

# A Deep Learning-Based Person Search System for Real-World Camera Images

Chih-Ta Yen<sup>1\*</sup>, Guan-Yu Chen<sup>2</sup>

<sup>1</sup> Department of Electrical Engineering, National Taiwan Ocean University, Taiwan

<sup>2</sup> Department of Electrical Engineering, National Formosa University, Taiwan  
chihtayen@gmail.com, 10665143@gm.nfu.edu.tw

## Abstract

A person search system was developed to identify the query person from images captured by cameras at four scenes in the study. This study analyzed three network architectures called Model Basic, Model One, and Model Two. To verify the validity of the model design, the models in the public data set and in the recorded system data set were compared to determine whether the results of the proposed model exhibited consistent performance between the camera images from the public data set and the recorded, unprocessed system data set. The detected pedestrian images then underwent distance matching relative to query person images by using the online instance matching (OIM) loss function. Based on Model Basic, Model One and Model Two were designed to further improve accuracy by incorporating different convolutional neural networks. In CUHK-SYSU data set, the testing results of Model Basic, Model One and Model Two achieved the accuracies of 72.38%, 75.96% and 75.32%, respectively. The testing results of Model Basic, Model One, and Model Two with the system data set achieved accuracies of 63.745%, 68.80%, and 69.33%, respectively.

**Keywords:** Deep learning, Object detection, OIM loss function, Person search, ResNet50

## 1 Introduction

Cameras are ubiquitous, and dynamic camera images have become part of the mainstream approach for monitoring systems. Through long-term operation and numerous recorded dynamic images, real-time tracking or tracking by searching for specific persons, events, or objects has become possible. Because the human eye lies at the core of tracking identification, dynamic image monitoring is only feasible through human visual observation. However, such a use of human eye is constrained by limited visual attention, difficulties in data analysis, numerous missing reports or false alarms, and slow response. Thus, observation with the human eye cannot cope with processing the exponentially growing quantity of data. In real-time tracking, to identify the target to be tracked, personnel must be constantly deployed to continually watch real-time images. In tracking by searching, a substantial number of personnel is required to rewind and examine recorded videos before identifying the target to be searched. Such monitoring systems are not cost effective,

whether in time or human resources. Thus, many scholars have aimed to realize real-time online tracking and highly accurate object detection, addressing the aforementioned problems using object detection and tracking in machine vision. In particular, deep learning has been used to replace traditional machine learning computations to improve identification accuracy.

Since 2012, following the increasingly wide application of convolutional neural networks (CNNs), the field of person reidentification (Re-ID) has had a resurgence. Bromley et al. proposed the Siamese neural network, and some variants have been developed [1]. The Siamese structure is composed of two independent CNN network models that extract features separately and compare the features. Subsequently, the output results of the two networks are combined to produce the final result. Chopra et al. modified the Siamese network structure and used it for identifying image similarity [2]. Ahmed et al. integrated the results of previous studies and designed a special CNN model for person Re-ID. By inputting a group of person images to two identical CNN networks and using the verification of loss functions to train parameters, the model more accurately identifies whether two images are of the same person [3]. Ding et al. proposed a method for identifying whether two images are of the same person based on person data from different scenarios. Their method mainly involves the use of similarity measurement to enable the neural network to learn image features. The aim is to minimize the feature distance for the same person while maximizing the feature distance between different persons [4]. Based on existing person Re-ID architectures, recent studies have focused on improving the formulation of environmental variables. Because, in practice, different cameras are installed at different distances, camera image resolution vary, which makes person comparison difficult. To address the problem, Li et al. proposed a person Re-ID method for cameras with low resolution (LR). Their aim was to compare LR person images with their high-resolution counterparts in an existing data set. To enhance accuracy in person identification from traditional LR images with different resolutions, image processing techniques are used prior to the input of training data, where LR images are rescaled to produce images with greater resolution and clarity before they are input into the model for computation. However, this method is inapplicable to Re-ID problems. To solve the LR image problem, a unified multiscale Re-ID learning architecture is required, instead of direct comparison and identification using Re-ID [5-6].

Person Re-ID is aimed at solving problems of identifying a person whose images are shot by different cameras at various angles. As a rapid developing field of research, person Re-ID has been widely applied in dynamic image monitoring and multimedia applications, such as pedestrian detection [7], cross-camera visual tracking [8], and object tracking analysis [9]. However, images of the same person may exhibit different patterns; this is because of different camera shooting conditions, such as perspective, posture, lighting, shading, identification rate, camera setup, and a messy background. These environmental variables make pedestrian detection more difficult. To solve this challenging problem, researchers have proposed many Re-ID data sets and algorithms to gradually increase basic identification performance. Nevertheless, much work remains to be done before these tools can be applied.

Traditional person Re-ID comprises two steps: pedestrian detection and reidentification. First, pedestrian detection is conducted to identify whether a pedestrian is present in the image. Subsequently, and second, when a pedestrian is present, a linear classification method is adopted to perform reidentification. Because of its linear classification characteristics, this method often has the problems of overfitting data, an inability to identify unknown data, and difficulties in feature collection, which bottleneck the development of the technology. To address these problems, many studies have investigated object detection and tracking in machine vision. Deep learning has been used in place of traditional machine learning computation to improve identification accuracy. However, person Re-ID and pedestrian detection have different purposes. The purpose of person Re-ID is to classify individual persons to distinguish between different persons and to make the identity of persons traceable at any time point. The purpose of pedestrian detection is to detect features belonging to the pedestrian and locate the pedestrian's position. At present, most person Re-ID algorithms emphasize considerations pertaining to the query and gallery, in which most images are cropped [10-11]. In real-life application, query pedestrians must be searched in a complex and realistic scenario. Xiao et al. proposed combining person Re-ID and pedestrian detection to develop a person search architecture that can be applied in real-world images [12]. To facilitate detection in real application scenarios, a comprehensive set of environmental interference factors are retained in the image in addition to the query person.

This study designed a person Re-ID search and tracking system for images captured from multiple cameras. Studies on person Re-ID and tracking more generally have mainly discussed the themes of collecting and using person data; the improvement of model identification accuracy; the analysis of differences between camera-captured person images; and the effect on identification accuracy from environmental factors, such as lighting, shading, posture, and perspective. These themes have been discussed theoretically in the absence of a complete system flow for combining the methods explored and for applying them in practical person Re-ID and tracking. The goal of this study was to use cross-camera real-time images and recorded images as the input data for training and testing. Subsequently, semisupervised learning was adopted to label a small proportion of the training data set. Thereafter, the deep learning model was improved to obtain greater identification accuracy. Finally, the user interface was designed, culminating in a complete system. When a user

inputs the query person, the system identifies each image that possibly contains the person and informs the user of the probability that a given pedestrian is the query person.

## 2 System Flow and Model Flow

The system is composed of a data set and model. The data input method involves two channels to facilitate computation. The following two items are uploaded to the server for computation: first, a data set that contains the real scenario images of the search person from (Camera 1, ...Camera N) and second, the image of the search person. The model uses the deep learning approach of faster region-based CNN (R-CNN) object detection as the basis in combination with the Re-ID method. The output is a match with a person. The matching threshold values are set in the model. Matching results greater than the threshold values represent successful matches and are selected and stored, whereas those results lower than the threshold values are considered failed matches and stored (Figure 1).

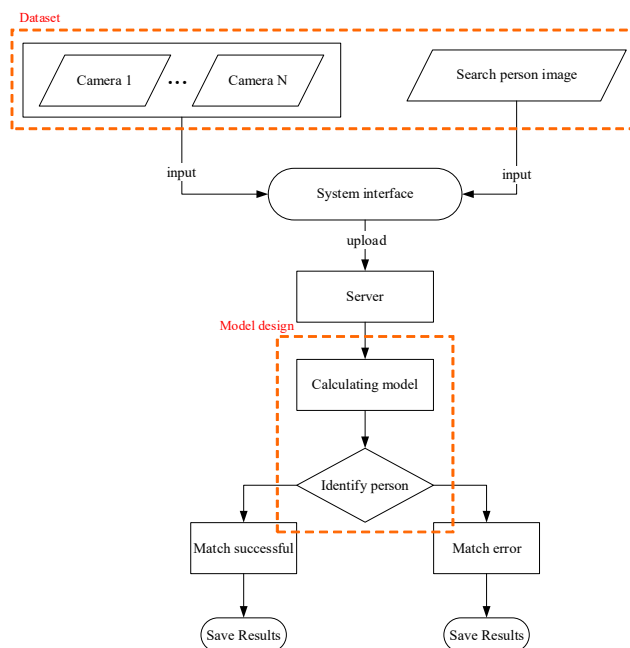


Figure 1. System flow

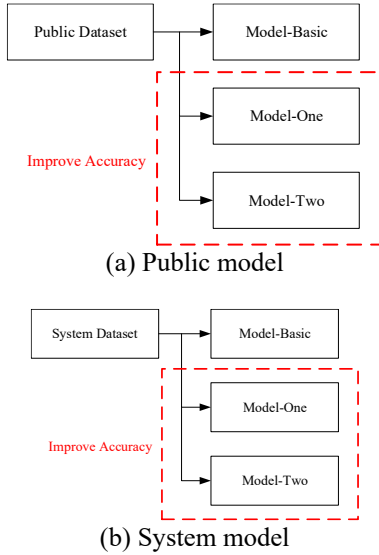
All computations were conducted on the Ubuntu 16.04 operating system; the computer had an Intel i7-2600 four-core CPU, 12 GB of DDR3 RAM, and a GTX 1060 GPU with 6 GB of video memory (Table 1).

Table 1. Computer settings

Computer	
Operating system	Ubuntu 16.04
Processor	Intel i7-2600
Memory	DDR3 12G
Graphics card	GTX1060

The model design flow is illustrated in Figure 2. First, the public data set CUHK-SYSU [12] was used for training and testing to increase the reliability of the model design. The model was divided into three categories, which were Model Basic, Model One, and Model Two. Model One and Model

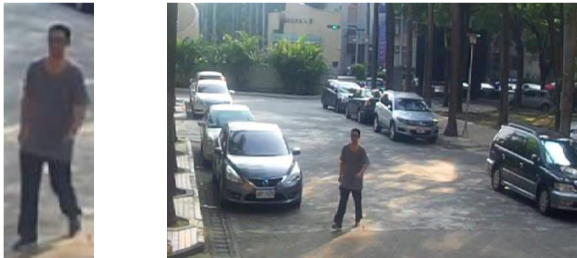
Two were models modified from Model Basic for improved accuracy, as shown in Figure 2(a). Subsequently, the data set collected in this study (referred to as the system data set) was used to train and test the model, which was again divided into Model Basic, Model One, and Model Two. Finally, the analysis results of the public data set and the system data set were compared to identify whether they exhibited consistent tendencies. Among the analysis results of the system data set, the optimal solution was selected as the system model, as shown in Figure 2(b).



**Figure 2.** Model design flow

### 3. Data Set and Model Design

Present-day Re-ID references and algorithms mostly focus on matching cropped pedestrian images; thus, nearly all images in the data set were cropped [10-11], as shown in Figure 3(a). However, these images are unsuitable for testing in person search because real-world images are affected by various background and external factors, as shown in Figure 3(b). The purpose of person search is to identify the query person from all these factors.

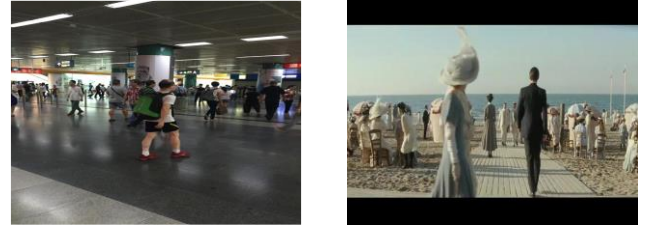


(a) Cropped images (b) Real scenario images  
**Figure 3.** Data set difference for images

#### 3.1 Public Data Set

To solve the problem, the person-search data set CUHK-SYSU, which Xiao et al. applied to analyzing real-world images, was adopted as the public data set. CUHK-SYSU is a large-scale person-search data set collected under diverse scenarios. The data set comprises two sections: street snapshots and movie or TV series snapshots. In street snapshots, hundreds of scenario images are collected from

various perspectives and under various lighting conditions, identification rates, shadings, and backgrounds. The data set of snapshots contains snapshots from movies or television series. These two snapshot data sets provide diverse scenarios and challenging perspectives. In total, 8432 labeled persons and 18 184 images are provided by the data set. Some CUHK-SYSU images of street snapshots (StreetSnap) and movie or TV series snapshots (Movie&TV) are shown in Figure 4(a) and Figure 4(b), respectively.



(a) StreetSnap (b) Movie&TV  
**Figure 4.** CUHK-SYSU images

The architecture of the CUHK-SYSU [12] data set is detailed in Table 2. In total, 11260 images were used for training, where 55 272 pedestrian bounding boxes were labeled. The bounding boxes contained pedestrian positions  $(x_1, y_1, x_2, y_2)$ . A total of 5532 pedestrian identities (id) were labeled.


**Table 2.** CUHK-SYSU data set architecture

Data set	Images	Pedestrians	Identities
StreetSnap	12,490	7,5845	6057
Movie&TV	5,694	2,0298	2,375
Training	11,260	5,5272	5,532
Test	6,978	4,0871	2,900
Overall	18,184	9,6143	8,432

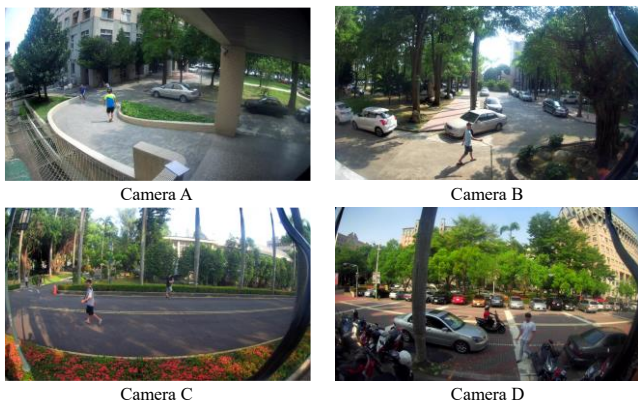
#### 3.2 System Data Set

The person images in the system data set were collected through a method similar to that used for the public data set. The images were primarily collected using a SJ4000 camera that supports image capture at 4K resolution. The maximum image resolution was 12 million pixels. Image data were collected at 60 frames per second. The shooting environment mimicked those of surveillance cameras along streets and roads. The person image data comprising the system data set were mainly collected from four sites on a campus. The camera specifications are listed in Table 3.

**Table 3.** SJ4000 Camera Specifications

	Specifications
Camera appearance	
Image resolution	Supports 4K video recording
Camera pixels	12 million
Frame per second	60Fps
USB port	USB2.0

In response to the system's needs, this study adopted four scenarios named Camera A, Camera B, Camera C, and Camera D, each for a given image capture scenario, when collecting pedestrian images for training the network to identify persons (Figure 5). The installation of each camera was purposeful. To capture images affected by simple to complex environmental factors, the cameras were installed at locations that were near to distant from the query person, respectively. Cameras A and B were nearest and furthest to the query person, respectively, where images from Camera B are likely to have problems with lighting and messy backgrounds. These different cameras were used to determine whether the proposed network is robust under such complex environmental factors.



**Figure 5.** Images corresponding to the four scenarios

The system data set contained images of 54 query persons in four real-world scenarios and was used to compare the accuracy of identifying query persons under different scenarios. System tests were conducted, which involved a small-scale data set based on the 54 query persons. The data set contained 164 images for the four scenarios. The images were evenly distributed among the four scenarios. Each person with the same pedestrian identity was present in three images in a scenario on average. Among the 164 images, 272 pedestrians and 54 identities were identified. The data set architecture is shown in Table 4.

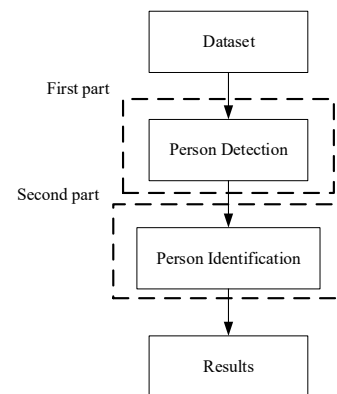
**Table 4.** System data set architecture

Data set	Images	Pedestrians	Identities
Camera A	39	64	11
Camera B	43	73	16
Camera C	40	67	14
Camera D	42	68	13
Training	114	209	38
Test	50	63	16
Overall	164	272	54

The major difference between the system data set and the public data set lies in whether the same person appears for different scenarios. In the public data set, a person only appears at a single scenario; thus, none of the persons had the same identity. By contrast, in the system data set, the same person may appear repeatedly at different scenarios. The challenge faced by the present study was ensuring successful identification of the same person under different scenarios. This study's analysis results on the advantages and disadvantages of each scenario can be used by future studies to improve the identification rate.

## 4 Model Design

The model design mainly comprised two parts: person detection and person identification (Figure 6). First, the data are input into the data set. Then, for person detection, all pedestrians that appear in the real scenario images are located with their coordinate positions through the use of bonding boxes. Subsequently, for person identification, all pedestrians undergo matching computation with the search person to determine the probabilities that any given pedestrian is the search person. The model design is completed by combining the two parts.



**Figure 6.** Brief flow of model

As required by the present study's experiment, the network must comprise two inputs of query person images and gallery images [12]. Therefore, the model network was designed based on the Siamese network [1]. The Siamese network satisfied this study's requirements: it is a similarity measurement network that is mainly applied to cases with diverse classes of samples, where a few samples are in each category. Traditional classification methods require specific information on the category each sample is in. Each sample must be labeled specifically and a massive sample data set is required. Traditional methods are unsuited to tasks with few samples classified in diverse categories. The basic operation of the Siamese network is depicted in Figure 7. First, a pair of data points  $X_1, X_2$  were input into the network. Through a pair of identical CNN networks, a pair of features  $G_W(X_1), G_W(X_2)$  sharing an identical weight  $W$  were extracted. After the data were input, the similarity of the features was computed to determine the probability that  $X_1$  and  $X_2$  belonged in the same category. When the feature distance between  $X_1$  and  $X_2$  decreases, their probability of being in the same category increases and the loss decreases. By contrast, when the feature distance between  $X_1$  and  $X_2$  increases, their probability of being in the same category decreases and the loss increases.

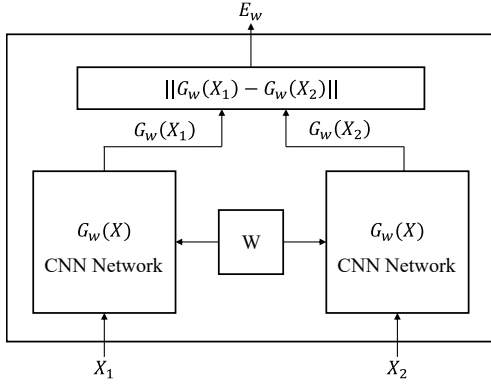


Figure 7. Siamese network architecture

## 4.1 Model Basic

The design principles of the network model pertained to detecting persons in a real-world image, matching the detected pedestrian with the search person, determining the person's identity, and calculating the probability of the detected pedestrian being the search person. The pedestrian detection network was based on faster R-CNN [13-15]. The person Re-ID method features the use of the online instance matching (OIM) loss function [16]. The joint network design for person search was developed by combining the two methods (Figure 8).

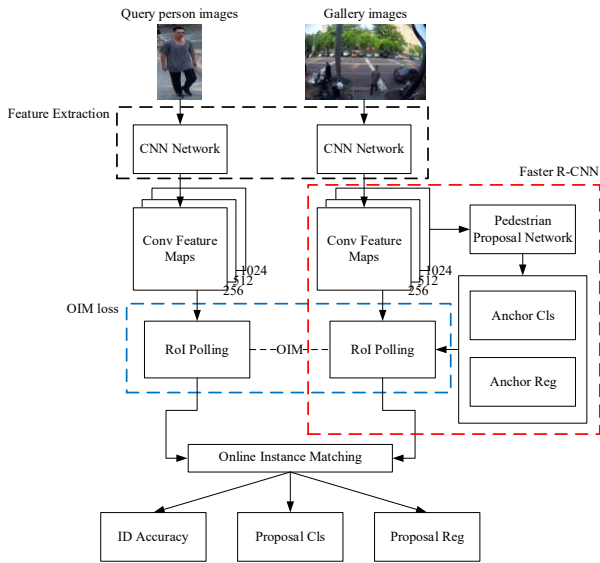


Figure 8. Model Basic network architecture

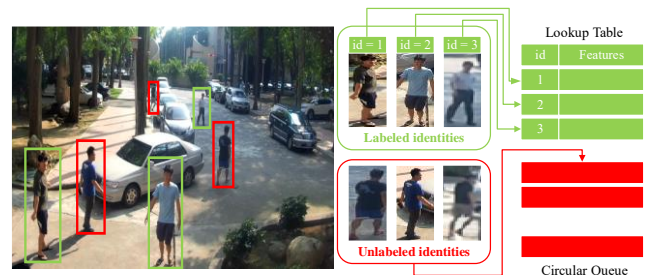
The network model of Model Basic is described with respect to the architecture and tuned hyperparameters in Figure 8, as follows.

- (1) The ResNet50 model was selected for the CNN network architecture [11]. The 256–1024 kernels in the layers from Convolution1\_1 to Convolution4\_3 in the ResNet50 model were output as the features. The query person images and gallery images were input into the CNN network to extract features and generate a series of convolution feature maps as the output.
- (2) In the pedestrian proposal network, each position of the feature maps was set with nine anchor bounding boxes. The softmax classifier was used to determine the reliability of pedestrian existence in the bounding boxes.

Regression analysis was used to modify the position of the bounding boxes. After a regression analysis of anchors containing pedestrians, nonmaximum suppression was conducted to reduce the number of repeated bounding boxes. Finally, these bounding boxes were used as the output of the pedestrian proposal network, which is called Proposal.

- (3) The region of interest (ROI) pooling mapped the bounding boxes obtained from Proposal onto the feature maps. After the coordinates of the bounding boxes on the feature maps were obtained, pooling was conducted to acquire the maximum or mean values. If  $L$  categories of different query persons exist in the training set, after matching Proposal with the query person, Proposal may present three prediction results. A successful match is one where Proposal is assigned to a category identity from 1 to  $L$  categories (one among the  $L$  categories); such a Proposal is termed a Proposal with label identities. By contrast, proposals with a failed match are called Proposals with unlabeled identities and Proposals containing nonpedestrian or background clutter. The three results are input into the ROI pooling and mapped to the feature maps.
- (4) The person Re-ID method adopts the OIM, which was constructed by Xiao et al. for person Re-ID in a joint network [16]. Figure 9(a) depicts the OIM detection results, which are represented in green (containing label identities) and red (containing unlabeled identities) bounding boxes. Two supportive structures, namely the lookup table (LUT) and circular queue (CQ), were used on the OIM, as shown in Figure 9(b). In general person Re-ID models, the cropped person images result in the existence of only positive samples. Therefore, the models were less generalizable and thus less suited to practical use.

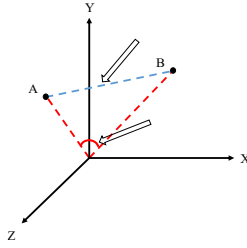
1. LUT: The LUT is where the positive samples are stored; it comprises eigenvectors of label identities. To determine the similarity between the label identities and the query person, subsequent calculations only require a comparison between the distances of the mini-batch samples of the query person and the LUT samples of the label identities.
2. CQ: CQ is where negative samples are stored. It is composed of the eigenvectors of unlabeled identities. To determine the similarity between the unlabeled identities and the query person, subsequent calculations only require a comparison between the distances of the mini-batch samples of the query person and the CQ samples of the unlabeled identities.



(a) OIM detection results (b) LUT and CQ used on the OIM

Figure 9. OIM operating architecture

- (5) In OIM, the distance comparison uses the cosine distance to reduce the computational load. The cosine distance differs from the Euclidean distance (where the absolute distance of each point is used to compare the similarity of images) in that the cosine distance uses the angles between vectors to calculate image similarity, where a greater angle indicates lower similarity (Figure 10). The cosine distance is defined in (1).  $A_i$  and  $B_i$  represent the components of vectors A and B, respectively.



**Figure 10.** Cosine distance and Euclidean distance

$$\cos \theta = \frac{\sum_{i=1}^n A_i \times B_i}{\sqrt{\sum_{i=1}^n (A_i)^2} \times \sqrt{\sum_{i=1}^n (B_i)^2}} \quad (1)$$

The purpose of OIM is to distinguish between persons with different identities, specifically by reducing the feature distance between images of the same person while increasing the feature distance between images of different persons. First, in (2),  $V$  represents the LUT extracted features;  $L$  represents the number of persons with different identities, and  $D$  represents the feature dimensions that are extracted for a single person.

$$V \in R^{L \times D} \quad (2)$$

In (3),  $x$  is the features extracted from the mini-batch samples, and  $M$  is the mini-batch size.

$$x \in R^{D \times M} \quad (3)$$

Consequently, the similarity  $d_l$  between the mini-batch samples and all persons with label identities can be calculated using (4).

$$d_l = V^T x \quad (4)$$

In addition to label identities, many unlabeled identities are valuable aids to feature learning. They can be safely used as the negation category of all label identities, as shown in (5), where  $U$  is the CQ extracted features and  $Q$  represents the queue size. Per [11], the queue size was set at 5000. After each iteration, new eigenvectors were pushed into the queue, and they eject outdated eigenvectors to maintain the loop of 5000 computations.  $D$  is the feature dimensions extracted for a single person.

$$U \in R^{D \times Q} \quad (5)$$

On basis of (5), the similarity  $d_u$  between the mini-batch samples and persons with unlabeled identities is calculated using (6).

$$d_u = U^T x \quad (6)$$

According to the LUT and CQ, the probability of a mini-batch  $x$  belonging to the  $i$ th category identity is calculated. The probability  $p_i$  of  $x$  belonging to a label identity is calculated using (7).  $\tau \in [0,1]$  is used to maintain the flatness of the probability distribution. The probability  $q_i$  of  $x$  belonging to an unlabeled identity is calculated using (8).

$$p_i = \frac{\exp(v_i^T x / \tau)}{\sum_{j=1}^L \exp(v_j^T x / \tau) + \sum_{k=1}^Q \exp(u_k^T x / \tau)} \quad (7)$$

$$q_i = \frac{\exp(u_i^T x / \tau)}{\sum_{j=1}^L \exp(v_j^T x / \tau) + \sum_{k=1}^Q \exp(u_k^T x / \tau)} \quad (8)$$

To identify the expected value of the optimal parameter solutions of the model and to maximize the OIM probabilities, the statistical maximum likelihood estimation method was used to acquire the maximized expected log-likelihood, as shown in (9).  $p_i$  is the predicted value of the distribution.

$$L = E_x [\log p_i] \quad (9)$$

Subsequently, the gradient of  $L$  to  $x$  can be expressed in (10).

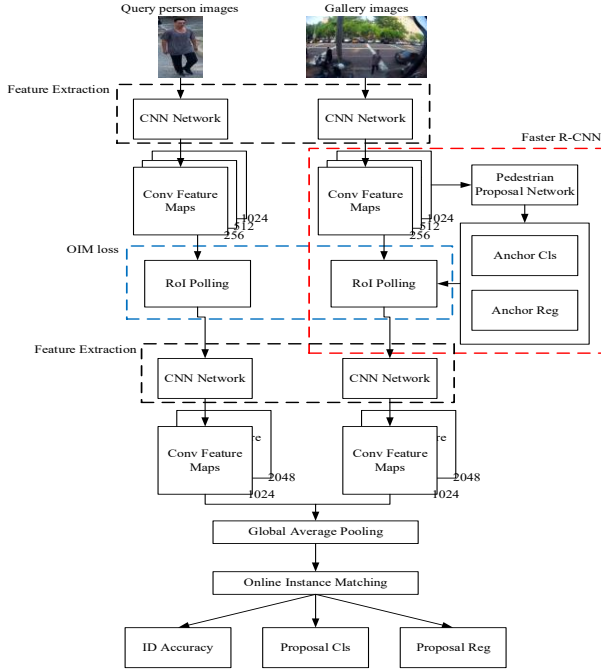
$$\frac{\partial L}{\partial x} = \frac{1}{\tau} \left[ (1 - p_i) v_i - \sum_{j=1, j \neq i}^L p_j v_j - \sum_{k=1}^Q q_k u_k \right] \quad (10)$$

According to the equation, the OIM loss effectively compared the mini-batch samples with all persons with label and unlabeled identities. The results reduced the feature distance between persons with same identities and maximized the feature distance between persons with different identities. The person matching analysis results and the regression prediction results for person detection bounding boxes were thus obtained.

## 4.2 Model One

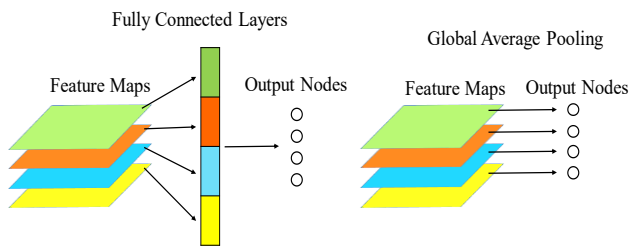
Model One is a modification of Model Basic. Because person identification revolves around analyzing similarities between different persons, more detailed local features are required to obtain improved identification results. Therefore, Model One was designed to obtain improved local features. After the ROI pooling was output, a six-layer CNN network was added to double the kernel number from 1024 to 2048, thereby generating 2048 feature maps. The model is aimed at obtaining subtle eigenvectors between localized pedestrians and the query persons and at examining whether more detailed eigenvectors improve the identification of similarity between pedestrians and the query persons. The increase in the kernel number increased the number of eigenvectors and corresponding overfitting rates. Therefore, global average pooling was added to reduce the likelihood of overfitting. Thereafter, the data were input to OIM to facilitate subsequent computation (Figure 11) [17]. The general basic CNN procedure is composed of convolutional, pooling, and fully connected layers. The present study adopted the global average pooling approach proposed by Lin et al. to replace the final fully connected layers [18]. That study noted two advantages of using global average pooling: (1) it effectively

mitigates the incidence of model overfitting and (2) it reduces the computational load of the model parameters. Therefore, the final step in the training involved the input of all eigenvectors obtained from the network into global average pooling to complete the model training.



**Figure 11.** Model One network architecture

The difference comparison for global average pooling is shown as follows. As shown in Figure 12(a), the outputs of the fully connected layers are eigenvectors of all the feature maps. As shown in Figure 12(b), the global average pooling outputs are the mean values of the eigenvectors of each feature map. The number of feature maps is equal to that of the outputs.

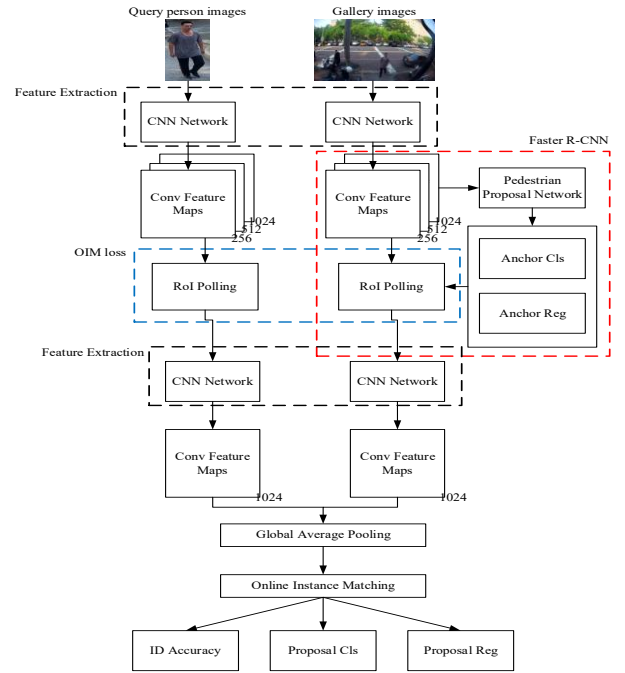


(a) Fully connected layer (b) Global average pooling  
**Figure 12.** Architecture comparison

### 4.3 Model Two

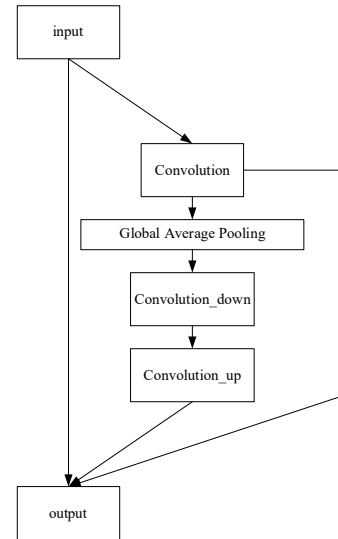
Similar to Model One, Model Two is a modification of Model Basic. Because person identification revolves around the similarity between different persons, detailed local features are required to obtain improved identification results. In Model Two, a six-layer CNN network was also added following the ROI pooling output. However, the CNN network was designed using the squeeze-and-excitation (SE) network method [19]. The model was modified without changing the kernel number (Figure 13). Thus, Convolution\_down and Convolution\_up  $1 \times 1$  convolution kernels were added to the convolution layer to integrate eigenvectors of different sizes. The output can yield three

eigenvectors. Thereafter, the axpy function  $\mathbf{a} * \mathbf{x} + \mathbf{y}$  was used to calculate the final eigenvalue of the output.



**Figure 13.** Model Two network architecture

Figure 14 shows the SE network architecture with the axpy function added.



**Figure 14.** SE Network architecture [19]

## 5 Experimental Procedures and Results

First, Caffe was selected among numerous deep learning frameworks as the framework for writing the network model; this was because Caffe has multiple supports and a fast operating speed. It is suitable for the person search system. The version of multi-GPU Caffe based on OpenMPI was adopted because this version supports multicalculation. Before installing Caffe, certain libraries must be installed (Table 5), which provided the settings used in this experiment.

**Table 5.** Library version

Library	Version
Python	2.7
CUDA	8.0
Cudnn	5.1
OpenMPI	1.8.8
boost	1.58

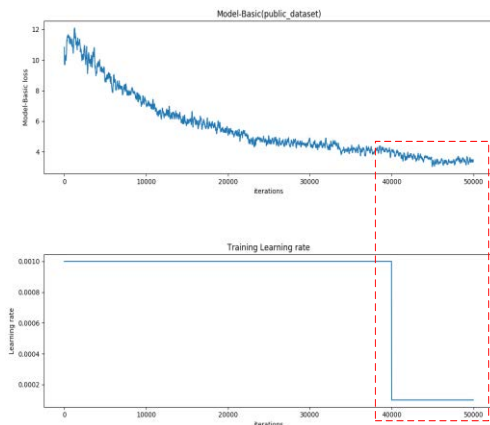
## 5.1 Public Data Set

The training data were inserted into the network for training to define the training parameters (Table 6). This experiment adopted the following training parameters: 50 000 iterations and an initial learning rate of 0.001. The display was enabled every 20 iterations.

**Table 6.** Training parameters for public data set

Parameter	Value
Iterations	50000
Learning rate	0.001
Display	20
Gamma	0.1
Stepsize	40000

The experiment executed three person search network model architectures in the public data set to facilitate training and testing. Three experimental results were obtained. Figure 15 shows the convergence curves of Model Basic after 50 000 iterations of the public data set.

**Figure 15.** Model Basic's training loss and learning rate curves for public data set

As learning rate increases, learning speed increases; and as learning rate decreases, local learning efficiency increases. Therefore, the experimental learning rate did not involve a fixed parameter of 0.001; instead, the learning rate changes with iterations to further converge the loss. Thus, (11) was applied to change the learning rate and converge the loss after further reduction of the loss.

$$Learning\ rate_{step} = (base\_lrgammar)^{\lfloor floor(iter/step) \rfloor} \quad (11)$$

As shown in the red area in Figure 14, gamma represents the variation value of the learning rate in each iteration; stepsize represents the iteration when the learning rate changes.

Thus, when the iteration reached 40 000, (11) is employed to further change the learning rate; convergence is enabled after further reduction of the loss. Figure 16 shows the convergence curves of Model One after 50 000 iterations of the public data set.

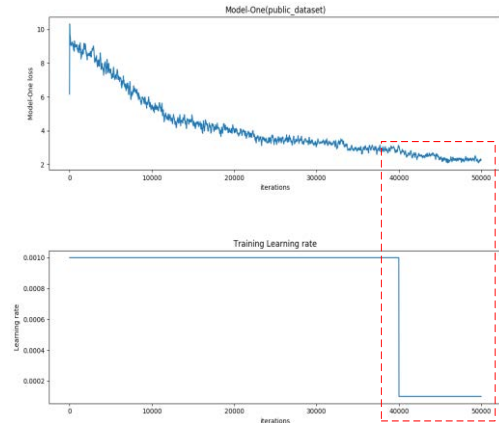
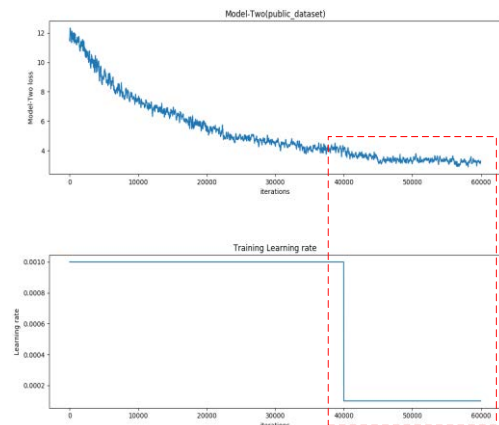
**Figure 16.** Model One's training loss and learning rate curves for public data set

Figure 17 shows the convergence curves of Model Two after 60 000 iterations of the public data set.

**Figure 17.** Model Two's training loss and learning rate curves for public data set.

The training time and total testing time of the three networks with the public data set are shown in Table 7. Model Basic had simpler network layers compared with Model One and Model Two; thus, the eigenvalues were fewer and the training time and total testing time were the shortest. In Model One, a six-layer CNN network was added and the kernel number increased to obtain more local eigenvalues. Therefore, the online parameter computation load increased, and the training time and total testing time of Model One were longer than those of Model Basic. In Model Two, the dimensional changes of Convolution\_down and Convolution\_up were added to the network to acquire eigenvalues of different sizes. This method was found to face challenges in convergence; thus, the number of iterations increased to 60 000, thereby prolonging the training time to longer than those of Model Basic and Model One. Because the proposed CNN network differs from the conventional CNN network, which only acquires the smallest eigenvalues, by adding eigenvalues of different sizes, the computation is longer. Thus, the training time of Model Two was the longest.

**Table 7.** Time analysis of three network models for the public data set

Network model	Training time	Total testing time
Model Basic	9 hr 43 min 02 s	35 min 07 s
Model One	24 hr 3 min 05 s	1 hr 3 min 36 s
Model Two	29 hr 54 min 02 s	1 hr 21 min 32 s

The three networks were compared in terms of evaluation indicators for the public data set model (Table 8). Recall is the proportion of all targets being correctly classified as the query person. Average precision (AP) is the quotient obtained by dividing the accuracy of all query persons in the category by the total number of images containing the category of query persons. The mean AP (mAP) represents the value of averaged accuracy calculated on all categories. The mAP of Model One and Model Two increased relative to that of Model Basic, verifying that the networks of Model One and Model Two were well designed. In Model One, the more detailed eigenvalues improved the mAP. In Model Two, the combinations of eigenvalues of different sizes also improved mAP. Methods based on Model One and Model Two had nearly identical effectiveness, which improved mAP. Thus, the system data set was used to determine which model was more suited to the needs of the system environment.

**Table 8.** Evaluation indicators for three network models for the public data set

Network model	Recall	Ap	mAP
Model Basic	78.96%	70.28%	72.38%
Model One	76.68%	71.90%	75.96%
Model Two	76.02%	70.69%	75.32%

## 5.2 System Data Set of Self-recorded Images

The training data were input into the network for training, and the training parameters are detailed in Table 9. The training parameters of the present experiment were for 8000 iterations. The learning rate was set initially at 0.001, and the display was enabled every 20 iterations.

**Table 9.** Training parameters for system data set

Parameter	Value
Iterations	8000
Learning rate	0.001
Display	20
Gamma	0.1
Stepsize	6000

The experiment involved the execution of three person search models, with distinct architectures, in the system data set to facilitate training and testing. Three experimental results were thus generated. Figure 18 displays the convergence curves of Model Basic after 8000 iterations on the system data set.

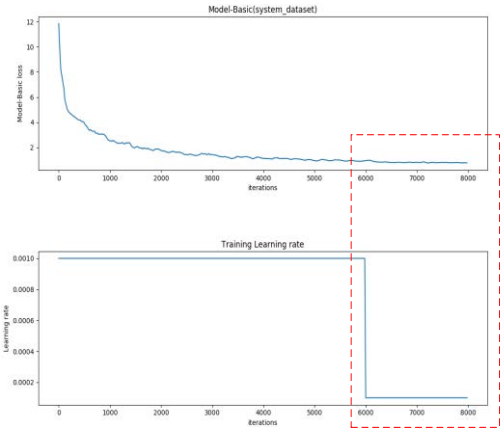
**Figure 18.** System data set Model Basic training loss and learning rate curves

Figure 19 displays the convergence curves of Model One after 8000 iterations on the system data set.

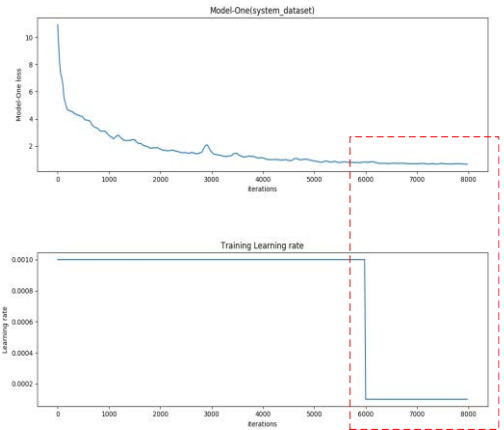
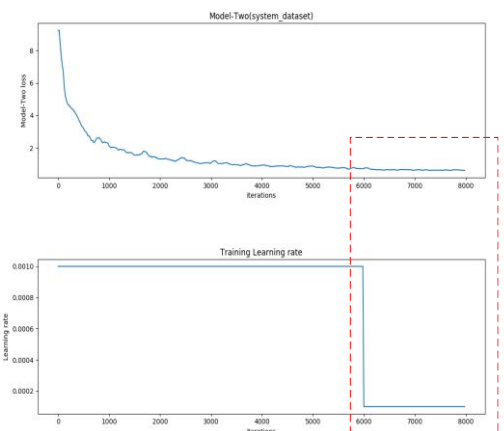
**Figure 19.** System data set Model-One training loss and learning rate curves

Figure 20 displays the convergence curves of Model Two after 8000 iterations on the system data set.

**Figure 20.** System data set Model Two training loss and learning rate curves

The training time for each of the three networks based on the system data set was then compared. As shown in Table 10, The results revealed that the training time of Model Basic was the shortest, corresponding to the result obtained for the public data set. The training times of Model One and Model Two

were similar, suggesting a nearly identical computational time for the two models. However, in the public data set, Model Two needed to undergo 10 000 more iterations than did the other networks to achieve convergence. Thus, the convergence of Model Two becomes increasingly challenging and requires additional iterations as the data quantity increases. When the data quantity was lower, the training times of Model One and Model Two were equivalent.

**Table 10.** Time analysis of three network models for the system data set

Network model	Training time
Model Basic	1 hr 11 min 01 s
Model One	1 hr 13 min 33 s
Model Two	1 hr 13 min 33 s

In the following test for the system data set, the mAP was calculated without distance ranking. Instead, the person search system was tested by determining the system accuracy in identifying the query person in the four scenarios. The testing method is illustrated in Figure 21. The search person was input for person analysis in the four scenarios.



**Figure 21.** Query person and four scenario examples in system tests

Table 11 lists the ten query persons for person search in the test of Model Basic's network for the system data set; the table also lists the successful matching rate and average accuracy of successful matches in the four scenarios. For example, the average accuracy of Query Person 1 being correctly identified in the four scenarios was 60.50%.

**Table 11.** Model Basic's person search accuracy for the system data set

System test	Model Basic's person search accuracy in test for the system data set				
	Camera A	Camera B	Camera C	Camera D	Average accuracy
Query person 1	80%	45%	51%	66%	60.50%
Query person 2	83%	47%	64%	69%	66%
Query person 3	82%	51%	55%	63%	62.75%
Query person 4	73%	48%	53%	73%	61.75%
Query person 5	80%	55%	73%	74%	70.50%
Query person 6	75%	46%	53%	48%	55.50%
Query person 7	69%	41%	60%	63%	58.25%
Query person 8	74%	51%	47%	62%	59%

Query person 9	81%	53%	70%	75%	70%
Query person 10	80%	65%	70%	82%	74.25%

Table 12 lists the four scenarios for person search in the test of the Model Basic network model for the system data set. The results in Table 11 were substituted into (12) to calculate the successful matching rate of query persons in each scenario. The results revealed that the query persons were most easily identified in the Camera A scenario (average accuracy: 78%), whereas the query persons were most unlikely to be identified in the Camera B scenario (average accuracy: 50%).

$$\frac{\text{Camera } x(\text{Query persons 1 to 10 in sum})}{10}, x \in A, B, C, D \quad (12)$$

**Table 12.** Model Basic's successful matching rate of each scenario for the system data set

Four scenarios	Model Basic's successful matching rate of each scenario			
	Camera A	Camera B	Camera C	Camera D
Average accuracy of successful matching	78%	50%	60%	68%

Table 13 lists the ten query persons for person search in the test of Model One's network for the system data set; the table also lists the successful matching rate and average accuracy of successful matches in the four scenarios. For example, the average accuracy of Query Person 1 being correctly identified in the four scenarios was 74%.

**Table 13.** Model One's person search accuracy for the system data set

System test	Model One's person search accuracy in test for the system data set				
	Camera A	Camera B	Camera C	Camera D	Average accuracy
Query person 1	82%	68%	72%	74%	74%
Query person 2	85%	45%	61%	69%	65%
Query person 3	83%	53%	52%	75%	65.75%
Query person 4	83%	45%	66%	73%	66.75%
Query person 5	79%	61%	72%	67%	69.75%
Query person 6	79%	58%	61%	65%	65.75%
Query person 7	70%	44%	65%	62%	60.25%
Query person 8	86%	59%	72%	73%	73%
Query person 9	85%	53%	78%	73%	72%
Query person 10	82%	66%	77%	79%	76%

Table 14 lists the four scenarios for person search in the test of Model One's network for the system data set. The results on matching accuracy in each scenario were substituted into (12). The results revealed that the query persons were most easily identified in the Camera A scenario (average accuracy: 81%), whereas the query persons were most unlikely to be identified in the Camera B scenario (average accuracy: 55%).

**Table 14.** Model One person search successful matching rate in each scenario for the system data set

	Model One's successful matching rate of each scenario			
Four scenarios	Camera A	Camera B	Camera C	Camera D
Average accuracy of successful matching	81%	55%	68%	71%

Table 15 lists the ten query persons for person search in the test of Model Two's network for the system data set; the table also lists the successful matching rate and average accuracy of successful matches in the four scenarios. For example, the average accuracy of Query Person 1 being correctly identified in the four scenarios was 75%.

**Table 15.** Model Two's person search accuracy for the system data set

	Model Two's person search accuracy in test for the system data set				
System test	Camera A	Camera B	Camera C	Camera D	Average accuracy
Query person 1	85%	68%	75%	72%	75.00%
Query person 2	89%	54%	56%	68%	67%
Query person 3	87%	51%	55%	69%	65.50%
Query person 4	81%	60%	66%	71%	69.50%
Query person 5	83%	57%	66%	69%	68.75%
Query person 6	84%	55%	61%	64%	66.0%
Query person 7	69%	53%	60%	61%	60.75%
Query person 8	89%	60%	69%	76%	74%
Query person 9	85%	59%	73%	70%	72%
Query person 10	86%	68%	72%	77%	75.75%

Table 16 lists the four scenarios for person search in the test of Model Two's network for the system data set. The results on matching accuracy in each scenario were substituted into (12). The results revealed that the query persons were most easily identified in the Camera A scenario (average accuracy: 84%), whereas the query persons were most unlikely to be identified in the Camera B (average accuracy: 70%).

**Table 16.** Model Two's successful matching rate in each scenario for the system data set

	Model Two's successful matching rate of each scenario			
Four scenarios	Camera A	Camera B	Camera C	Camera D
Average accuracy of successful matching	84%	59%	65%	70%

Finally, the evaluation indicators of the three networks in the public data set model were compared (Table 17). The self-recorded system data set had fewer data points and lower category diversity relative to the public data set; therefore, the Recall value of the system data set was higher than that of the public data set. Almost all detected images were of pedestrians.

In addition, when the kernel number was changed to 2048, the Recall and Ap values were the lowest. The Recall and Ap values of 1024 kernels were higher than those of Model One. The comparison of the public data set evaluation indicators revealed that when data quantity and complexity were high, the detailed local eigenvalues of Model One yielded the most effective Ap values. However, in scenarios with low data quantity and complexity, the different eigenvalues in Model Two facilitated the achievement of the most effective Ap values. In the test on person search using the system data set, among the three designed network models (namely, Model Basic, Model One, and Model Two), Cameras A and B had the highest and lowest, respectively, average accuracy and successful matching rates. This was because Camera A was closer to the pedestrian and its images had less background and environmental noise than did Cameras B, C, and D. Camera B was more affected by lighting, shooting distance, and environmental noise compared with Cameras A, C, and D. The average accuracy was highest in Model Two at 69.33% and lowest in Model Basic at 63.75%. Corresponding to the model design, Model One and Model Two generated results identical for both the public and system data sets and effectively improved the accuracy of Model Basic.

**Table 17.** Evaluation indicators used in analyzing the three network models for the system data set

Network model	Recall	Ap	mAP
Model Basic	86.21%	80.85%	63.75%
Model One	82.76%	80.35%	68.80%
Model Two	87.66%	81.19%	69.33%

## 6 Conclusion

The locations of the cameras are quite difference, how to find the optimal models in different scenarios are most important. To obtain the optimal network model, three network models, namely, Model Basic, Model One, and Model Two, were used for analysis. The model design concepts were verified through the public data set CUHK-SYSU and the system data set. Models were designed based on Model Basic through two channels of the Siamese network. First, all pedestrians were detected among real-world images for person search before the search person image was incorporated for distance calculation and matching. Pedestrian detection was conducted using faster R-CNN. The OIM loss function was used to match distance calculation results. Model Basic combines the two methods to complete person search tasks. Model One and Model Two were designed to improve the accuracy of Model Basic. Model One incorporated an additional six-layer CNN network and constantly increased the kernel number to obtain more numerous and more detailed local features. Model Two also incorporated a six-layer CNN network; however, it differed from the more detailed local features in Model One, Model Two was combined with feature values with different sizes to attain more comprehensive outputs.

In the public data set, Model Basic, Model One, and Model Two achieved 72.83%, 75.96%, and 75.32% accuracy, respectively. In the system data set, Model Basic, Model One, and Model Two achieved 63.75%, 68.80%, and 69.33%

accuracy, respectively. Different results generated from images affected by different environmental factors were verified. For example, Camera A images and Camera B images had the highest and lowest successful matching rates, respectively. According to the different characteristics of Model Basic, Model One, and Model Two, two models were adopted for backend computation in the final system. When the data quantity and complexity is low at the initial stage, Model Two is most suitable; as the data size and complexity increase, Model One is more suitable and should be used to accelerate data training time.

## Acknowledgment

We thank for supporting of the Ministry of Science and Technology MOST (grant no. MOST 108-2221-E-150-022-MY3, MOST 110-2634-F-019-002, MOST 111-2221-E-019-074) and the National Taiwan Ocean University. And we also thank for editor kind coordination. Moreover, we are grateful the reviewers for constructive suggestions.

## References

- [1] J. Bromley, J. W. Bentz, L. Bottou, I. Guyon, Y. LeCun, C. Moore, E. Säckinger, R. Shah, Signature verification using a Siamese time delay neural network, *International Journal of Pattern Recognition and Artificial Intelligence*, Vol. 7, No. 4, pp. 669-688, August, 1993.
- [2] S. Chopra, R. Hadsell, Y. LeCun, Learning a similarity metric discriminatively, with application to face verification, *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, San Diego, CA, USA, 2005, pp. 539-546.
- [3] E. Ahmed, M. Jones, T. K. Marks, An improved deep learning architecture for person re-identification, *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, USA, 2015, pp. 3908-3916.
- [4] S. Ding, L. Lin, G. Wang, H. Chao, Deep feature learning with relative distance comparison for person re-identification, *Pattern Recognition*, Vol. 48, No. 10, pp. 2993-3003, October, 2015.
- [5] X. Li, W. Zheng, X. Wang, T. Xiang, S. Gong, Multi-Scale Learning for Low-Resolution Person Re-Identification, *2015 IEEE International Conference on Computer Vision (ICCV)*, Santiago, Chile, 2015, pp. 3765-3773.
- [6] Y. Wu, J. Lim, M. Yang, Online Object Tracking: A Benchmark, *2013 IEEE Conference on Computer Vision and Pattern Recognition*, Portland, OR, USA, 2013, pp. 2411-2418.
- [7] C. C. Loy, T. Xiang, S. Gong, Multi-camera activity correlation analysis, *2009 IEEE Conference on Computer Vision and Pattern Recognition*, Miami, FL, USA, 2009, pp. 1988-1995.
- [8] X. Wang, Intelligent multi-camera video surveillance: A review, *Pattern recognition letters*, Vol. 34, No. 1, pp. 3-19, January, 2013.
- [9] X. Wang, K. Tieu, E. L. Grimson, Correspondence-Free Activity Analysis and Scene Modeling in Multiple Camera Views, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 32, No. 1, pp. 56-71, January, 2010.
- [10] W. Li, R. Zhao, T. Xiao, X. Wang, DeepReID: Deep Filter Pairing Neural Network for Person Re-identification, *2014 IEEE Conference on Computer Vision and Pattern Recognition*, Columbus, OH, USA, 2014, pp. 152-159.
- [11] L. Zheng, L. Shen, L. Tian, S. Wang, J. Wang, Q. Tian, Scalable Person Re-identification: A Benchmark, *2015 IEEE International Conference on Computer Vision (ICCV)*, Santiago, Chile, 2015, pp. 1116-1124.
- [12] T. Xiao, S. Li, B. Wang, L. Lin, X. Wang, End-to-End Deep Learning for Person Search, April, 2016. <https://arxiv.org/abs/1604.01850v1>
- [13] R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation, *2014 IEEE Conference on Computer Vision and Pattern Recognition*, Columbus, OH, USA, 2014, pp. 580-587.
- [14] R. Girshick, Fast R-CNN, *2015 IEEE International Conference on Computer Vision (ICCV)*, Santiago, Chile, 2015, pp. 1440-1448.
- [15] S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 39, No. 6, pp. 1137-1149, June, 2017.
- [16] T. Xiao, S. Li, B. Wang, L. Lin, X. Wang, Joint Detection and Identification Feature Learning for Person Search, *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, 2017, pp. 3376-3385.
- [17] K. He, X. Zhang, S. Ren, J. Sun, Deep Residual Learning for Image Recognition, *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, 2016, pp. 770-778.
- [18] M. Lin, Q. Chen, S. Yan, Network In Network, *2nd International Conference on Learning Representations, ICLR 2014*, Banff, AB, Canada, 2014, pp. 1-10.
- [19] J. Hu, L. Shen, G. Sun, Squeeze-and-Excitation Networks, *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, 2018, pp. 7132-7141.

## Biographies



**Chih-Ta Yen** received his Ph.D. degree from the Department of Electrical Engineering, National Cheng Kung University, Taiwan, in 2008. He is currently an associate professor in the areas of communication, optical design, and artificial intelligence technologies at the Department of Electrical Engineering, National Taiwan

Ocean University.



**Guan-Yu Chen** was born in Taoyuan City, Taiwan. He received his M.S. degree from the Department of Electrical Engineering, National Formosa University, Taiwan, in 2018. His major interests are in the areas of Deep learning, Machine learning, Convolutional neural network, Image recognition.