electronics

MDPI

Article

Evolutionary-Fuzzy-Integral-Based Convolutional Neural Networks for Facial Image Classification

Cheng-Jian Lin [1,*], Chun-Hui Lin [2], Chi-Chia Sun [3] and Shyh-Hau Wang [2]

[1] Department of Computer Science and Information Engineering, National Chin-Yi University of Technology, Taichung 411, Taiwan
[2] Department of Computer Science and Information Engineering, National Cheng Kung University, Tainan 701, Taiwan
[3] Department of Electrical Engineering, National Formosa University, Yunlin 632, Taiwan
* Correspondence: cjlin@ncut.edu.tw; Tel.: +886-4-23924505 (ext. 8753)

Received: 23 July 2019; Accepted: 27 August 2019; Published: 5 September 2019

Abstract: Various optimization methods and network architectures are used by convolutional neural networks (CNNs). Each optimization method and network architecture style have their own advantages and representation abilities. To make the most of these advantages, evolutionary-fuzzy-integral-based convolutional neural networks (EFI-CNNs) are proposed in this paper. The proposed EFI-CNNs were verified by way of face classification of age and gender. The trained CNNs' outputs were set as inputs of a fuzzy integral. The classification results were operated using either Sugeno or Choquet output rules. The conventional fuzzy density values of the fuzzy integral were decided by heuristic experiments. In this paper, particle swarm optimization (PSO) was used to adaptively find optimal fuzzy density values. To combine the advantages of each CNN type, the evaluation of each CNN type in EFI-CNNs is necessary. Three CNN structures, AlexNet, very deep convolutional neural network (VGG16), and GoogLeNet, and three databases, computational intelligence application laboratory (CIA), Morph, and cross-age celebrity dataset (CACD2000), were used in experiments to classify age and gender. The experimental results show that the proposed method achieved 5.95% and 3.1% higher accuracy, respectively, in classifying age and gender.

Keywords: convolutional neural network; fuzzy integral; particle swarm optimization; image processing; classification

1. Introduction

Image recognition technology has continued to develop chiefly due to deep learning technology. The origins of convolutional neural networks (CNNs) can be traced back to 1998. LeCun et al. [1] proposed the LeNet-5 model and used the back propagation (BP) algorithm to adjust the parameters of the neural networks; this model is a successful convolutional neural network even today. Later, a deeper network architecture, AlexNet [2], was proposed by Alex Krizhevsky and opened up the development of deep learning.

For the convolutional layers, AlexNet contains 60 million parameters and uses rectified linear unit (ReLU) as the activation function, which is different from LeNet, and it adds dropout, avoiding model overfitting. To enhance the abilities of each convolutional layer, Lin et al. [3] replaced conventional convolutional layers with multilayer perceptron (MLP) convolutional layers. GoogLeNet [4] improved the MLP convolutional layer by using a 1×1 convolutional kernel which achieves cross-channel message exchange and reduces dimensions. Due to the degradation problem, these networks cannot learn the features in deeper layer networks. In order to solve this problem, ResNet [5] was proposed to

map the low-level features directly to high-level networks, which means that instead of learning from the beginning, the deeper layer networks contain the previous few layers' representation ability.

The latter part of a CNN is a classifier, which includes the most parameters in the CNN. Based on the concept of dropout, Wan et al. [6] proposed DropConnect, which closes neurons randomly, and its performance is slightly better than the dropout method. Network in Network [3] changed fully connected layers to global average pooling. Since global average pooling has no parameters to train, it can reduce the burden of the networks without the problem of overfitting.

For the activation function, sigmoid, tanh, and other saturation functions were used in the early networks. However, gradient disappearance problems occur in the deeper layer networks; rectified linear unit (ReLU) [2] was therefore proposed as the activation function. It reduces the calculation time and makes the model more resistant to overfitting. However, some neurons cannot be triggered using ReLU; thus, some parameters cannot be updated. There are many improved activation functions used in CNNs, such as Leaky ReLU [7], parametric rectified linear unit (PReLU) [8], randomized leaky ReLU (RReLU), and Swish [9].

The most common optimization method in the training process is stochastic gradient descent (SGD), which randomly selects a set of training samples each time for training. This method usually learns effectively; however, it relies on the learning rate setting and the training time is relatively long. Some improved optimization methods [10–12] have been proposed in order to solve these problems. The adaptive learning rate was proposed by Adam [13] and can predict a result faster.

The aforementioned architectures and optimization methods have different advantages and features in different applications. Kornblith et al. [14] questioned whether the results attained from ImageNet can be used in other problems, and the result was confirmed, which proves that different network architectures have their representation abilities. Therefore, maximizing the performance of systems by combining multiple features or multiple classifiers has become a popular topic. Classifier selection and classifier interaction modules within a Bayesian framework were used to recognize an object [15]. A multiple classifier system that incorporates a global optimization technique based on a genetic algorithm which adopts the weighted majority vote by maximizing the performance of the system was tested on a handwritten digit recognition problem [16]. Based on these ideas, in order to leverage advantages from different CNNs, the integration of multiple convolutional neural networks is proposed in this paper. The fuzzy integral is used to evaluate different CNNs in this paper; however, instead of using the conventional fuzzy integral which sets fuzzy density values by heuristic experiments, an evolutionary fuzzy integral is proposed in the paper which chooses the fuzzy density adaptively by particle swarm optimization (PSO), and the final classification results are computed using both Sugeno's and Choquet's rules. Three CNN structures, AlexNet, very deep convolutional neural network (VGG16), and GoogLeNet, and three databases, computational intelligence application laboratory (CIA), Morph, and cross-age celebrity dataset (CACD2000), were used in experiments to classify age and gender.

In this study, the major contributions of the proposed evolutionary-fuzzy-integral-based CNNs (EFI-CNNs) are as follows:

- Integrating multiple CNNs is proposed to leverage advantages from different CNNs.
- An evolutionary fuzzy integral is proposed to choose optimal fuzzy density values by PSO, avoiding the effects of manual setting.
- The experimental results indicate that the proposed method exhibited superior accuracy compared to other methods.

The rest of this paper is organized as follows. Section 2 introduces the conventional CNN structure. The proposed EFI-CNNs are described in Section 3. Section 4 presents the experimental results of three facial databases, and Section 5 gives the conclusions.

2. Overview of CNNs

Conventional CNNs consist of two parts, as shown in Figure 1. The first part contains the steps of convolution and pooling which are used for feature extraction. The second part is a classifier which uses fully connected layers. Conventional CNNs still use feature extraction plus a classifier; however, lately, CNNs have replaced fully connected layers with average pooling. This can reduce the large number of required parameters and lower the degree of overfitting as well.

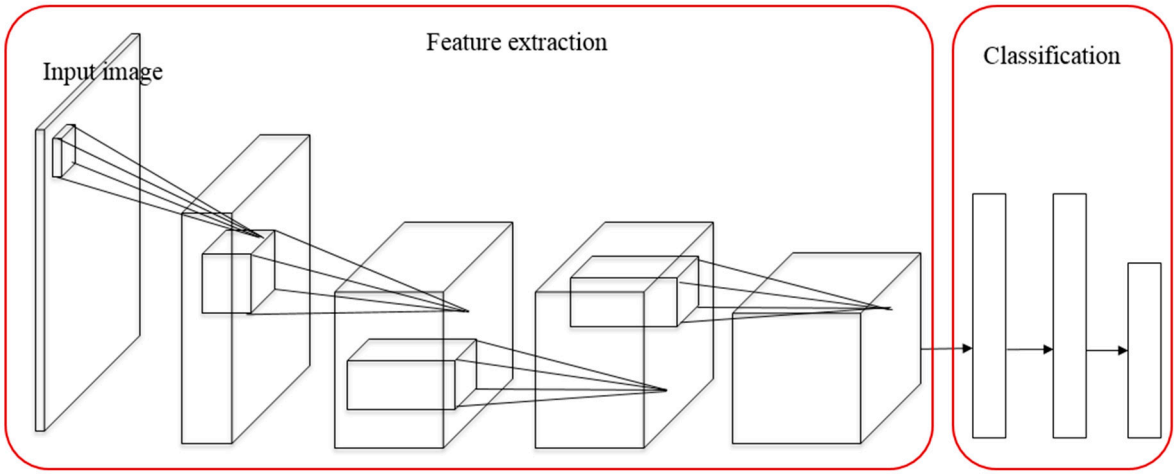

Figure 1. The structure of a convolutional neural network (CNN).

In the following sections, the feature extraction steps, convolutional layer, pooling, and activation function will be introduced.

2.1. Convolutional Layer

Convolution is the concept of a receptive field in the feature extraction process. It is used by many conventional filters, such as Sobel and Gabor. The convolution kernel is used on the input matrix like a sliding window, as shown in Figure 2. A two-dimensional convolution operation with a moving step equal to 1 is presented in Equation (1):

$$Y_{IJ} = \sum_{i=0}^{K_w} \sum_{j=0}^{K_h} x_{(I+i-1)(J+j-1)} * k_{ij}, \quad (1)$$

where Y_{IJ} is an output matrix, and K_w and K_h are the width and height of a convolution kernel, respectively. In general, $K_w = K_h$ represents a square convolution kernel, x_{ij} is the input matrix, and k_{ij} is the weight of the convolution kernel which needs to be updated during training. In order to maintain a constant size after convolution, zero-padding is used at the edge of the input matrix.

2.2. Pooling

After the convolution operation, the extracted features can theoretically be classified directly. However, this requires a huge number of parametric operations, which makes the training process difficult and prone to overfitting. Pooling is one way to reduce the dimensions.

The pooling process applies a mask operation on the input matrix with a sliding window. During the process, the moving step is equal to the width of the convolution kernel. Each element is calculated only one time. Therefore, an N × N mask can lower the input feature matrix 1/N times to achieve the effect of reducing the dimensions.

There are two pooling operations: Max pooling and average pooling. Max pooling takes the maximum value in a region R_{ij} and ignores other values. Average pooling calculates the average value in a region. The max pooling function is expressed as

$$a_{kij} = \max_{(p,q)\in R_{ij}} \left(a_{kpq}\right), \tag{2}$$

and the average pooling function is shown by the following equation:

$$a_{kij} = \frac{1}{|R_{ij}|} \sum_{(p,q)\in R_{ij}} a_{kpq}, \tag{3}$$

where a_{kij} is the output activation of the kth feature map at (i,j), a_{kpq} is the input activation at (p,q) within R_{ij}, and $|R_{ij}|$ is the size of the pooling region [17]. In order to retain the most prominent features, max pooling is the pooling method most often used during the convolution process. In contrast, global average pooling is usually applied in the output layer, because when classifying a target, average pooling contains physical meaning in the calculation process; therefore, a larger target can get a higher output value in the calculation.

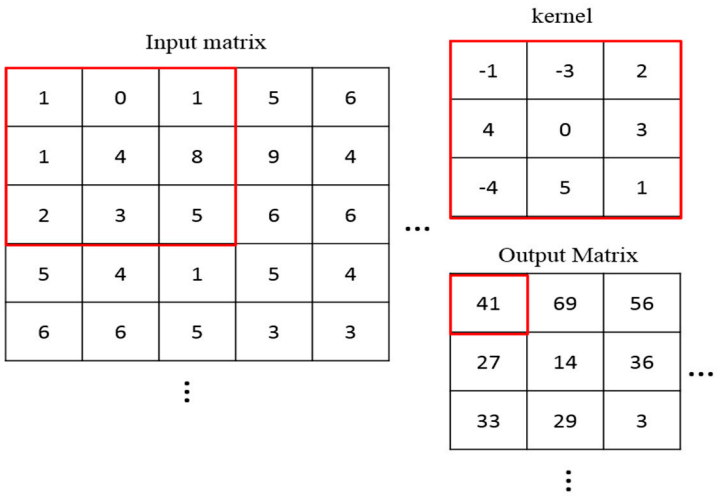

Figure 2. Procedure of convolution.

2.3. Activation Function

The purpose of the activation function is to get nonlinear outputs from linearly combined networks. The disadvantage of sigmoid is that when the network gets deeper, gradient disappearance problems occur during back propagation. Therefore, rectified linear unit (ReLU) is used herein as the activation function. The definition of ReLU is illustrated as follows:

$$f(x) = \begin{cases} 0, \ if \ x < 0 \\ x, \ if \ x \geq 0 \end{cases}, \tag{4}$$

where if input x is smaller than 0, then the output is 0. If input x is bigger than 0, then the output is x.

3. The Proposed EFI-CNNs

In order to leverage advantages from different CNNs, the integration of multiple convolutional neural network architectures with different optimal methods through the evolutionary fuzzy integral is proposed in this paper. The flowchart of an EFI-CNN is depicted in Figure 3.

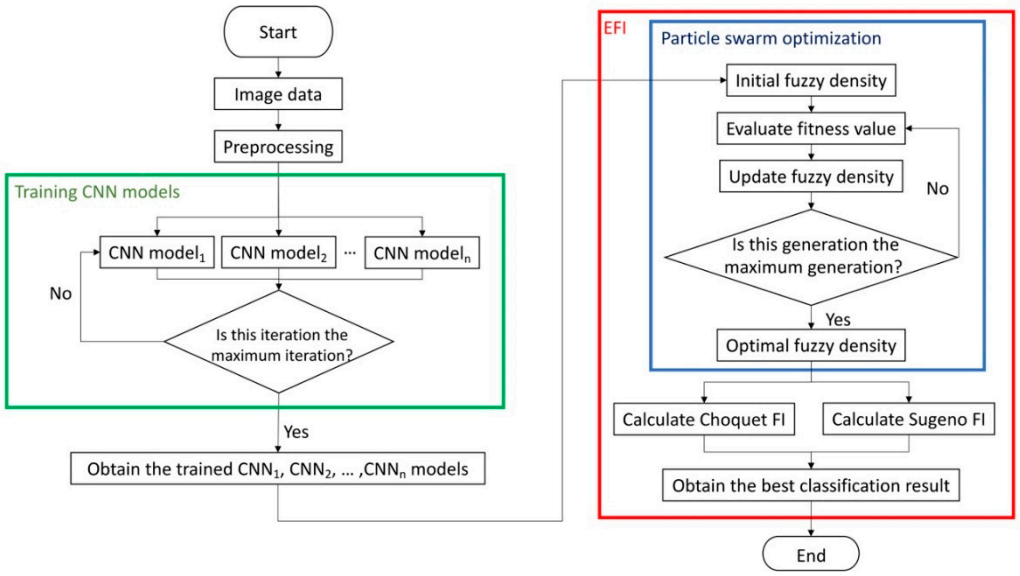

Figure 3. Flowchart of an evolutionary-fuzzy-integral (EFI)-based CNN.

First, image preprocessing is required based on the dataset in use. Second, we input images to train the CNN models with different CNN architectures or different optimization methods. Third, we choose optimal fuzzy density values based on the best fitness value via the particle swarm optimization (PSO) method. Later, we use the obtained optimal fuzzy density values to calculate the fuzzy measure. Then, we sort the classifiers and define the set A for calculating Sugeno and Choquet, two fuzzy integral rules. At the end, the result with higher accuracy from either Sugeno or Choquet is chosen.

As Figure 3 shows, the outputs are obtained from the CNNs and treated as the inputs of the evolutionary fuzzy integral. The evolutionary fuzzy integral is described as follows.

Assuming $\mathrm{X} = \{x_i\}_{i=1:n}$ represents a set of n classifiers (in this study, n was set to 3), the fuzzy measure $\mathrm{g}(\{x_i\})$ can be considered as the worth value of subset $\{x_i\}$, and its value is between 0 and 1. The fuzzy measure must fit the three following conditions:

(1) $g(X) = 1$ represents that the outputs of all classifiers are consistent; the results can be fully trusted.
(2) $g(\emptyset) = 0$ represents that the outputs of all classifiers are not considered; the result has no reference value.
(3) The fuzzy measure is an increasing monotonic function:

$$\text{If } \mathrm{A} \subset \mathrm{B} \subset \mathrm{X}, \text{ then } 0 \leq \mathrm{g}(A) \leq \mathrm{g}(B) \leq 1. \tag{5}$$

When there is only one element in set X, the fuzzy measure $g(X)$ is called the fuzzy density. Before calculating other fuzzy measures, the fuzzy density must be decided first. In the conventional fuzzy integral, the fuzzy density is set by users based on their experience. In general, the fuzzy density represents the worth value of the outputs in each network classifier. Some studies have used the accuracy of non-training data as the fuzzy density [18]; however, the accuracy of non-training data is usually not the optimal fuzzy density. The relation between fuzzy measure and fuzzy density is shown in Figure 4.

In this paper, the fuzzy density was chosen by particle swarm optimization (PSO) [19], avoiding the effects of manual setting. PSO was derived from the behavior patterns of birds. Flying birds are interpreted as countless particles moving continuously in the solution space and undergoing multiple iterations to find the best solution.

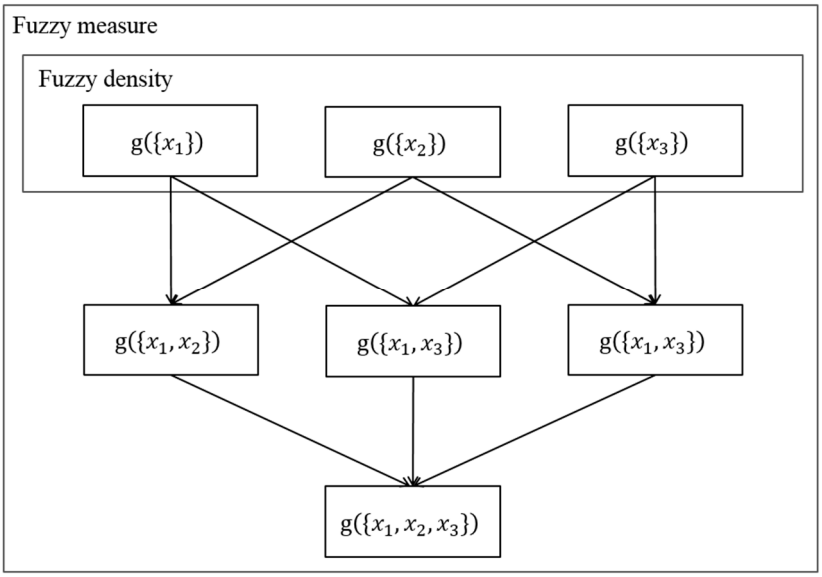

Figure 4. Relation between fuzzy measure and fuzzy density.

The fuzzy density of the evolutionary fuzzy integral is $n \times k$ floating point numbers, where n is the number of classifiers, i.e., the number of CNNs in the system, and k is the number of categories in a classification problem. The dimension of the solution space is $n \times k$ and ranges from 0 to 1.

Figure 5 shows the flowchart of the particle swarm optimization (PSO). The PSO is used for determining the fuzzy density in fuzzy integral. Initially, M number of particles p are randomly generated in the solution space, and each particle represents a set of fuzzy densities, as shown in Figure 6.

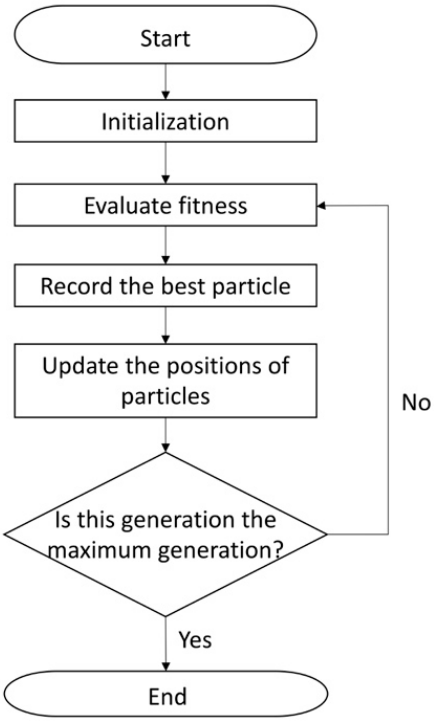

Figure 5. Flowchart of the particle swarm optimization.

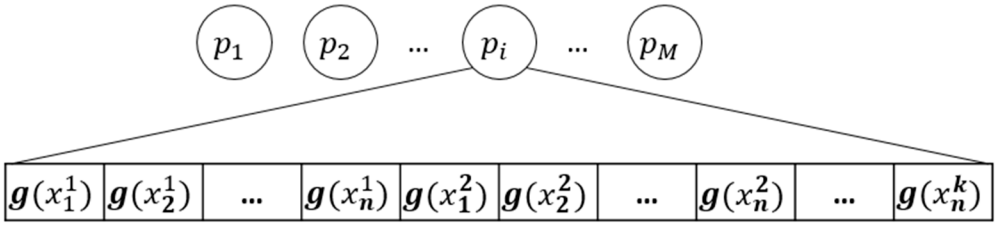

Figure 6. Encoding of particles.

The fitness value of each particle is evaluated as follows:

$$Fitness = \frac{TP}{TP + FP}, \tag{6}$$

where TP is the number of samples that are predicted correctly, and FP is the number of samples that are prejudged to be wrong.

Among the whole particle swarm, the best fitness value among all the particles is denoted $Gbest$, and the best fitness value in one particle during an iteration is denoted $Pbest_i$ for $i = \{1, 2 \ldots, M\}$. When updating the position of a particle, the particle's fitness value will be compared with $Gbest$ and $Pbest_i$. If the particle's current location is better than its historical locations, the particle will be updated and recorded. The location update formula for a particle swarm is as follows:

$$\begin{aligned} v_{id}(t) = w * v_{id}(t-1) + c_1 * rand(\,) * (Pbest_i - p_{id}(t)) + c_2 * rand(\,) \\ *(Gbest - p_{id}(t)), \end{aligned} \tag{7}$$

$$p_{id}(t+1) = p_{id}(t) + v_{id}(t), \tag{8}$$

where $v_{id}(t)$ represents the velocity of the ith particle in the dth dimension at the tth generation, $p_{id}(t)$ is the location of the ith particle in the dth dimension at the tth generation, and $rand()$ is a random number from 0 to 1. The parameter w is linearly decreasing from 0.8 to 0.2 according to the number of generations, and c_1 and c_2 are usually both set to 2.

In Equations (7) and (8), in order to find the optimal fuzzy density, each particle changes its location according to its past experiences and group experiences. The searching process stops when the maximum generation number is reached. After calculating the fuzzy density, the fuzzy measure can be determined. The fuzzy measure formula is as follows:

$$g(A \cup B) = g(A) + g(B) + \lambda g(A) g(B), \text{ where } A,\ B \subset X, \tag{9}$$

$$\lambda + 1 = \prod_{i=1}^{n} \left(1 + \lambda g^i\right). \tag{10}$$

The value of λ is equal to the output dimension in the evolutionary fuzzy integral, which means that each output category can calculate one λ, where $\lambda \in (-1, \infty)$ and with the following characteristics:

(1) If $\sum_{i=1}^{n} g^i = 1$, then $\lambda = 0$;
(2) If $\sum_{i=1}^{n} g^i < 1$, then $\lambda > 0$;
(3) If $\sum_{i=1}^{n} g^i < 1$, then $-1 \leq \lambda > 0$.

There are many methods for calculating the fuzzy integral, such as Sugeno [20], Choquet [21], ordered weighted averaging AND operator (OWA-AND) [22], ordered weighted averaging OR operator (OWA-OR) [23], and Fuzzy min–max [24]. The Sugeno and Choquet methods were chosen in this study. The steps to calculate Sugeno and Choquet fuzzy integrals are as follows:

First, we sort the n classifiers according to their output:

$$h(x_{\pi_1}) \geq h(x_{\pi_2}) \geq \cdots h\left(x_{\pi_j}\right) \geq \cdots \geq h(x_{\pi_n}), \tag{11}$$

where π_j represents the jth largest output belonging to a classifier. For example, $\pi_2 = 3$ reveals that the output of the third classifier is the second largest number in all the outputs. $\mathrm{h}(x_{\pi_1})$ indicates the maximum output value of the classifier.

Second, we define set A_i as follows:

$$A_0 = \{\varnothing\}, A_i = A_{i-1} + \left\{x_{\pi_i}\right\}. \tag{12}$$

Lastly, we calculate the Sugeno and Choquet fuzzy integrals as follows.

Sugeno FI:

$$Y_s = \bigvee_{i=1}^{N} \left(h(x_{\pi i}) \bigwedge g(A_i)\right). \tag{13}$$

Choquet FI:

$$Y_c = \sum_{i=1}^{n} h(x_{\pi i})(g(A_i) - g(A_{i-1})). \tag{14}$$

The two fuzzy integrals are calculated at the same time; the fuzzy integral with higher accuracy is then chosen.

4. Experimental Results

In order to evaluate the proposed method of EFI-CNNs, three CNNs—AlexNet [2], VGG16 [25], and GoogLeNet [4]—were used to experiment on face classification by both gender and age range with three popular gradient descent optimization algorithms—stochastic gradient descent (SGD), adaptive subgradient (Adagrad), and adaptive moment estimation (Adam) [26]. In addition, to prove that the evolutionary fuzzy integral works, the conventional fuzzy integral (FI) was also compared with the experimental results.

To evaluate the recognition rate of fuzzy integrals objectively, the CIA, Morph, and CACD2000 face databases were used in this study. Each database was divided into three parts: Training data, verified data, and test data. Only training data were involved in the convolutional neural networks. Then, the verification data were used to adjust the fuzzy density evolutionally. Finally, we used the test data to compare the final accuracy. In the evolution of the fuzzy density, the maximum number of iterations was set to 1000.

4.1. CIA Database

The CIA database is a small Chinese face database collected by a research laboratory. Some examples from the CIA database are shown in Figure 7. The age distribution is from 6 to 80 years old. The numbers of images per gender and age range are shown in Tables 1 and 2. Two identified problems, age and gender, were classified using the different CNN architectures and different optimization methods for training. The results are shown in Tables 3 and 4. Table 5 reveals the results using EFI-CNN to identify age and gender. The results explain that in the examined gender and age ranges, EFI has higher recognition rates.

Table 1. Age ranges in the CIA database.

Class	<12	13~19	20~45	>45
Quantity	313	592	934	248

Figure 7. Examples from the computational intelligence application laboratory (CIA) face database.

Table 2. Genders in the CIA database.

Class	Male	Female
Quantity	1080	1008

Table 3. Age classification results of the CIA database by the three CNNs.

Age	AlexNet [2]	VGG16 [25]	GoogLeNet [4]
SGD	69.26	69.82	69.31
Adagrad	63.48	67.66	68.43
Adam	69.18	**70.57**	69.78

SGD—stochastic gradient descent; Adagrad—adaptive subgradient; Adam—adaptive moment estimation.

Table 4. Gender classification results of the CIA database by the three CNNs.

Gender	AlexNet [2]	VGG16 [25]	GoogLeNet [4]
SGD	84.29	84.35	84.30
Adagrad	81.78	82.10	83.52
Adam	83.28	83.47	**85.02**

Table 5. Classification results of the CIA database by the different methods.

Methods	Age	Gender
AlexNet (SGD) [2]	69.26	84.29
VGG16 (Adam/SGD) [25]	70.57	84.35
GoogLeNet (Adam) [4]	69.78	85.02
AlexNet (SGD + SGD + SGD) with EFI	72.17	86.47
VGG16 (Adam + Adam + Adam) with EFI	71.26	87.28
GoogLeNet (Adam + Adam + Adam) with EFI	72.61	86.10
AlexNet (SGD + Adagrad + Adam) with FI	71.92	85.37
VGG16 (SGD + Adagrad + Adam) with FI	72.67	85.47
GoogLeNet (SGD + Adagrad + Adam) with FI	72.51	85.97
AlexNet (SGD) + VGG16 (Adam) + GoogLeNet (Adam) with FI	74.36	86.66
AlexNet (SGD + Adagrad + Adam) with EFI	73.34	87.39
VGG16 (SGD + Adagrad + Adam) with EFI	74.48	88.51
GoogLeNet (SGD + Adagrad + Adam) with EFI	73.91	89.85
AlexNet (SGD) + VGG16 (Adam) + GoogLeNet (Adam) with EFI	**76.02**	**90.61**

4.2. Morph Database

The Morph database is a Western-based face database [27]. It contains 55,000 face photos taken of 13,000 people. Figure 8 displays some photos from the Morph database. The age distribution is

between 16 and 77 years old. The average interval between photos of each person is 164 days, and there is no continuous shooting. The data on age ranges and gender are shown in Tables 6 and 7. Tables 8 and 9 illustrate the age ranges and gender accuracy in classification of the Morph database using various CNNs. Table 10 illustrates the results of EFI-CNN on the Morph database. From the classification results of age ranges, the different CNN architectures with EFI have a better recognition rate, and the gender classification accuracy reached 98.56%.

Figure 8. Examples from the Morph face database.

Table 6. Age ranges in the Morph database.

Class	<25	25~35	36~45	>45
Quantity	16,632	14,141	15,609	8752

Table 7. Genders in the Morph database.

Class	Male	Female
Quantity	46,645	8489

Table 8. Age classification results of the Morph database by the three CNNs.

Age	AlexNet [2]	VGG16 [25]	GoogLeNet [4]
SGD	70.89	67.49	70.58
Adagrad	69.35	68.12	69.67
Adam	68.07	70.47	68.63

Table 9. Gender classification results of the Morph database by the three CNNs.

Gender	AlexNet [2]	VGG16 [25]	GoogLeNet [4]
SGD	96.55	97.40	96.91
Adagrad	97.57	96.81	97.30
Adam	95.71	97.38	**98.04**

4.3. CACD2000 Database

The CACD2000 database [28] consists of 160,000 photos taken of 2000 celebrities at different times. Five photos from the CACD2000 database are displayed in Figure 9. Table 11 shows that the age distribution is from 16 to 62 years old. In this database, there is no gender label, so no gender classification was performed in this experiment. The celebrities in the database are mainly non-Asian people. Each photo was taken in different lighting and contains some noise. Table 12 shows the accuracy of CACD2000 database age classification using different CNNs. Since the photo source of the CACD2000 database was not manually filtered, much noise is mixed in, so the age recognition rates were lower than those for the CIA database and Morph. Table 13 reveals the identification results by EFI-CNN on the CACD2000 database. From the age group classification results, CNNs with different architectures combined with EFI had a better identification rate—7% higher than that of GoogLeNet CNNs.

Table 10. Classification results of the Morph database by the different methods.

Methods	Age	Gender
AlexNet (SGD/Adagrad) [2]	70.89	97.57
VGG16 (Adam/SGD) [25]	70.47	97.40
GoogLeNet (SGD/Adagrad) [4]	70.58	98.04
AlexNet (SGD + SGD + SGD) with EFI	71.31	97.64
VGG16 (Adam + Adam + Adam) with EFI	72.85	98.02
GoogLeNet (Adam + Adam + Adam) with EFI	72.77	**98.56**
AlexNet (SGD + Adagrad + Adam) with FI	72.52	97.75
VGG16 (SGD + Adagrad + Adam) with FI	72.27	98.03
GoogLeNet (SGD + Adagrad + Adam) with FI	72.72	98.22
AlexNet (SGD) + VGG16 (Adam) + GoogLeNet (Adam) with FI	74.48	98.28
AlexNet (SGD + Adagrad + Adam) with EFI	74.92	97.83
VGG16 (SGD + Adagrad + Adam) with EFI	74.66	98.28
GoogLeNet (SGD + Adagrad + Adam) with EFI	74.52	**98.56**
AlexNet (SGD) + VGG16 (Adam) + GoogLeNet (Adam) with EFI	**76.68**	98.52

Figure 9. Examples from the cross-age celebrity dataset (CACD2000) face database.

Table 11. Age ranges in the CACD2000 database.

Class	<25	25~35	36~45	>45
Quantity	29,029	38,595	42,588	53,234

Table 12. Age classification results of the CACD2000 database by the three CNNs.

Age	AlexNet [2]	VGG16 [25]	GoogLeNet [4]
SGD	59.08	58.53	**63.11**
Adagrad	61.95	57.31	60.13
Adam	62.13	61.39	61.41

Table 13. Classification results of the CACD2000 database by the different methods.

Methods	Age
AlexNet (Adam) [2]	62.13
VGG16 (Adam) [25]	61.39
GoogLeNet (SGD) [4]	63.11
AlexNet (Adam + Adam + Adam) with EFI	64.93
VGG16 (Adam + Adam + Adam) with EFI	63.58
GoogLeNet (SGD + SGD + SGD) with EFI	65.55
AlexNet (SGD + Adagrad + Adam) with FI	63.33
VGG16 (SGD + Adagrad + Adam) with FI	63.25
GoogLeNet (SGD + Adagrad + Adam) with FI	64.53
AlexNet (Adam) + VGG16 (Adam) + GoogLeNet (SGD) with FI	66.41
AlexNet (SGD + Adagrad + Adam) with EFI	67.04
VGG16 (SGD + Adagrad + Adam) with EFI	66.38
GoogLeNet (SGD + Adagrad + Adam) with EFI	67.15
AlexNet (Adam) + VGG16 (Adam) + GoogLeNet (SGD) with EFI	**69.71**

Further experiments with the databases UTKFace, face and gesture recognition network (FGNET), internet movie database – Wikipedia (IMDB-WiKi) were also implemented to support the proposed method. Different architectures LeNet, AlexNet, and GoogLeNet were used but with the same optimization methods mentioned above and the same parameter settings during the experiments. The age range classification results from the three datasets are listed in Table 14. The gender classification results from the UTKFace database and IMDB-WIKI database are displayed in Table 15. Both classification results reveal that integrating multiple CNNs with EFI allows for better accuracy than a single CNN.

Table 14. Age classification results in three databases.

Methods	UTKFace	FGNET	IMDB-WIKI
AlexNet (SGD) [2]	69.59	63.63	60.88
GoogLeNet (SGD) [4]	67.06	55.55	65.12
LeNet (SGD) [1]	62.88	54.54	51.84
AlexNet (SGD) + LeNet (SGD) + GoogLeNet (SGD) with EFI	**71.74**	**72.72**	**65.15**

Table 15. Gender classification results in UTKFace and IMDB-WiKi databases.

Methods	UTKFace	IMDB-WIKI
AlexNet (Adam) [2]	92.81	91
GoogLeNet (SGD) [4]	92.32	91.78
LeNet (Adagrad) [1]	90.45	85.84
AlexNet (Adam) + LeNet (Adagrad) + GoogLeNet (SGD) with EFI	**93.73**	**91.9**

4.4. Discussions

From Figure 10, we can see that the method proposed in this paper can effectively improve age and gender identification. Figures 11–13 illustrate one of the test images from each of the three databases. The results of these images are shown in Tables 16–18, and there are two wrong CNN predictions. This means that if the system uses a conventional voting concept, the prediction will be wrong. Instead of using a voting concept, the FI mechanism can get the correct prediction.

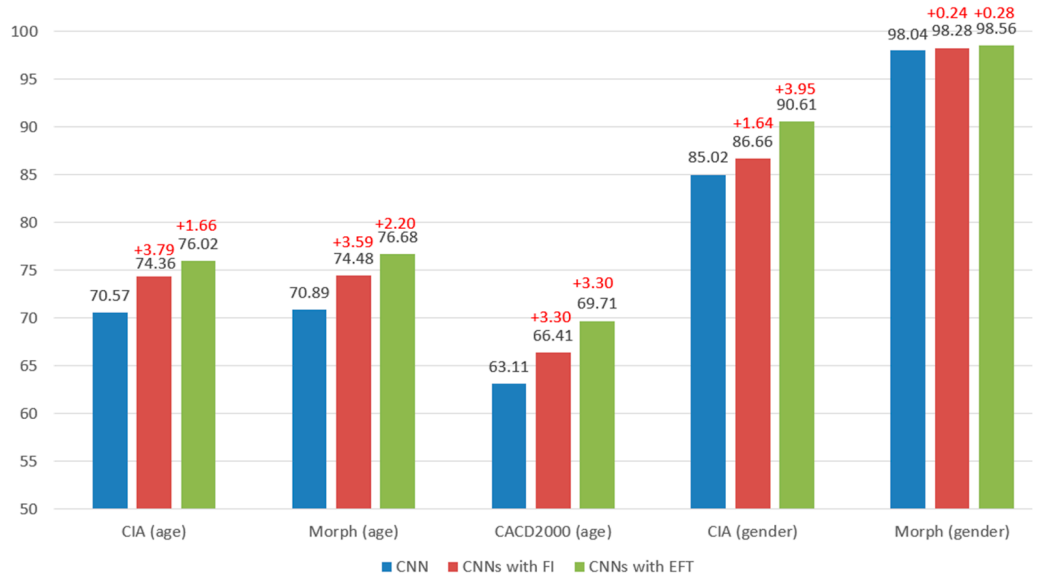

Figure 10. Accuracy in different CNNs on the three databases.

Figure 11. An example photo from the CIA database.

Figure 12. An example photo from the Morph database.

Figure 13. An example photo from the CACD2000 database.

Table 16. Recognition results of the CIA photo.

Classifier \ Classes	<12	13~19	20~45	>45
Ground truth	0	0	**1**	0
AlexNet [2]	0.00	**0.67**	0.30	0.03
VGG16 [25]	0.02	**0.70**	0.22	0.07
GoogLeNet [4]	0.00	0.21	**0.79**	0.00
Proposed method	0.00	0.58	**0.59**	0.06

Table 17. Recognition results of the Morph photo.

Classifier \ Classes	<25	25~35	36~45	>45
Ground truth	0	0	0	**1**
AlexNet [2]	0.05	0.18	**0.43**	0.35
VGG16 [25]	0.14	0.22	**0.37**	0.27
GoogLeNet [4]	0.01	0.08	0.26	**0.66**
Proposed method	0.01	0.19	0.40	**0.58**

Table 18. Recognition results of the CACD2000 photo.

Classifier \ Classes	<25	25~35	36~45	>45
Ground truth	0	0	**1**	0
AlexNet [2]	0.06	0.11	**0.72**	0.11
VGG16 [25]	0.19	**0.43**	0.32	0.07
GoogLeNet [4]	0.21	**0.39**	0.38	0.02
Proposed method	0.06	0.42	**0.55**	0.03

Figure 14 illustrates the accuracy of EFI combined with multiple CNNs. However, combining more than three CNNs does not significantly improve accuracy. This means that we cannot continuously increase the number of CNNs to improve the accuracy of the system, which will greatly increase the demand for hardware.

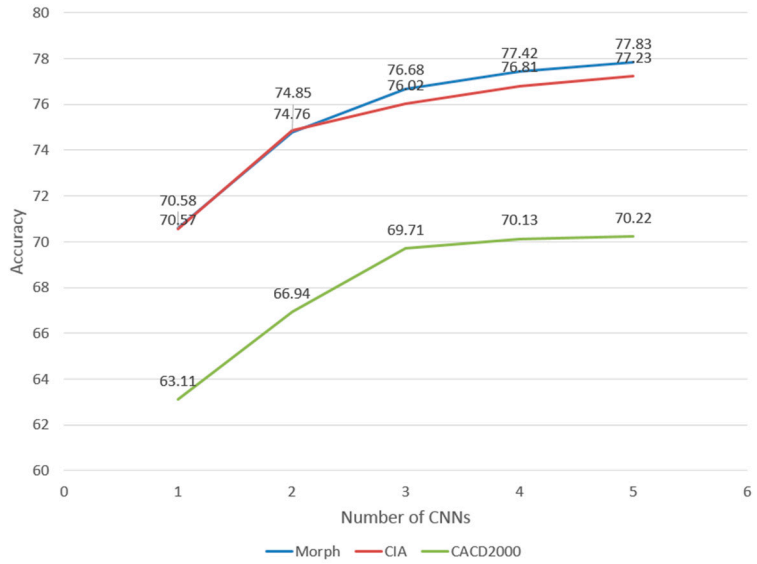

Figure 14. Age range recognition accuracy in multiple (1~5) CNNs with EFI.

5. Conclusions

In this paper, we utilized EFI-CNNs for age and gender classification of faces from the CIA, Morph, and CACD2000 databases based on fuzzy integral theory. The outputs acquired from trained CNNs were used as input to the EFI, and the final fuzzy integral rule was chosen from either Sugeno or Choquet during testing. Particle swarm optimization was applied to automatically search for the optimal fuzzy density value. The experimental results show that the proposed method obtains the best result when compared with the CNNs AlexNet, VGG16, and GoogLeNet. The proposed method improved the accuracy rates of age and gender classification by 5.95% and 3.1%, respectively. To further validate the proposed method, experiments and analysis were performed using more databases UTKFace, FGNET, and IMDB-WiKi. The experimental results show that the proposed method has better accuracy than other methods. In the future work, how to automatically determine the number of CNN and adjust the combination of CNN is a research direction that can effectively improve the classification accuracy.

Author Contributions: Conceptualization and methodology, C.-J.L., C.-C.S. and S.-H.W.; software, C.-J.L. and C.-H.L.; writing–original draft preparation, C.-J.L. and C.-H.L.

Funding: This research was funded by the Ministry of Science and Technology of the Republic of China, Taiwan (no. MOST 108-2634-F-005-001).

Conflicts of Interest: The authors declare that there are no conflicts of interest regarding the publication of this manuscript.

References

1. LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. *Gradient-based Learning Applied to Document Recognition*; Wiley-IEEE Press: Hoboken, NJ, USA, 1998; Volume 86, pp. 2278–2323.
2. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet Classification with Deep Convolutional Neural Networks. In Proceedings of the 25th International Conference on Neural Information Processing Systems, Lake Tahoe, NV, USA, 3–6 December 2012; pp. 1097–1105.
3. Lin, M.; Chen, Q.; Yan, S. Network in Network. In Proceedings of the International Conference on Learning Representations, Banff, AB, Canada, 14–16 April 2014.
4. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going Deeper with Convolutions. In Proceedings of the Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1–9.
5. Wu, S.; Zhong, S.; Liu, Y. Deep Residual Learning for Image Steganalysis. *Multimed. Tools Appl.* **2018**, *77*, 10437–10453. [CrossRef]
6. Wan, L.; Zeiler, M.; Zhang, S.; LeCun, Y.; Fergus, R. Regularization of Neural Networks Using Dropconnect. In Proceedings of the 30th International Conference on Machine Learning, Atlanta, GA, USA, 16–21 June 2013; pp. 1058–1066.
7. Maas, A.L.; Hannun, A.Y.; Ng, A.Y. Rectifier Nonlinearities Improve Neural Network Acoustic Models. In Proceedings of the International Conference on Machine Learning, Atlanta, GA, USA, 16–21 June 2013.
8. He, K.; Zhang, X.; Ren, S.; Sun, J. Delving Deep into Rectifiers: Surpassing Human-level Performance on Imagenet Classification. In Proceedings of the 2015 IEEE International Conference on Computer Vision, Washington, DC, USA, 7–13 December 2015; pp. 1026–1034.
9. Ramachandran, P.; Zoph, B.; Le, Q.V. Searching for Activation Functions. In Proceedings of the 6th International Conference on Learning Representations, Vancouver, BC, Canada, 30 April–3 May 2018.
10. Asness, C.S.; Moskowitz, T.J.; Pedersen, L.H. Value and Momentum Everywhere. *J. Financ.* **2013**, *68*, 929–985. [CrossRef]
11. Duchi, J.; Hazan, E.; Singer, Y. Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. *J. Mach. Learn. Res.* **2011**, *12*, 2121–2159.
12. Neural Networks for Machine Learning—Lecture 6a—Overview of Mini-Batch Gradient Descent. Available online: https://www.cs.toronto.edu/~{}hinton/coursera/lecture6/lec6.pdf (accessed on 27 August 2019).
13. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. In Proceedings of the International Conference on Learning Representations, San Diego, CA, USA, 7–9 May 2015.
14. Kornblith, S.; Shlens, J.; Le, Q.V. Do Better ImageNet Models Transfer Better? In Proceedings of the Computer Vision and Pattern Recognition, San Diego, CA, USA, 16–20 June 2019.
15. Tang, J.; Wen, G. Object Recognition via Classifier Interaction with Multiple Features. In Proceedings of the 8th International Conference on Intelligent Human-Machine Systems and Cybernetics, Hangzhou, China, 27–28 August 2016.
16. De Stefano, C.; Della Cioppa, A.; Marcelli, A. An adaptive weighted majority vote rule for combining multiple classifiers. In Proceedings of the 16th International Conference on Pattern Recognition, Quebec City, QC, Canada, 11–15 August 2002.
17. Williams, T.; Li, R. Wavelet Pooling for Convolutional Neural Networks. In Proceedings of the 6th International Conference on Learning Representations, Vancouver, BC, Canada, 30 April–3 May 2018.
18. Tong, B.; Liu, Y. An Speech and Face Fusion Recognition Method Based on Fuzzy Integral. In Proceedings of the IEEE International Conference on Robotics and Biomimetics, Qingdao, China, 3–7 December 2016.
19. Pandey, S.; Wu, L.; Guru, S.; Buyya, R. A Particle Swarm Optimization-based Heuristic for Scheduling Workflow Applications in Cloud Computing Environments. In Proceedings of the 24th IEEE International Conference on Advanced Information Networking and Applications, Perth, Australia, 20–23 April 2010.
20. Abbaszadeh, S.; Eshaghi, M.; de la Sen, M. The Sugeno fuzzy integral of log-convex functions. *J. Inequal. Appl.* **2015**, *2015*, 362. [CrossRef]

21. Torra, V.; Narukawa, Y. The interpretation of fuzzy integrals and their application to fuzzy systems. *Int. J. Approx. Reason.* **2006**, *41*, 43–58. [CrossRef]
22. Cho, S.B. Fuzzy Aggregation of Modular Neural Networks with Ordered Weighted Averaging Operators. *Int. J. Approx. Reason.* **1995**, *13*, 359–375.
23. Cheng, C.H.; Chen, C.T.; Huang, S.F. Combining fuzzy integral with order weight average (OWA) method for evaluating financial performance in the semiconductor industry. *Afr. J. Bus. Manag.* **2012**, *6*, 6358–6368.
24. Mesiar, R.; Mesiarova, A. Fuzzy integrals and linearity. *Int. J. Approx. Reason.* **2008**, *47*, 352–358. [CrossRef]
25. Liu, S.; Deng, W. Very Deep Convolutional Neural Network Based Image Classification Using Small Training Sample Size. In Proceedings of the 3rd IAPR Asian Conference on Pattern Recognition, Kuala Lumpur, Malaysia, 3–6 November 2015.
26. Ruder, S. An overview of gradient descent optimization algorithms. *arXiv* **2017**, arXiv:1609.04747.
27. Morph Database. Available online: https://ebill.uncw.edu/C20231_ustores/web/classic/store_main.jsp?STOREID=4 (accessed on 27 August 2019).
28. Chen, B.C.; Chen, C.S.; Hsu, W.H. Face recognition and retrieval using cross-age reference coding with cross-age celebrity dataset. *IEEE Trans. Multimed.* **2015**, *17*, 804–815. [CrossRef]