costs. The SGD algorithm cannot satisfy this point. Second, because SGD updates frequently, it may cause severe oscillations. In addition, because the SGD algorithm uses the gradient descent of a random sample as the average gradient descent of the overall sample, this also makes it easy for SGD to converge to the local optimum, and in some cases may be trapped in the saddle point. This limits the optimization performance of SGD for convolutional neural networks thus affecting the positioning accuracy.

### 3.2. VNLSTM-PoseNet image positioning network

a. Image processing for larger receptive fields. In order to solve the problem of image information loss in PoseNet, this paper proposes to use the entire field of view of the image, that is, only need to resize the input image to 224×224 pixels, as shown in Figure 5. Figure 6 shows the difference in the receptive field of the input image of PoseNet and the network of this paper. There is significant key information in the red box area in the Figure 6(b), but the PoseNet network discards this information. The direct resizing method proposed in this paper will lead to different aspect ratios, but considering that the changes of aspect ratio are consistent with all images in the dataset, we think that the loss of the original aspect ratio will not have a great impact on the network performance. On the other hand, this resize method would reduce the

(a) Original image (b) Image after preprocessing

**Figure 5.** Effect of improved image preprocessing. (a) Original image (b) Image after preprocessing.

(a) PoseNet (b) VNLSTM-PoseNet

**Figure 6.** Comparison figure of the field of view of PoseNet and our method. (a) PoseNet (b) VNLSTM-PoseNet.

resolution of the image, but compared to the image resolution, the receptive field is more important, because the pooling layer in the network can smooth the high-frequency details of the high-resolution image. Therefore, a higher positioning accuracy can be obtained by adopting a direct resize method. Our experimental results also proved the hypothesis.

b. LSTM-based network structure. GoogleNet uses an average pooling layer after the convolutional layer to collect the information of each feature channel in the entire image. PoseNet uses a fully connected layer after average pooling layer to learn the correlation between features. But the regression pose is not optimal after the high-dimensional output of the fully connected layer. By the way, in order to overcome the gradient vanishing problem and have faster training speed, the Relu activation function is chose. Specifically, compared with the amount of available training data, the dimension of the 2048-dimensional image through the fully connected layer is usually relatively large. Therefore, the linear pose regressor has multiple degrees of freedom, and overfitting is likely to cause inaccurate prediction of the test image. We can directly reduce the dimensionality of fully connected layer, but studies have shown that it is more effective to use LSTM memory block networks for dimensionality reduction (Walch et al. 2017). Compared with PoseNet applying dropout to avoid overfitting, the method in this paper estimates more accurate position, which proves the rationality of our use of LSTMs. In the network of this paper, we output the 2048-dimensional feature vector as a sequence and insert four LSTM units after the fully connected layer. They serve as dimensionality reduction of feature vectors in a structured manner, and recognize the most useful feature correlations to complete the pose estimation task. In fact, directly inputting the 2048-dimensional vector into the LSTM does not work well for the reason that even though the storage unit of the LSTM can remember the features in the distance, the 2048 length vector is too long for the LSTM. To solve this problem, we resize the vector to a 32×64 matrix and apply four LSTMs in four directions: up, down, left, and right. Then connect these four LSTMs and put them into the fully connected pose prediction layer, which serves as structured dimensionality

reduction and greatly improves the accuracy of pose prediction. The network structure is shown in Figure 7, where the blue part represents the module inherited from PoseNet and the yellow part represents the improved module of this paper.

c. Improved network optimizer. Nadam method is a method to obtain better performance by calculating the adaptive learning rate of each hyperparameter. It combines the idea of Nesterov Accelerated Gradient (NAG) on the basis of Adam (Dozat 2016).

Adam method adds first-order momentum and second-order momentum on the basis of SGD. The first-order momentum is shown in Equation (2), where $\beta_1$ is a hyperparameter, often taking an empirical value of 0.9 and $g_t$ is the gradient of objective function with respect to the parameters. The first-order momentum is the average value of the exponential movement of the gradient direction at each time, which is approximately equal to the average value of the sum of the gradient vectors at the latest $1/((1\text{-}\beta_1))$ time.

$$m_t = \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot g_t \tag{2}$$

In other words, the descending direction at time $t$ is not only determined by the gradient direction of the current point, but also by the descending direction accumulated before. This means that the descending direction is mainly the descending direction accumulated before, and is slightly biased to the descending direction at the current moment, thereby avoiding training problems caused by the extreme current gradient.

The second-order momentum solves the problem of learning rate, and the historical update frequency is measured by the second-order momentum-the sum of the squares of all gradient values so far in this dimension. The second-order momentum is shown in Equation (3), where $\beta_2$ is a hyperparameter.

$$V_t = \beta_2 \cdot V_{t-1} + (1 - \beta_2) g_t^2 \tag{3}$$

For parameters that are updated frequently, we have accumulated a lot of knowledge about them. We do not want to be affected too much by a single sample. We hope that the learning rate will be slower. For parameters that are updated occasionally, we know too little information. We hope to learn more from every occasional sample, that is, the learning rate is higher. This can be achieved through second-order momentum.

Considering that $m_t$ and $v_t$ are biased to the initial value at the initial stage of the iteration, bias correction can be made to the first-order momentum and second-order momentum, as shown in Equations (4) and (5).

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t} \tag{4}$$

$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t} \tag{5}$$

Finally, the updating rule of Adam algorithm after introducing first-order momentum and second-order momentum is shown in Equation (6).

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t} + \varepsilon} \hat{m}_t \tag{6}$$

Among them, $\theta_t$ is the model parameter at the current moment, $\theta_{t+1}$ is the model parameter at the next moment to be predicted, $\varepsilon$ is the smoothing term to prevent the denominator from being zero, and $\eta$ is the learning rate.

The gradient direction in Adam method is determined by the accumulated momentum and the current gradient. But the core idea of NAG is to consider the influence of the future position on the current gradient when calculating the gradient. That is, in order to make the descent process more intelligent, the algorithm must be able to slow down the update rate before the objective function has a tendency to increase. The result is to prevent the algorithm from being too fast, thus increasing responsiveness, and effectively solving the problem of SGD easily falling into local optimality.

It uses the momentum of the future moment when calculating the model parameters $\theta_t$ in the previous iteration, namely,

$$\theta_t = \theta_{t-1} - m_t \tag{7}$$

Finally, the parameter update rule of Nadam is shown in Equation (8).

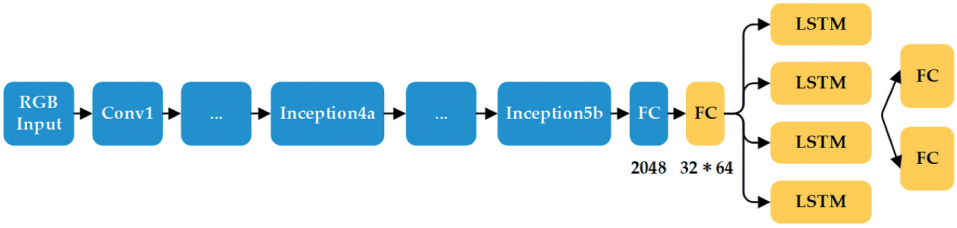



Figure 7. Our image positioning deep ConvNet.

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t} + \varepsilon}\left(\beta_1 \hat{m}_{t-1} + \frac{(1-\beta_1)g_t}{1-\beta_1^t}\right) \quad (8)$$

Obtaining the learning rate of each hyperparameter in an adaptive manner through the above improvements can effectively solve the problem of low training accuracy and high cost caused by the consistent learning rate of all hyperparameters of SGD and the problem of easily falling into local optimum.

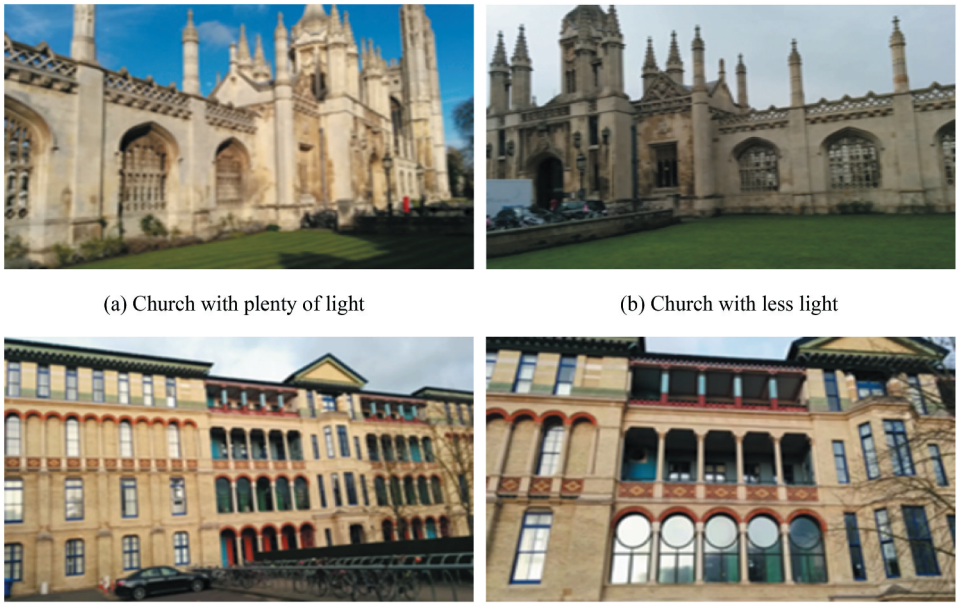
## 4. Experiments and analysis

### 4.1. Experimental data and computing environment

In order to verify the effectiveness of VNLSTM-PoseNet method proposed in this paper, we use the method to conduct experiments and compares the experimental results with that of PoseNet open-source code. This paper uses Pytorch to program the proposed new methods. In the experiment, the processor used is Intel(R) Core(TM) i7-8750 H, the memory is 8GB, and the GPU we use is GeForce GTX 1060, and the network is fine-tuned with a batch size of 75. Set the initial learning rate of 500 epochs to 0.0005, and the learning rate adjustment strategy to lamda. For the network structure, the hidden size of LSTM layer is 256. For the Nadam optimization algorithm, set $\beta_1 = 0.9$ and $\beta_2 = 0.999$. The image data used in this paper comes from Cambridge Landscape dataset. The two sets of data used are Kings College and Old Hospital. The Kings College dataset has a shooting area of 5600 $m^2$, with a total of 1220 training images and 343 test images. The Old Hospital dataset has a shooting area of 2000 $m^2$, with a total of 895 training images and 182 test images, respectively. The sample images of the datasets are shown in Figure 8. Figure 8 (a and b) are selected from the Kings College dataset while Figure 8(c and d) are selected from the Old Hospital dataset.

### 4.2. Experimental results and analysis

To compare these methods, we present results for urban streets image-based localization on the two publicly available Cambridge landmarks datasets in Table 1. Figure 9 shows the training loss of each epoch during the network training process. The figure on the left shows the change of position error with epoch, and the figure on the right shows the change of orientation error with epoch. It can be seen from the Figure 9 that although the training loss fluctuates in a small range during the training process, the overall trend is to steadily decrease and reach a relatively stable value around 500 epochs, so we set the final training epoch to 500. The five different colors represent PoseNet and the four improved methods proposed in this paper: Bv-PoseNet, LSTM-PoseNet, Nadam-PoseNet and VNLSTM-PoseNet. It can be seen that the improved method proposed in this paper has reduced loss compared with the original PoseNet. Especially, the training loss of VNLSTM-PoseNet is greatly reduced compared with PoseNet, it is also the final method proposed in this paper. The other three methods are single improvements in method of VNLSTM-PoseNet, their experiments were used to verify the effectiveness of each individual improvement. When the network is stable, the



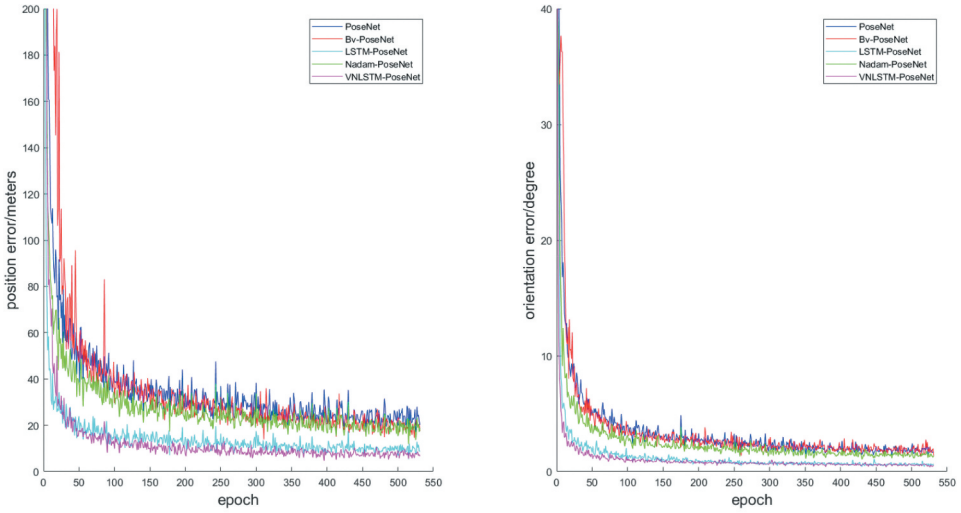
(a) Church with plenty of light (b) Church with less light

(c) Big view of the building (d) Close-up view of building

**Figure 8.** Sample images of datasets. (a) Church with plenty of light (b) Church with less light (c) Big view of the building (d) Close-up view of building.

**Table 1.** Localization results of several methods.

| | | PoseNet | Bv-PoseNet | LSTM-PoseNet | Nadam-PoseNet | VNLSTM-PoseNet |
|---|---|---|---|---|---|---|
| Kings College | Position(m) | 2.86 | 2.83 | 2.42 | 1.91 | 1.71 |
| | Orientation(°) | 6.24 | 6.14 | 6.11 | 5.03 | 4.77 |
| Old Hospital | Position(m) | 2.20 | 2.12 | 1.85 | 1.00 | 0.89 |
| | Orientation(°) | 5.05 | 4.94 | 4.71 | 4.47 | 3.83 |




(a) Position error of different epoch (b) Orientation error of different epoch

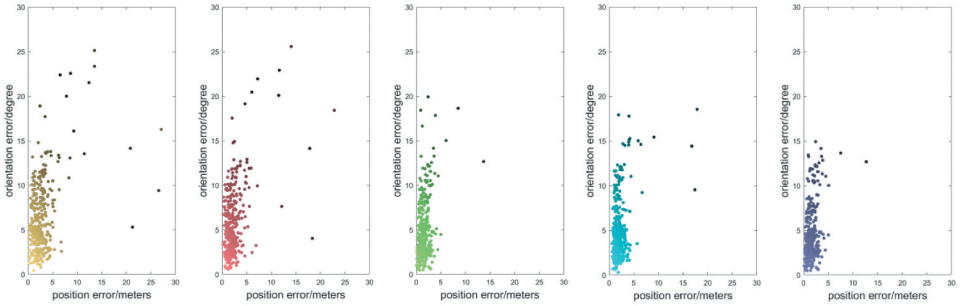
**Figure 9.** Training loss over different epochs. (a) Position error of different epoch (b) Orientation error of different epoch.

position error and orientation error are reduced by about 15 m and 2 degrees, respectively.

In order to compare the five methods fairly, we provide the average position accuracy and average orientation accuracy results of the five methods for positioning the two datasets, as shown in Table 1. The average position and orientation accuracy of the original PoseNet is about 2.5 m and 6 degrees respectively. The average accuracy of the VNLSTM-PoseNet method proposed in this paper is significantly higher than that of PoseNet. The position accuracy and orientation accuracy are improved by about 1.2 m and 1.5 degrees while the error is reduced by about 48% and 24%.

Figure 10 is the scatter plot of the position and orientation errors after using five methods to locate the query images. The horizontal axis is the position error in meters, and the vertical axis is the orientation error with the unit of degree. Figure 10(a) is the result of PoseNet, Figure 10(b) is the result of Bv-PoseNet with improved image preprocessing method, Figure 10(c) is the result of LSTM-PoseNet with improved LSTM-based network structure, Figure 10 (d) is the result of Nadam-PoseNet using the Nadam optimizer, and Figure 10(e) is the result of the finally positioning method VNLSTM-PoseNet proposed in this paper.

It can be seen from the Figure 10 that the error point of the improved method in this paper is closer to the origin point, that is, the position error and the orientation error are smaller, which shows that the method in this paper has significantly improved the



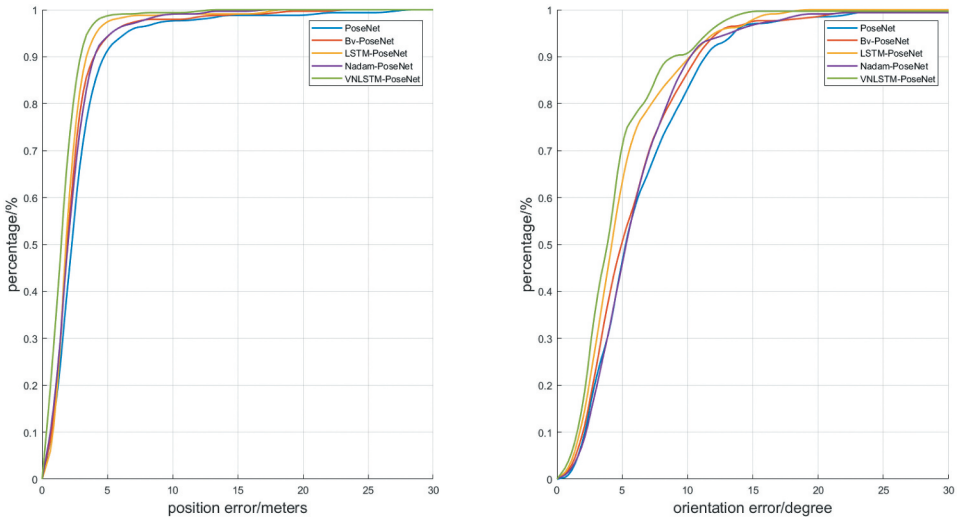

(a) PoseNet (b) Bv-PoseNet (c) LSTM-PoseNet (d) Nadam-PoseNet (e) Our method

**Figure 10.** Error scatter plot of query image localization. (a) PoseNet (b) Bv-PoseNet (c) LSTM-PoseNet (d) Nadam-PoseNet (e) Our method.

accuracy and robustness of the PoseNet method. Especially for the images difficult to position, PoseNet method has many images with large positioning errors. These position errors and orientation errors can reach about 20–30 m and 20–30 degrees, respectively, and the number of such images is large. The position errors of images positioned by Bv-PoseNet method is less than 25 m. The position errors of LSTM-PoseNet are greatly improved compared with PoseNet as well. The position errors of most images are within 10 m and the orientation errors are within 20 degrees. Nadam-PoseNet has only a few images whose position errors are between 15 m to 20 m but the orientation errors are within 18 degrees. For VNLSTM-PoseNet, the maximum position errors and orientation errors are only slightly larger than 15 m and 15 degrees while the number of such images is very small, and the position accuracy and orientation accuracy are greatly improved compared with PoseNet.

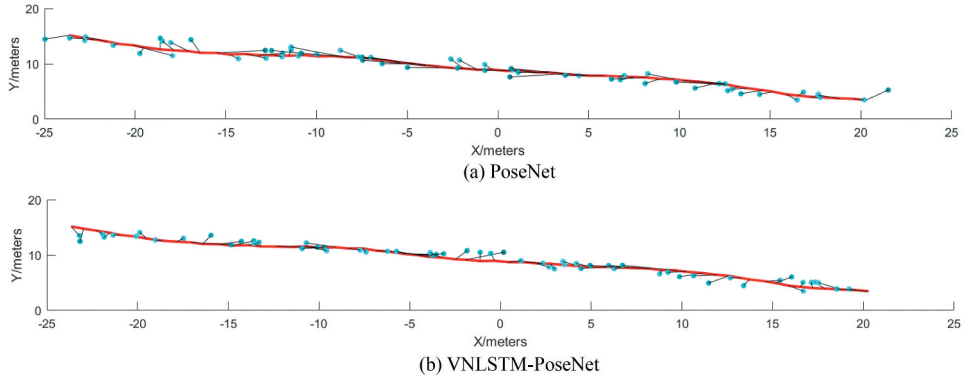
Figure 11 is the cumulative error histogram of the Kings College test dataset showing the positioning performance of the five methods from a more quantitative and intuitive perspective. Figure 11(a) shows the position error, and Figure 11(b) shows the orientation error. Generally speaking, compared with PoseNet algorithm, the method proposed in this paper is more competitive. From the perspective of position error, PoseNet has about 9% of images whose position errors are within 1 m, Bv-PoseNet and LSTM-PoseNet have about 12%, Nadam-PoseNet has about 14%, and VNLSTM-PoseNet has reached 30%. For images with position errors within 2 m, the percentages of the five methods are 42%, 54%, 60%, 52%, and 72%, respectively. Bv-PoseNet, LSTM-PoseNet, Nadam-PoseNet have 12%, 18%, and 10% more images with position errors within 2 m than PoseNet, and the corresponding percentage of the high-precision positioning network VNLSTM-PoseNet proposed in this paper is 30%. Similarly, when the position error is 5 m, the corresponding percentages are 91%, 94%, 97%, 94%, 98%. It can be obtained that the position accuracy of the method proposed in this paper is greatly improved compared with PoseNet. For images that are difficult to locate (position error greater than 10 m), the method in this paper also has a better positioning effect. The percentages of the five methods for this type of images are 97%, 98%, 98%, 98%, and 99%. For the orientation error, the percentages of the orientation errors within 5 degrees of the five methods are 47%, 51%, 64%, 47%, and 73%, respectively. The percentage of VNLSTM-PoseNet's orientation error within 5 degrees is 26% higher than that of PoseNet, that is, the orientation accuracy is significantly improved. When the orientation error is 15 degrees, the percentages corresponding to the five methods are 96%, 97%, 97%, 96%, 100%, that is, the orientation errors of all images of VNLSTM-PoseNet are within 15 degrees, but at this time in the PoseNet method, there are still many images whose orientation errors are greater than 15 degrees.

Figure 12 shows the positional relationship between the estimated camera pose and the real camera pose. The red line is the real trajectory of the image sequence, and the blue point represents the camera pose of each frame calculated by the two experimental deep ConvNets in this paper. Use a black line to connect the estimated position with the real position of corresponding image and the length of the black line represents the difference between the calculated position and the real position. Figure 12(a) is the result of PoseNet, and Figure 12(b) is the result of VNLSTM-PoseNet. On the whole, the black line of the VNLSTM-PoseNet method is shorter than that of




(a) Position error (b) Orientation error

**Figure 11.** Cumulative histogram of localization error. (a) Position error (b) Orientation error.




(a) PoseNet




(b) VNLSTM-PoseNet

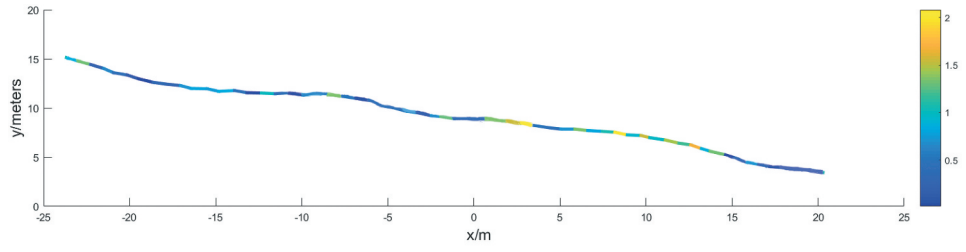
**Figure 12.** Comparison of real pose and calculated pose. (a) PoseNet (b) VNLSTM-PoseNet.

PoseNet, that is, the positioning accuracy of the method in this paper is significantly improved compared with PoseNet. In particular, in the area of the abscissa from 0 to 5 m, that is, the front door area in the dataset Kings College, there are many vehicle occlusions, which affect the image positioning, resulting in uneven distribution of estimated positions; thus, clusters are formed near the part of the trajectory, which indicates that there is a positional deviation. But the VNLSTM-PoseNet method proposed in this paper improves the problem of position distribution. The black line is shorter, that is, the estimated camera pose is closer to the real camera pose. These differences are more easily observed in the visualization results in Figure 12.

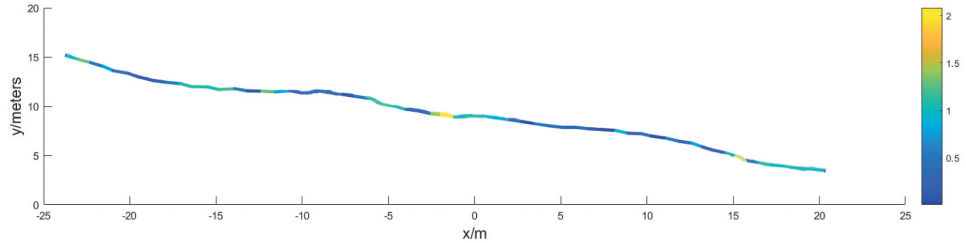
Figure 13 is a visualization map of the errors of each part of the predicted trajectory, the color represents the magnitude of the errors. It shows the positioning accuracy of different areas more intuitively. The trajectory is the true ground trajectory calculated by SfM method. The difference between the trajectory calculated by different networks and the real trajectory is drawn from dark blue to yellow, where the color provides a measure of error. Among them, Figure 13 (a) is the result of PoseNet, and Figure 13(b) is the result of VNLSTM-PoseNet. Overall, every part of VNLSTM-PoseNet trajectory shows more blue and green color than PoseNet, that is, higher accuracy, while PoseNet has more yellow trajectories, that is, the errors of these parts are large. From the perspective of trajectory, the overall position error is relatively large in the range of x-coordinates 0 to 5 m and 10 to 15 m. This is due to the positioning difficulties caused by the presence of vehicles and vegetation in this part. However, the method in this paper reduces the influence of these occlusions on the positioning to a certain extent, which shows that they have higher accuracy in these parts.

## 5. Discussion

According to the above experimental results, the positioning accuracy of the four improved methods proposed in this paper are all higher than that of PoseNet.




(a) PoseNet




(b) VNLSTM-PoseNet

**Figure 13.** Visualizing error for the different parts of the trajectory. (a) PoseNet (b) VNLSTM-PoseNet.

Among them, Bv-PoseNet can obtain more image feature information by increasing the receptive field of the image, thereby improving the success rate and accuracy of image positioning. For LSTM-PoseNet, by introducing the LSTM structure to perform structural dimensionality reduction on the fully connected layer, the most useful feature correlation can be selected for pose estimation, and the Nadam optimizer in Nadam-PoseNet is conducive to selecting more suitable network hyperparameters to improve the accuracy of image positioning. The VNLSTM-PoseNet, which integrates the three methods, has a significant improvement in image positioning accuracy compared to PoseNet. In addition, VNLSTM-PoseNet method proposed in this paper has fewer images that are difficult to locate, that is, the images whose position accuracy and orientation accuracy are significantly greater than other images. Judging from the actual scene, these difficult-to-locate images are mostly images with sudden changes in the shooting direction, images blocked by weakly textured objects or images with dark light. For images with sudden shooting direction changes, as shown in Figure 14(a and b), this is because in actual shooting, when the track direction suddenly changes, the image definition is poor due to motion blur, and the corresponding image quality is poor, which affects the positioning accuracy. However, our VNLSTM-PoseNet network reduces the impact of motion blur on the positioning results to a certain extent. The specific performance is that the images difficult to position are less than PoseNet and the accuracy is relatively higher. At the same time, there are some vegetation or vehicle occlusions in the dataset. Due to the weak texture of vegetation and vehicles or repetitive texture structure in the entire space, it is easy to cause ambiguity when these features are used as positioning features and cause positioning failure, as shown in Figure 14(c–f). The texture of the occluder is weak and repetitive, which is a challenge for image positioning. In this case, PoseNet

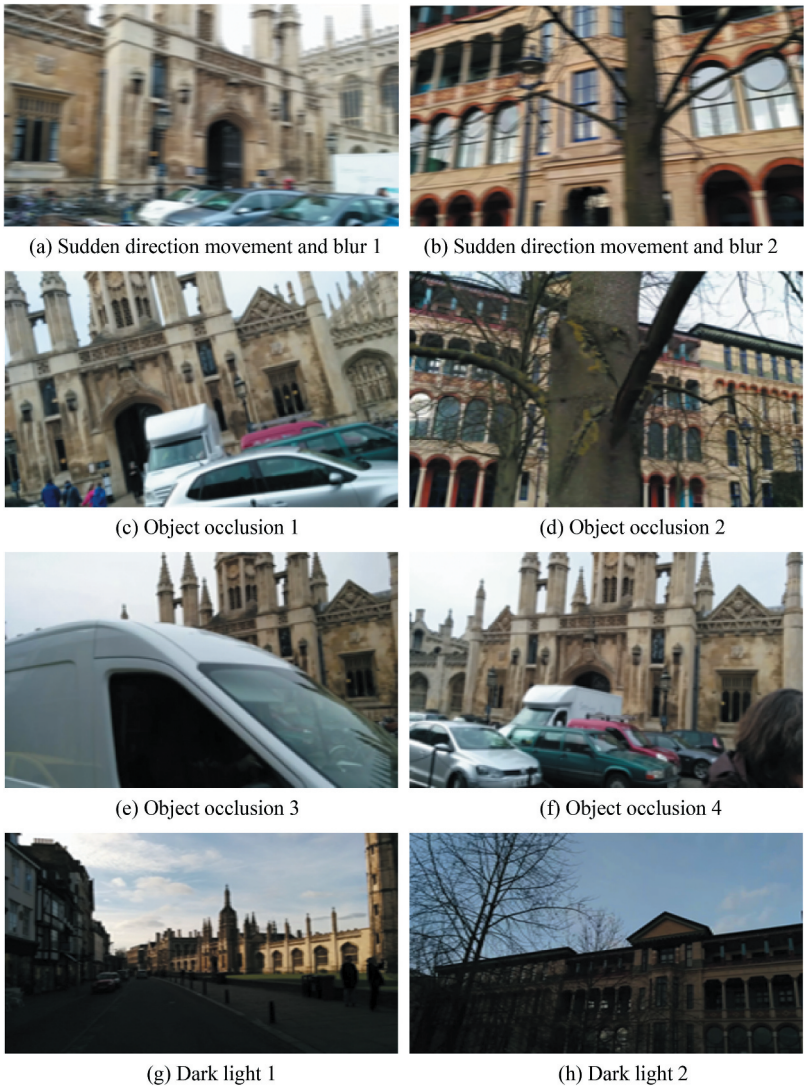



**Figure 14.** Example of query images difficulty to position. (a) Sudden direction movement and blur 1 (b) Sudden direction movement and blur 2 (c) Object occlusion 1 (d) Object occlusion 2 (e) Object occlusion 3 (f) Object occlusion 4 (g) Dark light 1 (h) Dark light 2.

cannot calculate the accurate pose. However, the method proposed in this paper can more effectively use high-level features such as contours for positioning to compensate for the impact of texture loss on the positioning results, it can better calculate camera pose. Especially after improving the image preprocessing method, as the receptive field becomes larger, more image feature information can be obtained, so the calculated pose is more accurate. In addition, the weak texture caused by the dark light also brings challenges to the positioning, as shown in Figure 14 (g and h). Based on the above analysis, the VNLSTM-PoseNet method proposed in this paper has better robustness to situations, such as the motion blur, the weak texture, the illumination variation, and the occlusions. It can effectively use high-dimensional features for image positioning, and can better adapt to challenging situations in real environments.

On the other hand, from the perspective of running time, it takes about 52 ms to locate an image using PoseNet with the environment of this paper, and it takes about 58 ms to locate an image using VNLSTM-PoseNet method with the same environment, which far meets the real-time requirements.

## 6. Conclusions

In this paper, we presented a new deep ConvNet learning architecture that address the big challenge of image-based camera relocalization in urban streets from only RGB images. Rather than precomputing local feature points and building a 3D photo-realistic map as done in traditional matching-based relocalization techniques. In our proposed VNLSTM-PoseNet method, firstly, the training images can obtain a larger receptive filed by adopting a new clipping method, which can get more key image information to enhance positioning. Then, in order to obtain more suitable deep ConvNet hyperparameters, the Nadam optimizer is used to optimize the network based on the Pytorch framework. At last, the LSTM structure is introduced into the PoseNet network to perform structural dimensionality reduction on the fully connected layer and select the most useful relevant features for real-time camera pose regression. With a systematic evaluation on the two existing outdoor datasets through experiments, we show that VNLSTM-PoseNet can lead to drastic improvements in positioning performance compared to other PoseNet-based methods, and achieving a localization accuracy of approximately less than 0.9 m in the dataset of Old Hospital. To our knowledge and analysis, the localization errors are mainly caused by the motion blur, the texture-less, the illumination variation, and the occlusions. We demonstrate that our approach succeeds in these challenging scenarios where the other CNN-based methods perform less well.

To this end, that is no doubt that exploring CNN-based camera relocalization in hard scenarios is a promising research direction. Besides aiming to close the gap in accuracy between local feature matching-based image localization, it has a vast advantage with robustness and efficiency. Of course, the localization errors definitely can be affected by those challenging scenarios. Alternatively, the errors could be an effect of the features learnt by the deep ConvNet for localization. In future work, we will conduct more in-depth research and exploration on the correlation of these problems, and introduce more constraints and information to improve the accuracy of camera pose regression based on convolutional neural network.

## ORCID

Ruizhi Chen http://orcid.org/0000-0001-6683-2342

## Data availability statement

The data used to support the findings of this study are available from the corresponding author upon request. http://mi.eng.cam.ac.uk/projects/relocalisation/

## ORCID

Ruizhi Chen http://orcid.org/0000-0001-6683-2342

## Notes on contributors

*Ming Li* is an associate professor of Wuhan University and a postdoctoral research fellow of ETH Zürich. His main research interests are the principles and methods of machine learning, photogrammetric computer vision, robotics, and underwater photogrammetry and remote sensing.

*Jiangying Qin* is a master of Wuhan University. Her main interests are machine learning and photogrammetric computer vision, especially in indoor positioning and navigation based on geometry and machine learning.

*Deren Li* is an academician of the Chinese Academy of Sciences and the Chinese Academy of engineering. The main research contents are the theoretical innovation, integrated innovation and collaborative innovation of geospatial informatics.

*Ruizhi Chen* is a professor of Wuhan University, and his research interests include ubiquitous positioning of smart phones and satellite navigation.

*Xuan Liao* received her master's degree from Wuhan University of Photogrammetry and Remote Sensing in 2020. She is currently a research assistant and doctoral student at Hong Kong Polytechnic University. Her current research focuses on global solar computing based on remote sensing and space information technology, deep learning, and change detection.

*Bingxuan Guo* is a professor of Wuhan University, mainly engaged in digital photogrammetry, computer vision,

graphics and imaging, indoor positioning and artificial intelligence.

## Funding

This work is supported by the National Key R&D Program of China [grant number 2018YFB0505400], the National Natural Science Foundation of China (NSFC) [grant number 41901407], the LIESMARS Special Research Funding [grant number 2021] and the College Students' Innovative Entrepreneurial Training Plan Program [grant number S2020634016].

## References

Acharya, D., M. Ramezani, K. Khoshelham, and S. Winter. 2019a. "BIM-Tracker: A Model-based Visual Tracking Approach for Indoor Localisation Using A 3D Building Model." *ISPRS Journal of Photogrammetry and Remote Sensing* 150: 157–171. doi:10.1016/j.isprsjprs.2019.02.014.

Acharya, D., S. Roy, K. Khoshelham, and S. Winter. 2019b. "Modelling Uncertainty of Single Image Indoor Localisation Using a 3D Model and Deep Learning." *ISPRS Journal of Photogrammetry and Remote Sensing*. doi:10.1016/j.isprsjprs.2019.02.020.

Asadi, K., H. Ramshankar, M. Noghabaei, and K. Han. 2019. "Real-Time Image Localization and Registration with BIM Using Perspective Alignment for Indoor Monitoring of Construction". *Journal of Computing in Civil Engineering* 33: 5. doi:10.1061/(ASCE)CP.1943-5487.0000847.

Bay, H., A. Ess, T. Tuytelaars, and L. Van Gool. 2008. "Speeded-up Robust Features (SURF)." *Computer Vision and Image Understanding* 110 (3): 346–359. DOI:10.1016/j.cviu.2007.09.014.

Behera, R.K., D. Naik, S.K. Rath, and R. Dharavath. 2020. "Genetic Algorithm-based Community Detection in Large-scale Social Networks." *Neural Computing and Applications* 32 (13): 9649–9665. DOI:10.1007/s00521-019-04487-0.

Brachmann, E., A. Krull, S. Nowozin, E Brachmann, A Krull, S Nowozin, J. Shotton, F. Michel, S. Gumhold, and C. Rother. 2017. "Dsac-differentiable Ransac for Camera Localization." In IEEE Conference on Computer Vision and Pattern Recognition, Hawaii, July 21- 26.

Brachmann, E., F. Michel, A. Krull, M.Y. Yang, and S. Gumhold. 2016. "Uncertainty-driven 6D Pose Estimation of Objects and Scenes from a Single RGB Image." In IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, June 27–30.

Cavallari, T., S. Golodetz, N.A. Lord, J. Valentin, L. Di Stefano, and P.H. Torr. 2017. "On-the-fly Adaptation of Regression Forests for Online Camera Relocalisation." In IEEE Conference on Computer Vision and Pattern Recognition, Hawaii, July 21–26.

Chum, O., and J. Matas. 2005. "Matching with PROSAC-progressive Sample Consensus." In IEEE computer society conference on computer vision and pattern recognition, San Diego, June 20–26.

Deng, J., W. Dong, R. Socher, L.J. Li, K. Li, and L. Fei-Fei. 2009. "ImageNet: A Large-scale Hierarchical Image Database." IEEE Conference on Computer Vision and Pattern Recognition, Miami, June 20-25.

Dozat, T. 2016. "Incorporating Nesterov Momentum into Adam." In International Conference of Legal Regulators, San Juan,May 2–4.

Fischler, M.A, and R.C. Bolles. 1981. "Random Sample Consensus: A Para-digm for Model Fitting with Applications to Image Analysis and Automated Cartography." *Communications of the ACM* 24 (6): 381–395. doi:10.1145/358669.358692.

Glocker, B., J. Shotton, A. Criminisi, and S. Izadi. 2015. "Real-time Rgb-d Camera Relocalization via Randomized Ferns for Keyframe Encoding." *IEEE Transactions on Visualization and Computer Graphics* 21 (5): 571–583. doi:10.1109/TVCG.2014.2360403.

Guzman-Rivera, A., P. Kohli, B. Glocker, J. Shotton, T. Sharp, A. Fitzgibbon, and S. Izadi. 2014. "Multi-output Learning for Camera Relocalization." IEEE Conference on Computer Vision and Pattern Recognition, Columbus,June 23- 28.

Han, X.F., H. Laga, and M. Bennamoun. 2019. "Image-based 3D Object Reconstruction: State-of-the-Art and Trends in the Deep Learning Era." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 43 (5): 1578–1604. doi:10.1109/TPAMI.2019.2954885.

Hesch, J.A., and S.I. Roumeliotis. 2011. "A Direct Least-Squares (DLS) Method for PnP." In International Conference on Computer Vision, Barcelona, November 6–13.

Hu, Z., J. Tang, Z. Wang, K. Zhang, L. Zhang, and Q. Sun. 2018. "Deep Learning for Image-based Cancer Detection and Diagnosis – A Survey". *Pattern Recognition* 83: 134–149. doi:10.1016/j.patcog.2018.05.014.

Husain, S.S., and M. Bober. 2019. "REMAP: Multi-layer Entropy-guided Pooling of Dense CNN Features for Image Retrieval." *IEEE Transactions on Image Processing* 28 (10): 5201–5213. doi:10.1109/TIP.2019.2917234.

Jin, Y., L. Yu, G. Li, S. Fei. 2021. "A 6-DOFs Event-based Camera Relocalization System by CNN-LSTM and Image Denoising". *Expert Systems with Applications* 170: 114535. doi:10.1016/j.eswa.2020.114535.

Kendall, A., and R. Cipolla. 2016. "Modelling Uncertainty in Deep Learning for Camera Relocalization." In IEEE international conference on Robotics and Automation, Stockholm, May 16–21.

Kendall, A., and R. Cipolla. 2017. "Geometric Loss Functions for Camera Pose Regression with Deep Learning." In IEEE Conference on Computer Vision and Pattern Recognition, Hawaii, July 21–26.

Kendall, A., M. Grimes, and R. Cipolla. 2015. "Posenet: A Convolutional Network for Real-time 6-dof Camera Relocalization." In The IEEE international conference on computer vision, Boston, June 7–13.

Khoshelham, K., and S. Winter. 2019. "BIM-PoseNet: Indoor Camera Localisation Using a 3D Indoor Model and Deep Learning from Synthetic Images." *ISPRS Journal of Photogrammetry and Remote Sensing* 150: 245–258. doi:10.1016/j.isprsjprs.2019.02.020.

Kosowski, G., T. Rymarczyk, D. Wojcik, T. Cieplak, and P. Adamkiewicz. 2020. "Effect of Features Extraction on Improving LSTM Network Quality in ECG Signal Classification." *Przeglad Elektrotechniczny* 12 (12): 194–197.

Lepetit, V., F. Moreno-Noguer, and P. Fua. 2009. "EPnP: An AccurateO(n) Solution to the PnP Problem." *International Journal of Computer Vision* 81 (2): 155–166. doi:10.1007/s11263-008-0152-6.

Li, M., R. Chen, X. Liao, B. Guo, W. Zhang, and G. Guo. 2020a. "A Precise Indoor Visual Positioning Approach Using A Built Image Feature Database and Single User

Image from Smartphone Cameras." *Remote Sensing* 12 (5): 869–894. doi:10.3390/rs12050869.

Li, P., L. Han, X. Tao, X. Zhang, C. Grecos, A. Plaza, P. Ren, et al. 2020b. "Hashing Nets for Hashing: A Quantized Deep Learning to Hash Framework for Remote Sensing Image Retrieval." *IEEE Transactions on Geoscience and Remote Sensing* 58 (10): 7331–7345. DOI:10.1109/TGRS.2020.2981997.

Liu, X., S.J. Dyke, C.M. Yeum, I. Bilionis, A. Lenjani, and J. Choi. 2020. "Automated Indoor Image Localization to Support a Post- Event Building Assessment." *Sensors* 20 (6): 1610–1628. DOI:10.3390/s20061610.

Lock, O., and C. Pettit. 2020. "Social Media as Passive Geo-participation in Transportation Planning–how Effective are Topic Modeling & Sentiment Analysis in Comparison with Citizen Surveys?" *Geo-spatial Information Science* 23 (4): 275–292. doi:10.1080/10095020.2020.1815596.

Lowe, D.G. 2004. "Distinctive Image Features from Scale-invariant Keypoints." *International Journal of Computer Vision* 60 (2): 91–110. doi:10.1023/B:VISI.0000029664.99615.94.

Melekhov, I., J. Ylioinas, J. Kannala, and Rahtu E. 2017. "Image-based Localization Using Hourglass Networks." In IEEE international conference on computer vision, Venice, October 22–29.

Meng, L., J. Chen, F. Tung, J.J. Little, and C.W. de Silva 2016. "Exploiting Random Rgb and Sparse Features for Camera Pose Estimation." In Proceedings of the British Machine Vision Conference, York, September 19–22.

Miao, R., P. Liu, Z. Gong, W. Xue, X. Ji, and R. Ying. 2021. "Adaptive Stereo Direct Visual Odometry with Real-Time Loop Closure Detection and Relocalization." In IEEE International Symposium on Circuits and Systems, Daegu, Korea, May 23–26.

Ming, C., S. Chunhua, and L. Reid. 2018. "A Hybrid Probabilistic Model for Camera Relocalization." *The British Machine Vision Conference, Newcastle upon Tyne*, September 1–12.

Mur-Artal, R., J.M.M. Montiel, and J.D. Tardos. 2015. "ORB-SLAM: A Versatile and Accurate Monocular SLAM System." *IEEE Transactions on Robotics* 31 (5): 1147–1163. doi:10.1109/TRO.2015.2463671.

Mur-Artal, R., and J.D. Tardos. 2017. "ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo and RGB-D Cameras." *IEEE Transactions on Robotics* 33 (5): 1255–1262. doi:10.1109/TRO.2017.2705103.

Nguyen, A., T.T. Do, D.G. Caldwell, and N. G. Tsagarakis. 2017. "Real-Time 6DOF Pose Relocalization for Event Cameras with Stacked Spatial LSTM Networks." In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, Hawaii, July 21–26.

Nister, D., and H. Stewenius. 2006. "Scalable Recognition with a Vocabulary Tree." In IEEE Computer Society Conference on Computer Vision and Pattern Recognition, New York, June 17–22.

Niu, Q., M. Li, S. He, C. Gao, S H. Gary Chan, and X. Luo. 2019. "Resource-efficient and Automated Image-based Indoor Localization." *ACM Transactions on Sensor Networks* 15 (2): 1–31. doi:10.1145/3284555.

Prins, A.J., and A. Van Niekerk. 2021. "Crop Type Mapping Using LiDAR, Sentinel-2 and Aerial Imagery with Machine Learning Algorithms." *Geo-spatial Information Science* 24 (2): 215–227. doi:10.1080/10095020.2020.1782776.

Qin, J., M. Li, X. Liao, and J. Zhong. 2019. "Accumulative Errors Optimization for Visual Odometry of ORB-SLAM2 Based on RGB-D Cameras." *ISPRS International Journal of Geo-Information* 8 (12): 581–600. DOI:10.3390/ijgi8120581.

Rublee, E., V. Rabaud, K. Konolige, and G. Bradski. 2011. "ORB: An Efficient Alternative to SIFT or SURF." In International conference on computer vision, Barcelona, November 6–13.

Sattler, T., B. Leibe, and L. Kobbelt. 2011. "Fast Image-based Localization Using Direct 2d-3d Matching." In International Conference on Computer Vision, Barcelona, November 6–13.

Sattler, T., B. Leibe, and L. Kobbelt. 2016. "Efficient & Effective Prioritized Matching for Large-scale Image-based Localization." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39 (9): 1744–1756. doi:10.1109/TPAMI.2016.2611662.

Seifi, S., and T. Tuytelaars. 2019. "How to Improve CNN-Based 6-DoF Camera Pose Estimation." In IEEE/CVF International Conference on Computer Vision Workshop, Seoul, October 27–28.

Shi, X., Z. Chen, H. Wang, D. Y. Yeung, W.K. Wong, and W. C. Woo. 2015. "Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting." *International Conference on Neural Information Processing Systems*, Montreal, December 7–12.

Shotton, J., B. Glocker, C. Zach, S. Izadi, A. Criminisi, and A. Fitzgibbon. 2013. "Scene Coordinate Regression Forests for Camera Relocalization in RGB-D Images." In IEEE Conference on Computer Vision and Pattern Recognition, Portland, June 23–38.

Shukla, S., R.K. Behera, S. Misra, and S.K. Rath. 2018. "Software Reliability Assessment Using Deep Learning Technique." *Towards Extensible and Adaptable Methods in Computing* 5 (1): 57–68.

Silpa-Anan, C., and R. Hartley. 2008. "Optimised Kd-trees for Fast Image Descriptor Matching."In IEEE Conference on Computer Vision and Pattern Recognition, Anchorage, June 23–28.

Singh, R., R. Khurana, A.K.S. Kushwaha, and R. Srivastava. 2020. "Combining CNN Streams of Dynamic Image and Depth Data for Action Recognition." *Multimedia Systems* 26 (5): 1–10.

Szegedy, C., W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. 2015. "Going Deeper with Convolutions." In IEEE Conference on Computer Vision and Pattern Recognition, Boston, June 8–10.

Tateno, K., F. Tombari, I. Laina, and N. Navab. 2017. "Cnn-slam: Real-time Dense Monocular Slam with Learned Depth Prediction." In The IEEE Conference on Computer Vision and Pattern Recognition, Hawaii, July 21–26.

Valada, A., N. Radwan, and W. Burgard. 2018. "Incorporating Semantic and Geometric Priors in Deep Pose Regression." In Workshop on learning and inference in robotics: Integrating structure, priors and models at robotics: Science and systems, Pennsylvania, June 26–30.

Valentin, J., M. Niebner, J. Shotton, and P. Torr. 2015. "Exploiting Uncertainty in Regression Forests for Accurate Camera Relocalization." In IEEE Conference on Computer Vision and Pattern Recognition, Boston, June 8–10.

Walch, F., C. Hazirbas, L. Leal-Taixe, T. Sattler, S. Hilsenbeck, and D. Cremers 2017. "Image-based Localization Using LSTMs for Structured Feature Correlation." In IEEE International Conference on Computer Vision, Venice, October 22–29.

Wu, J., L. Ma, and X. Hu. 2017. "Delving Deeper into Convolutional Neural Networks for Camera Relocalization." In IEEE international conference on Robotics and Automation, Singapore, May 29–June 3.

Zhang, M., W. Li, and Q. Du. 2018. "Diverse Region-based CNN for Hyperspectral Image Classification." *IEEE Transactions on Image Processing* 27 (6): 2623–2634. doi:10.1109/TIP.2018.2809606.

Zinkevich, M., M. Weimer, A.J. Smola, and L. Li. 2011. "Parallelized Stochastic Gradient Descent." In Conference on Neural Information Processing Systems, Granada, December 12–14.