INDOOR POSITIONING USING CONVOLUTION NEURAL NETWORK TO REGRESS CAMERA POSE

Jing-Mei Ciou and Eric Hsueh-Chan Lu

Department of Geomatics, National Cheng Kung University, Taiwan – (p66064049, luhc)@mail.ncku.edu.tw

KEY WORDS: Convolutional Neural Network, Indoor Positioning, Camera Positioning, Computer Vision, Navigation

ABSTRACT:

In recent years, the issue of indoor positioning has become more and more popular and attracted more attention. Under the absence of GNSS, how to more accurately position is one of the challenges on the positioning technology. Camera positioning can be calculated by image and objects. Therefore, this study focuses on locating the user's camera position, but how to calculate the camera position efficiently is a very challenging problem. With the rapid development of neural network in image recognition, computer can not only process images quickly, but also achieve good results. Convolution Neural Network (CNN) can sense the local area of the image and find some high-resolution local features. These basic features are likely to form the basis of human vision and become an effective means to improve the recognition rate. We use a 23-layer convolutional neural network architecture and set different sizes of input images to train the end-to-end task of location recognition to regress the camera's position and direction. We choose the sites where are the underground parking lot for the experiment. Compared with other indoor environments such as chess, office and kitchen, the condition of this place is very severe. Therefore, how to design algorithms to train and exclude dynamic objects using neural networks is very exploratory. The experimental results show that our proposed solution can effectively reduce the error of indoor positioning.

1. INTRODUCTION

Human beings can quickly and easily judge the three-dimensional structure of moving objects and scenes through their eyes, and then calculate their position and direction. For example, when walking in the street, we can quickly judge where and where we are going by the surrounding buildings. How to effectively estimate the position and posture of users is a major challenge in the field of computer vision. Nowadays, positioning technology is progressing step by step, and outdoor positioning has become more mature. With more and more indoor positioning services proposed, positioning technology can be roughly divided into WiFi, Bluetooth positioning, infrared positioning, and camera and so on. WiFi positioning can achieve complex and large-scale positioning, but the positioning accuracy can reach up to 2 meters, which cannot achieve more accurate positioning. In complex space environment, Bluetooth and infrared positioning will be affected by noise, occlusion and other factors. The location of camera can be calculated by image and object.

In the past few years, there have been a large number of deep neural networks which have successfully learnt from large data on the network. They can predict the position and attribute accuracy of objects in classified images, and have greatly improved the accuracy by more than 90%. In the field of image recognition, neural networks have developed rapidly. Convolution Neural Network (CNN), Artificial Neural Networks (ANN), Recurrent Neural Network (RNN) and other methods have contributed to image recognition, especially convolution neural network. The main function of convolution neural network (CNN) is to zoom in and recognize two-dimensional figures with invariant shape distortion, such as displacement, and to learn some invariant features, such as translation and rotation. This is an effective recognition algorithm. In recent years, it has been widely used in image processing and pattern recognition. Its structure has strong adaptability, simple structure and few training parameters. It has fewer learning parameters, because it does not need to use the whole image pixels as input as the traditional recognition method, only need to learn a part of the patch. The local weight sharing structure of convolutional neural network has good performance in video recognition and image processing, especially in multi-dimensional input vector images, which reduces the complexity of the network. The advantage of convolutional neural networks and other kinds of neural networks is that they can extract high-resolution feature points for local areas. By using these feature points as the visual basis of human beings, the recognition error rate can be effectively reduced. A large number of literatures, whether single camera or dual camera technology, have been gradually discussed. We referred to a paper which called ‘Posenet: A convolutional network for real-time 6-dof camera relocalization’ by Alex Kendall *et al.* The 23-layer convolution neural network structure is used to input 224x224 color images, train the end-to-end position recognition task, and output 7-dimensional vector, respectively, 3-dimensional position and 4-dimensional quaternion. Finally, the output position and direction errors in the test stage. The accuracy of this method is not the best compared with other related papers such as SCoRe Forest which presented by Jamie Shotton *et al.* but for the performance of severe environment, the errors of other papers will increase a lot, while the accuracy of Posenet is relatively stable, even better than other papers.

Based on Kendall's method, before the training stage, we did some experiments in the pre-processing part of image input values. We adjusted different input size formats and combined with whether or not to load pre-training models to explore and analyse their convergence, and then compared the position and direction errors between the models. Through experiments, we finally know that the image is directly scaled to 224 x 244 size and the pre-training model is loaded as the input value. The pose error predicted by the training model is less precise and

This contribution has been peer-reviewed.
https://doi.org/10.5194/isprs-archives-XLII-2-W13-1289-2019 | © Authors 2019. CC BY 4.0 License.

more in line with our requirements. Details of the experiment will be described in Section 4.2. Since our method is supervised learning, we need to prepare precise pose data as the true value of the model. We use the navigation-level inertial navigation system with the loop camera LadyBug5 to collect the images, positions and directions needed for the experiment. In order to be close to the real environment, the panorama will do some pre-processing actions in the experimental stage, which will be explained in detail in Section 4.1. The experimental site is the underground parking lot behind the Engineering Department Hall of Sheng-Li Campus of National Cheng Kung University. Compared with the outdoor environment, the light in the underground parking lot is dim and the scene is monotonous without obvious characteristics. As an experiment, it is very challenging.

2. RELATED WORK

Localization problems can be solved in two traditional ways. Based on structural localization techniques, a common method is to use Structure from Motion (SfM) to represent scenes by implementing three-dimensional motion reconstruction, Agarwal *et al.* (2009), Snavely *et al*. (2006) and Kendall *et al*. (2016) have shown they use structure from motion algorithms on photo collections. The reason why humans can find their three-dimensional information from moving objects is because the brain finds a match in the continuous 2D image, that is, the corresponding point. The corresponding depth information is then obtained by the difference between the matching points. SfM derives the 3D information from the 2D image of the time series, where it does not need to input any camera parameters. Through the matching features between the 2D images, the parameters of the camera can be inferred. By establishing a set of 2D-3D correspondence relationships between matching features found in the query and descriptors related to 3D points, they get the complete 6DoF camera pose of the query photos.

Our localization recognition method tends to represent scenes through a database of geographically tagged photographs. We use the indoor mobile mapping platform provided by the work plan to collect scene images and their geographic locations. The platform is equipped with precision instruments. The images can be obtained by panoramic cameras, and the precise geographic location data can be obtained by IMU inertial measurement instrument and GNSS satellite receiver. In addition, there are also papers that use machine learning to localization. Jamie Shotton *et al.* (2013) proposed the scene coordinates for re-location to return to the forest in the context of the relocalization of RGB-D images. They use depth images to label scene coordinates. The algorithm maps each pixel value of depth images from camera coordinates to global coordinates. Then these coordinates are used as input values to train the regression forest model. Finally, the model regresses these coordinate labels to localize the camera. Generally speaking, it is to train the random forest model, predict the position of each pixel in the image, and then estimate the camera pose by generating 2D-3D matching, rather than relative position.

Convolutional neural network (CNN) is an efficient recognition algorithm, which has been widely used in image recognition and object detection in recent years. Integral neural network has three advantages in image processing: (i) The input image can match the network structure to a certain extent. (ii) Weight sharing reduces training parameters and makes the neural network structure more adaptable and simpler. (iii) feature extraction and pattern classification can be carried out simultaneously and simultaneously. To Accurately estimate the pose of a 6-DOF Camera, Kendall *et al.* (2016) proposed a CNN model, PoseNet, for regression pose estimation. They input 224x224 RGB images into PoseNet model for training, and then load pre-trained model before training, so that the training process can converge quickly, indicating that the model can be used in the dataset without overfitting. Then the training model can regress the 7-dimensional poses vector and locate the camera. In this paper, we will refer the architecture of PoseNet. In this paper, we will refer to PoseNet architecture, adjust different image size formats in the pre-processing part of input values, and analyse and discuss the accuracy of its 7-dimensional pose vector.

3. PROPOSED METHOD

3.1 Camera Pose

In this section, we referred the method of Kendall *et al.*, using the convolutional neural network to train a model directly from images to estimate camera pose. The neural network outputs a pose vector, including position and direction, as follows:

$$p = [x, q] \tag{1}$$

Where p = pose vector
x = position of the 3D camera
q = quaternion

Select quaternions to represent directions, because by normalizing to unit length, 4-D values can easily be mapped to rotation, which is simpler than the orthogonalization required for rotation matrices.

3.2 Architecture

In this paper, we refer to the PoseNet architecture derived from a deep neural network, GoogLeNet, proposed by Szegedy *et al.* (2015). GoogLeNet is a 22-layer convolutional neural network for image classification. It contains six inception modules and three classifiers. The classifier is used in the test phase. An inception module is a method of grouping filters in a convolution layer to achieve better and more useful features through filters of different scales in the same convolution layer. PoseNet's architecture is designed to make some micro-adjustments to a 23-layer convolutional neural network for GoogLeNet (see Figure 1). The adjustments are mainly divided into three parts: (i) First, three multi-classifiers are replaced by affine regression. Each final full connection layer is modified to output 7-dimensional pose vector, including 3-dimensional position vector and 4-dimensional direction vector. (ii) Before the final affine regenerator, insert a full connection layer with feature size 2048. This is to generate a localization vector, which can be explored by PoseNet. (iii) Normalized quaternion direction vector to unit length is in the test stage, because only the quaternion can represent rotation. Before entering the training phase, PoseNet resizes the image to 256 pixels and then crops it to 224 x 224 pixels on center. In Section 4.2, we will show different image scaling and compare its pose error with PoseNet.

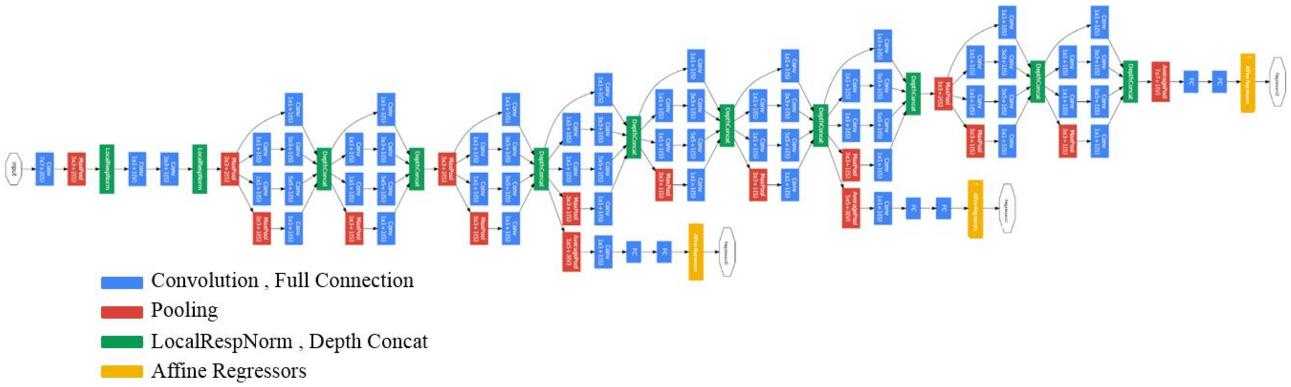
🟦 Convolution , Full Connection
🟥 Pooling
🟩 LocalRespNorm , Depth Concat
🟨 Affine Regressors

Figure 1. PoseNet architecture.

For comparison, we will use the same data set, and some basic settings (such as training samples, test samples, iterations, learning rates, etc.) will be set to the same. Generally, training a neural network requires a lot of training data. In classification problems, each output labels contains at least one training sample, but for regression problems, the output labels are mostly continuous or infinite. We used a pre-training model, GoogLeNet, and set it up in the same way as Kendall *et al.* did. In Section 4.2, we also demonstrate the convergence and error comparisons with or without the use of pre-training models.

3.3 Loss Function

In order to regress to the camera pose, the Euclidean loss is obtained by using the stochastic gradient descent algorithm in the training process. The loss function $L_p(I)$ is as follows:

$$L_p(I) = \|x-\hat{x}\|_2 + \beta \left\| q - \frac{\hat{q}}{\|\hat{q}\|} \right\|_2 \qquad (2)$$

where x and $\hat{x}$ are ground truth and estimated positions, while q and $\hat{q}$ are ground truth and estimated directions. Beta is a proportional factor, which is used to keep the position and direction errors approximately equal. In the process of calculating loss function for quaternion, it is necessary to normalize the direction vector of quaternion to the unit length first, because only the United quaternion can be used to describe rotation (oriented). Finally, the output of the model is position and direction errors. In order to maintain its balance without causing too large output errors for either side, we follow the Kendall *et al.* settings, and set the indoor environment beta ratio factor between 120 and 750.

4. EXPERIMENTAL EVALUATIONS

4.1 Experimental Setting and Dataset

For machine learning, supervised learning must require not only data but also ground truth labelling. Today's technologies such as image classification or image-based outdoor scene localization have released large training and test data sets. For indoor environment data sets, most of them are room-sized spatial ranges. Based on the localization of local features, we adopt the location recognition method of photographs with geographic markers.

The indoor scene dataset is collected by the indoor mobile mapping platform, which is provided by the mobile platform survey and mapping technology development plan and combines the mobile platform and precision instruments in a combined way. Electric farm machinery is used as mobile platform. The platform is an electric platform. It has forward and backward switches and loads an automatic braking system. When it is in a static state, it will automatically brake. Because the platform is not fully automatic, it needs to be manually operated in the direction of turning. The platform is charged by 12V DC, and the overall weight is about 800 kg. The setting packages of time pulse are 1-10 Hz, 1-10 seconds, 1-10 cycles of wheel speed and manual output. The part of precision instrument is divided into positioning system and mapping system. The positioning system is iNAV-RQH-10018-iMAR, which integrates IMU inertial measurement instrument and GNSS satellite receiver for the inertial navigation system with navigation-level laser gyroscope. The mapping system uses Ladybug 5 panoramic camera, which can take pictures from six angles simultaneously, and can be mosaic to form a panoramic image. The indoor mobile mapping platform is shown in Figure 2.

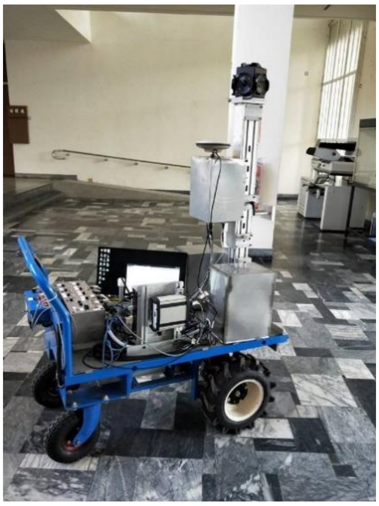

Figure 2. The indoor mobile mapping platform.

Because all the images taken by Ladybug 5 camera are panoramic images, considering the popularity of the public, the panoramic images will be pre-processed first. We use the image taken by Ladybug 5 camera (the camera with the zenith angle removed) to simulate the ASUS zenfone 2 mobile phone images (4096 x 3072) by program. These mobile phone images have geographic location reference pose information of their respective images. The experimental site is the underground parking lot behind the Engineering Department Hall of Sheng-Li Campus of National Cheng Kung University.

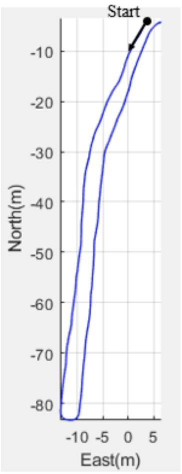

Figure 3. The path trajectory of the underground parking lot.

Figure 4. Sample images selected randomly from dataset.

The dataset spans the entire underground parking lot with a total area of 1200 m^2. The path trajectory of the dataset is shown in Figure 3, and the sample images are shown in Figure 4. Underground parking lots are dim and monotonous compared with other indoor environments such as chess, office and kitchen. Because of the bad environment in the field, it is more difficult for convolutional neural network to return to the user's position and direction, so it is very challenging as an experiment.

4.2 Experimental Result

Our pre-training model, Googlenet, uses a database for Places provided by Zhou *et al.* (2014), which contains about 7 million images and 476 scenes, and then lets the model train 800 iterations. We initialize the architecture and import the pre-training model to get the initial weights of random access to the network. We use a 23-layer convolutional neural network architecture and set different sizes of input images to train the end-to-end task of location recognition to regress the camera's position and direction. All experiments were performed using Tensorflow. Hardware devices used a single GeForce GTX 1080 Ti GPU display card to speed up the training and testing of neural network operation. The number of iterations was set to 30,000 times.

The dataset is collected in the underground parking lot behind the Engineering Department Hall of Sheng-Li Campus of National Cheng Kung University. The Training samples have 40,020 images, and the test samples have 500 images. Before the training phase, we designed several formats for the input image, including the format name and content details:

- Posenet_ori: The PoseNet’s method that images resize to 455 x 256 and then crop to 224 x 224 in centre, and this model will load pretrained model.
- Posenet_nonpy: The PoseNet’s method that images resize to 455 x 256 and then crop to 224 x 224 in centre, but this model will not load pretrained model.
- Resize224x224_npy: Images resize directly to 224 x 224, and this model will load pretrained model.
- Resize100x100: Images resize directly to 100 x 100
- Resize150x150: Images resize directly to 150 x 150
- Resize224x224: Images resize directly to 224 x 224
- Resize250x250: Images resize directly to 250 x 250
- Resize300x300: Images resize directly to 300 x 300
- Resize350x350: Images resize directly to 350 x 350
- Resize400x400: Images resize directly to 400 x 400

In Figure 5, the X axis is iteration and the Y axis is loss. These two bright blue and dark blue lines means ‘posenet_ori’ and ‘resize224_npy’. We can see that if we load the pre-training model, it can converge quickly in less iterations. And, the convergence of each model tends to be slow after the fifteen thousand iteration. In Table 1, among the intermediate errors in direction, the results from the format "resize 100x100" to "resize 400x400" seem to have a smile curve, and ‘resize250x250’ is the lowest point. Among the intermediate errors in position, there is no difference in the error between the models after the format 'resize150x150'.Comparing the two models of ‘resize250x250’ and ‘resize224x224_npy’, We can see that the two models have their own advantages, but from the point of view of the positioning system, the position is more important than the direction, so we will give priority to the minimum position error, that is, the ‘resize224x224_npy’ model. Finally, I compared the two models of ‘resize224x224_npy’ and ‘posenet_ori’. We can find that ‘resize224x224_npy’ is better than the PoseNet's method in this case. The improvement rate of orientation error is 33.4%, and the improvement rate of position error is 42%.

This contribution has been peer-reviewed.
https://doi.org/10.5194/isprs-archives-XLII-2-W13-1289-2019 | © Authors 2019. CC BY 4.0 License.

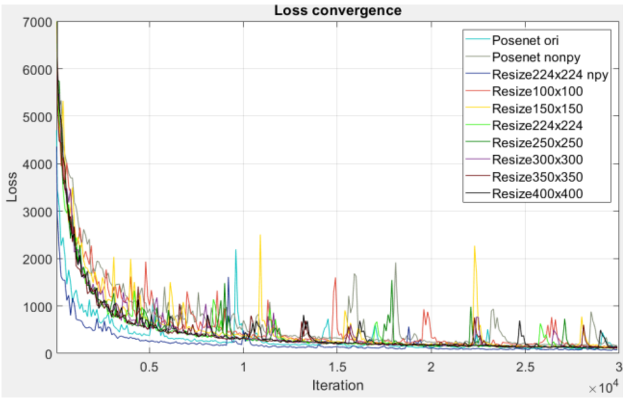

Figure 5. The loss convergence.

Table 1. Median error results on underground parking lot.

Format of input image	Position + Improvement (%)	Direction + Improvement (%)
posenet_ori	1.291 m	3.622°
Posenet_nonpy	1.400 m	3.139°
Resize224_npy-3	0.749 m (42%)	2.414° (33.4%)
Resize100x100	1.431 m	3.586°
Resize150x150	1.029 m	2.471°
Resize224x224	0.933 m	2.111°
Resize250x250	0.874 m	1.773°
Resize300x300	0.901 m	2.060°
Resize350x350	0.896 m	2.140°
Resize400x400	0.851 m	2.334°

5. CONCLUSIONS AND FUTURE WORK

In this paper, we propose a solution for indoor positioning based on CNN model. We first use indoor mobile mapping platform to collect high-quality panoramic images and high-precision location data, and then simulate mobile phone images with different directions through programs. Finally, these positions and directions are used as the ground truth of mobile phone image dataset. Underground parking lot is used in this experiment area. Compared with other office or kitchen areas with obvious characteristics, the underground parking lot has dim light and monotonous scenes, which is very challenging for the experiment. We use these datasets to train the 23-layer convolution neural network, get the training model, and analyse the training results. Before the training phase, we use input images of different size formats to analyse the difference between the convergence and output errors of different size images in the model. Finally, the results show that the error of images directly resize 224x224 is smaller than that of PoseNet's method. The improvement rate of direction error and position error is 33.4% and 42% respectively. In the future, we can analyse and understand which environmental factors will cause larger position or direction errors, which may be due to the lack of feature points, dim light or moving objects occupying too many areas of the whole image. In addition, we can try to improve the training model of the neural network to get more accurate position and direction.

ACKNOWLEDGEMENTS

This research was partially supported by Ministry of Science and Technology, Taiwan, R.O.C. under grant no. MOST 107-2119-M-006-028 - and The Mobile Platform Survey and Mapping Technology Development Plan under grant no. 107SU1207, for providing indoor mobile mapping platform and helping us to collect the necessary information and related data for the experiment.

REFERENCES

iMAR Navigation GmbHi, 2012. iNAV-RQH: Inertial Gyro Navigation System (ring laser gyro based). https://www.imar-navigation.de/de/produkte-uebersicht/product-overview-by-product/item/inav-rqh-inertial-laser-gyro-navigation-system (6 June 2018).

FLIR Integrated Imaging Solutions Inc.,2013. Ladybug5 USB3 Camera. https://www.ptgrey.com/ladybug5-360-degree-usb3-spherical-camera-systems (1 September 2018).

N. Andrew, N. Jiquan, F. Chuan Yu, M. Yifan, S. Caroline, C. Adam, M. Andrew, H. Awni, H. Brody, W. Tao and T. Sameep, 2013. UFLDL Tutorial. http://deeplearning.stanford.edu/wiki/index.php/UFLDL_Tutorial (8 November 2017).

N. Andrew, N. Jiquan, F. Chuan Yu, M. Yifan, S. Caroline, C. Adam, M. Andrew, H. Awni, H. Brody, W. Tao and T. Sameep, 2013. Convolutional Neural Networks. http://ufldl.stanford.edu/tutorial/supervised/ConvolutionalNeuralNetwork/ (28 November 2017).

S. Agarwal, N. Snavely, I. Simon, S. M. Seitz and R. Szeliski, 2009. Building Rome in A Day. *IEEE 12th International Conference on Computer Vision*, pp. 72-79.

N. Snavely, S. M. Seitz and R. Szeliski, 2006. Photo Tourism: Exploring Photo Collections in 3D. *ACM Transactions on Graphics (TOG)*, Vol. 25, No. 3, pp. 835-846.

J. T. Lalis, B. D. Gerardo and Y. Byun, 2014. An Adaptive Stopping Criterion for Backpropagation Learning in Feedforward Neural Network, Vol. 9, No. 8, pp.149-156.

A. Kendall, M. Grimes and R. Cipolla, 2016. Posenet: A Convolutional Network for Real-Time 6-DOF Camera Relocalization. *IEEE International Conference on Computer Vision (ICCV)*, pp. 2938-2946.

J. Shotton, B. Glocker, C. Zach, S. Izadi, A. Criminisi and A. Fitzgibbon, 2013. Scene Coordinate Regression Forests for Camera Relocalization in RGB-D Images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2930-2937.

C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke and A. Rabinovich, 2015. Going Deeper with Convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1-9.

B. Zhou, A. Lapedriza, J. Xiao, A. Torralba and A. Oliva, 2014. Learning Deep Features for Scene Recognition using Places Database. In *Advances in Neural Information Processing Systems*, pp. 487-495.

https://doi.org/10.5194/isprs-archives-XLII-2-W13-1289-2019