International Journal of Geo-Information

MDPI

Article

Integration of Convolutional Neural Network and Error Correction for Indoor Positioning

Eric Hsueh-Chan Lu * and Jing-Mei Ciou

Department of Geomatics, National Cheng Kung University, Tainan City 701, Taiwan; p66064049@mail.ncku.edu.tw

* Correspondence: luhc@mail.ncku.edu.tw

Received: 13 November 2019; Accepted: 27 January 2020; Published: 29 January 2020

Abstract: With the rapid development of surveying and spatial information technologies, more and more attention has been given to positioning. In outdoor environments, people can easily obtain positioning services through global navigation satellite systems (GNSS). In indoor environments, the GNSS signal is often lost, while other positioning problems, such as dead reckoning and wireless signals, will face accumulated errors and signal interference. Therefore, this research uses images to realize a positioning service. The main concept of this work is to establish a model for an indoor field image and its coordinate information and to judge its position by image eigenvalue matching. Based on the architecture of PoseNet, the image is input into a 23-layer convolutional neural network according to various sizes to train end-to-end location identification tasks, and the three-dimensional position vector of the camera is regressed. The experimental data are taken from the underground parking lot and the Palace Museum. The preliminary experimental results show that this new method designed by us can effectively improve the accuracy of indoor positioning by about 20% to 30%. In addition, this paper also discusses other architectures, field sizes, camera parameters, and error corrections for this neural network system. The preliminary experimental results show that the angle error correction method designed by us can effectively improve positioning by about 20%.

Keywords: Indoor positioning; image registration; convolutional neural network; deep learning; computer vision

1. Introduction

1.1. Background

With the rapid development of surveying and spatial information technology, more and more attention has been paid to the research and application of positioning. In outdoor environments, people can acquire accurate information of the ground position through global navigation satellite systems (GNSS). The emergence of GNSS has also resulted in more convenient location-based services (LBSs) [1] in many fields, such as satellite navigation systems, intelligent parking systems, and various geodetic surveys. Although GNSS brings convenience to people's lives, when a satellite signal is obscured, the application of GNSS will also fail. For example, signals are most likely to be lost indoors or in basements. Once these signals are obscured, the GNSS will not be able to continue to provide positioning services. Therefore, the problem of determining how to continuously calculate a location after the failure of a satellite signal has made indoor positioning technology a popular research topic. Indoor positioning is widely used and has high commercial value. Common application fields include the route guidance of a station, augmented reality (AR) interactions in art galleries, intelligent guides in department stores, and cargo monitoring in factories. Therefore, more and more scholars are discussing the technology related to indoor positioning.

1.2. Motivation

Indoor positioning technology can be roughly divided into three categories: dead reckoning, wireless signal positioning, and image positioning, but the first two have their own shortcomings. After dead reckoning is used for a period of time, error propagation will continue to expand, resulting in poor positioning accuracy. In a large-scale complex space, the position accuracy of wireless signals is unsatisfactory because of unstable factors such as interference and obstruction. Therefore, the third image positioning technology is becoming the focus of the main research axis. In the field of computer vision, whether for monocular or binocular camera technology, the applications of deep learning in positioning issues have been widely discussed and studied. This research refers to the architecture of PoseNet [2], proposed by Alex Kendall et al. When users employ mobile phones to shoot an image, this architecture can estimate the position and orientation of the image through the trained model. Mobile phones are easily accessible devices for the public. This method not only considers the universality of the device but also eliminates the interference problems that other signal locations will encounter. In addition, this paper relates to positioning with other deep learning applications; the accuracy of this method will maintain stable performance in harsh environments. However, according to our initial research results, the accuracy of PoseNet in a more complicated indoor environment is not good. There are three reasons for this problem, which also supports our motivation. First, this type of environment is inconsistent with our emphasis on position from the loss function of PoseNet, so the loss function needs to be modified appropriately. Secondly, the authors only cut the middle part of the training image, which resulted in information loss for the whole image. Finally, if the training image data and the actual positioning image are from different cameras, it is easy to produce errors between different platforms. As far as we know, there is no literature that discusses the errors between different shooting platforms, so overcoming this problem is one of the motivations of this paper.

1.3. Problem

Humans can rapidly and easily distinguish the moving objects and three-dimensional structure of a scene they see through their eyes and then calculate their position and orientation. Image indoor positioning technology is used to replace human eyes with a camera lens as the main core concept, to identify the position of people in an indoor space. Thus, how to estimate people's positions in an indoor space as accurately as possible is one of the problems discussed in the field of computer vision. In the past, many studies on deep neural networks have researched large data and predicted the accuracy of attributes in classification images. Moreover, this accuracy has also been improved to more than 90%. In recent years, deep neural networks have been used to predict spatial position accuracy. Deep neural networks are classified into many types, including the well-known convolutional and recurrent neural networks. In the field of image recognition, a convolutional neural network (CNN) is the most commonly used. CNN has two characteristics: local features and weight sharing. In image processing and video recognition, a CNN can extract important feature values from local images through filters. Weight sharing can reduce the complexity of a network in multi-dimensional vector image computation. In order to achieve image-based indoor positioning, such as when walking indoors (Figure 1), the user can take an image in a certain direction, extract the feature value of the image through CNN, and finally calculate the user's position at that time. Our goal is to predict this position as accurately as possible. Thus, this research refers to PoseNet, which best meets our situational needs, while the architecture of deep learning is adjusted to explore image scaling, cross-camera simulated mobile phones, and error correction.

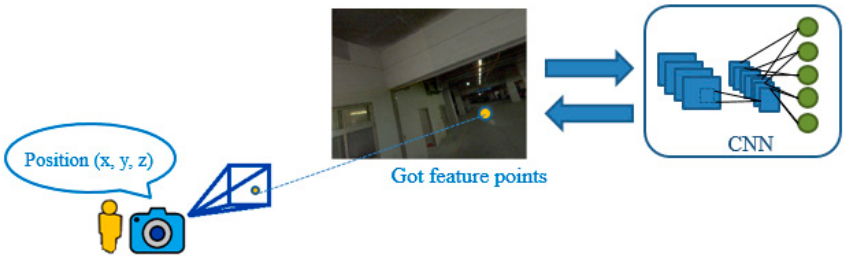


Figure 1. Sketch map of an indoor positioning application.

1.4. Contributions

1. Re-planning a 23-layer CNN architecture that is more suitable for indoor environments, adjusting the weight of loss function, and focusing on an accurate prediction of indoor positioning.
2. Before the training stage, the size of the image is changed to preserve the whole image as the input value of CNN.
3. In practical indoor positioning applications, most of the indoor images come from cameras on different platforms. The angle error correction of the position prediction results is first discussed between different platforms.
4. A mobile cartographic platform with a positioning system and a mapping system is used to collect a dataset of the underground parking lot and the South Palace Museum, including image and geographic location information, which can be used for related research in the future.
5. The preliminary experimental results of changing the image size show that the proposed method can effectively improve the indoor positioning accuracy by about 20% to 30%. After changing the loss function, the indoor positioning accuracy can be improved by about 80% to 90%. Preliminary experimental results show that our error correction method can effectively improve the indoor positioning accuracy by about 20%.

2. Related Work

This section reviews some important studies related to positioning issues. All related work can be divided into three parts including indoor positioning, image positioning, and convolutional neural networks.

2.1. Indoor Positioning

Dead reckoning, wireless signal positioning, and image positioning are common indoor positioning technologies. The first is based on Inertial Measurement Unit (IMU) technology. Li et al. [3] used hand-held mobile devices for indoor positioning. In this process, the step size and azimuth angle are first estimated according to the trajectory of user; then, the particle filter algorithm is used, and the plane is added. In this way, the initial estimate is corrected, and the final position is obtained. A particle filter is a non-parametric Bayesian estimation, which is often used in dynamic tracking and computer vision to calculate the position of the next moment. Lan et al. [4] proposed an indoor positioning system using sensors in hand-held mobile devices to collect available information. They applied indoor positioning in parking lots and use a pedestrian dead reckoning (PDR) system to track the trajectory of the user to detect when the user leaves the parking space, so the next user can use the handheld mobile device to obtain the location service for querying the parking space. Based on PDR, they abandon magnetometers and use accelerometers and gyroscopes only to detect the pace of the user and calculate the step size and azimuth of the user. In addition, they also calibrate the orientation errors caused by using only gyroscopes to obtain the azimuth.

The second common type of wireless signal positioning includes infrared, WiFi, and Bluetooth. This calculation method includes proximity positioning, an intersection method, and feature matching. The proximity positioning usually uses a Bluetooth low energy (BLE) device as a signal transmitter

and transmits the recognition signal to a nearby handheld mobile device. The intersection method uses the information obtained from the distance between the signal transmitter with multiple known coordinates and other sensors and then uses the concept of the resection method to obtain the desired sensor coordinates. Feature matching, also known as fingerprinting technology, is mainly divided into two stages. The first stage produces training data. It first measures the signal intensity of the position of multiple signal transmitters in an indoor space and then records the received signal intensity and the location mark of the transmitter to complete a radio map. The second stage is to locate the sensor needed to measure the point position, that is to say, to compare the results with the feature map of the first stage after measuring the signal intensity of the sensor, and to finally obtain the position of the sensor. Grossmann et al. [5] set up an Access Point (AP) for Wireless Local Area Networks (WLAN) in the exhibition hall of a museum and used the Received Signal Strength Index (RSSI) to obtain the position information. Subhan et al. [6] estimated the position of the indoor positioning system based on the Bluetooth using feature comparison. The accuracy of the indoor positioning system depends largely on the parameters of the alignment and the measurement results of the surrounding environment. The authors mentioned that external environmental factors include temperature, signal reflection, and obstacle interference. All these factors affect the feature matching and the decrease the accuracy. Thus, before obtaining their fingerprinting results, the authors proposed a method to measure wireless signals, which is called the standard radio propagation model, and used this method to estimate the real distance between the Bluetooth and the devices. Thereafter, the intersectional method of trilateration was used to obtain the coordinates for the position, which is constrained as the input of the filter to reduce the positional error of fingerprint.

The third image positioning technology is based on photogrammetry. The relative relationship between the camera and image control points is calculated through indoor control points. The camera position is calculated by geometric principles. Image positioning technology would not be affected by environmental factors such as temperature or wireless signal. In addition, Bluetooth positioning needs to install some signal transmitters in the surrounding environment. In response to some indoor environments, this device cannot be installed. Hence, image positioning provides the main axis of this study. Image positioning technology will be described in detail in Section 2.2.

2.2. Image Positioning

The problem of image positioning can be solved in two ways: feature matching positioning technology and machine learning positioning technology. Simultaneous localization and mapping (SLAM), scale invariant feature transform (SIFT), structure from motion (SfM), and other common methods of feature matching positioning technology are used. SLAM locates the pose and position by repeatedly collecting map features during the process of motion and then constructs and locates an image according to its own position. Incremental maps are used to achieve real-time location and map construction. Engel et al. [7] proposed a direct (non-feature) monocular SLAM algorithm, which allows large-scale construction of environmental maps. On the basis of high precision attitude estimation based on image alignment, the authors reconstructed a three-dimensional environment space into an attitude map with a semi-dense depth. However, SLAM needs to be operated by light detection and ranging (LiDAR) and images. At the application level, the mobile devices of almost all users do not have LiDAR elements, which is not in line with the goal of indoor positioning by mobile devices. Therefore, this paper does not use SLAM technology for exploration and experimentation. SIFT is used to detect and describe the local features in images. SIFT is mainly used in machine vision using sensory vision instruments. It searches for key points in scale space and extracts information such as position, scale, and rotation invariants as SIFT-like points. These SIFT-like points can be used for feature matching, and some papers discussed this positioning problem based on feature matching. Liang et al. [8] and Hao et al. [9] used SIFT for map feature matching and location information, but the disadvantage of SIFT is that it requires many feature databases and effective sifting of SIFT-like

points. This huge amount of computation will make it difficult for indoor location services to achieve real-time application.

SfM represents scenes through reconstructed three-dimensional motion. Agarwal et al. [10], Snavely et al. [11], Wu et al. [12], and Kendall et al. [2] used motion recovery structure algorithms to collect image postures (i.e., positions and directions). People can find matches from continuous 2D images in the brain and use them to find three-dimensional information on moving objects. Matching uses the corresponding point based on the difference between the matching points; the corresponding depth information can thus be obtained. SfM obtains 3D information from time series 2D images without the input of any camera parameters. Camera parameters can be deduced by matching features between 2D images. By establishing the corresponding relationship between a set of features and their 3D coordinates, the complete 6-DoF camera attitude of the corresponding image can be obtained. SfM has the same disadvantages as SIFT: Both require a long execution time. On the practical application level, they cannot achieve the condition of real-time application.

Machine learning includes many theories, such as decision tree, random forest, deep learning, and so on. J. Shotton et al. [13] employed scene coordinates to reposition the returned data based on RGB-D images. The authors used image scene coordinates with depth information. This method converted the coordinate information of the depth images from the camera to the whole area, and then input these coordinates to train the regression forest model; they then model regression to locate the camera. Furthermore, this method trains the random forest model to predict the position of pixels in the image and estimates the camera's attitude by creating 2D-3D matching instead of relative positioning. In-depth learning subdivides many kinds of artificial neural networks (ANNs), such as convolutional neural network (CNN), recurrent neural network (RNN), and so on. For image recognition, CNN is the most widely discussed technology. At present, there are more technologies able to apply the CNN model to mobile devices and implement the function of real-time identification. A CNN model that requires significant time training can be deployed offline. Users only operate CNN models deployed on mobile devices for positioning purposes, which solves the above-mentioned technical problems, including the lack of LiDAR and long computation. The next section will discuss CNN-based positioning technology in detail.

2.3. Convolutional Neural Network

A convolutional neural network, or CNN [14], is an effective and useful image identification algorithm which is widely used in pattern recognition, object detection, and image positioning. The CNN has three strengths in image processing: (1) It can extract high-resolution feature values from local areas through filters. (2) Its weight sharing structure can reduce the training parameters, thus reducing the complexity of the network. (3) Both allow extraction and predictive classification to be implemented simultaneously. CNN models, such as AlexNet, GoogLeNet [15], and residual network (ResNet) [16], are all well-known network architectures in the field of image recognition. AlexNet was the champion of the ImageNet ILSVRC competition in 2012, and is also a model that has attracted CNN attention. GoogLeNet was the winner of the ILSVRC classification competition in 2014. Szegedy et al. used an inception module to reduce training parameters, which will be detailed in Section 4.3.2. ResNet was the champion of the ILSVRC competition in 2015. He et al. found that if the number of network layers reaches a certain number, the accuracy of training will begin to decline, that is to say, if the network is too deep, it will become difficult to train. The authors designed a residual network to improve the gradient vanishing problem in the training process, through which the input value can be approximated to the output value, which greatly reduces the error rate of training. In this section, CNN-based positioning is discussed for two purposes: depth estimation for generating depth maps and regression pose estimation. A depth map is a 2D image, and each pixel on the image records the distance from the perspective of the viewer to the surface of the object. Here, the object refers to the object generated by a shadow. Godard et al. [17] proposed a different model and showed that no depth data was needed, but synthetic depth maps were directly trained as intermediate values.

Furthermore, this model uses unsupervised monocular depth estimation, whose purpose is to estimate binocular stereo images using a single image. This method provides a new training loss value for image reconstruction, which can enhance left-right consistency in the neural network and finally generate disparity depth maps. Zhou et al. [18] proposed an unsupervised learning model which uses continuous images to estimate self-motion and scene reconstruction. These models use a single image depth network and a multi-image attitude network. The loss function calculates the loss between the image and the target from the estimated depth and attitude and finally generates the predicted depth map of a single image.

There are also studies on positioning in combination with a variety of indoor positioning technologies. Ashraf et al. [19] combined mobile phone sensor data with a CNN to predict the current position of pedestrians, with the purpose of reducing the dependence of devices in magnetic field positioning system. Firstly, a CNN model was trained to recognize indoor scenes, which helped to identify specific floors and reduce the search space. Then a modified K Nearest Neighbor (mKNN) was proposed to calculate the current position of pedestrian as the starting point of PDR; then, the Extended Kalman Filter (EKF) was implemented from the position of PDR and the database, and the final position was obtained. Kang et al. [20] proposed a new architecture to improve PDR performance. A segmented signal frame was used to determine the pedestrian speed for CNN and RNN, and the walking distance was estimated by calculating the speed and moving time. Wang et al. [21] extracted channel state information (CSI) data from WiFi as the input for deep CNN to predict the position of mobile devices. Mittal et al. [22] also used WiFi data. They proposed a method for transforming WiFi signatures into images to build a CNN with an extensible fingerprint recognition framework. Niitsoo et al. [23] and Bregar et al. [24] used the latest wireless signal ultra-wideband (UWB). They proposed a CNN framework to estimate the absolute position of labels by learning channel impulse response (CIR) data.

In order to accurately regress the pose of a monocular camera, Kendall et al. [2] used a CNN model called PoseNet in 2015 to regress the pose estimation. They use the SfM algorithm designed by Furukawa et al. [25] to obtain the required position information. During the training phase, the authors scaled 455 × 256 pixels for color images and cut the center into 224 × 224 pixels. The authors then input the scaled pixels into the model for training. The theses proposed by Bengio et al. [26], Oquab et al. [27], and Razavian et al. [28] have shown the feasibility of transfer learning. Kendall et al. learned the weights of other models before training based on transfer learning. Transfer learning shows that loading a pre-training model can accelerate the convergence of training processes and help neural networks learn local features quickly; this model can also be applied in the dataset without over-fitting. Finally, the training model regresses a 7-dimensional pose vector to achieve camera positioning. Kendall et al. [29] used a Bayesian CNN to estimate the uncertainty of the model for the subsequent year, to detect the presence of scenes in the input image, and to improve the positioning accuracy of a large-scale outdoor dataset. Kendall et al. [30] adjusted the loss function of the PoseNet estimation orientation in 2017 to improve the performance of the model. Walch et al. [31] proposed a new architecture based on the PoseNet model, which combines CNN with long short-term memory (LSTM). LSTM is a kind of neural network used for processing sequential data. This model's advantage is that it can preserve the weights of the previous layers to ensure that useful features are not lost during training. The authors' intuition is that PoseNet estimates a pose from high dimensions, so using a full connected layer is not a good choice. The high dimension of a full connection output may lead to over-fitting during training, so the authors designed LSTM after a full connection to reduce the structural dimensions and select useful features for pose estimation. PoseNet is not entirely suitable for indoor locations. Although Kendall et al. have continued to update this model, since the focus of this research is on positioning accuracy, adjusting the internal loss function is inevitable. Furthermore, PoseNet has not been discussed in relation to the position error issues of different image sizes, field sizes, cross-platforms, etc. Therefore, this paper discusses these factors in depth.

3. Problem Statement

In order to clearly explain the problems and objectives of this study, some terms used in this paper must be formally defined; then, the problem statement is summarized in this section. Table 1 summarizes the symbols used in this paper.

Table 1. The symbols used in this paper.

Symbol	Description
I	RGB Images
P^v	Position vector
d	A data
D	A series of d compose a dataset
S	Several datasets compose a scene.

Definition 1. *A Datum. Parameter D represents a datum, and this datum contains a set of N × M pixels of RGB image information. This datum can be divided into historical data and future data. Historical data are a type of data composed of known data, which are used in training the samples needed for model learning. Historical data* $d^h = \{I,\ P^v\}$ will contain the observed three-dimensional position vector P^v. *On the other hand, future data are a data type featuring unknown data, which are used in the test samples needed for model prediction. For future data* $d^f = \{I,\ null\}$, the P^v value is unknown. The position vector P^v *will be evaluated in the CNN model.*

Definition 2. *A Dataset of a camera device. We use a camera device to collect one or more trajectory data* $D = \{d_o,\ d_1,\ d_2, \ldots,\ d_n\}$ *with image information. The trajectory data will be divided into historical data (training samples) and future data (test samples). The test samples need to be covered by training samples so that the CNN model can effectively predict the position vector* P^v *of the future data. Figure 2 shows a series of data comprising a dataset; each camera device has a dataset.*

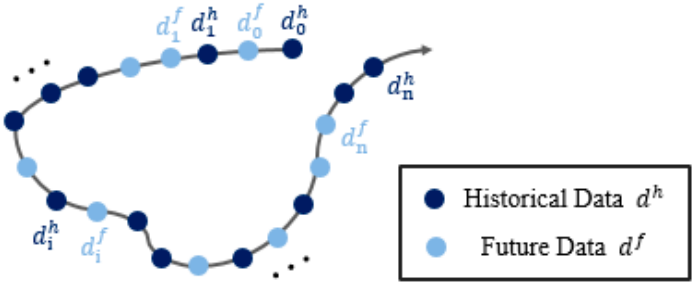


Figure 2. The concept of dataset *D*.

Definition 3. *A Scene. A scene* $S = \{D_1,\ D_2,\ D_3,\ \ldots,\ D_n\}$ *can be composed of one or more datasets. Figure 3 shows an example of a scenario in which there can be many camera devices.*

Research Objective. This research focuses on the prediction of camera position. I and P^v in dataset d^h will be used as input values, and the CNN will train the model. Finally, the model will be used to predict the position vector P^v of each d^f. Minimizing the median of the predicted position error (m) is the main goal of this study. Further, many test samples are predicted, and all the errors of these samples are collected to evaluate the median errors according to the ground exact position of the dataset. This research also attempts to use different camera devices (datasets) as training and test samples in the same scenario to achieve cross-platform camera positioning.

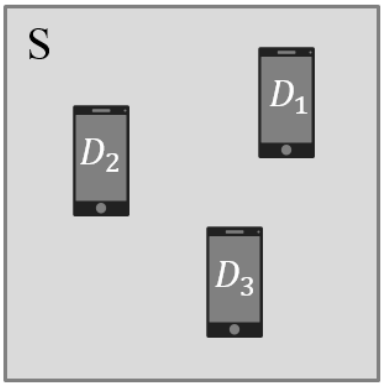


Figure 3. An example of a scene.

4. Methodology

This section introduces the proposed methodology for the deep learning model to regress camera positioning in the indoor scene.

4.1. Framework

The steps and processes of the overall architecture are shown in Figure 4. Based on the PoseNet model, a convolution neural network (CNN) is designed for indoor positioning. Moreover, we discuss the range of the main CNN models for indoor data collection, adjustment and training, as well as the error correction between different camera parameters. Indoor data mainly use a mobile cartographic platform to collect the image and position information needed by the model. Before actual operation, the image collected by the mobile cartographic platform will be simulated as a mobile phone image. The structure and loss function of the CNN model are adjusted, and different image size matching pre-training models are designed before the training stage. In addition, this study also designs an angle error correction algorithm to correct the position errors between different camera platforms. The following is a detailed description of the research steps of each part.

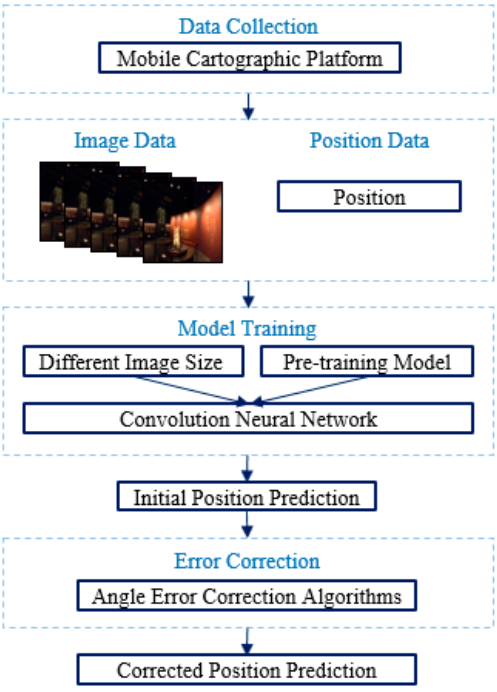


Figure 4. The framework of the methodology.

4.2. Data Preparation

Our paper adopts supervised learning, which requires data and corresponding ground exact position labels. In the problem of position, image positioning recognition with geographic coordinates is used. In recent years, many datasets have been used in image processing to deal with classification or outdoor scene positioning. For datasets used in indoor environments, most feature room-sized spaces. To experiment with specific areas, data must be collected independently. In order to cope with large-scale and long-term purposes, the indoor mobile cartographic platform is used to collect indoor scene data. This mobile cartographic platform is offered by the project of mobile platform development on surveying and mapping technology [32]. The mobile cartographic platform is an electric agricultural machinery cart with two precision instruments. As shown in Figure 5, this platform is equipped with an electric switch to control its forward or backward movement. In addition, this platform has an automatic braking device, which can automatically start the braking function even if the platform is stationary on the slope without causing danger. Because this platform requires manual operation to control direction, it takes considerable time to collect the dataset. Precision instruments are equipped on the platform to collect the data needed for the experiment. Precision instruments are divided into a positioning system and mapping system. The positioning system uses INAV-RQH-10018-IMAR, including a GNSS satellite receiver and an IMU inertial measurement instrument. The GNSS signals are detected outdoors, and then the platform is pushed indoors to collect position information and image information. The position information is processed by the PointerMMS software developed by the participant in the project of mobile platform development on surveying and mapping technology. According to the test results in 2015, the accuracy is less than 1 m in 5 min. The mapping system uses a LadyBug5 camera. This camera is equipped with six-angle lenses to take photos at the same time and form a panoramic image output. Considering the universality of life and the application of indoor positioning technology, the panorama captured by the LadyBug5 camera is pre-processed and simulated for the images with different mobile phone parameters. To further explain, the panorama is spliced into a plane by a program, and the images shot in different orientations are simulated for three angles. The three angles, yaw, pitch, and roll, represent the absolute orientation of the image. For simplicity, this paper does not use normal conventions for these angles; instead, it uses "easy human readable" conventions:

- Yaw is defined as the angle between the optical axis of the camera and an Eastern direction. If the simulated image "looks" toward the East, it has a yaw of 0°; if the image is in the North direction, it has a yaw of 90°.
- Pitch is defined as the angle between the optical axis and the horizontal plane, which is defined as positive when the simulated image "looks" toward the top.
- Roll is defined as the rotation of the image around the optical axis. When the camera is tilted to the left, it is positive.

Each camera has a different focal length and image size. In this paper, these two parameters are used to simulate different cameras, where image size indicates the width and height of a cut image in the panorama that can be used to determine the view breadth. To sum up, these simulated mobile phone images all have their own geographic reference position information.

4.3. CNN Model Training

This section introduces the CNN model based on camera position and explains in detail how to re-adjust the architecture of a 23-layer CNN so that the model can learn new loss functions to update the weight and ultimately output the position vector. The following three sections describe the camera positioning, CNN architecture, and loss function.

Figure 5. Mobile cartographic platform.

4.3.1. Camera Positioning

Camera positioning is based on the relative relationship between image control points and cameras, and the geometric principle is used to calculate the position of cameras. In addition to acquiring good images and precise position coordinate information, how to extract important features from an image and how to regress the positioning accuracy close to the ground exact position is the most important goal of this paper for indoor positioning based on a CNN. This model refers to the architecture of PoseNet and make some minor adjustments to it to calculate the camera position directly from the image training model. Then, the onvolutional neural network outputs position vectors (i.e., three-dimensional position information), as shown in Formula (1):

$$P = [x, y, z] \tag{1}$$

4.3.2. CNN Architecture

The architecture of PoseNet also refers to the 22-layer CNN of GoogLeNet [15], which is a classification model and used to be the champion of the Large Dataset ImageNet Challenge (ILSVRC14). Although this CNN has a 22-layer deep network structure, the sizes of its parameters are much smaller than those of other networks with fewer layers (such as the Visual Geometry Group (VGG) or AlexNet networks). In order to increase the number of layers and reduce the number of parameters, only sparse connections can be used, but most of the algorithms are based on a dense matrix. Therefore, in order to achieve high computational performance, only the stacking of neurons in the human brain can be simulated, and the sparse matrix can be aggregated into the dense matrix to achieve this goal. GoogLeNet proposes a network structure called "Inception modules" to build a sparse and highly computational network structure. These Inception modules group the filters in the convolution layer. That is to say, in the same layer, filters with different scales are used to obtain better and more useful eigenvalues. Kendall et al. compared the effectiveness of the AlexNet and GoogLeNet architectures and finally decided on the GoogLeNet architecture. Figure 6 illustrates the revised architecture of PoseNet based on GoogLeNet. The main adjustments are marked with purple and light green boxes.

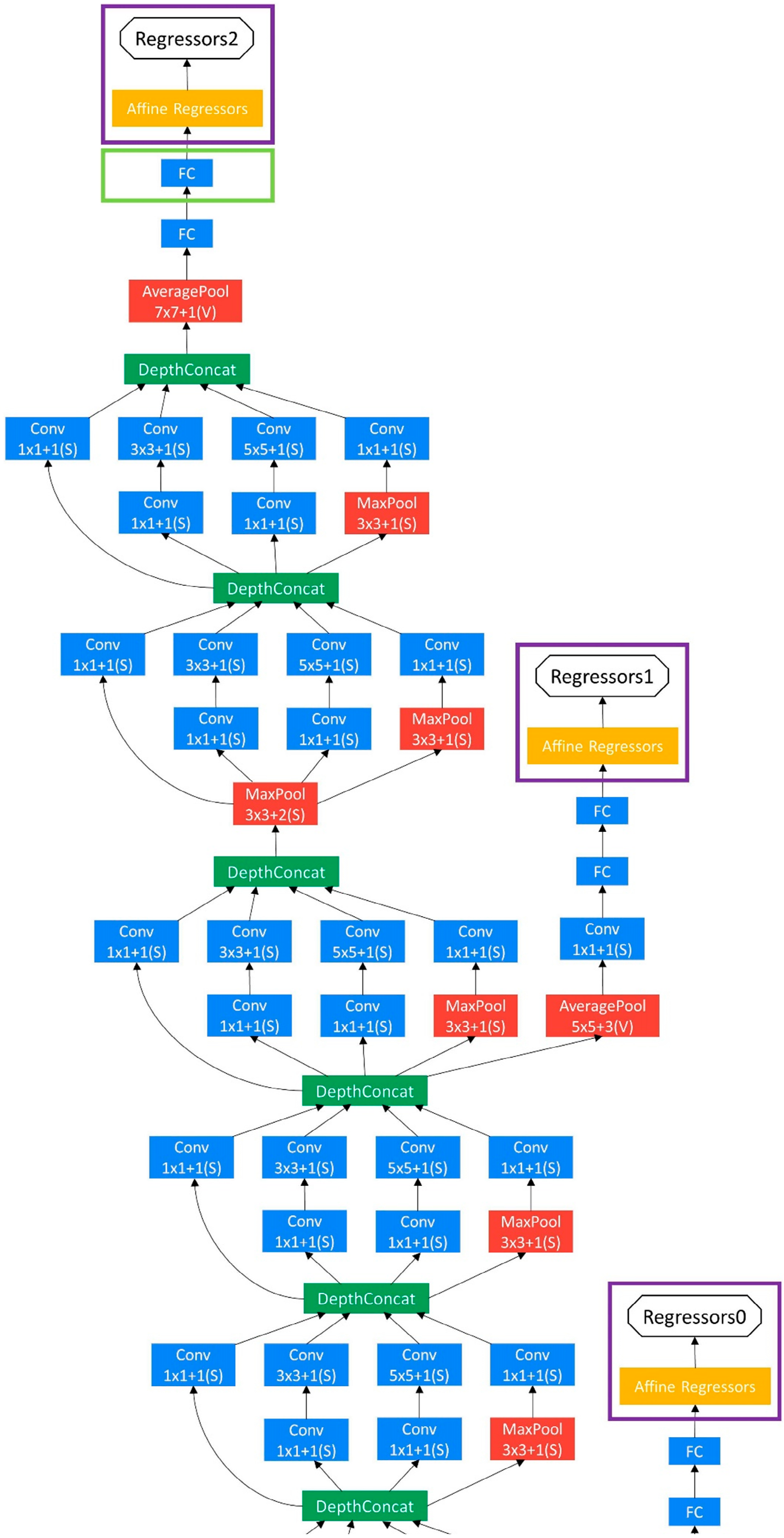


Figure 6. *Cont.*

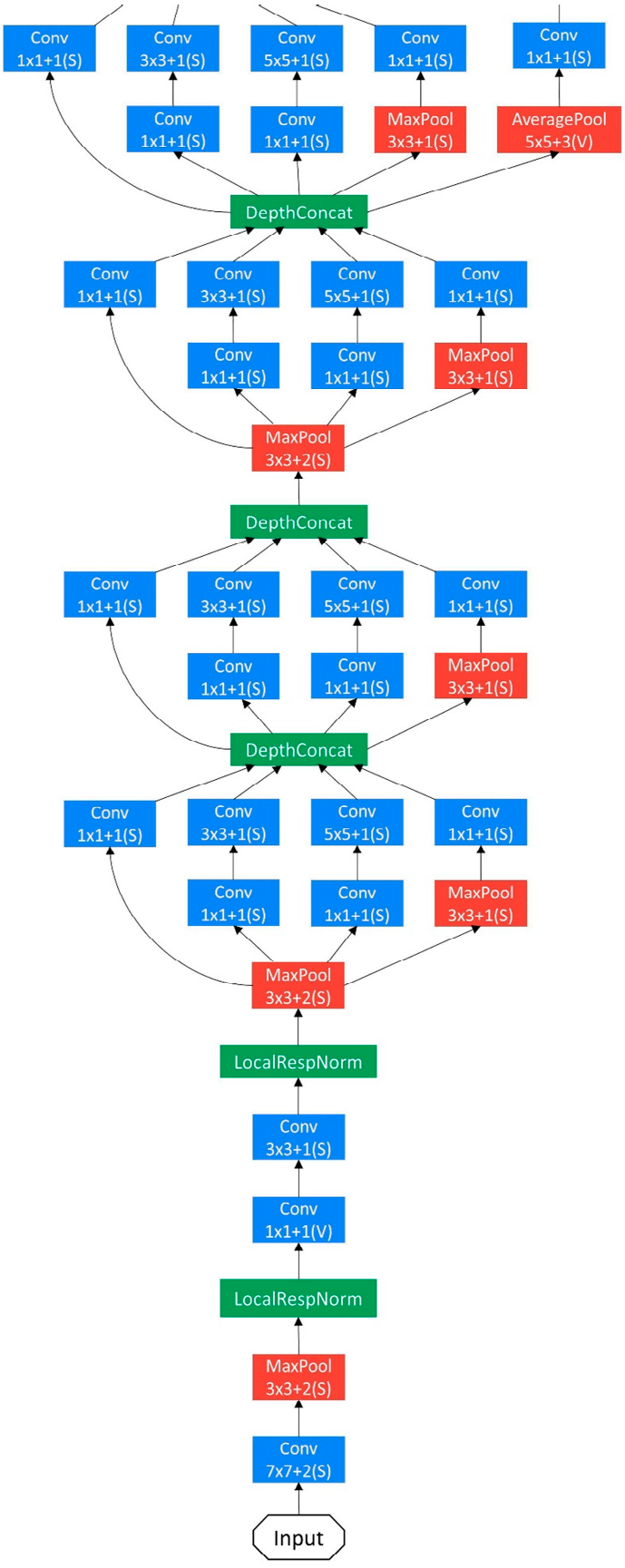


Figure 6. The architecture of PoseNet.

- Purple box: The three multi-classifiers replaced with the regression affine. Each final full-connection layer outputs a seven-dimensional pose, including a three-dimensional position and a four-dimensional quaternion.
- Light green box: A full connection layer is inserted with a feature size of 2048 before the final affine regenerator to form a 23-layer architecture. This process generates a location vector that can be explored by PoseNet. For classification problems, the output label is finite since each output label is come from one of training samples. However, in regression problems, the output label is a continuous value and the regression output is infinite and complicated.

This research has also made some adjustments to the architecture of PoseNet. The final output of the PoseNet model includes position and orientation information, but the orientation is not necessary for our goal. Our ultimate goal is to let users use mobile phones for image positioning, focusing on position accuracy. A mobile phone has an accelerometer, gyroscope, and magnetometer three-axis sensor within itself. Only the information from the gyroscope in a mobile phone is needed to determine the orientation angle, so a CNN is not needed to estimate orientation. That is to say, as shown in Figure 7, our CNN model ultimately discards the orientation and only outputs position information.

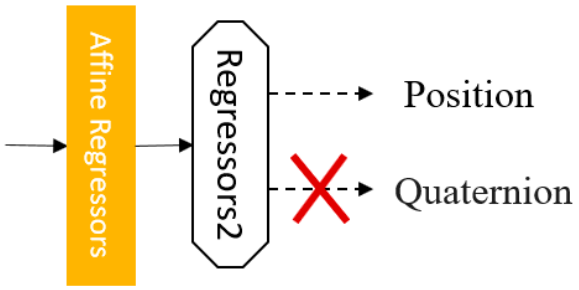


Figure 7. The convolutional neural network model only outputs the position vector.

In addition, in the image pre-processing stage before training, PoseNet scales 455 × 256 pixels for the input color image and then crops the center into 224 × 224 pixels. Although this method of image cutting cannot deform the image, this method will result in the loss of important features of the surrounding local environment. Especially for indoor environments, every pixel value of each image is very important, which is an important foundation for the CNN to calculate its position. As shown in Figure 8, the cropping method is removed, and the color image is allowed to be scaled directly to $N \times N$ pixels as input values to preserve the integrity of the image. In addition to image pre-processing, PoseNet also uses a pre-training model before the training stage. This pre-training model enables the CNN model to learn the initial weights and allows the model training to converge quickly. This process is like learning English letters (the pre-training model), which entails giving children basic knowledge of the letters (weight), followed by learning words or conversations through training, which is faster than learning English first. The experimental results of the pre-training model for scaling size matching will be described in detail in Section 4.

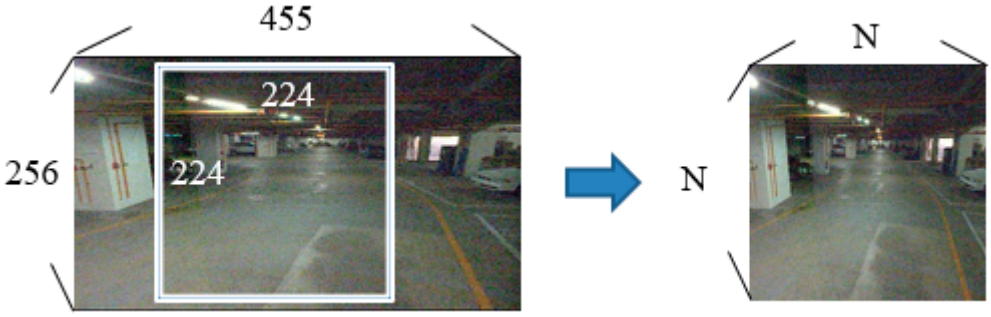


Figure 8. Adjust the input size of image preprocessing.

4.3.3. Loss Function

The architecture of the CNN model was described in the previous section. In this section, if the loss function calculates the orientation loss, it will affect the accuracy of the position prediction. Our research focuses on positional accuracy, so the orientation is not used for prediction. Since the orientation of the estimation has been removed, the loss function is rewritten for Equation (2). This research uses the stochastic gradient descent (SGD) algorithm in the training process to obtain the Euclidean loss so as to return to the camera position. The loss function is shown in Equation (2), where $\hat{P}$ and P are the position predicted values and ground exact position, respectively. Many image recognition datasets are trained to explore features in advance, and then the pre-training model is loaded into other models for training. This method helps converge to a lower loss in less training time. The pre-training model is then used (called GoogLeNet) and experiment using the same settings as Kendall et al.

$$\text{loss(I)} = \left\| \hat{P} - P \right\| \tag{2}$$

4.4. Error Correction

In this section, an error correction method is introduced to correct the initial position predictions from the deep learning model. Based on the preliminary experimental results, the position estimation between the cross-camera simulated mobile phones may have a large position error, and the median position error of the measured results can be as high as 3–4 m, using one simulated mobile phone as the training sample for model training and another simulated mobile phone as the test sample for regression position. This kind of high error estimation cannot be used at all for indoor positioning, so a method has been designed to use the known angle for clustering. Since the different angles of the test samples have been found, the error results will have different distributions of displacement. After the preliminary experimental tests, the possible fixed distribution is divided into a group, and the three-axis position's prediction value from the ground exact position of each group of images is subtracted to obtain the three-axis position error value. Then, the error average $\text{P}_{\text{error}}[x,\ y,\ z]$ of each group is averaged, as shown in Formula (3). After calculating the average error of each group, the initial position error will be corrected according to the angle of the shooting at that time, and according to which angle the angle falls in the group, using the average error previously calculated. This calculation method is used to subtract the average error from the initial predicted value and to obtain the predicted value after error correction $C[x,\ y,\ z]$, as shown in Formula (4).

$$P_{error}[x,\ y,\ z] = \frac{\sum_{i=1}^{N}\left(\hat{P} - P\right)}{N} \tag{3}$$

$$C[x,\ y,\ z] = \hat{P} - P_{error} \tag{4}$$

Five types of angle grouping combinations with error correction has been designed (Figure 9). The average number of samples for each angle grouping combination should not be too small, and the sampling range should not be too centralized. The best method is to average the whole trajectory. If the above conditions are satisfied, the average error can be obtained to correct the initial prediction position effectively. For example, as shown in Figure 10, the dataset uses a cutting area with a circular trajectory. The training sample simulates Zenfone2 mobile phone images (totaling 5500), while the test sample simulates R11s mobile phone images, totaling 1250. Using 10 random position points, each position point has 10 directional images. A total of 100 samples are grouped and averaged. The experimental results will be described in detail in Section 5.7.

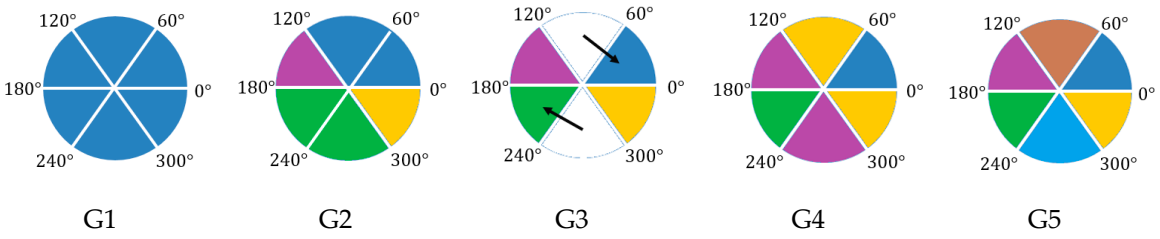


Figure 9. Five angle grouping combinations for error correction.

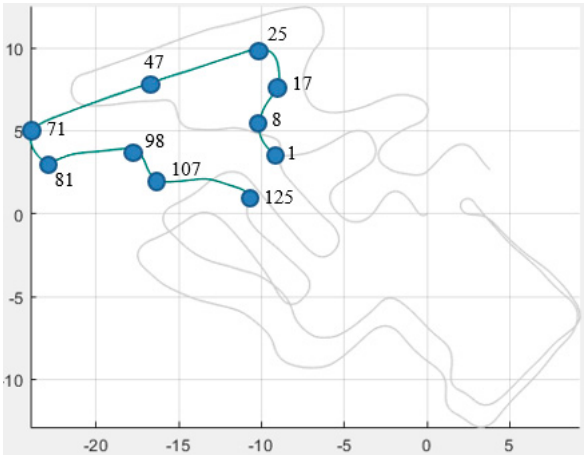


Figure 10. Average number of samples for the grouping combinations.

As shown in Figure 9, G1 is an intuitive method, wherein all angles are averaged together. The design concept of G2 to G4 is based on the preliminary experimental results. After the test samples in the range of 60° to 120° and 240° to 300° are regressed through the CNN model, their estimated positions have no fixed distribution in space relative to the ground exact position. Therefore, three different clusters were designed to carry out the experiment. G5 does not divide the angles between 60° to 120° and 240° to 300° into a single area, rather, they are a cluster. The details of each angle group are described below.

- G1: The first group is non-clustering, and the average error values of all angles are made together. This combination is the most intuitive, and its efficiency is less than that of other combinations.
- G2: Each color is a group. For example, the angle between 0° to 120° is a group. There are four groups in total, and the average error values of each group are calculated.
- G3: This group is also divided into four groups. In particular, the angles between 60° to 120° and between 240° to 300° are not used to calculate the average error. However, the angle of these two intervals will be corrected by using the average error values calculated from 0° to 60° and from 180° to 240°.
- G4: Each color is a group, such as the angles between 60° and 120° and those between 300° and 0°. There are four groups in total, and the average error values of each group are calculated.
- G5: 60° to 120° and 240° to 300° are clustered, respectively. Thus, there are six groups in total, and each group calculates its average error value.

5. Experimental Evaluation

This section collects images and geographic position information through the indoor mobile cartographic platform and simulates various mobile phone images. An underground parking lot and the South Palace Museum were chosen as the experimental sites. The position accuracy of the convolution neural network (CNN) is evaluated by varying the different image sizes, and the issues of field size, cross-camera, and error correction are discussed. All the experiments were carried out on a

Tensorflow platform. There were three hardware devices used in this experiment: Geforce GTX 1080 TI, Geforce GTX 1080, and Geforce GTX 2080 TI GPUs, which accelerated the CNN. Each CNN model was trained for 30,000 iterations.

5.1. Experimental Design

The experiment used a pre-trained model called GoogLeNet, using the Places database [33], which includes about 7 million images and 476 scene categories. The model was trained for 800 iterations. The architecture is initialized, and the pre-trained model is imported to obtain the random initial weight of the neural network. The experimental field collected the dataset of the underground parking lot and the South Palace Museum. The 23-layer CNN structure was used to train the end-to-end location identification task and regress to the camera's position. The panorama images of the indoor scene and the geographic position information were collected through the indoor mobile cartographic platform [32] acquired for the experimental model training. In this experiment, MATLAB programming was used to splice the panorama into planes, and through five parameters, including roll, pitch, yaw, focal length, and image size, to simulate various camera images with various orientations. This experiment uses the underground parking lot behind the Engineering Department Hall of the Cheng Kung Campus of National Cheng Kung University and the South Courtyard of National Palace Museum as the experimental field. The specifications and other experimental designs are described in detail in the following sections.

The experimental results are presented as errors; for each test image, the Euclidean distance is calculated after subtracting the corrected predicted value from the ground exact position, and the final error value $Error_i$ is obtained, as shown in Formula (5).

$$Error_i = \sqrt{(P[x_i]-C[x_i])^2 + (P[y_i]-C[y_i])^2 + (P[z_i]-C[z_i])^2} \tag{5}$$

There are two main methods for evaluating experiments: One uses the median error, and the other uses accuracy. The median error is used to find the median value of N ordered numbers in the dataset. The formula is $(N+1)/2$. Further, for odd values, the intermediate number is given, and for even values, the intermediate point between two intermediate values is given. As shown in Formula (6), when there are N numbers of X sequence, the formula will give two intermediate values, i.e., x^{upper} and x^{lower}, and the average of the two intermediate values will be the final result, Median(x). For example, for six values, the formula will give an index of 3.5 and average the second (upper) and third (lower) values to take the intermediate values:

$$\text{Median(x)} = \frac{x^{upper}[(N+1)/2] + x^{lower}[(N+1)/2]}{2} \tag{6}$$

The method of calculating the ratio is to design a threshold T. In N test samples, the position error is obtained by subtracting the ground exact position P from the position predicted value $\hat{P}$. If the position error is less than the threshold, it is regarded as accurate positioning. Finally, the ratio is calculated by dividing the accurate number by the total number. Formula (7) is as follows:

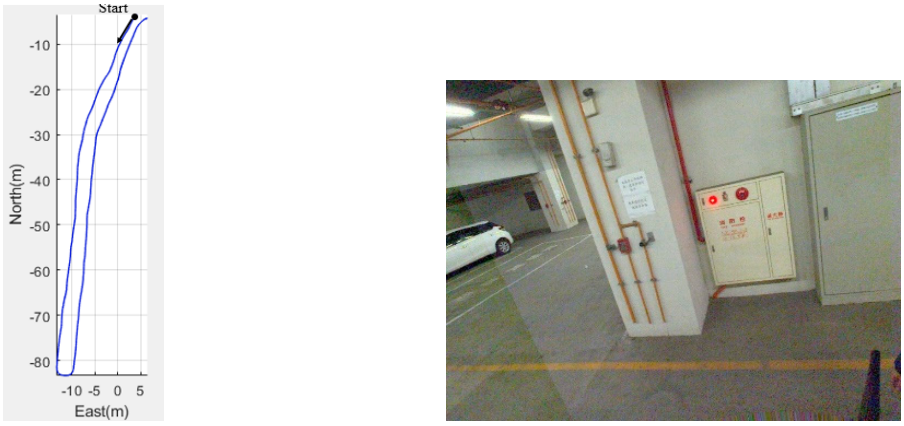
$$\begin{cases} Ratio = \sum \frac{count}{N} \times 100\% \\ count = count + 1 \text{ , if } \hat{P} - P < T \end{cases} \tag{7}$$

5.1.1. Underground Parking Lot

The underground parking lot behind the Engineering Department Hall of Cheng Kung Campus of National Cheng Kung University covers an area of about 80×15 square meters. Its trajectory is shown in Figure 11a, starting from the starting point to the bottom, then turning left to return to the starting point. Figure 11b is one of the image samples, totaling 135,240 images, including 230 positions,

each with 588 orientation angles. The orientation angle of each image is generated by the program when it simulates the mobile phone image. The details are as follows:

- Roll: −45°~45° (There are 7 types of images generated every 15°)
- Pitch: −18°~18° (There are 7 types of images generated every 6°)
- Yaw: −180°~150° (There are 12 types of images generated every 30°)



(**a**) (**b**)

Figure 11. (**a**) Trajectory of the underground parking lot; (**b**) simulated mobile phone image.

The underground parking lot environment is monotonous, and moving objects like cars and locomotives occupy a large area of the image. In harsh environments, CNN can only locate users through obvious features, such as fire hydrants, pipelines, and escape gates, which is a challenging experiment.

5.1.2. Palace Museum

The South Courtyard of the National Palace Museum covers an area of about 25 × 35 square meters. Its trajectory is shown in Figure 12a and circles the whole exhibition hall from the beginning and then turns left and back to the origin once. Figure 12b is one of the image samples. There are 46,548 images in total, including 862 positions, each with 54 orientation angles. The orientation angle of the image is generated by the program when it simulates the mobile phone image. The details are as follows:

- **Roll:** −45°~45° (There are 3 types of images generated every 45°)
- **Pitch:** −6°~6° (There are 3 types of images generated every 6°)
- **Yaw:** −180°~150° (There are 6 types of images generated every 60°)

(**a**) (**b**)

Figure 12. (**a**) Trajectory of the exhibition hall; (**b**) simulated mobile phone image.

Compared with other small areas, such as offices, kitchens, and other indoor environments, the Southern Palace Museum is dim, and some exhibits have problems with reflecting light. Like the underground parking lot mentioned in the preceding section, this area belongs to a harsh environment. For a CNN, it is very difficult to extract important eigenvalues, which will affect the position estimation. Thus, this experiment remains very exploratory.

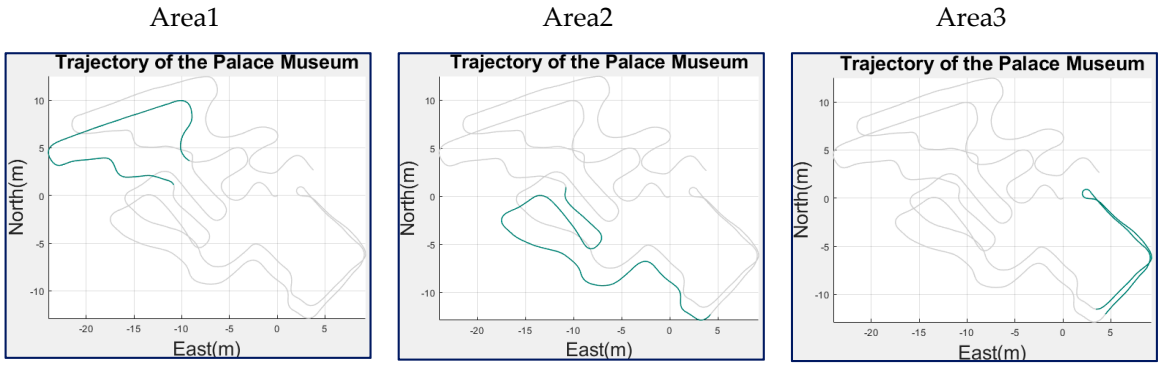
5.1.3. Different Types of Simulated Mobile Phones and Cutting Areas

Considering the image size and focal length of different simulated mobile phone models, four different mobile phone models are simulated. Images from the simulated mobile phones are shown in Figure 13. The names, focal lengths, and image sizes of the simulated mobile phones are described below:

- **Zenfone2:** The focal length is 3.8, and the image size is 4096 × 3072 pixels.
- **R11s:** The focal length is 4.10, and the image size is 1920 × 1080 pixels.
- **Tango:** The focal length is 3.38, and the image size is 3840 × 2160 pixels.
- **Zenfone3:** The focal length is 4.04, and the image size is 3840 × 2160 pixels.

Figure 13. Images from four different simulated mobile phones.

To evaluate the model's accuracy under various area sizes, three kinds of regional cutting were performed for the field of the South Courtyard of the Palace Museum. The trajectories of circle, bend walking, and back-and-forth are considered, as shown in Figure 14.



Area1	Area2	Area3
Training set:	Training set:	Training set:
125 × 44 = 5,500 images	159 × 44 = 6,996 images	121 × 44 = 5,324 images
Testing set:	Testing set:	Testing set:
125 × 10 = 1,250 images	159 × 10 = 1,590 images	121 × 10 = 1,210 images

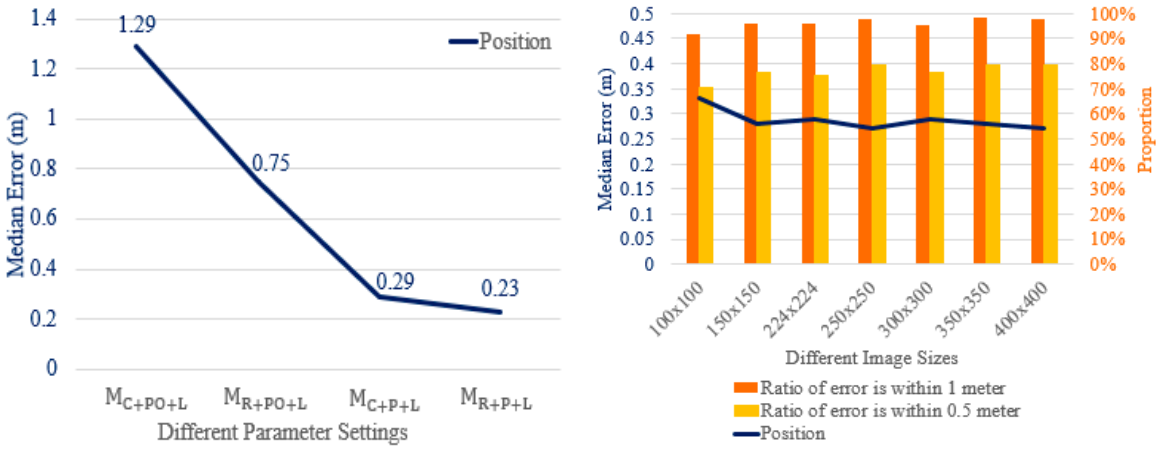
Figure 14. Circle, bend walking, and back-and-forth trajectories.

5.2. Impact of Various Image Sizes

Before the training stage, different sizes of input images are set for analysis, and the accuracy errors between the different sizes are discussed. For fairness, the loss function of PoseNet not only adjusts to these settings but also retains the original version of the loss function. The following describes the meaning of each character. M represents the architecture of PoseNet. C and R represent the image processing. C represents the image size, resized to 455 × 256 pixels (with the center cropped to 224 × 224 pixels), and R represents directly resizing the image to 224 × 224 pixels. Loss functions are divided into P and PO. P means only calculating position loss, while PO means calculating position and orientation loss. Finally, L stands for loading the pre-training model. The following details describe the settings of each image size. The experiments in the following two fields all use the cross-validation method to cut the data into five alternate training and testing models. Thus, the images of each size will be trained five times to obtain five models and then averaged to achieve data fairness.

- $\mathbf{M_{C+PO+L}}$: Setting up the original paper. This model uses the image processing method to crop the image into 224 × 224 pixels. The loss function calculates the position and orientation and then loads the pre-training model.
- $\mathbf{M_{R+PO+L}}$: This model uses the image processing method to directly resize the image into 224 × 224 pixels. The loss function calculates the position and orientation and then loads the pre-training model.
- $\mathbf{M_{C+P+L}}$: This model uses the image processing method to crop the image into 224 × 224 pixels. The loss function calculates the position and then loads the pre-training model.
- $\mathbf{M_{R+P+L}}$: This model uses the image processing method to directly resize the image into 224 × 224 pixels. The loss function calculates the position and then loads the pre-training model.
- $\boldsymbol{N \times N}$**:** The image is directly resized into $N \times N$, where N is the edge length of the square image.

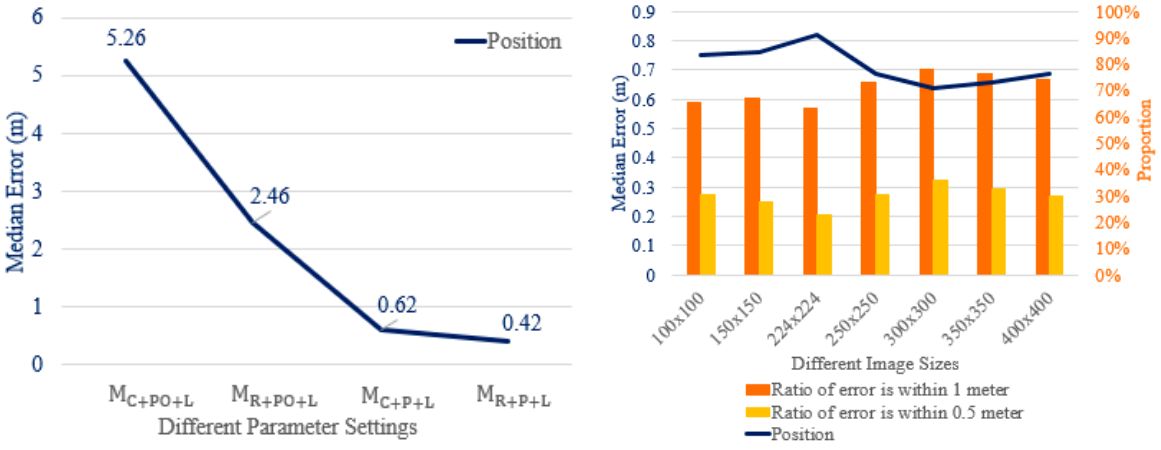
The dataset for the underground parking lots totals 135,240 images, including 230 positions, while each position point has 588 orientations. These 588 orientations are cut into five equal parts, each with about 118 orientation angles. A total of 108,330 images were trained, and 26,910 images were tested. In Figure 15, the experiments of different parameter settings (a) and different image sizes (b) are analyzed. In Figure 15a, the x-axis represents different parameter settings, and the y-axis represents the median error (unit: meters). When the pre-training model is loaded and the image size is 224 × 224 pixels, M_{R+P+L} obtains the best position for the median error, which is about 0.23 m. When the loss function is adjusted to the calculated position only, the position's median error for the M_{R+P+L} model is about 0.06 m smaller than that of the M_{C+P+L} model, and the improvement rate is about 20.6%. Compared with model M_{C+PO+L} (the setting in the original paper), the position's median error of M_{R+P+L} is smaller at about 1.06 m, and the improvement rate is as high as 82.2%. In Figure 15b, the x-axis represents different image sizes, the y-axis is biaxial, the broken-line map compares the median error on the left, and the bar chart compares the ratio on the right. The results show that the median error between 100 × 100 and 400 × 400 pixels is not much different (about 0.3 m), and the ratio of the error is almost the same. The ratio of the error in 1 m is about 97%, and the ratio of error in 0.5 m is about 75%.



(**a**) (**b**)

Figure 15. Dataset for the underground parking lots: (**a**) the position of the median errors of different parameter settings; (**b**) the position of the median errors and ratio of different image sizes.

The dataset for the Palace Museum contains 46,548 images, including 862 positions. Each position point has 54 orientations. These 54 orientations are cut into 5 equal parts, each of which has about 11 orientation angles. The training samples included 37,928 images, and the test samples included 8620 images. In Figure 16, the experiments of different parameter settings (a) and different image sizes (b) are analyzed. In Figure 16a, the *x*-axis represents different parameter settings, and the *y*-axis represents the median error (unit: meters). When the pre-training model is loaded, and the image size is 224 × 224 pixels, M_{R+P+L} obtains the best position median error, which is about 0.42 m. When the loss function is adjusted to the calculated position only, the position median error of the M_{R+P+L} model is about 0.2 m smaller than that of the M_{C+P+L} model, and the improvement rate is about 32.6%. Compared with the model M_{C+PO+L}, the setting in the original paper, the position median error of M_{R+P+L} is smaller at about 4.48 m, and the improvement rate is as high as 92%. In Figure 16b, the *x*-axis represents different image sizes, the *y*-axis is biaxial, the broken-line map compares the median error on the left, and the bar chart compares the ratio on the right. The results show that the median error between 100 × 100 and 400 × 400 pixels is not much different at about 0.71 m, and the ratio of the error is almost the same. The ratio of error in 1 m is about 71%, and the ratio of error in 0.5 m is about 30%.



(**a**) (**b**)

Figure 16. Dataset for the Palace Museum: (**a**) the position median errors of different parameter settings; (**b**) the position median errors and ratios of different image sizes.

5.3. Impact of Various Architecture

In this section, the position accuracy of the training models under different architectures is discussed. In addition to the PoseNet architecture used, another model, ResNet50, is used to modify the output position error of the architecture to compare it with our model. ResNet is divided into several layers for scholars to use. ResNet50 (with 50-layers) was selected as the experimental object, because the full connection layer of the last layer is 2048, the same size as the full connection layer of PoseNet. Table 2 compares the position error and error ratio of ResNet50, PoseNet, and our model. The database uses the zenfone2 simulation image of the South Palace Museum for 30,000 iterations. In order to be fair, PosNet and our model do not load a pretrained model. The results show that the position error of our model is much smaller than that of resnet50.

Table 2. Position median error and error ratio under different architectures.

Architecture	Position Error (m)	Ratio (<1 m)	Ratio (<0.5 m)
ResNet50	11.27	0.3%	0%
GoogLeNet (PoseNet)	1.13	43.3%	13%
GoogLeNet (Ours)	0.82	63.3%	23%

5.4. Impact of Various Area Sizes

The experimental results of Section 5.2 show that M_{R+P+L} is the best image input size before the training stage. Then, the errors of various simulated mobile phones in the Palace Museum field are discussed. The whole field and the cutting area are experimentally studied separately. The cutting area chooses the first circle trajectory to present the experimental results. Finally, the cross-platform error accuracy of different simulated mobile phones is tested in the cutting area. Table 3 provides training for each simulated mobile phone image to test the median error of the location of the same simulated mobile phone (in meters). Based on the results of the four simulated mobile phone tests in the whole field, the median error of the simulated Zenfone2 mobile phone location is the smallest. Its error is 0.42 m, and 91.3% of the test sample error is within 1 m, while the error of the simulated R11s mobile phone is the largest; its error is 1.14 m, which is only 43.6% of the test sample error (within 1 m). The reason for these errors may be related to the size and focal length of the simulated mobile phone image. The image of the simulated Zenfone2 mobile phone is far from the object and contains a great deal of information. The CNN model can calculate the precise location based on useful information. Conversely, the image of the simulated R11s mobile phone is too close to the object, so sometimes only the local area of the object is captured. It is difficult for the CNN model to calculate an accurate position based on these local features.

Table 3. Results of the simulated mobile phones tested on the same platform.

Training Model	Position Error (m)				Ratio (<1 m)			
	Area1	Area2	Area3	All	Area1	Area2	Area3	All
Zenfone2	0.15	0.17	0.12	0.42	99.0%	98.7%	97.4%	91.3%
R11s	0.31	0.37	0.40	1.14	87.3%	77.7%	77.7%	43.6%
Tango	0.19	0.17	0.14	0.47	97.8%	95.4%	96.7%	87.3%
Zenfone3	0.18	0.20	0.18	0.58	94.7%	92.3%	92.6%	80.2%

Then, the results of the three cutting areas and the whole path in the table are compared. Moreover, reducing the field range can reduce the overall position error by about 60%. In the experiment of the four different simulated mobile phones tested on the same platform, no matter the trajectory of the circle, bend walking, or back-and-forth trajectory, the position median error between the three cutting areas does not fluctuate much. Therefore, there is no need to design a specific trajectory when

collecting images and geographic position information. The test sample only needs to be encapsulated in the training sample to obtain a high position accuracy.

Figure 17 shows the position prediction value of the same camera simulated mobile phone based on each simulated mobile phone as a training model, and yellow is the position prediction value of all test samples. Area1 is used as the dataset, and a total of 1250 images were tested. As seen in the Figure 17, the position prediction of the simulated Zenfone2 phones approximates the real track, and the predictions of the simulated Tango and Zenfone3 phones are mostly close to the track, although a few of the test samples have a large error. Compared with the other three simulated mobile phones, the position prediction value of R11s is much farther from the real trajectory, resulting in an increase in the median position error. This error may be due to the fact that the image of the simulated R11s mobile phone contains too few useful features and too few objects in the image. Moreover, it is difficult for the neural network to judge the complete position information. Therefore, some test samples have been misestimated, thereby increasing the overall median error.

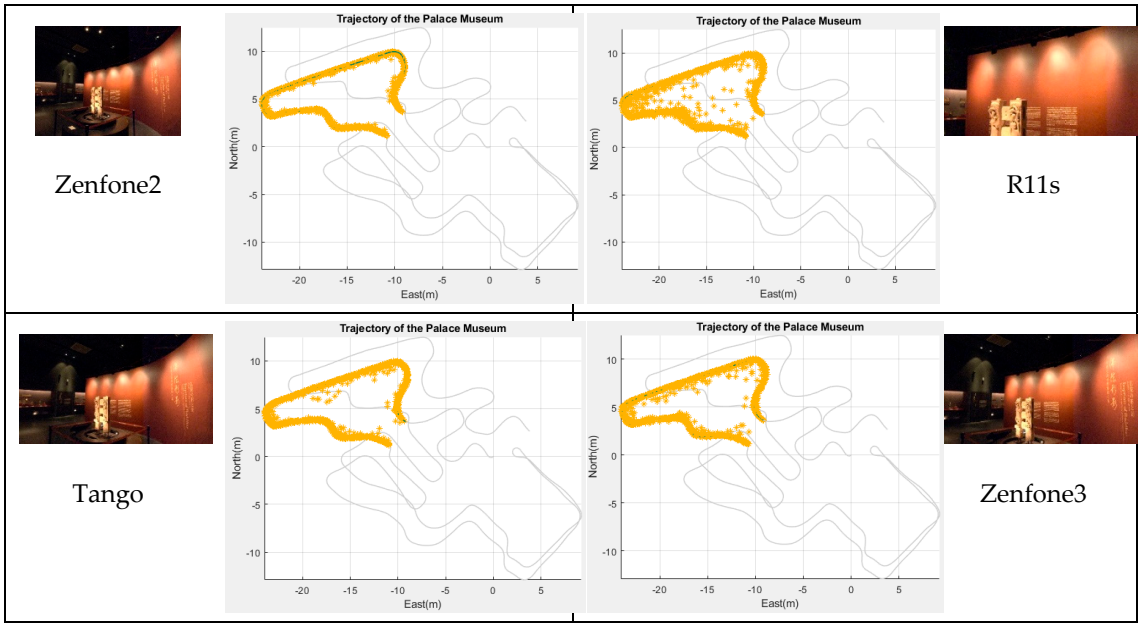


Figure 17. Using four simulated mobile phones as the training model and calculating the position prediction value of the mobile phones on the same camera.

This section explores how to re-divide the dataset into training samples and test samples via different sampling methods. During data collection, according to each piece of position information in the trajectory, the images are simulated in various orientations. The main sampling methods are divided into training and testing samples according to the orientation, but considering the impact of the other sampling methods on the neural network, the following two sampling methods were designed for the experiments:

- Random_division: All images are randomly divided.
- Position_division: Sampling is based on position. Four of the five positions are used for training samples and one for testing samples.

The images of the simulated zenfone2 mobile phone are used for the experiment, and the whole field is selected as the dataset. According to the method of using orientation for sampling, the median error of the position is 0.42 m. Based on the experimental results in Table 4, the position errors will not differ too much, regardless of whether the sampling is based on orientation, randomness, or position. Thus, in the following experiments, the method of sampling according to the orientation is still used.

Table 4. Position median error and error ratio under different sampling methods.

Training Model	Position Error (m)	Ratio (<1 m)	Ratio (<0.5 m)
Random_division	**0.38**	**94.5%**	**67.6%**
Position_division	0.41	89.3%	62.1%

5.5. Impact of the Cross Camera

Table 5 shows that each simulated mobile phone image is trained by cutting the area as a dataset to test the median position error (in meters) between the cross-camera simulated mobile phones. Based on the results, there is no significant difference in the median error of the position between the simulated Zenfone2, Tango, and Zenfone3 phones in cross-platform testing, but the error of position increases to 3–4 m when testing the image of the simulated R11s mobile phone. However, using the simulated R11s mobile phone as the training model, the images of the other three simulated mobile phones tested also rose to 3–4 m. The reason for this error may be that the image of the simulated R11s mobile phone is too different from that of the other three simulated mobile phones; that is to say, the image contains too much feature information. Using three simulated mobile phones as the training model, the neural network learns the complete image area, but the image of the simulated R11s mobile phone is too local, making it difficult for the neural network to match its features, thereby increasing errors. Conversely, using the simulated R11s mobile phone as the training model, the neural network can only learn local features, making it difficult to distinguish the larger ranges of images and positions.

Table 5. Using the cutting area as a dataset: The results of simulated mobile phones tested with the cross-camera.

Training Model	Test Data	Position Error (m)	Ratio (<1 m)	Ratio (<0.5 m)
Zenfone2	Zenfone2	**0.15**	**98.4%**	**92.2%**
	R11s	4.31	7.7%	2.5%
	Tango	0.36	87.0%	65.1%
	Zenfone3	1.01	49.4%	24.6%
R11s	Zenfone2	4.18	4.6%	1.4%
	R11s	**0.36**	**80.9%**	**62.3%**
	Tango	3.95	6.1%	1.6%
	Zenfone3	3.14	11.7%	4.2%
Tango	Zenfone2	0.26	94.1%	77.4%
	R11s	3.89	9.4%	4.0%
	Tango	**0.17**	**96.6%**	**88.9%**
	Zenfone3	0.55	76.7%	46.4%
Zenfone3	Zenfone2	0.77	62.2%	31.7%
	R11s	3.41	11.8%	4.4%
	Tango	0.47	82.9%	52.7%
	Zenfone3	**0.19**	**93.2%**	**83.4%**

Table 5 presents a small portion of the experiment with the position prediction map of all the test samples. Figure 18 shows that the position prediction values of the four simulated mobile phones are calculated using the simulated Zenfone2 mobile phone as the training model. The simulated Zenfone2 mobile phone's position prediction values are almost close to the real track, and the simulated Tango and Zenfone3 mobile phone's position prediction values are mostly close to the track. Because the images of the simulated R11s phone differ too much from those of the other three simulated mobile phones, the position prediction errors of many test samples are very large.

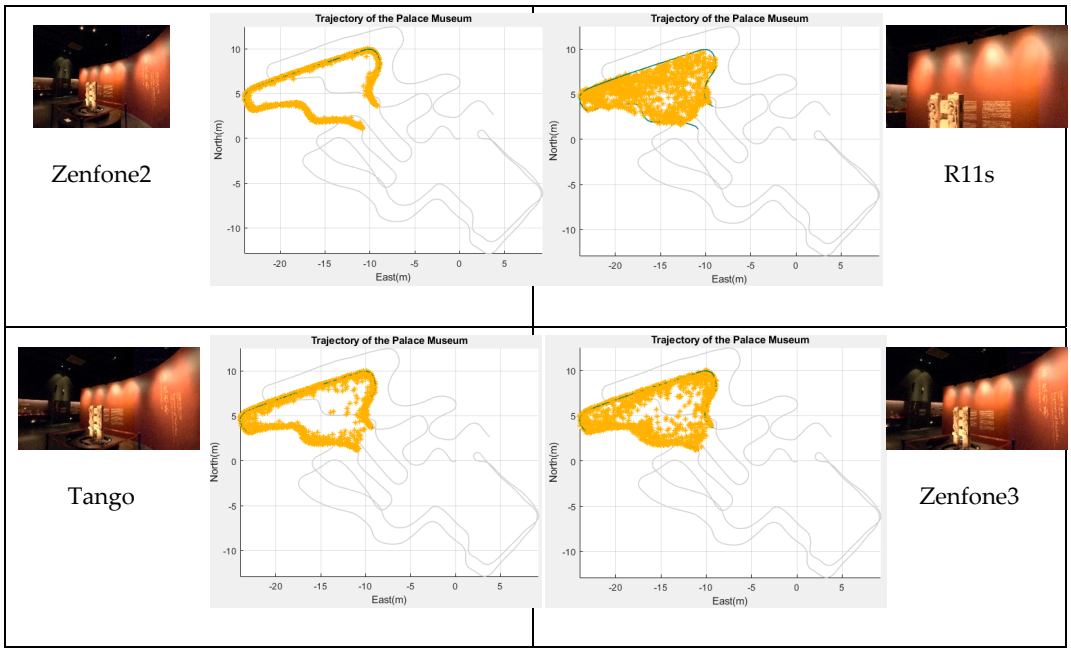
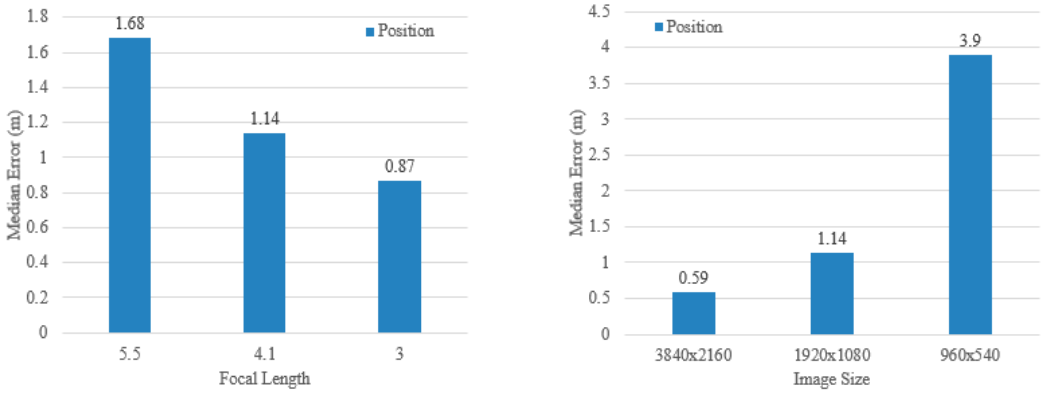


Figure 18. Using a simulated Zenfone2 mobile phone as the training model and calculating the position prediction values of the four simulated mobile phones.

In addition, this research performed some experiments for comparison with the simulated R11s mobile phone, in order to test the influence of focal length and image size on the CNN. The focal length of the simulated R11s is 4.10, and its image size is 1920 × 1080 pixels. Figure 19a illustrates an experiment for fixing the image size and adjusting the focal length. Experiments show that the larger the focal length is, the closer the distance between the image and the object is, and the worse the position prediction of model is. Figure 19b shows an experiment in which the focal length is fixed, and the image size is adjusted. This experiment shows that the image size adjustment is helpful for improving the accuracy of neural-like position prediction. However, the prediction accuracy is worse.



(**a**) (**b**)

Figure 19. (**a**) The position median errors of different focal lengths; (**b**) the position median errors of different image sizes.

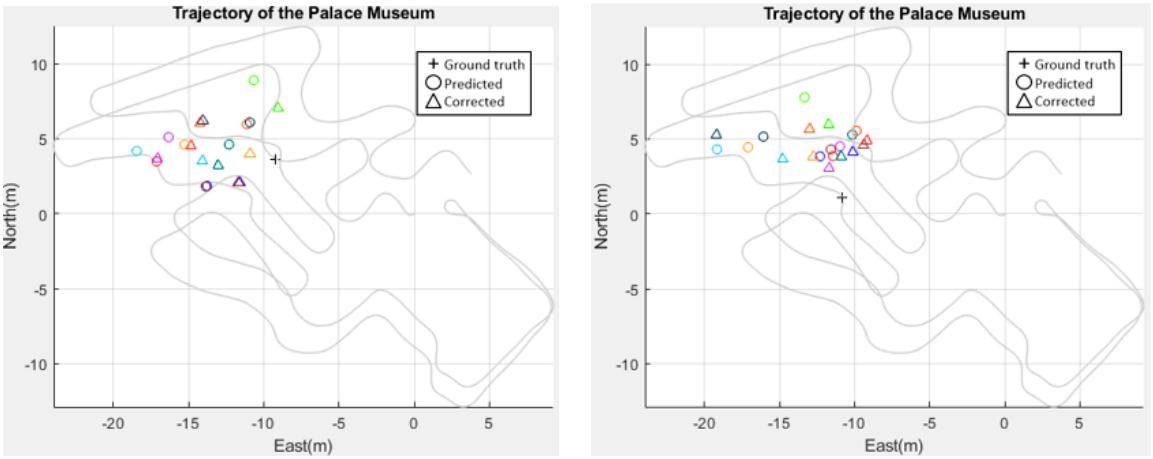
5.6. Impact of Error Correction

As Table 5 shows, as the error accuracy of cross-platform testing increases, an error correction method is designed to correct the initial position predictions from the CNN model output. The known angles are 30°, 90°, 150°, 210°, 270°, and 330°. According to the previous experiment, the dataset

choses a cutting area, Area1. The training sample features an image of the simulated Zenfone2 mobile phone, and the test sample is an image of a simulated R11s mobile phone. Ten random positions are used, and 10 orientation images are taken from each position point, so the samples total 100. These samples are used to perform grouping averaging, and the initial position error is 4.73 m. Five kinds of angle grouping are designed for the experiments. The experimental results are as follows:

- G1: The corrected median error of the position is 4.44 m, and the improvement rate is 6.1%.
- G2: The corrected median error of the position is 3.74 m, and the improvement rate is 20.8%.
- G3: The corrected median error of the position is 3.73 m, and the improvement rate is 21.1%.
- G4: The corrected median error of the position is 3.97 m, and the improvement rate is 16%.
- G5: The corrected median error of the position is 3.72 m, and the improvement rate is 21.3%.

In Figure 20, 10 test images of two positions are forecasted, and the error is corrected by the G5 method. Figure 20a shows the experimental results of the starting point of the trajectory, while Figure 20b illustrates the experimental result of the end point of the trajectory. The black plus sign is the value of the starting and ending points, blue is the initial position prediction value, and green is the position prediction value after error correction. Based on this figure, the position prediction value is close to the exact ground position after error correction, which proves that our error correction method can effectively correct the position prediction value.



(**a**) (**b**)

Figure 20. Using the G5 method to perform error correction for: (**a**) the initial position prediction of the start point; (**b**) the initial position prediction of the end point.

5.7. Experimental Summary

The original settings of PoseNet are the regression position and orientation, so the loss function calculates the loss of the position and orientation at the same time. In order to simultaneously obtain a good position and orientation, the weight between the two must be balanced; thus, the position error must be higher than the position error, which only calculates the position loss. The simulated mobile phone that the user holds has a sensor itself, and the orientation angle can be obtained by the gyroscope in the sensor, so there is no need to use a specially-designed neural network to predict the orientation. For this reason, the CNN architecture is adjusted so that the loss function only calculates the weight of the position and ultimately predicts the user's position. Experiments with different image sizes show that our new method can effectively improve positioning accuracy by about 20% to 30%. Compared with the original setup of PoseNet, our new method can effectively improve the positioning accuracy by about 80% to 90%. In the experiment of cutting the field, the field reduction of the dataset can help to reduce the overall position error. Moreover, it is unnecessary to design a special

trajectory to collect data. In addition, considering the problem that most of the images come from different types of simulated mobile phones in practical application, if the training image data and the actual positioning image are taken from different simulated mobile phones, it is easy to cause errors between different cameras. In cross-camera experiments, the size and focal length of the image have a great influence on the neural network. The objects and areas covered in the image are the main reasons for an increase in the position error. Finally, in order to further reduce the position error, this paper proposes an error correction algorithm. Through angle grouping, the error of the initial position error in each group is calculated and averaged to correct the error. Experiments show that the proposed five angle clustering methods can effectively reduce the error, and the optimal angle clustering method can reduce the position error by about 20%.

6. Conclusions and Future Work

This paper re-designed a 23-layer convolutional neural network (CNN) architecture suitable for indoor environments and adjusted the weight of loss function so that the loss function only calculates the weight of the position, focusing on accurately predicting indoor positioning. Before the training phase, the image is kept complete, and the size is resized directly as the input value of the CNN. Experiments on different image sizes show that the proposed method can effectively improve the positioning accuracy by about 20% to 30% under the same conditions of only calculating the position loss. Compared with the original setup of PoseNet, our new method can effectively improve the positioning accuracy by about 80% to 90%. Using the mobile cartographic platform with a positioning system and mapping system, the datasets are collected from the underground parking lot and the South Palace Museum. These datasets can be used for related research in the future. In practical indoor positioning applications, users mostly use the cameras of different platforms. This paper is the first to discuss the indoor positioning accuracy between different cameras. The experimental results show that the image size and focal length have great influence on a CNN. The objects and areas covered in the image are the main reasons for increases in position errors. A CNN and angle error correction have also been integrated for the first time. The experiments show that the proposed five angle grouping methods can effectively reduce errors, and the best angle grouping method can effectively improve indoor positioning accuracy by about 20%.

In the future, this research could be applied to various methods, such as image collection, image processing, and deep learning model architecture, so indoor positioning technology can develop more accurately. From the perspective of image collection, this study used a manual method to collect image data and the geographic position data needed by the deep learning model through a mobile cartographic platform. For supervised learning, obtaining a large amount of correct training material is always key. Scholars often spend a great deal of time collecting image data in the process of research. At present, there is no automatic and effective data collection mechanism. Thus, surveillance cameras, unmanned aerial vehicles (UAV), and customized Arduino devices will be able to automatically collect image data. Research on identification and deep learning will also produce a high degree of convenience. From the perspective of image processing, this paper does not analyze the moving objects that may appear in the training images. The most common moving objects in indoor scenes are cars and people. These moving objects are unnecessary features for deep neural network learning, therefore, will inevitably cause a decline in accuracy. This type of research could use neural networks, such as YOLO or Mask RCNN, to detect moving objects and ensure neural networks do not learn such features through modification to solve the problem of moving objects in images. From the perspective of the deep learning model, mobile phone photography contains many valuable parameters, such as sensor value, three-axis attitude, focal length aperture, etc. This paper does not integrate this information as a basis for prediction. Thus, some of this information could be potentially combined with position predictions from the CNN model, and a neural network model could be trained like a fully connected neural network or Long Short-Term Memory (LSTM) to obtain a more accurate position. In addition, the dataset can also increase its number of samples through image rotation or

data noise, so the learning of neural networks can be more comprehensive. Field selection also has a great impact on the neural network, such as light and shade, environmental complexity, and the number of eigenvalues. Therefore, different fields can also be experimented to test whether the neural network model can effectively achieve indoor positioning. Efficient indoor positioning may be applied to indoor intelligent parking or intelligent cities and may also be combined with augmented real-world applications. Indoor positioning is also expected to facilitate spatial information applications.

Author Contributions: Conceptualization, E.H.-C.L.; Methodology, E.H.-C.L. and J.-M.C.; Software, J.-M.C.; Validation, E.H.-C.L. and J.-M.C.; Formal Analysis, E.H.-C.L.; Investigation, J.-M.C.; Resources, E.H.-C.L.; Data Curation, J.-M.C.; Writing-Original Draft Preparation, E.H.-C.L. and J.-M.C.; Writing—Review & Editing, E.H.-C.L.; Visualization, J.-M.C.; Supervision, E.H.-C.L.; Project Administration, E.H.-C.L.; Funding Acquisition, E.H.-C.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Ministry of Science and Technology, Taiwan, grant number MOST 107-2119-M-006-028-.

Acknowledgments: The authors would like to acknowledge the financial support of the Minister of Interior, Executive Yuan of Taiwan, through National Cheng Kung University.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

References

1. Tan, J.S.-F.; Lu, E.H.-C.; Tseng, V.S. Preference-Oriented Mining Techniques for Location-Based Store Search. *Knowl. Inf. Syst.* **2013**, *34*, 147–169. [CrossRef]
2. Kendall, A.; Grimes, M.; Cipolla, R. PoseNet: A Convolutional Network for Real-Time 6-DOF Camera Relocalization. In Proceedings of the IEEE ICCV, Santiago, Chile, 7–13 December 2015; pp. 2938–2946.
3. Li, F.; Zhao, C.; Ding, G.; Gong, J.; Liu, C.; Zhao, F. A Reliable and Accurate Indoor Localization Method Using Phone Inertial Sensors. In Proceedings of the 2012 ACM Conference on Ubiquitous Computing, Pittsburgh, PA, USA, 5–8 September 2012; pp. 421–430.
4. Lan, K.C.; Shih, W.Y. An Indoor Locationtracking System for Smart Parking. *Int. J. Parallel Emergent Distrib. Syst.* **2014**, *29*, 215–238. [CrossRef]
5. Grossmann, U.; Gansemer, S.; Suttorp, O. RSSI-Based WLAN Indoor Positioning Used within A Digital Museum Guide. *Int. J. Comput.* **2014**, *7*, 66–72.
6. Subhan, F.; Hasbullah, H.; Rozyyev, A.; Bakhsh, S.T. Indoor Positioning in Bluetooth Networks Using Fingerprinting and Lateration Approach. In Proceedings of the IEEE ICISA, Jeju Island, Korea, 26–29 April 2011; pp. 1–9.
7. Engel, J.; Schöps, T.; Cremers, D. LSD-SLAM: Large-Scale Direct Monocular SLAM. In *ECCV*; Springer: Cham, Switzerland, 2014; pp. 834–849.
8. Liang, J.Z.; Corso, N.; Turner, E.; Zakhor, A. Image-Based Positioning of Mobile Devices in Indoor Environments. In *Multimodal Location Estimation of Videos and Images*; Springer: Cham, Switzerland, 2015; pp. 85–99.
9. Hao, O.; Cai, R.; Li, Z.; Zhang, L.; Pang, Y.; Wu, F. 3D Visual Phrases for Landmark Recognition. In Proceedings of the IEEE Conference on CVPR, Providence, RI, USA, 16–21 June 2012; pp. 3594–3601.
10. Agarwal, S.; Furukawa, Y.; Snavely, N.; Simon, I.; Curless, B.; Seitz, S.M.; Szeliski, R. Building Rome in A Day. *Commun. ACM* **2011**, *54*, 105–112. [CrossRef]
11. Snavely, N.; Seitz, S.M.; Szeliski, R. Photo Tourism: Exploring Photo Collections in 3D. *ACM TOG* **2006**, *25*, 835–846. [CrossRef]
12. Wu, C. Towards Linear-Time Incremental Structure from Motion. In Proceedings of the IEEE International Conference on 3D Vision-3DV, Seattle, WA, USA, 29 June–1 July 2013; pp. 127–134.
13. Shotton, J.; Glocker, B.; Zach, C.; Izadi, S.; Criminisi, A.; Fitzgibbon, A. Scene Coordinate Regression Forests for Camera Relocalization in RGB-D Images. In Proceedings of the IEEE Conference on CVPR, Portland, OR, USA, 23–28 June 2013; pp. 2930–2937.

14. Ng, A.; Ngiam, J.; Foo, C.Y.; Mai, Y.; Suen, C.; Coates, A.; Maas, A.; Hannun, A.; Huval, B.; Wang, T.; et al. Convolutional Neural Networks. 2013. Available online: http://ufldl.stanford.edu/tutorial/supervised/ConvolutionalNeuralNetwork/ (accessed on 28 August 2013).
15. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going Deeper with Convolutions. In Proceedings of the IEEE Conference on CVPR, Boston, MA, USA, 7–12 June 2015; pp. 1–9.
16. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the IEEE Conference on CVPR, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
17. Godard, C.; Aodha, O.M.; Brostow, G.J. Unsupervised Monocular Depth Estimation with Left-Right Consistency. In Proceedings of the IEEE Conference on CVPR, Honolulu, HI, USA, 21–26 July 2017; pp. 270–279.
18. Zhou, T.; Brown, M.; Snavely, N.; Lowe, D.G. Unsupervised Learning of Depth and Ego-Motion from Video. In Proceedings of the IEEE Conference on CVPR, Honolulu, HI, USA, 21–26 July 2017; pp. 1851–1858.
19. Ashraf, I.; Hur, S.; Park, Y. Application of Deep Convolutional Neural Networks and Smartphone Sensors for Indoor Localization. *Appl. Sci.* **2019**, *9*, 2337. [CrossRef]
20. Kang, J.; Lee, J.; Eom, D.S. Smartphone-Based Traveled Distance Estimation Using Individual Walking Patterns for Indoor Localization. *Sensors* **2018**, *18*, 3149. [CrossRef] [PubMed]
21. Wang, X.; Wang, X.; Mao, S. Deep Convolutional Neural Networks for Indoor Localization with CSI Images. *IEEE Trans. Netw. Sci. Eng.* **2018**. [CrossRef]
22. Mittal, A.; Tiku, S.; Pasricha, S. Adapting Convolutional Neural Net-works for Indoor Localization with Smart Mobile Devices. In Proceedings of the 2018 on GLSVLSI, Chicago, IL, USA, 23–25 May 2018; pp. 117–122.
23. Niitsoo, A.; Edelhäuβer, T.; Mutschler, C. Convolutional Neural Networks for Position Estimation in TDoA-Based Locating Systems. In Proceedings of the International Conference on IPIN, Nantes, France, 24–27 September 2018; pp. 1–8.
24. Bregar, K.; Mohorčič, M. Improving Indoor Localization Using Convolutional Neural Networks on Computationally Restricted Devices. *IEEE Access* **2018**, *6*, 17429–17441. [CrossRef]
25. Furukawa, Y.; Curless, B.; Seitz, S.M.; Szeliski, R. Towards Internet-Scale Multi-View Stereo. In Proceedings of the IEEE Computer Society Conference on CVPR, San Francisco, CA, USA, 13–18 June 2010; pp. 1434–1441.
26. Bengio, Y.; Courville, A.; Vincent, P. Representation learning: A Review and New Perspectives. *IEEE TPAMI* **2013**, *35*, 1798–1828. [CrossRef] [PubMed]
27. Oquab, M.; Bottou, L.; Laptev, I.; Sivic, J. Learning and Transferring Mid-Level Image Representations Using Convolutional Neural Networks. In Proceedings of the IEEE Conference on CVPR, Columbus, OH, USA, 23–28 June 2014; pp. 1717–1724.
28. Razavian, A.S.; Azizpour, H.; Sullivan, J.; Carlsson, S. CNN Features Off-the-Shelf: An Astounding Baseline for Recognition. In Proceedings of the IEEE Conference on CVPRW, Columbus, OH, USA, 23–28 June 2014; pp. 806–813.
29. Kendall, A.; Cipolla, R. Modelling Uncertainty in Deep Learning for Camera Relocalization. In Proceedings of the IEEE ICRA, Stockholm, Sweden, 16–21 May 2016; pp. 4762–4769.
30. Kendall, A.; Cipolla, R. Geometric Loss Functions for Camera Pose Regres-sion with Deep Learning. In Proceedings of the IEEE Conference on CVPR, Honolulu, HI, USA, 21–26 July 2017; pp. 5974–5983.
31. Walch, F.; Hazirbas, C.; Leal-Taixé, L.; Sattler, T.; Hilsenbeck, S.; Cremers, D. Image-Based Localization Using LSTMs for Structured Feature Correlation. In Proceedings of the IEEE ICCV, Venice, Italy, 22–29 October 2017; pp. 627–637.
32. Chiang, K.W.; Tseng, Y.H.; Lu, H.C. *The Project of Mobile Platform Development on Surveying and Mapping Technology*; Department of Land Administration, Ministry of the Interior: Tainan City, Taiwan, 2018.
33. Zhou, B.; Lapedriza, A.; Xiao, J.; Torralba, A.; Oliva, A. Learning Deep Features for Scene Recognition Using Places Database. In Proceedings of the 27th International Conference on Neural Information Processing Systems, Montreal, QC, Canada, 8–13 December 2014; Volume 1, pp. 487–495.