

Uniform Experimental Design for Optimizing the Parameters of Multi-input Convolutional Neural Networks

Cheng-Jian Lin,^{1*} Chen-Hsien Wu,¹ Chi-Chia Sun,^{2,3} and Cheng-Hsien Lin⁴

¹Department of Computer Science and Information Engineering, National Chin-Yi University of Technology, Taichung 411, Taiwan

²Department of Electrical Engineering, National Formosa University, Yunlin 632, Taiwan

³Smart Machinery and Intelligent Manufacturing Research Center, National Formosa University, Yunlin 632, Taiwan

⁴Department of Electrical Engineering, National Chung Hsing University, Taichung City 402, Taiwan

(Received January 17, 2020; accepted May 25, 2020)

Keywords: image recognition, gender classification, convolutional neural network, uniform experimental design

In this paper, a multi-input convolutional neural network (CNN) based on a uniform experimental design (UED) is proposed for gender classification applications. The proposed multi-input CNN uses multiple CNNs to obtain output results through individual training and concatenation. In addition, to avoid using trial and error for determining the architecture parameters of the multi-input CNN, a UED was used in this study. The experimental results confirmed that the dual-input CNN with a UED achieved accuracies of 99.68 and 99.06% for the CIA and MORPH datasets, respectively. The accuracy of the proposed CNN increased significantly when increasing the number of inputs.

1. Introduction

In traditional machine learning methods, image features must be defined and captured by the user in advance.^(1,2) Recently, convolutional neural networks (CNNs) have been used to automatically capture features for overcoming the aforementioned problem. Therefore, CNNs have been widely and successfully applied in image recognition,^(3–5) speech recognition,⁽⁶⁾ colorimetric models,⁽⁷⁾ and face recognition.^(8,9) CNNs are the most commonly used architecture for deep learning, and they exhibit superior performance in image recognition. In 1998, LeCun *et al.* proposed the first CNN architecture called LeNet-5⁽¹⁰⁾ and applied this architecture to handwriting recognition. However, owing to problems such as excessive parameters, gradients, and lack of hardware equipment, the costs of the architecture exceeded its benefits. Deep learning was not popular with users in 1989. Krizhevsky *et al.* proposed the AlexNet⁽¹¹⁾ architecture and introduced the dropout method⁽¹²⁾ to prevent the network from falling into overfitting. Many researchers have proposed deep CNN architectures. Popular CNNs, such as GoogLeNet, have been proposed by Szegedy *et al.*⁽¹³⁾ and Simonyan and Zisserman.⁽¹⁴⁾

*Corresponding author: e-mail: cjlin@ncut.edu.tw
<https://doi.org/10.18494/SAM.2020.2771>

Although CNNs have been successfully used in various fields, most of them only use a single input. Therefore, some researchers have explored dual-input CNNs. In 2015, Su *et al.*⁽¹⁵⁾ proposed a multiview CNN for classifying 3D models. Through 3D model acquisition, 2D images with different perspectives were used as network inputs. The image features for multiple perspectives of an object were then combined. In 2017, Sun *et al.*⁽¹⁶⁾ used a dual-input CNN for flower grading. They used three flower images at different positions as the input and combined the image features after a single convolution and pooling operation. In 2019, Li *et al.*⁽¹⁷⁾ developed a dual-input neural network architecture for detecting coronary artery disease. Two types of signals, namely, electrocardiogram and phonocardiogram signals, were used as the network input. The features of the two signal types were combined to improve classification accuracy. In the aforementioned studies, the architecture parameters were selected by the user through trial and error.

The basic CNN architecture is shown in Fig. 1. The architecture comprises an input layer, an output layer, and multiple hidden layers. The kernel size, stride, and padding of the filters in the convolution layer and the pooling window in the pooling layer are determined by users according to experience. However, major design problems occur as the depth of a CNN increases. The CNN parameters selected by the user are not the optimal parameters. To determine the optimal parameters of a CNN architecture, continuous learning experimentation is required. In the engineering field, two methods are commonly used for optimizing parameters: the Taguchi method^(18–21) and uniform experimental design (UED) method.^(22–25) The Taguchi method is simpler in design than the UED method; however, the Taguchi method is only suitable for experiments with few levels and factors. The minimum number of runs required in the Taguchi method is equal to the square of the level. Compared with the Taguchi method, the UED method requires fewer runs. The UED method uses multiple regression to find the optimal parameters in the shortest possible time.

To overcome the drawbacks of a single-input architecture, in this paper, we propose a multi-input CNN based on UED for gender classification applications. The proposed CNN uses

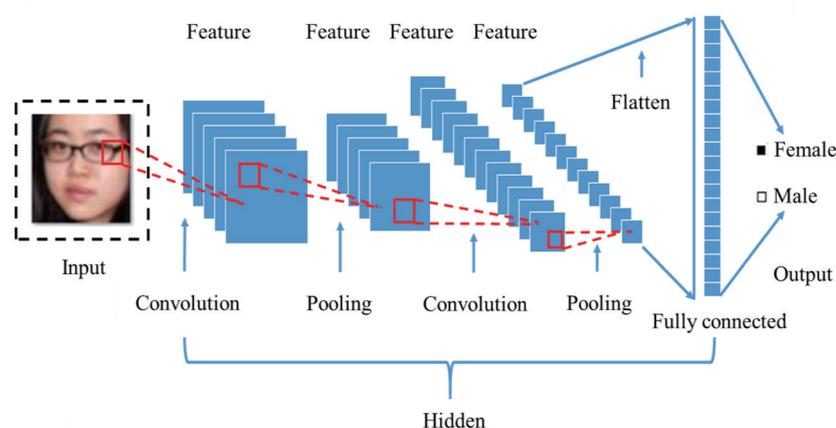


Fig. 1. (Color online) Structure of basic CNNs.

multiple CNNs to obtain output results through individual training and concatenation. To avoid using trial and error for determining the architecture parameters of the CNN, UED was used in this study. Under UED, multiple regression analysis is used to obtain the optimal parameters. Different numbers of inputs and different CNNs were used in the experiments of this study to verify the suitability of the proposed method for application to the CIA and MORPH datasets.

The remainder of this paper is organized as follows. Section 2 introduces the UED method. The architecture of the multi-input CNN, which comprises a convolutional layer, pooling layer, and fully connected layer, is described in Sect. 3. Section 4 presents the experimental results of the dual-input CNN for the CIA and MORPH datasets. Section 5 describes the effects of different numbers of inputs and different CNN architectures. Section 6 presents the conclusions and future research directions.

2. UED Method

In the UED method, multiple regression analysis is used to determine the optimal parameters. The number of runs required in the UED method is considerably lower than that required in the Taguchi method. A small amount of time is required to find the optimal parameters in the UED method. For an experiment with three factors and three levels, the Taguchi method requires at least nine runs, whereas the UED method requires only five runs. A uniform layout (UL) is denoted by $U_a(a^b)$, where U is the UL symbol, a is the number of levels and experiments, and b is the number of parameters. The overall design process is displayed in Fig. 2. The steps in the design process are given as follows:

The first step involves selecting the factor to be improved. Consider the basic CNN displayed in Fig. 1. The basic CNN has six affecting factors, including the kernel size, stride, and padding. The values of these parameters are preset (Table 1).

The number of experiments is determined according to an affecting factor as follows:

$$n > 2 \times S, \quad (1)$$

where n is the number of experiments and S is the affecting factor. If the number of experiments is less than 12, the uniformity is poor. Therefore, the number of experiments must be greater than 12. Consequently, the number of experiments is set to 13, and the affecting factor is 6.

The second step involves calculating the total number of columns from the number of experiments.

$$m = n - 1 \quad (2)$$

After determining the numbers of experiments and columns, the content of table $x_{i,j}$ can be obtained as follows:

$$x_{i,j} = (i \times j) \bmod n, \quad (3)$$

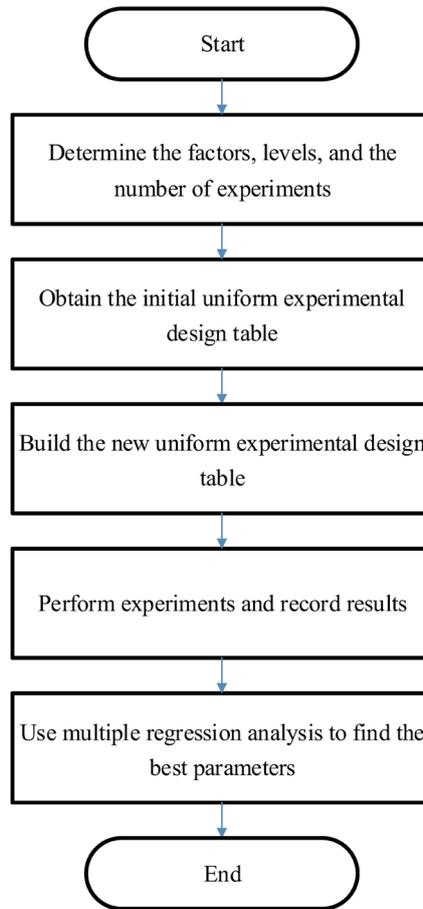


Fig. 2. Flow chart of the UED method.

Table 1
Level values of the six parameters of the CNN structure.

Level	First-layer convolution			Second-layer convolution		
	Kernel	Stride	Padding	Kernel	Stride	Padding
1	5	0	0	3	1	1
2	7	1	1	5	2	2
3	9	2	2	7	3	—

where $i = 1, 2, 3, \dots, m, j = 1, 2, 3, \dots, n$. For example, if m is 12 and n is 13, UL is represented as $U_{13}(13^{12})$. The initial UED table is presented in Table 2.

The third step involves selecting the use table according to $U_{13}(13^{12})$. As presented in Table 3, when the affecting factor is 6, the row comprising the numbers 1, 2, 6, 8, 9, and 10 is selected. The new UED table is presented in Table 4.

The fourth step involves performing an experiment and recording the results.

The fifth step involves performing multiple regression analysis on the obtained experimental results to solve the optimization parameters:

Table 2
Thirteen-level UL of $U_{13}(13^{12})$.

n	m											
	1	2	3	4	5	6	7	8	9	10	11	12
1	1	2	3	4	5	6	7	8	9	10	11	12
2	2	4	6	8	10	12	1	3	5	7	9	11
3	3	6	9	12	2	5	8	11	1	4	7	10
4	4	8	12	3	7	11	2	6	10	1	5	9
5	5	10	2	7	12	4	9	1	6	11	3	8
6	6	12	5	11	4	10	3	9	2	8	1	7
7	7	1	8	2	9	3	10	4	11	5	12	6
8	8	3	11	6	1	9	4	12	7	2	10	5
9	9	5	1	10	6	2	11	7	3	12	8	4
10	10	7	4	1	11	8	5	2	12	9	6	3
11	11	9	7	5	3	1	12	10	8	6	4	2
12	12	11	10	9	8	7	6	5	4	3	2	1
13	13	13	13	13	13	13	13	13	13	13	13	13

Table 3
Table used for $U_{13}(13^{12})$.

Number of factors	Number of columns											
2			1	5								
3			1	6	10							
4			1	6	8	10						
5			1	6	8	9	10					
6			1	2	6	8	9	10				
7			1	2	6	8	9	10	12			

Table 4
Thirteen-level UL of $U_{13}(13^{16})$ used to allocate the six design parameters with 13 levels.

	1	2	3	4	5	6
1	1	2	6	8	9	10
2	2	4	12	3	5	7
3	3	6	5	11	1	4
4	4	8	11	6	10	1
5	5	10	4	1	6	11
6	6	12	10	9	2	8
7	7	1	3	4	11	5
8	8	3	9	12	7	2
9	9	5	2	7	3	12
10	10	7	8	2	12	9
11	11	9	1	10	8	6
12	12	11	7	5	4	3
13	13	13	13	13	13	13

$$\varepsilon = Y - \left[\alpha_0 + \sum_{i=1}^f \alpha_{1i} \beta_i + \sum_{i=1}^f \alpha_{2i} \beta_i^2 + \sum_{i=1}^f \alpha_{3i} \beta_i^3 + \sum_{i=1}^{f-1} \sum_{j=i+1}^f \alpha_{4ij} \beta_i \beta_j \right], \tag{4}$$

where ε is the error. When ε approaches 0, its coefficient is the optimal weight. This optimal weight is used to find the optimal parameters. After training and testing the optimal parameters, the optimized parameters of the final UED table are obtained. The parameter α_0 is a constant, and α_{1i} , α_{2i} , α_{3i} , and α_{1ij} are coefficients of β .

3. Multi-input CNN

This section describes the proposed multi-input CNN. The term “multi-input” refers to the training of CNNs by using multiple inputs. This section uses the dual-input AlexNet network architecture as an example. Figure 3 displays a dual-input AlexNet network architecture. Two different inputs are fed into two identical CNN architectures. After the AlexNet calculation is completed, the data are combined through concatenation and the characteristic information is then passed to the fully connected layer for classification.

The CNN architecture can be freely selected in the proposed network. Three well-known CNN architectures, namely, LeNet, AlexNet, and GoogLeNet, are commonly used by researchers. In this study, we focused on the AlexNet network architecture. The AlexNet architecture is more popular in applications than the LeNet and GoogLeNet architectures because its size is between those of LeNet and GoogLeNet.

AlexNet has two main characteristics. First, it uses a nonlinear activation function [i.e., rectified linear unit (ReLU)] with a high convergence speed. Prior to the development of AlexNet, most neural networks used the sigmoid function with the vanishing gradient problem as the activation function. The ReLU function has a simpler operation than the sigmoid function, and only a threshold is required to obtain the activation value with the ReLU function. Second, the use of the dropout method in the first and second fully connected layers of the AlexNet architecture can effectively reduce the occurrence of overfitting.

To determine the optimal parameters of the multi-input CNN, the UED was used in this study. The entire experimental process is displayed in Fig. 4. In the first step, the parameters to be optimized are selected in the CNN architecture. In the second step, a UED is used to find the optimal weight through multiple regression analysis. The third step involves determining

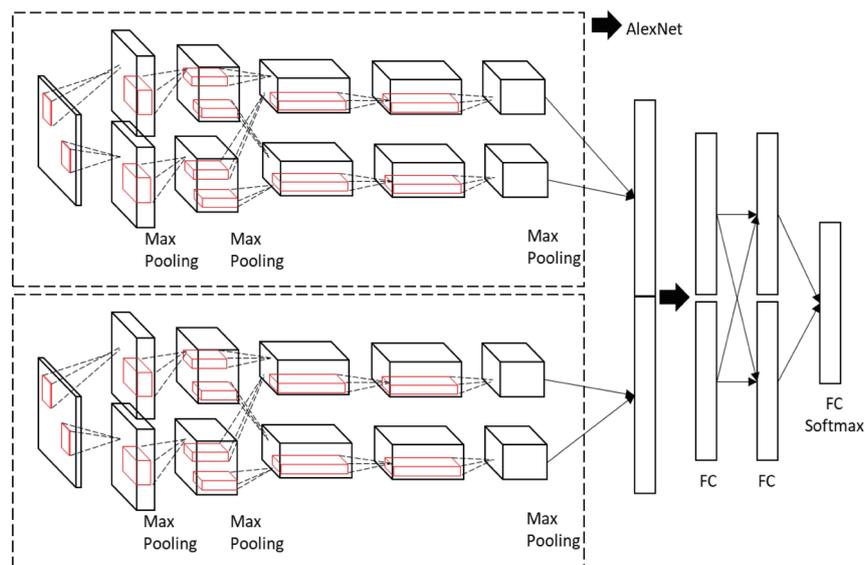


Fig. 3. (Color online) Structure of a dual-input CNN.

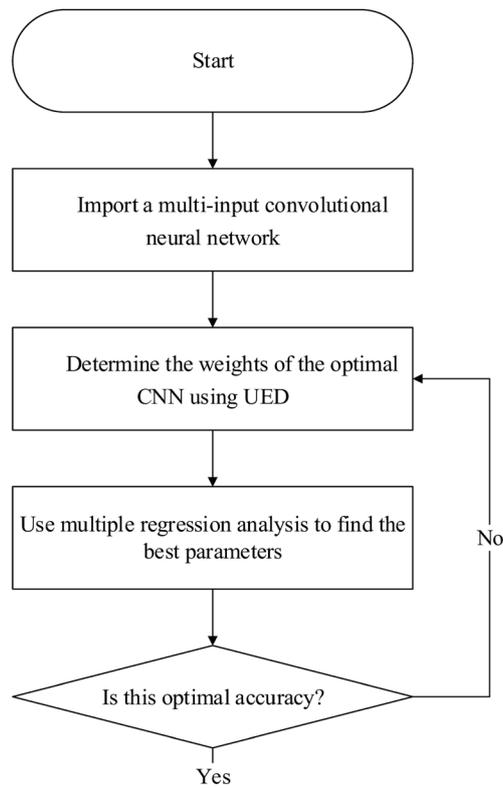


Fig. 4. Flow chart for obtaining the optimal CNN parameters under a UED.

the optimal parameters by using the optimal weight. The fourth step involves confirming whether the UED provides the highest possible accuracy rate. If yes, the process is completed; otherwise, the process returns to the second step.

3.1 Convolution layer

In the convolution layer, the mask of the convolution kernel is used to perform a convolution operation on the input matrix through the sliding window method. Figure 5 illustrates the convolution process. In Fig. 5, the length and width of the input image are both 5, the length and width of the convolution kernel are both 3, and the stride is 1. The output matrix size is obtained using the following equation:

$$W_o = \left[\frac{(W_i - k_w) + 2p}{s} \right] + 1, H_o = \left[\frac{(H_i - k_h) + 2p}{s} \right] + 1, \quad (5)$$

where W_o and H_o represent the width and height of the output matrix, respectively; W_i and H_i represent the width and height of the input matrix, respectively; p is the padding size; and s is the stride size. The output matrix (O_{RC}) of the convolution operation is expressed as follows:

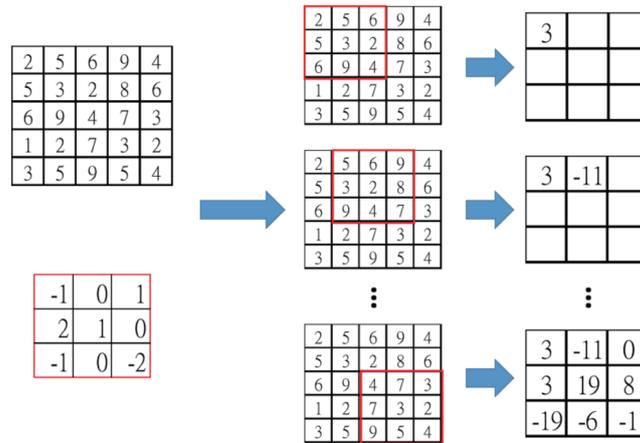


Fig. 5. (Color online) Convolution operation.

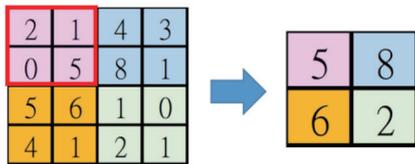


Fig. 6. (Color online) Maximum pooling operation.

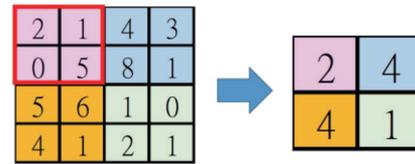


Fig. 7. (Color online) Average pooling operation.

$$O_{RC} = \sum_{i=0}^{K_h} \sum_{j=0}^{K_w} x_{(R+i-1)(C+j-1)} \times k_{ij} + b, \tag{6}$$

where K_h and K_w represent the width and height of the convolution kernel, respectively. In general, the size of the convolution kernel is equal to its width and height (i.e., $K_h = K_w$). The term k_{ij} represents the weight of the convolution kernel and x_{ij} denotes the input image matrix.

3.2 Pooling layer

In the pooling process, a mask is used to perform operations on the input matrix with a sliding window. This process is similar to the convolution operation. The only difference is that the mask does not overlap elements in the pooling process. In other words, each element in the input matrix is only covered once by the mask. Therefore, the dimensionality of the matrix can be reduced through the pooling process.

Two types of pooling operations exist, namely, maximum and average pooling. In the maximum pooling operation, the largest value in the mask is used as an output, as displayed in Fig. 6. In the average pooling operation, the average of all the values in the mask is used as the output, as depicted in Fig. 7.

3.3 Activation function

The ReLU function is used as the activation function in the convolutional and fully connected layers of the proposed CNN architecture. The final outputs are obtained through the softmax layer. The ReLU function is a nonlinear function. If the input a is greater than 0, the output is equal to a . Conversely, if the input a is less than or equal to 0, the output is 0. The formula for the ReLU function is given as follows:

$$f(a) = \begin{cases} a, & \text{if } a > 0 \\ 0, & \text{if } a \leq 0 \end{cases} \quad (7)$$

4. Experimental Results

To evaluate the proposed multi-input CNN, the AlexNet network and two face datasets, namely, the CIA and MORPH datasets, were used. The image data were incremented by performing the brightness reduction, brightness increase, rotate left, and rotate right operations on the two datasets, as displayed in Fig. 8. Therefore, the amount of incremented data was five times that of the original data. In the experiments, to perform cross-validation, three sets of training and testing data were randomly generated from the data. The average values obtained for the optimized parameters from three experiments were used to ensure overall fairness.

4.1 Parameter definition in the UED method

To obtain the optimized parameter structure of the multi-input AlexNet, the UED method and multiple regression analysis were used. In this subsection, a dual-input CNN is used as an example. Table 5 shows that the improvement factors selected in AlexNet included the kernel



Fig. 8. (Color online) Image increment: (a) original image and images obtained after using the (b) brightness reduction, (c) brightness increase, (d) rotate left, and (e) rotate right operations.

Table 5
Levels of the improvement factors.

Levels	Convolution of the first layer			Convolution of the fifth layer		
	Kernel	Stride	Padding	Kernel	Stride	Padding
1	9	2	0	3	1	1
2	11	4	1	5	2	2
3	13	—	2	7	—	—

size, step size, and padding of the convolution kernel in the first and fifth convolutional layers. The UED table presented in Table 6 was obtained using the steps mentioned in Sect. 2.

4.2 CIA dataset

The CIA dataset is a small facial image database that was collected by our laboratory. The database mainly comprises the facial images of Chinese individuals aged between 6 and 80 years, as displayed in Fig. 9. The amount of data obtained after the image increment was five times the amount of original data, as presented in Table 7. Under the UED method, experiments were performed under 13 sets of parameter values. The optimized parameters and classification

Table 6
Affecting factors used in the dual-input AlexNet.

	Convolution of the first layer			Convolution of the fifth layer		
	Kernel	Stride	Padding	Kernel	Stride	Padding
1	9	4	2	5	1	2
2	11	2	1	7	1	1
3	13	4	1	5	1	2
4	9	4	0	7	2	1
5	11	4	0	3	2	1
6	13	2	0	7	2	1
7	9	2	2	3	1	1
8	11	2	2	7	1	2
9	13	2	1	3	1	2
10	9	4	1	5	2	1
11	11	4	0	3	1	2
12	13	4	0	5	2	1
13	9	4	1	3	1	1

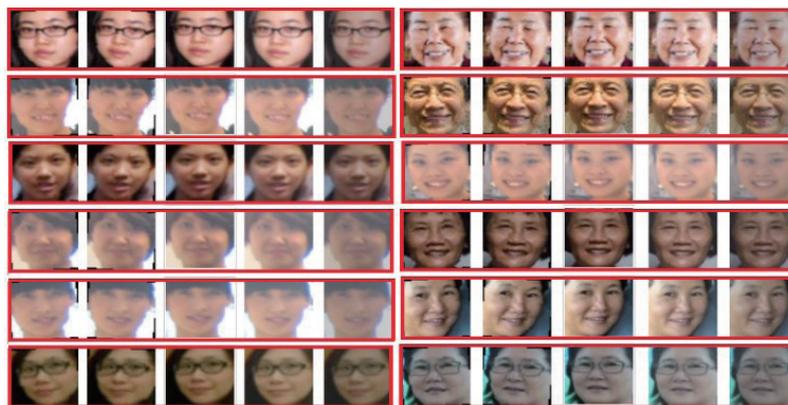


Fig. 9. (Color online) Sample images in the CIA dataset.

Table 7
Numbers of images before and after the increment for the CIA dataset.

	Male	Female
Number of images before the increment	1080	1008
Number of images after the increment	5400	5040

results for the CIA dataset are presented in Table 8. The highest experimental accuracy rate (99.60%) was obtained for the second set of experimental parameters. Multiple regression analysis was then performed on the CIA dataset to achieve an accuracy rate of 99.68% for the optimized parameters. The experimental results indicate that the UED method improved the accuracy by 0.08%.

4.3 MORPH dataset

The MORPH dataset is a collection of the facial images of various people aged between 16 and 77, as displayed in Fig. 10. The numbers of images before and after the increment process for the MORPH database are presented in Table 9. In the MORPH database, the average interval between successive image captures for each person is 164 days. The database does not include any continuously shot images.

Table 8
Experimental results of the dual-input AlexNet for the CIA dataset.

	Cross-validation 1	Cross-validation 2	Cross-validation 3	Average accuracy		
1	97.939626	97.843795	96.645903	97.476441		
2	99.664590	99.760422	99.377096	99.600703		
3	99.377096	99.568759	99.377096	99.440984		
4	99.472928	99.520843	98.706277	99.233349		
5	96.023000	96.550072	93.148059	95.240377		
6	99.616675	99.568759	99.377096	99.520843		
7	97.508385	98.179205	95.879253	97.188948		
8	99.377096	99.520843	99.137518	99.345152		
9	98.945855	99.377096	98.514614	98.945855		
10	99.520843	99.616675	99.185434	99.440984		
11	98.802108	99.712506	98.802108	99.105574		
12	99.568759	99.856253	99.329181	99.584731		
13	97.939626	99.185434	97.891711	98.338924		
UED	99.616675	99.808337	99.616675	99.680562		
Parameters in UED	Convolution of the first layer			Convolution of the fifth layer		
	Kernel	Stride	Padding	Kernel	Stride	Padding
	11	2	0	5	1	2



Fig. 10. (Color online) Sample images in the MORPH dataset.

Table 9

Numbers of images before and after the increment for the MORPH dataset.

	Male	Female
Number of images before the increment	46659	8492
Number of images after the increment	233295	42460

Table 10

Experimental results obtained with the dual-input CNN for the MORPH dataset.

	Cross-validation 1	Cross-validation 2	Cross-validation 3	Average accuracy		
1	98.213994	98.335479	98.221247	98.256907		
2	98.433392	98.674548	98.344545	98.484162		
3	98.498667	98.612899	98.400754	98.504107		
4	98.393501	98.282896	98.226687	98.301028		
5	98.226687	98.203115	98.262951	98.230918		
6	98.422513	98.398941	98.322787	98.381414		
7	98.647350	98.692680	98.710812	98.683614		
8	97.896684	97.615637	97.856793	97.789705		
9	97.728056	98.282896	97.907563	97.972838		
10	98.201302	98.165038	98.039927	98.135422		
11	98.393501	98.464216	98.163225	98.340314		
12	98.302841	98.212181	97.815089	98.110037		
13	98.333666	98.502294	98.175917	98.337292		
UED	99.107936	99.017279	99.058947	99.061387		
Parameters in UED	Convolution of the first layer			Convolution of the fifth layer		
	Kernel	Stride	Padding	Kernel	Stride	Padding
	13	4	2	7	1	2

The 13 experimental results obtained in the UED method are presented in Table 10. The accuracy of the seventh experiment was 98.68%, which was the highest accuracy among all the experiments. The accuracy of the optimized parameter combination was 99.06%, which was 0.38% higher than that of the seventh experiment.

5. Discussion and Analysis

5.1 Multi-input CNN

In this subsection, we discuss the accuracy of the proposed multi-input CNN architecture. Figure 11 shows a three-input AlexNet architecture. The definition of the parameters and the UED table for this architecture are the same as those for the dual-input architecture in Sect. 4. The training and testing data comprise images from the MORPH dataset. Table 11 presents the UED results for the three-input AlexNet. The average accuracy of the optimal parameter combination for this architecture was 99.16%, which is higher than the highest accuracy in the UED table by approximately 0.47%.

Figure 12 shows the architecture of a four-input AlexNet. Table 12 presents the UED results for the four-input CNN. The average accuracy of the optimal parameter combination for this CNN was 99.19%, which is higher than the highest accuracy in the UED table by approximately 0.57%.

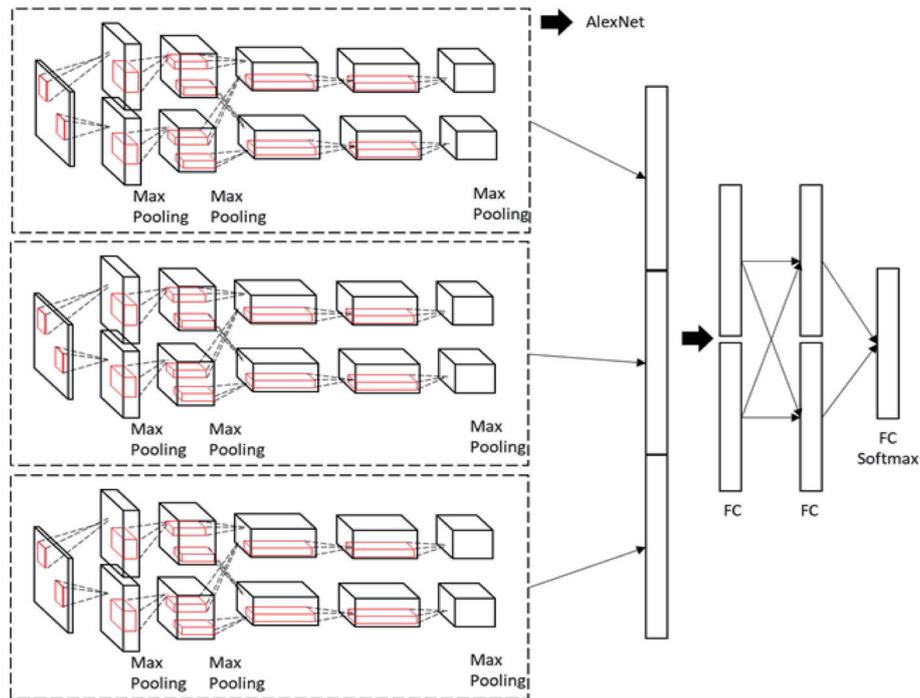


Fig. 11. (Color online) Architecture of a three-input AlexNet.

Table 11
Experimental results of the three-input AlexNet for the MORPH dataset.

	Cross-validation 1	Cross-validation 2	Cross-validation 3	Average accuracy		
1	98.299215	98.500480	98.340919	98.380204		
2	98.500480	98.585701	98.429766	98.505316		
3	98.413447	98.660042	98.456964	98.510151		
4	98.478722	98.406194	98.226687	98.370534		
5	98.308281	98.478722	98.219434	98.335479		
6	98.585701	98.426139	98.340919	98.450920		
7	98.692680	98.687240	98.710812	98.696911		
8	97.842288	97.898497	97.786078	97.842288		
9	98.136027	97.833222	98.059872	98.009707		
10	98.554877	98.670922	98.585701	98.603833		
11	98.437018	98.404381	98.491414	98.444271		
12	98.335479	98.446084	98.335479	98.372347		
13	98.444271	98.440645	98.179543	98.354820		
UED	99.122409	99.276532	99.104277	99.167739		
Parameters in UED	Convolution of the first layer			Convolution of the fifth layer		
	Kernel	Stride	Padding	Kernel	Stride	Padding
UED	11	4	0	7	1	2

The aforementioned results indicate that the proposed architecture can arbitrarily increase the number of inputs to form a multi-input AlexNet. The experimental results for the optimized parameters in the UED method are displayed in Fig. 13. The accuracy rates of the dual-input, three-input, and four-input AlexNet were 99.06, 99.17, and 99.20%, respectively. The accuracy

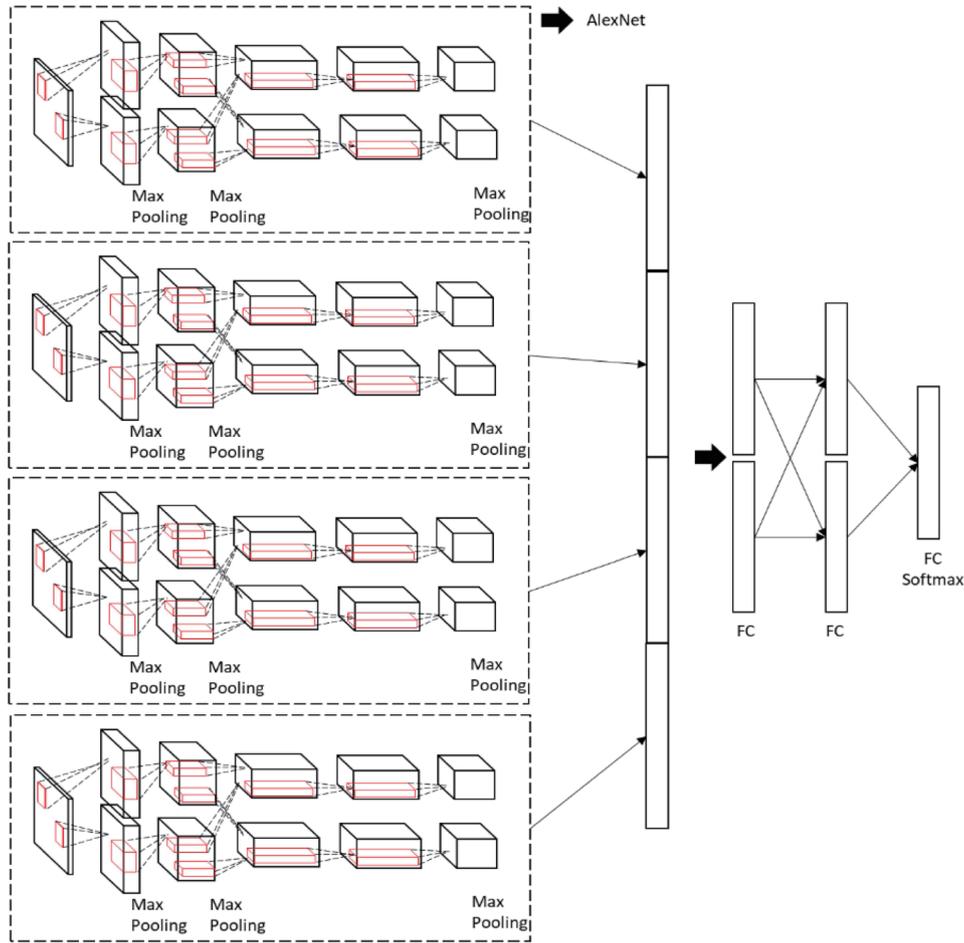


Fig. 12. (Color online) Architecture of a four-input AlexNet.

Table 12
Experimental results of the four-input AlexNet for the MORPH dataset.

	Cross-validation 1	Cross-validation 2	Cross-validation 3	Average accuracy		
1	98.393501	98.576635	98.475096	98.481744		
2	98.687240	98.411634	98.576635	98.558503		
3	98.507733	98.705373	98.386249	98.533118		
4	98.411634	98.340919	98.386249	98.379601		
5	98.582075	98.411634	98.565756	98.519822		
6	98.467843	98.609273	98.446084	98.507733		
7	98.705373	98.710812	98.692680	98.702955		
8	98.009102	97.945640	98.016355	97.990366		
9	98.009102	98.136027	98.039927	98.061685		
10	98.602020	98.609273	98.610482	98.607258		
11	98.393501	98.456964	98.509547	98.453337		
12	98.509547	98.625592	98.417073	98.517404		
13	98.456964	98.643724	98.420700	98.507129		
UED	99.272905	99.211256	99.107936	99.197366		
Parameters in UED	Convolution of the first layer			Convolution of the fifth layer		
	Kernel	Stride	Padding	Kernel	Stride	Padding
	13	4	1	7	1	1

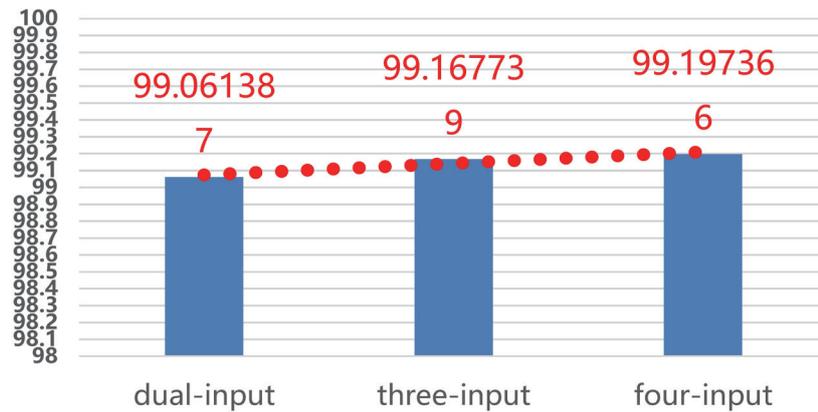


Fig. 13. (Color online) Accuracy curve of the multi-input AlexNet with optimal parameters.

increased with an increase in the number of inputs; however, the overall network speed decreased and the hardware costs increased.

5.2 Effect of various networks

In the aforementioned experiments, we adopted AlexNet. Because the AlexNet network is deeper than the LeNet network, it has a higher accuracy rate than the LeNet network. In addition, the AlexNet network is shallower than the GoogLeNet network. We replaced AlexNet with LeNet and GoogLeNet in the dual-input CNN. Table 13 indicates that the average accuracies of the LeNet, GoogLeNet, and AlexNet networks were 99.18, 99.07, and 99.30%, respectively. As displayed in Fig. 14, AlexNet had a higher average accuracy than the other two architectures. In theory, the accuracy rate of GoogLeNet is higher than that of AlexNet. However, because GoogLeNet has more layers than AlexNet, image features disappear when using the GoogLeNet architecture.

Different features can be obtained using multi-input CNNs. Determining which network features improve the accuracy of the entire system is a crucial task. Suitably integrating these features is also critical. Because the individual features obtained by multiple networks provide different interpretations of the same image, some features may allow the network to determine the correct result, whereas others may cause serious misjudgment. To solve this problem, a multilayer network fusion mechanism is added to the output of a feature network. This mechanism partially enhances or suppresses the original output features to perform a fusion operation. Thus, multiple features can be combined together to improve the overall recognition rate. Many researchers^(26–28) have presented feature fusion techniques for multi-input CNNs. In this study, we compared the proposed method with other methods.^(26–28) Table 14 presents the accuracy results of the proposed method and other methods for the MORPH and CIA datasets. The experimental results indicate that the proposed method has a higher average accuracy than the other methods^(26–28) for the MORPH and CIA datasets.

Table 13

Experimental results obtained when using the LeNet, GoogLeNet, and AlexNet architectures.

	Cross-validation 1	Cross-validation 2	Cross-validation 3	Average accuracy
Lenet	99.137518	98.185434	99.225363	99.182772
GoogLeNet	99.069637	99.025715	99.109567	99.068306
AlexNet	99.377096	99.329181	99.185434	99.297237

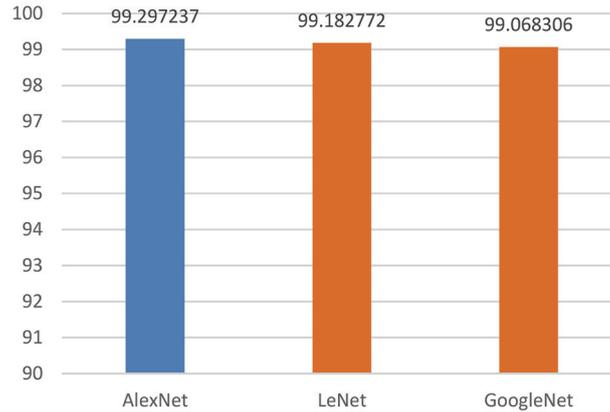


Fig. 14. (Color online) Average accuracies obtained using the LeNet, GoogLeNet, and AlexNet architectures.

Table 14

Comparison of the accuracy of various methods for the MORPH and CIA datasets.

	Feichtenhofer <i>et al.</i> ⁽²⁶⁾ (%)	Wu <i>et al.</i> ⁽²⁷⁾ (%)	Aygün <i>et al.</i> ⁽²⁸⁾ (%)	Proposed method (%)
MORPH	97.57	98.70	98.52%	99.06
CIA	94.20	99.60	99.65%	99.68

5.3 Results of single- and dual-input networks

This subsection presents the results of single- and dual-input networks for the CIA dataset. Table 15 lists the experimental results of single- and dual-input AlexNet architectures. The average accuracy of the dual-input AlexNet was 1.14% higher than that of the single-input AlexNet.

5.4 Experimental results for other datasets when using the proposed method

The CIFAR-10,⁽²⁹⁾ CIFAR-100,⁽²⁹⁾ Birdsnap,⁽³⁰⁾ Stanford cars,⁽³¹⁾ Flowers,⁽³²⁾ FGVC aircraft,⁽³³⁾ Oxford-IIIT pets,⁽³⁴⁾ and Food-101⁽³⁵⁾ datasets were also used to verify the proposed method. Table 16 presents the experimental results for the aforementioned datasets when using the proposed method. The proposed method exhibited a suitable average accuracy for these well-known datasets.

Table 15

Experimental results of single- and dual-input networks for the CIA dataset.

	CIA dataset			Average accuracy
	Cross-validation 1	Cross-validation 2	Cross-validation 3	
Single-input AlexNet	98.31%	98.46%	98.37%	98.38%
Dual-input AlexNet	99.62%	99.57%	99.38%	99.52%

Table 16

Experimental results for various other datasets when using the proposed method.

	Training data	Testing data	Classes	Average accuracy (%)
CIFAR-10 ⁽²⁹⁾	50000	10000	10	99.0
CIFAR-100 ⁽²⁹⁾	50000	10000	100	91.7
Birdsnap ⁽³⁰⁾	47386	2443	500	84.3
Stanford cars ⁽³¹⁾	8144	8041	196	94.8
Flowers ⁽³²⁾	2040	6149	102	98.8
FGVC aircraft ⁽³³⁾	6667	3333	100	92.9
Oxford-IIIT pets ⁽³⁴⁾	3680	3369	37	95.9
Food-101 ⁽³⁵⁾	75750	25250	101	93.0

5. Conclusions and Future Work

To overcome the drawbacks of single-input network architectures, a multi-input CNN based on the UED method is proposed in this paper for gender classification applications. The proposed multi-input CNN uses multiple CNNs to obtain output results through individual training and concatenation. To avoid using trial and error for determining the architecture parameters of the proposed network, a UED was used. In a UED, multiple regression analysis is used to determine the optimal parameters. The accuracy rates of the dual-input, three-input, and four-input AlexNet were 99.06, 99.17, and 99.20%, respectively. The average accuracies obtained when using the LeNet, GoogLeNet, and AlexNet networks with a dual input were 99.18, 99.07, and 99.30%, respectively.

In future studies, multiple input signals with different characteristics can be used in practical problems. For example, image, radar, and lidar signals can be used for car detection, and temperature, current, and vibration signals can be used as input signals in smart manufacturing.

References

- 1 C. H. Hsia, J. S. Chiang, and C. Y. Lin: *Scientia Iranica* **22** (2015) 2081. http://scintiairanica.sharif.edu/article_3756.html.
- 2 J. M. Guo, Y. F. Liu, C. H. Hsia, S. Y. Su, and H. Lee: *IEEE Trans. Inf. Forensics Security* **9** (2014) 1953. <https://ieeexplore.ieee.org/document/6894132>
- 3 A. Kumar, J. Kim, D. Lyndon, M. Fulham, and D. Feng: *IEEE J. Biomed. Heal. Informatics* **21** (2017) 31. <https://ieeexplore.ieee.org/document/7769199>
- 4 I. Sevo and A. Avramovic: *IEEE Geosci. Remote Sens. Lett.* **13** (2016) 740. <https://ieeexplore.ieee.org/document/7447728>
- 5 J. M. Ponce, A. Aquino, and J. M. Andujar: *IEEE Access* **7** (2019) 147629. <https://ieeexplore.ieee.org/document/8867877>
- 6 J. C. Hou, S. S. Wang, Y. H. Lai, Y. Tsao, H. W. Chang, and H. M. Wang: *IEEE Trans. Emerg. Top. Comput. Intell.* **2** (2018) 117. <https://ieeexplore.ieee.org/document/8323326>
- 7 P. T. Wang, J. J. Chou, and C. W. Tseng: *Sens. Mater.* **31** (2019) 1513. <https://www.mdpi.com/1424-8220/19/14/3207/htm>

- 8 C. Ding and D. Tao: *IEEE Trans. Pattern Anal. Mach. Intell.* **40** (2018) 1002. <https://ieeexplore.ieee.org/document/7917252>
- 9 P. Rasti, T. Uiboupin, S. Escalera, and G. Anbarjafari: *Lecture Notes in Computer Science* **9756** (2016) 175. https://link.springer.com/chapter/10.1007/978-3-319-41778-3_18
- 10 Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner: *Proc. IEEE* **86** (1998) 2278. <https://ieeexplore.ieee.org/document/726791>
- 11 A. Krizhevsky, I. Sutskever, and G. E. Hinton: *Commun. ACM* **60** (2017) 84. <https://dl.acm.org/doi/10.1145/3065386>
- 12 N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov: *J. Mach. Learn. Res.* **15** (2014) 1929. <http://jmlr.org/papers/v15/srivastava14a.html>
- 13 C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich: *2015 IEEE Conf. Computer Vision and Pattern Recognition (CVPR)* (2015) 1–9. <https://ieeexplore.ieee.org/document/7298594>
- 14 K. Simonyan and A. Zisserman: *3rd Int. Conf. Learn. Represent. (ICLR 2015)* 2014. <https://www.arxiv-vanity.com/papers/1409.1556/>
- 15 H. Su, S. Maji, E. Kalogerakis, and E. Learned-Miller: *2015 IEEE Int. Conf. Computer Vision (ICCV)* (2015) 945–953. <https://ieeexplore.ieee.org/document/7410471>
- 16 Y. Sun, L. Zhu, G. Wang, and F. Zhao: *J. Electr. Comput. Eng.* **2017** (2017) 1. <https://www.hindawi.com/journals/jece/2017/9240407>
- 17 H. Li, X. Wang, C. Liu, Y. Wang, P. Li, H. Tang, L. Yao, and H. Zhang: *IEEE Access* **7** (2019) 146457. <https://ieeexplore.ieee.org/document/8846698>
- 18 N. K. Meena, A. Swarnkar, J. Yang, N. Gupta, and K. R. Niazi: *IEEE Access* **7** (2019) 135689. <https://ieeexplore.ieee.org/document/8843953>
- 19 M. S. Ibrahim, A. Hanif, and A. Ahsan: *IEEE Access* **7** (2019) 129164. <https://ieeexplore.ieee.org/document/8824076>
- 20 X. Jia and G. Lu: *IEEE Antennas Wirel. Propag. Lett.* **18** (2019) 1878. <https://ieeexplore.ieee.org/document/8782624>
- 21 Y. Guo, J. Si, C. Gao, H. Feng, and C. Gan: *IEEE Trans. Magn.* **55** (2019) 1. <https://ieeexplore.ieee.org/document/8657754>
- 22 F. I. Chou, Y. K. Tsai, Y. M. Chen, J. T. Tsai, and C. C. Kuo: *IEEE Access* **7** (2019) 68316. <https://ieeexplore.ieee.org/document/8721236>
- 23 J. T. Tsai, P. Y. Yang, and J. H. Chou: *IEEE Access* **6** (2018) 40365. <https://ieeexplore.ieee.org/document/8412479>
- 24 Z. Hao, Z. Liu, and B. Feng: *Proc. 2014 IEEE Int. Conf. Progress in Informatics and Computing* (2014) 350–354. <https://ieeexplore.ieee.org/document/6972356>
- 25 Y. Wang, B. Xu, G. Sun, and S. Yang: *IEEE Trans. Evol. Comput.* **21** (2017) 665. <https://ieeexplore.ieee.org/document/7858752>
- 26 C. Feichtenhofer, A. Pinz, and A. Zisserman: *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2016) 1933–1941. <https://ieeexplore.ieee.org/document/7780582>
- 27 L. Wu, Y. Wang, X. Li, and J. Gao: *Pattern Recognit.* **76** (2018) 727. <https://www.sciencedirect.com/science/article/abs/pii/S0031320317304004>
- 28 M. Aygün, Y. H. Şahin, and G. Üna: *arXiv:1809.06191* (2018). <https://arxiv.org/abs/1809.06191>
- 29 A. Krizhevsky and G. Hinton: *Tech. Rep.* (2009). <https://www.cs.toronto.edu/~kriz/cifar.html>
- 30 T. Berg, J. Liu, S. W. Lee, M. L. Alexander, D. W. Jacobs, and P. N. Belhumeur: *IEEE Proc. Conf. Computer Vision and Pattern Recognition (IEEE CVPR2014)* (2014) 2011–2018. <https://ieeexplore.ieee.org/document/6909656>
- 31 J. Krause, J. Deng, M. Stark, and F.-F. Li: *2nd Workshop on Fine-Grained Visual Categorization* (2013). <http://ai.stanford.edu/~jkrause/papers/fgvc13>
- 32 M.-E. Nilsback and A. Zisserman: *2008 IEEE 6th Indian Conf. Computer Vision, Graphics & Image Processing (IEEE ICVGIP 2008)*, Bhubaneswar, India, (2008) 722–729. <https://ieeexplore.ieee.org/document/4756141>
- 33 S. Maji, E. Rahtu, J. Kannala, M. Blaschko, and A. Vedaldi: *arXiv:1306.5151v1 [cs.CV]* **21** (2013). <https://arxiv.org/pdf/1306.5151>
- 34 O. M. Parkhi, A. Vedaldi, A. Zisserman, and C. Jawahar: *Proc. Conf. Computer Vision and Pattern Recognition (CVPR)* (2012) 3498–3505. <https://dl.acm.org/doi/10.5555/2354409.2355061>
- 35 L. Bossard, M. Guillaumin, and L. V. Gool: *Computer Vision – ECCV 2014. Lecture Notes in Computer Science*, D. Fleet, T. Pajdla, B. Schiele, T. Tuytelaars Eds. **8694** (2014) 446–461. https://link.springer.com/chapter/10.1007/978-3-319-10599-4_29

About the Authors



Cheng-Jian Lin received his Ph.D. degree in electrical and control engineering from National Chiao-Tung University, Taiwan, in 1996. Currently, he is a Lifetime Distinguished Professor of the Computer Science and Information Engineering Department, National Chin-Yi University of Technology, Taichung City, Taiwan. His current research interests are machine learning, pattern recognition, intelligent control, image processing, and evolutionary robots. (cjlin@ncut.edu.tw)



Chen-Hsien Wu received her B.S. and M.S. degrees from the Computer Science and Information Engineering Department, National Chin-Yi University of Technology, Taichung City, Taiwan, in 2018 and 2020, respectively. Her current research interests are machine learning, pattern recognition, and image processing.



Chi-Chia Sun received his B.S. degree in computer science and engineering from National Taiwan Ocean University (2000–2004) and his M.S. degree in electronic engineering from National Taiwan University of Science & Technology (2004–2006). From April 2008 to March 2011, he worked as a research assistant at Dortmund University of Technology and received his doktor ingenieur degree with German Academic Exchange Service (DAAD) scholarship support. Currently, he is an associate professor in the Department of Electrical Engineering of National Formosa University, Taiwan. His research interests are in image processing, system integration, and VLSI FPGA design. He is also a member of IEEE and TICD society.



Cheng-Hsien Lin received his B.S. degree from the Department of Electrical Engineering, National Sun Yat-sen University, Kaohsiung City, Taiwan, in 2018. Currently, he is a graduate student of the Electrical Engineering Department, National Chung Hsing University, Taichung City, Taiwan. His current research interests are machine learning, pattern recognition, and FPGA.