




# Indoor Visual Positioning Based on Image Retrieval in Dense Connected Convolutional Network

Xiaomeng Guo, Danyang Qin<sup>(✉)</sup> , and Yan Yang

Heilongjiang University, Harbin 150080, People's Republic of China  
qindanyang@hlju.edu.cn

**Abstract.** As now available methods or systems based on image retrieval and visual researches are implemented in an indoor environment, their retrieval accuracy and real-time positioning still have their own limitations. For this reason, this paper designs a visual indoor positioning system based on densely connected convolutional network image retrieval. Combine visual positioning with DenseNet-based image retrieval method. The problem of excessively deep network layers caused by the original convolutional neural network in pursuit of high retrieval accuracy is improved. Under the advantage of ensuring the high accuracy of image retrieval based on depth features, the problem of low real-time positioning caused by the long training time of the convolutional network model is improved. The simulation results show the feasibility of the positioning method in indoor environment, and the comparison experiment verifies the improvement of accuracy and speed as well as the reliability of the method.

**Keywords:** DenseNet · Feature extraction · Image retrieval · Indoor positioning

## 1 Introduction

With the increasingly prominent advantages of visual indoor positioning, it has become one of the hot spots in the research field of indoor positioning due to its low cost, high accuracy, and high real-time positioning [1]. In recent years, regarding indoor visual positioning attempts, Kohoutek et al. used the CityGML, an internal architectural model with digital spatial semantics, to determine the location and direction of the distance imaging camera [2] at the highest level of detail (LoD 4). Hile and Borriello compared the floor plan with current cell phone images. Kitanov et al. The image taken by the camera installed on the robot is used to detect the image line and compare it with the 3D vector model [3]. The computer vision algorithm summarized by Schlaile et al.

---

This work is supported by the National Natural Science Foundation of China (61771186), University Nursing Program for Young Scholars with Creative Talents in Heilongjiang Province (UNPYSCT-2017125), Distinguished Young Scholars Fund of Heilongjiang University, and postdoctoral Research Foundation of Heilongjiang Province (LBH-Q15121), Outstanding Youth Project of Provincial Natural Science Foundation of China in 2020 (YQ2020F012).

It also depends on feature detection in the image sequence. Muffert et al. Determine the track of the omnidirectional camera according to the relative direction of the continuous image. Due to the limited indoor space, there is a high probability that the scene and the indoor layout tend to be highly similar [4]. At this time, the collected images and the extracted vector features will also be difficult to distinguish. In this case, it is easy to cause problems such as inaccurate estimation of camera parameters and mismatch of pose information. Therefore, high-precision feature information is very necessary for indoor positioning.

This paper proposes an indoor visual positioning scheme based on densely connected convolutional neural network [5] image retrieval. Through the extraction of image depth features, it combines the advantages of visual positioning and deep learning positioning, while ensuring high-precision positioning while ensuring it improves the real-time and robustness of indoor positioning [6].

## 2 System Description

The main thought of this method is to utilize the collected indoor scene images to determine the user's location through static objects. The system flowchart is shown in Fig. 1. The proposed visual indoor positioning system incorporate three portions: static image collection, image retrieval based on depth features, and user position estimation.

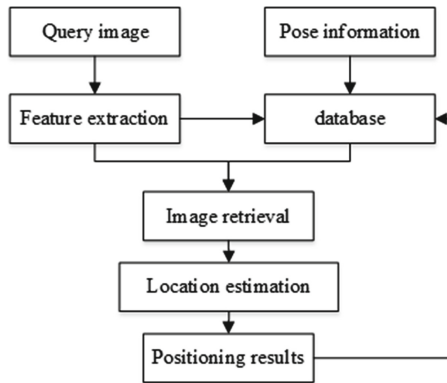


Fig. 1. Visual positioning system block diagram.

In the offline data set preparation stage, a camera is used to capture user RGB images. It contains the pose information of the captured image. The pre-trained DenseNet model extracts depth features from all collected images and sends them to the server. In the image retrieval stage, deep feature extraction is performed on images through densely connected convolutional networks, and distances are calculated for image features using metric learning methods such as Euclidean distance, and the distances of the images are sorted to obtain the primary search results, and then based on the context information of the image data and The manifold structure reorders the

image retrieval results to output the final retrieval results. The last part is the user position estimation stage, the two most similar images obtained from the image retrieval stage and the query image pose. Fuse the pose prediction with the relative distance and angle of the user. The final revised estimate and sensor deviation will be resent to the user to complete the result recall. The last two stages are completed online.

## 2.1 Resnet Structure

The main reason that the residual network works is that the residual block can easily learn the identity function. Therefore, it can be ensured that network performance will not be affected, and efficiency can even be improved in many cases. First, the residual element can be expressed as Formula 1 and 2:

$$y_l = h(x_l) + F(x_l, W_l) \quad (1)$$

$$x_{l+1} = f(y_l) \quad (2)$$

Where  $x_l$  and  $x_{l+1}$  represent the input and output of the first residual unit respectively, and each first residual unit usually contains a multilayer structure.  $F$  is the residual function, which means the residual of learning, and  $h(x_l) = x_l$  means the identity mapping, and  $F$  is the ReLU activation function. According to the above formula, the feature vector of similarity is obtained, and the calculation process is shown in Formula 3:

$$x_L = x_l + \sum_{i=1}^{L-1} F(x_i, W_i) \quad (3)$$

Using the chain rule, the gradient of the reverse process can be obtained as shown in Formula 4:

$$\frac{\partial loss}{\partial x_l} = \frac{\partial loss}{\partial x_L} \cdot \frac{\partial x_L}{\partial x_l} = \frac{\partial loss}{\partial x_L} \cdot \left( 1 + \frac{\partial}{\partial x_L} \sum_{i=1}^{L-1} F(x_i, W_i) \right) \quad (4)$$

The first factor  $\frac{\partial loss}{\partial x_L}$  represents the gradient of the loss function to  $L$ . This structure guarantees that even the worst results can get the same performance as the plain network, so as long as the residual block independently selects the working mechanism, the overall performance of the network can be improved, and the burden on the network itself is not increased. Figure 2 shows the residual structure [7]. In this case, the deep network should have at least the same performance as the shallow network without degradation.

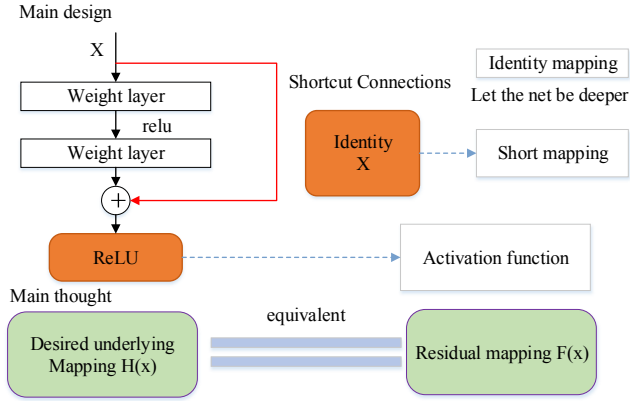


Fig. 2. Residual block principle.

### 2.2 DenseNet Network Structure

Compared with Resnet, Densenet’s design philosophy is as follows: a tighter network mechanism is created, that is, all layers are connected to each other, which means that each layer will accept the output of all previous layers as its additional input, which is the most obvious. The effect is to realize the repeated use of features, so the efficiency of the network is also significantly improved. The nonlinear transformation of DenseNet is shown in Eq. 5:

$$x_l = H_l([x_0, x_1, \dots, x_{l-1}]) \tag{5}$$

As shown in Fig. 3, when performing convolution operations on different layers, in order to ensure that the feature size is the same, the Densenet is divided into multiple Dense blocks, and the feature size in each Dense block remains the same. In addition, in order to ensure The connection between layers and the realization of downsampling function, set up transition layers between different dense blocks [8]. The problem of gradient loss is alleviated by tight connections, feature propagation is strengthened, feature maps are fully utilized, and the amount of parameters is greatly reduced.

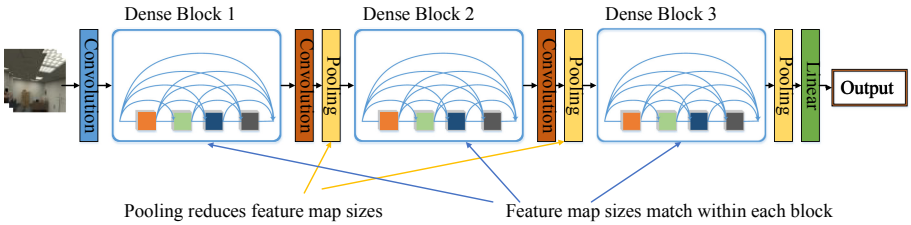


Fig. 3. Densenet network structure.

The core idea of DenseNet is to connect the network more closely, with relatively robust features, and the dependence between different layers is not too great, which further improves the accuracy of the network, and the training effect is very good. In addition, bottleneck layer, transition layer and small growth rate are used to narrow the network, reduce the parameters, effectively suppress overfitting, and reduce the calculation amount. DenseNet has many advantages, and the advantages are very clear when compared to Resnet.

### 3 Experimental Results

In this section, we will use the Pycharm simulation environment to simulate the performance of Densenet on the basis of the previous article, and then carry out the experimental test of the factors that affect the positioning accuracy, and finally evaluate the accuracy of image retrieval and indoor positioning. The research is based on densely connected convolutional network image retrieval. The visual positioning achieves the trade-off effect between positioning accuracy and real-time performance. The deep learning library uses Keras, TensorFlow, OpenCV, etc.

#### 3.1 Dataset Preparation

In order to compare with existing indoor positioning methods, we decided to choose classic open source data sets ICL-NUIM and TUM RGB-D, and select appropriate indoor scenarios from the data set as the test scenes of this experiment.

The ICL-NUIM dataset includes RGB-D images of camera trajectories from two indoor scenes. The images are collected by a handheld Kinect RGB-D camera, and Kintinuous is used to obtain the ground truth of the trajectory. The images are taken at a resolution of  $640 \times 480$  (Fig. 4).



Fig. 4. Interior scenes of the ICL-NUIM dataset.

The TUM RGB-D data set comprises the color and depth images of the Microsoft Kinect sensor and the real trajectory of the camera pose. The resolution of the image is  $640 \times 480$ . The data set consists of 89 sequences from different camera movements (Fig. 5).



Fig. 5. Interior scenes of the TUM RGB-D dataset.

### 3.2 Feature Extraction

Feature extraction refers to the extraction of more advanced features from the original pixels, these features can capture the difference between each category. This feature extraction uses an unsupervised method, and no image category labels are used when extracting information from pixels [9]. Commonly used traditional features include GIST, HOG, SIFT, LBP, etc. After feature extraction, use these features of the image and their corresponding category labels to train a classification model. Commonly used classification models include SVM, LR, random forest and decision tree.

Figure 6 shows the visualization of the depth feature vector (512 dimensions) extracted from the final convolutional layer of the Densenet-B and Densenet-C networks. The leftmost represents the vector of the input query image; the middle image represents the vector of the image with the highest score retrieved. Vector; The image on the right shows the vector of the feature vector image of the second matching image with the highest score in the same scene [10].

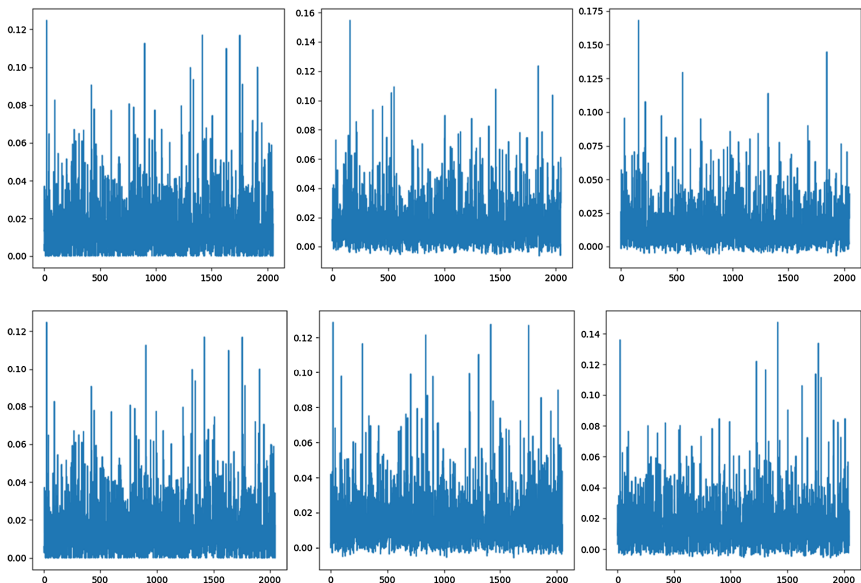


Fig. 6. Feature vector visualization.

For the CNN-based image retrieval method [11], it is mainly based on the pre-training network to extract the depth feature normalized and aggregated feature vectors to optimize the retrieval accuracy rate. Therefore, it is necessary to define an appropriate visual feature similarity measurement method, which undoubtedly has a great influence on the effect of image retrieval [12]. The accuracy of image feature similarity comparison lies in the depth feature extracted by the convolutional neural network. The public data set used in this paper is suitable for  $224 \times 224$  input. Since each training is a random sample pair, the training takes longer and the generalization is better, but it should not be too long. Iteration 3 W round, batch\_size = 128.

Euclidean distance is used to count the similarity between the feature vector  $v_c$  of the retrieved image (as shown in Formula 6) and the feature vector  $v_i$  of the query image (as shown in Formula 7). The smaller the distance, the higher the similarity. The distance between the two is shown in Formula 8:

$$V_c = [V_c^0, V_c^1, \dots, V_c^{512}] \quad (6)$$

$$V_i = [V_i^0, V_i^1, \dots, V_i^{512}] \quad (7)$$

$$D(V_c, V_i) = \sqrt{\sum_{c,i=1}^{512} (V_c - V_i)^2} \quad (8)$$

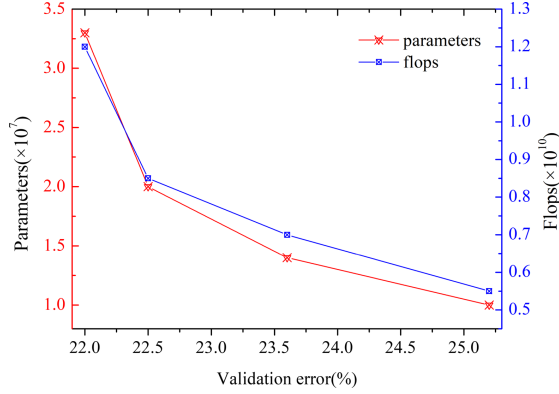
After feature extraction, image features are aggregated into fixed-length vectors. According to the current research status, similar to the general framework of most retrieval, we used the DenseNet model to extract the depth features of the input image of  $224 \times 224$ , and then mapped to the public space to obtain a unified representation and the same fixed-length feature vector. Then, the similarity is calculated according to the distance measurement method. In the final stage, the top ranked image is extracted as the retrieval result, which may be more than one image, and the most similar image [13] is found as the associated image.

### 3.3 Performance Analysis

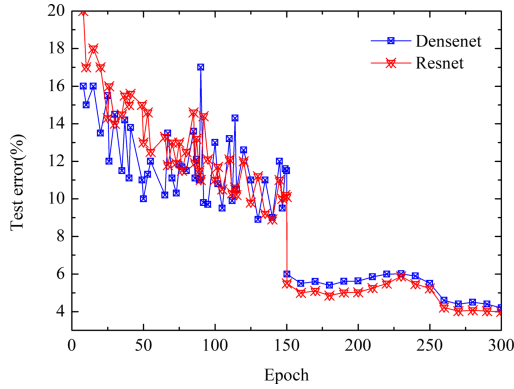
The experiment completed the positioning scheme on the Intel Corei5 6200U CPU @2.3 GHz. Figure 7 and Fig. 8 summarize the performance of the indoor positioning method based on depth feature extraction proposed in this paper.

It can be concluded that when the model implements the same test error, the parameters of the original DenseNet are usually 2–3 times more than those of the DenseNet. The figure on the right is the comparison between DenseNet and ResNet. With the same model accuracy, DenseNet only needs about one-third of the parameters of ResNet. Although they converged at about the same time they were training epochs, DenseNet needed less than a tenth of ResNet’s parameters. In the 100-layer Densenet, when  $k=12$ , the number of parameters is about 7.0 M, and when  $k=24$ , it is about 27.2 M. With the increase of  $K$  and network depth, parameter two and training difficulty also increase correspondingly.

In the offline stage, the pre-collected reference images are directly stored in the image data set, and the pose information, corresponding coordinates, and pre-extracted



**Fig. 7.** Densenet network performance.



**Fig. 8.** Comparison of dense net and resnet test errors.

depth feature vectors contained in these images are also recorded and stored. In the online phase, the query images in the test set are input into the pre-trained densenet, and the depth features extracted from the query images can be compared and analyzed with the previous features to obtain the feature vector with the highest similarity, and then output the most similar image. It also contains its location information, etc., and the coordinate information is fed back to the user. So in order to save query time, we need to pre-train the dense net network and collect coordinate and vector information. At the same recognition rate, Densenet's parameter complexity is about half that of ResNet.

For the ICL-NUIM dataset, the mean errors of the direction and translation of the method proposed in this paper are  $3.89^\circ$  and 0.446 m, and the median errors are  $0.142^\circ$  and 0.12 m. For the TUM RGB-D data set, the mean error was  $5.07^\circ$ , 0.401 m, and the median error was  $0.18^\circ$  0.141 m. For most of the indoor scenes [14] in the two data sets, our method has an average positioning success rate of more than 90% (Table 1).



**Table 1.** Localization performance in different scenarios from different datasets.

Dataset	Selected Scenario	The median error	The mean error	90% Accuracy
ICL-NUIM	Office room_1	0.097 m 0.03°	0.354 m 3.24°	0.354 m 1.37°
	Office room_2	0.142 m 0.12°	0.538 m 4.55°	0.436 m 0.89°
	All images	0.119 m 0.07°	0.446 m 3.89°	0.395 m 1.13°
TUM RGB-D	Office room_1	0.177 m 0.22°	0.435 m 4.62°	0.445 m 0.93°
	Office room_2	0.105 m 0.15°	0.367 m 5.52°	0.364 m 1.70°
	All images	0.141 m 0.18°	0.401 m 5.07°	0.405 m 1.31°

## 4 Conclusion

This paper proposes a system of assisting indoor visual positioning by image retrieval based on densely connected convolutional nets. Identify the given query image by retrieving matching database images that are marked in advance. The system combines deep learning algorithms and computer vision algorithms. Experimental results show that image retrieval based on depth features has a high retrieval accuracy and has a wide range of application potential in indoor scenes. Due to the complexity and instability of the indoor environment, the experimental results are significantly more robust than retrieval-based methods without deep feature extraction. Image retrieval based on deep features of DenseNet solves the previous plain convolutional neural network well. The retrieval time is slow and the positioning accuracy is not improved obviously due to the network being too deep, and the Resnet feature extraction is not obvious. We trade a more complex network for computational efficiency, and finally obtain high-precision posture information. It strikes a balance between positioning speed and accuracy, and can satisfy the pace requirements of indoor moving pedestrians under normal conditions. What needs to be considered in the future is a more efficient and robust method to be suitable for complex large-scale indoor scenes.

## References

1. Li, F., et al.: A reliable and accurate indoor localization method using phone inertial sensors. In: Proceedings of the 2012 ACM Conference on Ubiquitous Computing, pp. 421–430 (2012)
2. Ali, H.M., Omran, A.H.: Floor identification using smart phone barometer sensor for indoor positioning. *Int. J. Eng. Sci. Res. Technol.* **4**(2), 384–391 (2015)
3. Carrillo, D., Moreno, V., Beda, B., Skarmeta, A.F.: MagicFinger: 3D magnetic fingerprints for indoor location. *Sensors* **15**(7), 17168–17194 (2015)
4. Liu, J., Chen, R., Pei, L., Guinness, R., Kuusniemi, H.: A hybrid smartphone indoor positioning solution for mobile LBS. *Sensors* **2012**(12), 17208–17233 (2012)
5. Khan, S., Hayat, M., Bennamoun, M., Sohel, F., Togneri, R.: A discriminative representation of convolutional features for indoor scene recognition. *IEEE Trans. Image Process.* **25**, 3372–3383 (2016)
6. Xiao, A., Chen, R., Li, D., Chen, Y., Wu, D.: Indoor positioning system based on static objects in large indoor scenes by using smartphone cameras. *Sensors* **18**, 2229 (2018)

7. He, K.M., Zhang, X.Y., Ren, S.Q., et al.: Deep residual learning for image recognition. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
8. Huang, G, Liu, Z, Weinberger, K.Q, et al.: Densely connected convolutional networks. arXiv preprint [arXiv:1608.06993](https://arxiv.org/abs/1608.06993) (2016)
9. Gong, Y., Wang, L., Guo, R., Lazebnik, S.: Multi-scale orderless pooling of deep convolutional activation features. In: Proceedings of the European Conference on Computer Vision, 6–12 September 2014, Zurich, Switzerland, pp. 392–407 (2014)
10. Toliás, G., Avrithis, Y.: Image search with selective match kernels: aggregation across single and multiple images. *Int. J. Comput. Vis.* **116**, 262 (2016)
11. Sattler, T., Leibe, B., Kobbelt, L.: Efficient & effective prioritized matching for large-scale image-based localization. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**(39), 1744–1756 (2017)
12. Zakaria, L., Iaroslav, M., Surya, K., Juho, K. Camera relocalization by computing pairwise relative poses using convolutional neural network. In: Proceedings of the IEEE International Conference on Computer Vision Workshop, Venice, Italy, October, pp. 22–29, 920–929 (2017)
13. Gordo, A., Almazán, J., Revaud, J., Larlus, D.: Deep image retrieval: learning global representations for image search. In: Proceedings of the European Conference on Computer Vision, Amsterdam, pp. 241–257. The Netherlands, 11–14 October (2016)
14. Liang, J.Z., Corso, N., Turner, E., Zakhor, A.: Image based localization in indoor environments. In: proceedings of the Fourth International Conference on Computing for Geospatial Research and Application, San Jose, CA, USA, 22–24 July, pp. 71–75 (2013)