



Visual Localization Based on Deep Learning - Take Southern Branch of the National Palace Museum for Example

Chia-Hao Tu and Eric Hsueh-Chan Lu(✉)

Department of Geomatics, National Cheng Kung University, No. 1, University Rd,
701 Tainan City, Taiwan (R.O.C.)
luhc@mail.ncku.edu.tw

Abstract. Visual localization uses images to regress camera position and orientation. It has many applications in computer vision such as autonomous driving, augmented reality (AR) and virtual reality (VR), and so on. The convolutional neural network simulates biological vision and has a good image feature extraction ability, so using it in visual localization can improve regression accuracy. Although our team has built an image indoor localization model for Southern Branch of the National Palace Museum, this paper tries to use new network and loss function to achieve better positioning accuracy. In this paper, we use ResNet-50 as backbone network, and change the output layer to 3-dimensional position and 4-dimensional orientation quaternion, and use learnable weights loss function. We compare different pretrained models and normalization methods, and the best result improves the positioning accuracy by about 60%.

Keywords: Visual localization · Deep learning · Convolutional neural network

1 Introduction

With the mature development of global navigation satellite system (GNSS), people can obtain accurate ground position information through GNSS and provide various applications. If it is indoor, satellite positioning will fail, so indoor positioning technology is to solve the positioning problem when there is no satellite signal. Common technologies for indoor positioning include inertial navigation and wireless positioning. Inertial navigation uses an inertial measurement unit (IMU) to assist in positioning. IMU includes an accelerometer and a gyroscope, which can measure the change in pose of an object. The pose at this moment can be obtained by adding the pose change obtained by the measurement to the pose at previous moment. This method requires a given initial value, and as time increases, it will accumulate errors and make later trajectory drift. Positioning accuracy depends on the price of IMU. If higher positioning accuracy is required, IMU is more expensive. Wireless positioning, such as infrared, Wi-Fi, Beacon, ZigBee, etc., is to arrange wireless transmitters in indoor field. User uses wireless receiver to measure the strength of wireless signal from each transmitter to calculate the user's location.

This method needs to arrange transmitters in advance, and wireless signal is susceptible to interference and positioning failure. Visual localization uses visual characteristics of image to compare the features with images of database or 3D model of the field, and then calculate camera pose of the image. This method needs to collect many images or build a 3D model in advance, but there will be no accumulation of errors and less susceptible to interference. At present, visual localization has been widely used in the field of computer vision, such as robot positioning and navigation, autonomous driving, augmented reality (AR) and virtual reality (VR), and so on.

Visual localization methods can be roughly divided into the following three categories: image retrieval, learning model and 3D structure [19]. Image retrieval collects many images and their poses, and stores in a database. When positioning, an input image is compared with images in the database for similarity, and the camera poses are interpolated from similar images. This method is simple, fast, and easy to implement, but the accuracy is poor. Learning model is divided into training and testing stage. On training stage, we also need to collect many images and their poses to train the model to learn to regress camera pose. On testing stage, we only need to input an image to the model to get the camera pose. This method is more accurate than image retrieval, but it takes time to train model first. 3D structure will first establish a 3D model of the scene. During positioning, it will establish the match between two-dimensional feature points of the image and three-dimensional feature points of the 3D model, and then calculate the camera pose. This method has the highest accuracy, but the calculation time is longer and with the larger 3D model is, the longer the time is, and it fails easily in positioning on images with textureless features. Due to the rapid development of deep learning in recent years, convolutional neural networks (CNN) have been successfully applied to various visual tasks, such as image classification, object detection, etc. CNN simulates biological vision and has good image feature extraction capability, so the use of CNN in visual localization can more effectively extract features and improve accuracy of regression. At present, more and more scientists have invested in research of visual localization based on deep learning [20], showing that CNN is more capable of processing textureless features, and showing the accuracy of CNN visual localization can approach or even exceed those methods based on 3D structure and have real-time prediction speed simultaneously [16, 17].

Although our team has established an image indoor localization model based on deep learning for Southern Branch of the National Palace Museum [18], a better convolutional neural network and visual localization loss function have been proposed later, so this paper tries to use a new network and loss function to achieve better positioning accuracy. We use ResNet-50 as backbone network [5], change the output layer to a 7-dimensional camera pose, including a 3-dimensional position and a 4-dimensional orientation quaternion, and use learnable weights to combine position and orientation in loss function [8]. We Compare with last year results of our team's experiment in Southern Branch of the National Palace Museum, the best result in this paper improves positioning accuracy by about 60%.

In the following papers, Chapter 2 summarizes related works on CNN and deep learning-based visual localization. Chapter 3 describes the visual localization model

of this paper, including CNN architecture and loss function. Chapter 4 analyzes the experiments of pretrained model, normalization method and loss function.

2 Related Work

2.1 Convolutional Neural Network

LeCun et al. proposed the first CNN LeNet in 1998, which is composed of 2 convolutional layers and 3 fully connected layers [1]. Convolutional layers will downsample feature maps to reduce the size. The network uses convolution operations to extract image features, and automatically learns the parameters of convolution kernel through gradient calculation and backpropagation, so that the network learns to extract relevant features. With upgrading of computer hardware and emergence of big data, CNN has gained attention and caused a deep learning boom. Krizhevsky et al. proposed AlexNet in 2012, which is composed of 5 convolutional layers and 3 fully connected layers [2], and proposed ReLU activation function which accelerates network convergence and dropout method which reduces network overfitting. VGG proposed by Simonyan et al. in 2014 has a deeper network architecture [3]. For example, VGG16 is composed of 13 convolutional layers and 3 fully connected layers. In the same year, Google team Szegedy and others proposed GoogLeNet, not only to deepen the network, but also to increase the width by Inception architecture [4]. The same convolution layer extracts features of different scales through convolution kernels of different sizes, which helps to improve network performance. When the number of network layers increases, it is easy to cause the problem of back propagation gradient disappearance. In 2016, He et al. proposed residual learning to enable neural networks to have a deeper architecture such as the deepest ResNet has 152 layers [5]. Different from the network directly outputs features, residual network learns residual features and adds to the input as output features. Compared with the original direct learning features, residual learning is more stable. When residuals are zero, the network layer only performs identical mapping and will not degrade network performance. This paper will use ResNet as CNN model for visual localization, enabling CNN to have a deeper architecture to improve the performance of feature extraction.

2.2 Visual Localization Based on Deep Learning

In 2015, Kendall et al. first proposed PoseNet, a camera poses regression model based on GoogLeNet [6]. Pose includes 3-dimensional position and 4-dimensional orientation quaternion. The authors proved that transfer learning can reduce training time and achieve lower error. The following year, Kendall et al. used Bayesian CNN to estimate uncertainty of the model [7], detected whether there are scenes in the input image, and improved location accuracy of large-scale outdoor datasets. In 2017, Kendall et al. proposed learnable weights loss function to combine position and orientation to balance scale difference between position and quaternion values [8]. Wu et al. and Naseer et al. branched network into two parts to respectively output position and orientation [9, 10], and proposed data augmentation to increase the diversity of dataset by synthesizing images. Wu et al. further

used trigonometric function to solve that orientation has periodic problems [9]. Wang et al. and Clark et al. proposed a model combining recurrent neural network (RNN) [11, 12]. The model connects two LSTM (Long Short-Term Memory) layers after CNN to extract temporal features. The model Clark et al. proposed specifically uses two-way LSTM (Bi-LSTM) and adopts mixture density network to output pose distribution to solve perceptual aliasing problem [12]. Walch et al. combined LSTM in different way [13]. The authors thought that high dimensionality of the output from fully connected layer at the end of PoseNet may cause overfitting, so they design a different form of LSTM to reduce the dimensionality and select useful features for pose regression. Melekhov et al. proposed an hourglass-shaped encoder-decoder network [14]. Similar to fully convolutional network, hourglass network deconvolves feature maps to recover size, to maintain detailed features. In 2018, Brahmabhatt et al. proposed MapNet [15], which adds a restriction on relative pose between two images in loss function and uses additional sensor data such as IMU and GNSS to train MapNet without pose groundtruth. On testing stage, MapNet combines with pose graph optimization (PGO) to smooth the predicted trajectory. Valada et al. proposed VLocNet, which combines visual localization and visual odometry with multi-task learning [16]. Both tasks base on ResNet and share weights. VLocNet adds last pose prediction as feature into the model. The authors proposed a loss function to consider relative pose of continuous images. In the same year, Radwan et al. improved VLocNet and proposed VLocNet ++, which additionally considers semantic segmentation tasks [17]. Researches after 2020 apply the latest deep learning technology to visual localization. GrNet uses graph neural network (GNN) to aggregate information from nearby images [21]. AtLoc builds the network with self-attention [22]. MS-Transformer builds the network with transformer, and the network can locate in multi-scenes [23]. This paper will use learnable weights loss function to automatically learn the weights of positions and quaternions to balance scale difference.

3 Proposed Method

The model we proposed is composed of a CNN. The input is a 224×224 RGB image and the output is 7-dimensional camera pose. We use ImageNet pretrained model to speed up network convergence and achieve better performance. We perform z-score normalization on input image, so that training data can be scaled to the same scale, which is helpful for network to learn features.

3.1 Network Architecture

This paper uses ResNet as backbone network to extract image features [5], and its architecture is shown in Fig. 1. Convolution is mainly divided into 5 layers, and stride = 2 is used between each layer to reduce feature map. The Brackets in Fig. 1 are residual units. The second to fifth large layers are composed of several residual units. Global average pooling layer reduces feature maps to one-dimensional vector and then connects to output layer. ResNet was originally designed for image classification task. The architecture in Fig. 1 is used on ImageNet dataset with 1000 categories, so the dimension of output layer is 1000. After convolution layers extract image features, we

need to regress camera pose, so we change the dimension of output layer to 7, including 3-dimensional position and 4-dimensional orientation quaternion, and removed softmax layer.

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
		3×3 max pool, stride 2				
conv2_x	56×56	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$
conv3_x	28×28	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 8$
conv4_x	14×14	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 23$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 36$
conv5_x	7×7	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10 ⁹	3.6×10 ⁹	3.8×10 ⁹	7.6×10 ⁹	11.3×10 ⁹

Fig. 1. ResNet architecture (excerpt from [5])

Residual unit treats learned features as residuals. When input is x and learned residual is $F(x)$, add the input x and the residual $F(x)$ so that final output feature is $F(x) + x$. When the residual is zero, performance of the network will not be degraded, to solve disappearance or explosion of gradient caused by too many network layers.

It can be seen from Fig. 1 that when the size of feature maps reduces a half, the number of feature maps doubles. This principle is to maintain the complexity of network, but it will cause the input and output between residual units cannot be added due to different dimensions. Therefore, He et al. use 1×1 convolution to transform the input dimension to make it the same as the output dimension.

3.2 Loss Function

We give the ground truth of position and orientation quaternion of each image in the training dataset. The network learns to regress camera pose by supervised learning, such as loss function (1). \mathcal{L}_x is the loss function of position, \mathcal{L}_q is the loss function of quaternion, and γ is norm. We divide predicted quaternion by its vector length to make it a unit quaternion.

$$\mathcal{L}_x = \|x - \hat{x}_\gamma\|, \quad \mathcal{L}_q = \left\| q - \frac{\hat{q}}{\|\hat{q}\|} \right\|_\gamma \quad (1)$$

Due to the different unit of position and quaternion, \mathcal{L}_x and \mathcal{L}_q have different scale. If we direct add \mathcal{L}_x and \mathcal{L}_q as loss function, training may be difficult to output the optimal position and orientation. In PoseNet, the authors use a scale factor β to balance \mathcal{L}_x and \mathcal{L}_q [6]:

$$\mathcal{L} = \mathcal{L}_x + \beta \mathcal{L}_q \quad (2)$$

However, finding the best β requires a lot of experiments, which is time-consuming, and there will have different optimal β depending on training scene. Kendall et al. later proposed a method which automatically learns the weights of loss function between multi-tasks and apply it to camera pose regression [8]. It treats position and orientation regression as two tasks and using homoscedastic uncertainty between tasks as the weight of \mathcal{L}_x and \mathcal{L}_q . Homoscedastic uncertainty is a measure of uncertainty which is not dependent on input data. It stays constant for all input data and varies between different tasks. With this property, we can take homoscedastic uncertainty as the weight to combine losses of different tasks. The final loss function used in this paper is shown as Eq. (3). \hat{s}_x and \hat{s}_q are related to homoscedastic uncertainty, and they are automatic learning parameters. Only initial values need to be set, and then the best values will be found after network training converges.

$$\mathcal{L} = \mathcal{L}_x e^{-\hat{s}_x} + \hat{s}_x + \mathcal{L}_q e^{-\hat{s}_q} + \hat{s}_q \quad (3)$$

4 Experiments

This paper uses the dataset of Southern Branch of the National Palace Museum our team collected before [18]. The dataset has a total of 46,584 images, including 862 positions, and each position has 54 orientations. Our team collects scene images and their camera poses through indoor mobile mapping platform, which is divided into mapping system and positioning system. The mapping system uses the LadyBug5 panoramic camera, which can take images from 6 angles at the same time, stitch them into a panoramic image and then simulate it as mobile phone images; The positioning system includes IMU and GNSS instruments. It can obtain outdoor position through receiving GNSS signals and indoor position through IMU. This paper uses last year’s experimental results as a benchmark for comparison. Last year, our team used GoogLeNet as backbone network and used Places dataset pretrained model which pretrained for 800 iterations. The normalization method of input data was subtracting mean image of training dataset. The loss function only considered position, and the positioning error of training 30,000 iterations is 0.38 m.

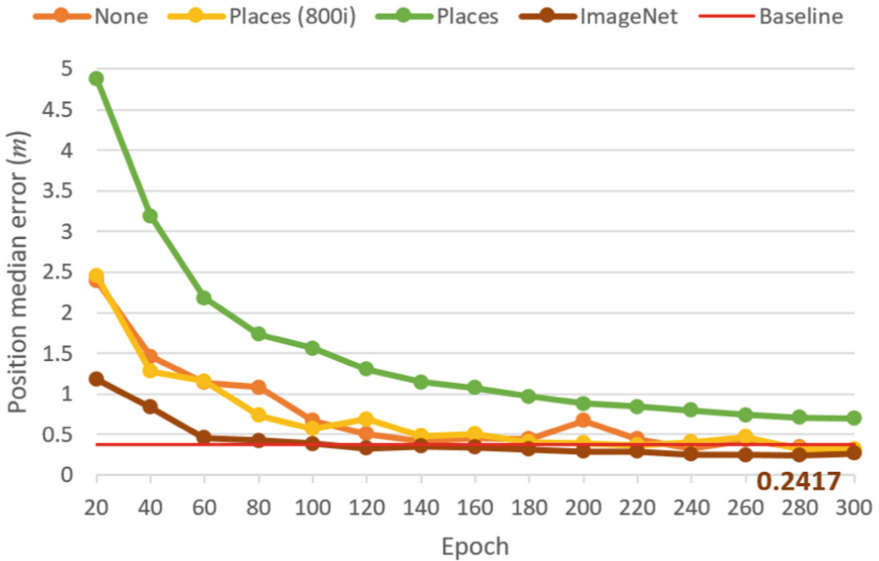
This paper uses ResNet-50 as backbone network for experiments, and compares the effects of different pretrained models, image normalization methods and loss functions on positioning error. The following experiments use Zenfone 2 simulation image of Southern Branch of the National Palace Museum. Image input size is 224×224 . All images are randomly cut into 5 parts for 5-fold cross validation. The following results are median of 5 experiments. Each experiment is trained for 300 epochs (about 100,000 iterations). This paper uses GeForce RTX 3090 for network training, and a cross validation training takes about 40 h.



(a) Mobile mapping platform



(b) Examples of the dataset

Fig. 2. Southern branch of the National Palace Museum dataset**Fig. 3.** Position error of different pretrained models

4.1 Pretrained Model

First, we want to find the best pretrained model. We first replace backbone network of the benchmark with ResNet-50, and then load different pretrained models. The positioning error comparison of different pretrained models is shown in Fig. 3. Baseline is the benchmark experiment. The result shows that the error of Places (800i) (Places dataset pretrained model pretrained for 800 iterations) and None (without pretrained model)

have no significant difference, which means that only pretraining 800 iterations does not achieve the effect of transfer learning. Place (Places dataset pretrained model pretrained to convergence) has the highest error, and ImageNet pretraining effect is the best, the error is 0.24 m. We believe that the reason may be the difference between two pretrained model tasks. Places dataset is scene recognition task, and ImageNet dataset is image recognition task. The feature extracted by image recognition relative to scene recognition is more related to visual localization. Using ImageNet pretrained model can accelerate the network convergence and achieve better accuracy.

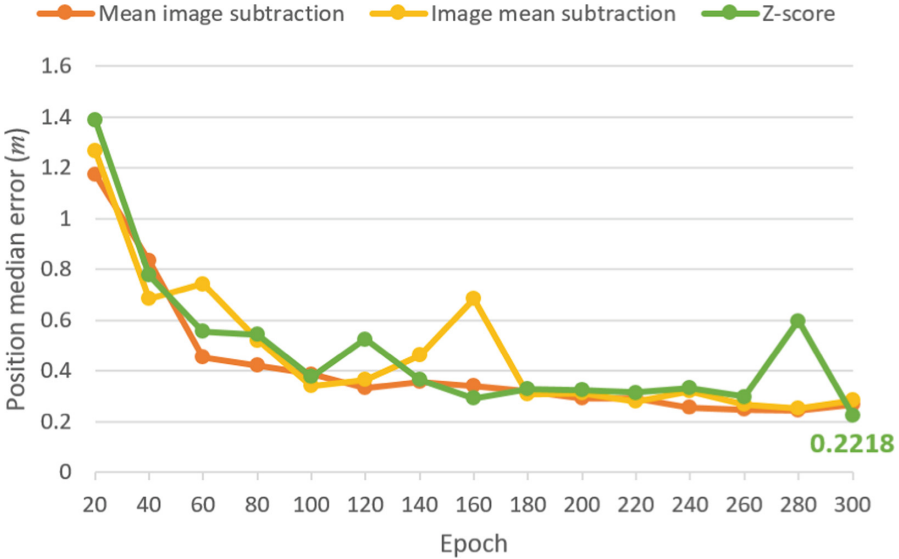


Fig. 4. Position error of different normalization methods

4.2 Normalization

We use ImageNet pretrained model and then want to find the best normalization method. The positioning error comparison of different normalization methods is shown in Fig. 4. Mean image subtraction is that input image subtracts mean image of training dataset, image mean subtraction is that input image subtracts mean RGB value of training dataset by channels, and z-score is that input image subtracts RGB mean and divides by RGB standard deviation. From the result, although mean image subtraction is relatively stable, it is found that three normalization methods have no obvious difference, which means that normalization has little effect on visual localization of Southern Branch of the National Palace Museum. Because z-score is common use and it has lowest error 0.22 m, we use z-score normalization for the next experiment.

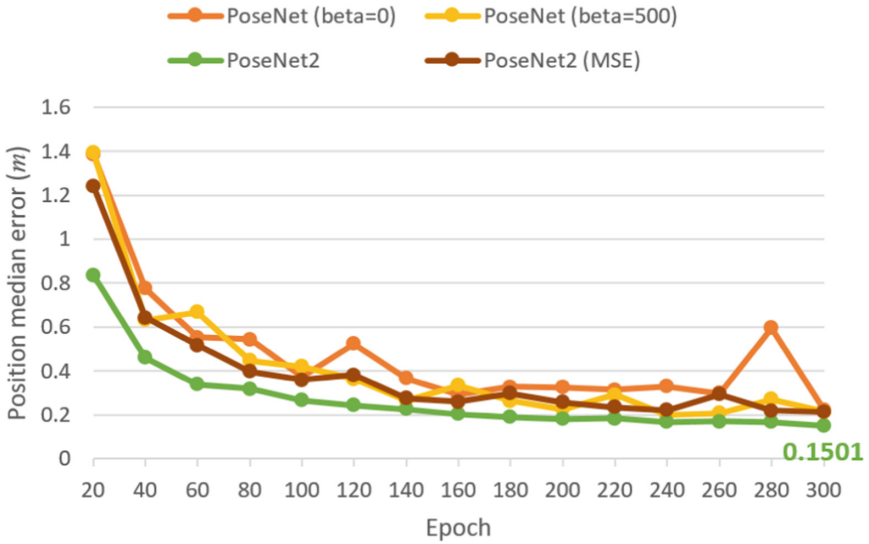


Fig. 5. Position error of different loss functions

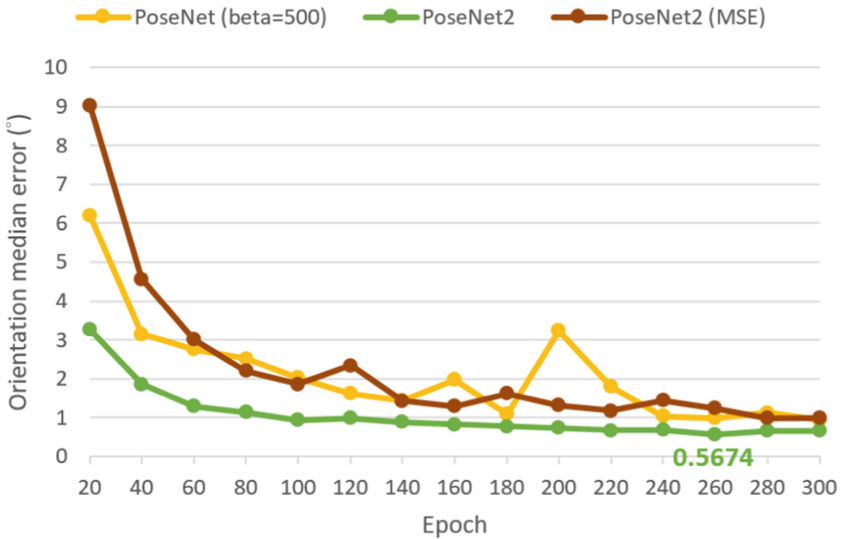


Fig. 6. Orientation error of different loss functions

4.3 Loss Function

In this section, we want to investigate the influence of different loss functions on positioning error. As shown in Figs. 5 and 6, we design four loss functions as shown in Eqs. 4, 5, 6, 7. The results show that the error of PoseNet ($\beta = 0$) is relatively higher than the other three. It means that regressing position and orientation at the same time

helps to reduce positioning error. We believe that the features extracted by regressing orientation are related to positioning. The position and orientation error of PoseNet2 are the lowest. The position error is 0.15 m, and the orientation error is 0.57° , which means that the loss function with learnable weights combined with position and orientation can effectively balance the scale between the two. The error of PoseNet2 (MSE) with square term is higher than that of PoseNet2. We speculate that the reason is the square term is more susceptible to errors, and the loss function cannot accurately reflect the difference between predicted value and ground truth.

1. PoseNet (beta = 0): Only consider position [6].

$$\mathcal{L} = (x - \hat{x})^2 + \beta \left(q - \frac{\hat{q}}{\|\hat{q}\|} \right)^2, \beta = 0 \quad (4)$$

2. PoseNet (beta = 500): Use the same value as the original paper [6].

$$\mathcal{L} = (x - \hat{x})^2 + \beta \left(q - \frac{\hat{q}}{\|\hat{q}\|} \right)^2, \beta = 500 \quad (5)$$

3. PoseNet2: Use the same initial values of parameters as the original paper [8].

$$\begin{aligned} \mathcal{L}_x &= |x - \hat{x}|, \mathcal{L}_q = \left| q - \frac{\hat{q}}{\|\hat{q}\|} \right|, \\ \mathcal{L} &= \mathcal{L}_x e^{-\hat{s}_x} + \hat{s}_x + \mathcal{L}_q e^{-\hat{s}_q} + \hat{s}_q, \hat{s}_x = 0, \hat{s}_q = -3 \end{aligned} \quad (6)$$

4. PoseNet2 (MSE): Replace MAE with MSE [8].

$$\begin{aligned} \mathcal{L}_x &= (x - \hat{x})^2, \mathcal{L}_q = \left(q - \frac{\hat{q}}{\|\hat{q}\|} \right)^2, \\ \mathcal{L} &= \mathcal{L}_x e^{-\hat{s}_x} + \hat{s}_x + \mathcal{L}_q e^{-\hat{s}_q} + \hat{s}_q, \hat{s}_x = 0, \hat{s}_q = -3 \end{aligned} \quad (7)$$

We summarize the last epoch results of all experiments in Table 1. The lowest position error is 0.15 m, which uses ResNet-50, ImageNet pretrained model, z-score normalization and learnable weights loss function. Compared with the baseline, 0.38 m, which uses GoogLeNet, Places dataset pretrained model which pretrained for 800 iterations, mean image subtraction and position loss function, we reduce the error by about 60%.

Table 1. Comparison of position median error with all experiments

Pretrained model	None	Places (800i)	Places	ImageNet
	0.2942 m	0.3223 m	0.7003 m	0.2665 m
Normalization	Mean image subtraction	Image mean subtraction		Z-Score
	0.2665 m	0.2844 m		0.2218 m
Loss function	PoseNet ($\beta = 0$)	PoseNet ($\beta = 500$)	PoseNet 2	PoseNet 2 (MSE)
	0.2218 m	0.2139 m	0.1501 m	0.2136 m

5 Conclusion and Future Work

This paper adopts new CNN model and loss function for visual localization of Southern Branch of the National Palace Museum. In terms of CNN, we use ResNet to replace the original GoogLeNet. Residual learning of ResNet enables CNN to have deeper architecture and improve performance. We also changed the output to regress position and orientation quaternion simultaneously, which the experiment result shows that can have better position accuracy than only regressing position. In terms of loss function, we use the loss function of learning weight which can automatically balance the scale difference between position and quaternion. Compared with the experiment result of South Branch of the Palace Museum last year, position accuracy has been improved by about 60%. In the future, we will combine visual localization with other related visual tasks such as visual odometry by multi-task learning. The features of related tasks will increase the performance of visual localization. We will also combine 3D information such as depth maps to learn 3D features of scene or add constraints to assist in localization.

Acknowledgement. This research was supported by Ministry of Science and Technology, Taiwan, R.O.C. under grant no. MOST 109–2121–M–006–013–MY2 and MOST 109–2121–M–006–005.

References

1. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. *Proc. IEEE* **86**(11), 2278–2324 (1998)
2. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. *Adv. Neural. Inf. Process. Syst.* **25**, 1097–1105 (2012)
3. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: *International Conference on Learning Representations* (2015)
4. Szegedy, C., et al.: Going deeper with convolutions. In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–9 (2015)
5. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778 (2016)

6. Kendall, A., Grimes, M., Cipolla, R.: Posenet: A convolutional network for real-time 6-dof camera relocalization. In: IEEE International Conference on Computer Vision, pp. 2938–2946 (2015)
7. Kendall, A., Cipolla, R.: Modelling uncertainty in deep learning for camera relocalization. In: IEEE International Conference on Robotics and Automation, pp. 4762–4769 (2016)
8. Kendall, A., Cipolla, R.: Geometric loss functions for camera pose regression with deep learning. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 5974–5983 (2017)
9. Wu, J., Ma, L., Hu, X.: Delving deeper into convolutional neural networks for camera relocalization. In: IEEE International Conference on Robotics and Automation, pp. 5644–5651 (2017)
10. Naseer, T., Burgard, W.: Deep regression for monocular camera-based 6-dof global localization in outdoor environments. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1525–1530 (2017)
11. Wang, S., Clark, R., Wen, H., Trigoni, N.: Deepvo: towards end-to-end visual odometry with deep recurrent convolutional neural networks. In: IEEE International Conference on Robotics and Automation, pp. 2043–2050 (2017)
12. Clark, R., Wang, S., Markham, A., Trigoni, N., Wen, H.: Vidloc: a deep spatio-temporal model for 6-dof video-clip relocalization. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 6856–6864 (2017)
13. Walch, F., Hazirbas, C., Leal-Taixe, L., Sattler, T., Hilsenbeck, S., Cremers, D.: Image-based localization using lstms for structured feature correlation. In: IEEE International Conference on Computer Vision, pp. 627–637 (2017)
14. Melekhov, I., Ylioinas, J., Kannala, J., Rahtu, E.: Image-based localization using hourglass networks. In: IEEE International Conference on Computer Vision Workshops, pp. 879–886 (2017)
15. Brahmabhatt, S., Gu, J., Kim, K., Hays, J., Kautz, J.: Geometry-aware learning of maps for camera localization. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 2616–2625 (2018)
16. Valada, A., Radwan, N., Burgard, W.: Deep auxiliary learning for visual localization and odometry. In: IEEE International Conference on Robotics and Automation, pp. 6939–6946 (2018)
17. Radwan, N., Valada, A., Burgard, W.: Vlocnet++: deep multitask learning for semantic visual localization and odometry. *IEEE Robot. Autom. Lett.* **3**(4), 4407–4414 (2018)
18. Lu, E.H.C., Ciou, J.M.: Integration of convolutional neural network and error correction for indoor positioning. *ISPRS Int. J. Geo. Inf.* **9**(2), 74 (2020)
19. Xin, X., Jiang, J., Zou, Y.: A review of visual-based localization. In: International Conference on Robotics, Intelligent Control and Artificial Intelligence, pp. 94–105 (2019)
20. Shavit, Y., Ferens, R.: Introduction to camera pose estimation with deep learning. arXiv preprint [arXiv:1907.05272](https://arxiv.org/abs/1907.05272) (2019)
21. Xue, F., Wu, X., Cai, S., Wang, J.: Learning multi-view camera relocalization with graph neural networks. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 11372–11381 (2020)
22. Wang, B., Chen, C., Lu, C.X., Zhao, P., Trigoni, N., Markham, A.: Atloc: attention guided camera localization. *AAAI Conf. Artif. Intell.* **34**(6), 10393–10401 (2020)
23. Shavit, Y., Ferens, R., Keller, Y.: Learning multi-scene absolute pose regression with transformers. In: IEEE/CVF International Conference on Computer Vision, pp. 2733–2742 (2021)