Engineering and Applied Science Research 2021;48(5):560-569 Research Article

EASR

Engineering and Applied Science Research

https://www.tci-thaijo.org/index.php/easr/index

Published by the Faculty of Engineering, Khon Kaen University, Thailand

Optimal weighted parameters of ensemble convolutional neural networks based on a differential evolution algorithm for enhancing pornographic image classification

Sarayut Gonwirat[1)] and Olarik Surinta*[2)]

[1)]Department of Computer and Automation Engineering, Faculty of Engineering and Industrial Technology, Kalasin University, Kalasin 46000, Thailand
[2)]Department of Information Technology, Faculty of Informatics, Mahasarakham University, Maha Sarakham 44150 Thailand

Received 18 December 2020
Revised 14 February 2021
Accepted 17 February 2021

Abstract

Use of ensemble convolutional neural networks (CNNs) has become a more robust strategy to improve image classification performance. However, the success of the ensemble method depends on appropriately selecting the optimal weighted parameters. This paper aims to automatically optimize the weighted parameters using the differential evolution (DE) algorithm. The DE algorithm is applied to the weighted parameters and then assigning the optimal weighted to the ensemble method and stacked ensemble method. For the ensemble method, the weighted average ensemble method is applied. For the stacked ensemble method, we use the support vector machine for the second-level classifier. In the experiments, firstly, we experimented with discovering the baseline CNN models and found the best models on the pornographic image dataset were NASNetLarge with an accuracy of 93.63%. Additionally, three CNN models, including EfficientNetB1, InceptionResNetV2, and MobileNetV2, also obtained an accuracy above 92%. Secondly, we generated two ensemble CNN frameworks; the ensemble learning method, called Ensemble-CNN and the stacked ensemble learning method, called StackedEnsemble-CNN. In the framework, we optimized the weighted parameter using the DE algorithm with six mutation strategies containing rand/1, rand/2, best/1, best/2, current to best/1, and random to best/1. Therefore, the optimal weighted was given to classify using ensemble and stacked ensemble methods. The result showed that the Ensemble-3CNN and StackedEnsemble-3CNN, when optimized using the best/2 mutation strategy, surpassed other mutation strategies with an accuracy of 96.83%. The results indicated that we could create the learning method framework with only 3 CNN models, including NASNetLarge, EfficientNetB1, and InceptionResNetV2.

Keywords: Pornographic image classification, Differential evolution algorithm, Mutation strategy, Convolutional neural networks, Ensemble convolutional neural networks, Stacked ensemble learning method, Ensemble learning method

1. Introduction

Image classification is an essential visual recognition process applied to many image recognition domains such as medical imaging, remote sensing, and pavement distress [1-4]. Beforehand, many complicated processes must be managed in the image classification system, such as preprocessing, feature extraction, and classification [1].

The convolutional neural networks (CNNs) architecture [5] was proposed to address the complicated processes of handwritten zip code recognition. The CNN architecture combined all the complex processes into the architecture. In the first part of the CNNs, the convolution layers are used to extract the spatial feature, called the deep feature method, by computing a convolution between a kernel and an input image. In the second part, the fully connected layers, also known as the neural network, are the classification task used to train the network and give the final probabilities for each image. In 2012, Krizhevsky et al. [6] proposed deep CNN, namely AlexNet architecture, to challenge the ImageNet classification task. Currently, CNNs have become more popular and widely used in image recognition, even detection and segmentation tasks. Many effective CNN architectures were proposed, such as MobileNets [7], DenseNet [8], and EfficeintNet [9].

To increasing the CNNs performance, an ensemble learning method was proposed, called ensemble CNN [4, 10, 11]. The ensemble CNN idea is to create parallel CNN models, which combined one or more CNN models, to learn from the training set and compute output probabilities. Subsequently, the output probabilities of each CNN model are then applied to the ensemble method as the input vector. The ensemble learning method is computed by various techniques, including majority vote, unweighted average, and weighted average techniques [4].

Furthermore, the second-level learning algorithm, the extended version of the ensemble method, was proposed, called the stacked ensemble learning method or meta-learning [12]. It was applied and improved the accuracy performance in many applications, such as surface electromyography (sEMG) of hand movements [13] and diagnosis of COVID-19 disease.

*Corresponding author. Tel.: +66 4375 4359
Email address: olarik.s@msu.ac.th
doi: 10.14456/easr.2021.58

According to the optimization of a weighted parameter, the differential evolution algorithm, a simple and effective optimization algorithm, is applied in many domains, such as feature selection, designing CNN architecture, optimizing weighted parameters [14-16].

Contribution of this research. This research mainly focuses on optimizing weighted parameters using the differential evolution (DE) algorithm for enhancing the accuracy and performance of pornographic image classification. The contributions are shown as follows:

1. We hypothesized that the ensemble CNNs would demonstrate a more robust performance with the pornographic classification system. In order to achieve the hypothesis, we proposed two ensemble CNN methods, including Ensemble-CNN and StackedEnsemble-CNN. These two ensemble methods were generated by combining the state-of-the-art CNN models, including MobileNetV2, InceptionResNetV2, EfficientNetB1, and NASNetLarge. For the Ensemble-CNN method, the unweighted average method was proposed, then the maximum probability was selected as the final prediction. For the StackedEnsemble-CNN method, we considered the support vector machine (SVM) with the radial basis function (RBF) kernel, the second-level classifier, to create the second model from the output probabilities of the CNN models. The final prediction was classified by the SVM algorithm.

2. For the Ensemble CNN methods, usually, the unweighted average method was proposed. In this paper, therefore, we examined the weighted average method using the differential evolution (DE) algorithm to optimize the weighted parameters of the CNNs automatically. As for the DE algorithm, to achieve the optimal weighted parameters, we evaluated six mutation strategies consisting of rand/1, rand/2, best/1, best/2, current to best/1, and random to best/1.

This paper is organized in the following way. The literature review is described in Section 2. Section 3 describes the proposed ensemble convolutional neural network. Section 4 presents the experimental setup, experimental results, and discussion. The conclusion and future work are presented in Section 5.

2. Literature review

In this section, we survey various techniques that have been proposed for image classification systems and relevant for the proposed method.

2.1 Ensemble convolutional neural networks (CNNs)

Wu et al. [10] applied an ensemble CNN for automatic classification of phonocardiograms. First, a Savitzky-Golay filter was proposed to denoise the heart sound and transformed to the frequency domain using a Fourier transform. Second, three feature extraction, including spectrogram, Mel spectrogram, and MFCC, were used as the input of each CNN model, called the VGGNet. Third, the output of the VGGNet models was sent to predict using the majority voting method.

Vasan et al. [11] proposed ensemble CNNs to classify image-based malware, called the IMCEC method. The Malimg dataset used in the experiment belonged to 25 different malware categories. In their method, the CNN models; VGG16 and ResNet50 were proposed to extract the deep feature and classification task. Consequently, the deep feature was given to the support vector machine (SVM) to classify. Also, the posterior probabilities from the previous step were then classified using the average ensemble method. Hence, the IMCEC method obtained more than 98% accuracy on the Malimg dataset.

Han et al. [4] proposed the ChexRadiNet algorithm, a light-weight deep CNN network, to classify chest X-Ray images of 14 chest diseases. For the ChexRadiNet algorithm, the ResNet-18 and ResNet-50 with the triplet mechanism were used. The set of feature maps were used as the output of this network. Moreover, the class activation mappings (CAMs) were then applied to generate the heatmaps that localize at the most indicative area.

2.2 Stacked ensemble learning method

Stacked ensemble learning is the method of enhancing the learning performance of a learning algorithm (namely second-level learning algorithm) over multiple learning (namely first-level learning algorithm), called learning to learn [12, 13, 17]. We can also call it the meta-learning method.

Shen et al. [13] proposed stacked ensemble learning to classify multi-channel surface electromyography (sEMG) hand movements. In their approach, the input data to the first-level classifiers, included time-domain data, frequency-domain data, and time-frequency-domain data. The CNN architectures were applied as the primary classifier. Further, the output data of three CNN models were concatenated and predicted using linear discriminant analysis, long short-term memory-CNN, support vector machine, and random forest. The stacked ensemble learning method obtained the best performance.

Gour and Jain [17] invented the stacked CNNs algorithm for the diagnosis of COVID-19 disease from chest X-ray images. In the first-level algorithm, the X-ray images were trained from the VGG16 and CovNet30 networks. The output of the CNNs was assigned as the input to the second-level algorithm. In the second-level algorithm, the output of CNNs was trained using a logistic regression algorithm and predicted as normal, pneumonia, or COVID-19 categories.

2.3 Differential evolution (DE) algorithm

Many researchers have recently employed evolutionary algorithms in feature selection, dimensionality reduction, classification, and unsupervised image classification tasks [18]. The DE algorithm has also been effectively applied to a continuous optimization problem suitable for discovering the weighted parameter of the ensemble CNNs. Wang et al. [19] proposed a hybrid DE approach with a new crossover operator to create deep CNNs. In their method, the DE was proposed to optimize the hyper parameters as follows; filter size, number of feature maps, stride size, and number of neurons at the fully connected layer.

Zhang et al. [14] proposed a DE algorithm for the weighted voting ensemble method to optimize the weighted parameter. First, five classifiers were selected as base models, including K-nearest neighbor, Naïve Bayes, C4.5, Bayesian networks, and ZeroR. Second, the weight outputs from five classifiers were then computed using the DE algorithm. Finally, the ensemble method with weighted voting was used to predict the output.

Furthermore, Dixit et al. [16] proposed to use the DE algorithm for feature selection and experimented on 25 benchmark datasets from the UCI repository. In their method, the mutation strategy with rand/2 was invented to control parameters. The optimal features that were selected using the DE algorithm were given to the SVM and Naïve Bayes for the classification process. Therefore, the experiments showed that using the DE algorithm as the feature selection improved the efficiency of the classification.

3. Ensemble convolutional neural network framework

This section describes the proposed ensemble convolutional neural network (CNN) framework to enhance the pornographic image classification performance, as shown in Figure 1. The proposed framework is divided into two main approaches; 1) Ensemble-CNN method (see Figure 1a) and 2) Stacked Ensemble-CNN method (see Figure 1b). Moreover, we include the differential evolution algorithm with six mutation strategies that aim to optimize the weighted parameters into both frameworks.

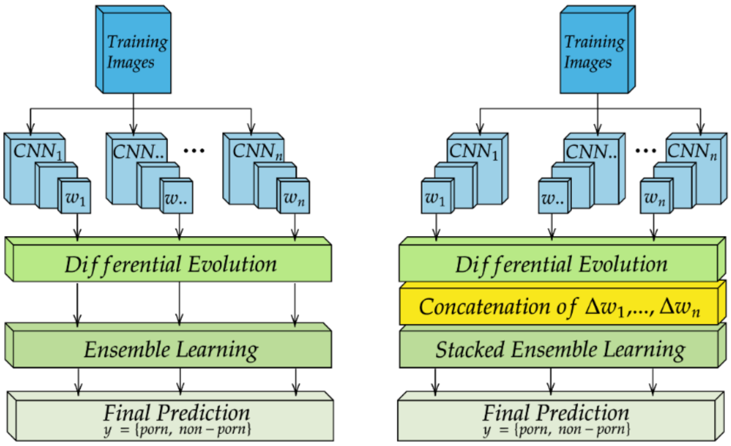

Figure 1 Illustration of the ensemble convolutional neural network architectures using differential evolution algorithms with mutation strategy. (a) Ensemble-CNN method and (b) Stacked Ensemble-CNN method.

In Figure 1a, we first propose different CNN models ($CNN_1,\ldots,CNN_n$; where n is the number of CNN models) to learn the pornographic image dataset. The output probabilities of the CNN models are computed using the softmax function [20]. In ensemble CNN, obtaining the optimal weighted parameters might be the main concern. We therefore propose using the DE algorithm to address the continuous number of the weighted parameters. Second, the weighted parameters $(w_1,\ldots,w_n)$ are automatically calculated using the differential evolution (DE) algorithm with different mutation strategies. Finally, for the ensemble method, we compute the output probabilities and optimal weighted parameters. The highest probability is decided as a final output(y). The differential evolution algorithm and ensemble learning method are described in Section 3.1 and Section 3.2.

Figure 1b describes the StackedEnsemble-CNN method. We apply the DE algorithms into this framework as well. Subsequently, the new optimal weighted parameters $(\Delta w_1,\ldots,\Delta w_n)$ are then calculated with the output probabilities of the CNN models and used as the training data. Furthermore, the support vector machine (SVM) [21] is proposed to train the data and create the second model. Also, the final decision is performed in this step. We will describe the stacked ensemble learning method in Section 3.3.

3.1 Differential evolution algorithm

Storn and Price [22] invented the heuristic algorithm to minimize continuous space functions and proved that it effectively finds an optimal solution for global optimization [14, 15], called the differential evolution (DE) algorithm. The DE algorithm contains four main operations; population initialization, mutation, crossover, and selection [14, 23]. The pseudocode of the DE algorithm is shown in Algorithm 1.

Algorithm 1 The differential evolution algorithm.

INPUT: The DE parameters: mutation factor (F), crossover rate (CR), and population (N)
    Population initialization: $X_G = \{x_{1,G}, x_{2,G},\ldots,x_{N,G}\}$
    Generation iteration: $G = 0$
WHILE the stopping condition is not satisfied
  FOR $i = 1,\ldots,N$
    Select random indexer: r_1, r_2, r_3, r_4
    $V_{i,G} = generateMutant(X_G)$
    $u_i = Crossover(x_G, V_{i,G})$
    IF $f(u_i) < f(x_G)$ THEN
      $x_i(G+1) = u_i$
    ELSE
      $x_{i,G+1} = x_G$
  END FOR
  Update generation iterator: $G = G + 1$
END WHILE
OUTPUT: The optimal weighted parameters $(w_1, w_2,\ldots,w_n)$

3.1.1 Population initialization

Initialization of the candidate number of populations is the first step of the DE algorithm. Let $x_{i,G}$, where $i = 1,2,...,N$, i is the index solution of the population at generation (G), and N represents the size of the population [14, 16]. We defined the population number as 30. Then, we have sufficient agents in the search-space. The solution vectors can be defined as follows;

$$x_i = rand_i(0,1) \cdot (x_i^{max} - x_i^{min}) + x_i^{min} \quad (1)$$

where x^{max} and x^{min} are boundary of the parameters, $rand_i(0,1)$ is an integer random number (0,1).

3.1.2 Mutation

Mutation operation is a benchmark strategy for evaluating the performance of the DE algorithms [16]. The mutant vector $V_{i,G}$ is generated using the mutation strategy and $X_{i,G}$ is a target vector in the current generation [14]. The benchmark mutation strategies are defined by the following equations;

DE/rand/1: $V_{i,G} = X_{r1,G} + F \times (X_{r2,G} - X_{r3,G})$ (2)

DE/rand/2: $V_{i,G} = X_{r1,G} + F \times (X_{r2,G} - X_{r3,G}) + F \times (X_{r4,G} - X_{r5,G})$ (3)

DE/best/1: $V_{i,G} = X_{best,G} + F \times (X_{r1,G} - Xr_{2,G})$ (4)

DE/best/2: $V_{i,G} = X_{best,G} + F \times (X_{r1,G} - X_{r2,G}) + F \times (X_{r3,G} - X_{r4,G})$ (5)

DE/current to best/1: $V_{i,G} = X_{i,G} + F \times (X_{best,G} - X_{i,G}) + F \times (X_{r1,G} - X_{r2,G})$ (6)

DE/random to best/1: $V_{i,G} = X_{r1,G} + F \times (X_{best,G} - X_{r1,G}) + F \times (X_{r2,G} - X_{r3,G})$ (7)

where $r_1, r_2, ..., r_5$ are the mutual integers that random within the range $1,2,...,N$. F is a mutation scaling factor, and $X_{best,G}$ is the vector with the best fitness in generation G. In these experiments, we defined the scaling factor as 0.5.

3.1.3 Crossover

The crossover operation is calculated as a new vector, called a trial vector $u_{i,G}$, it performs between the target vector $X_{i,G}$ and mutant vector $V_{i,G}$ using a crossover probability. The crossover operation is given as follows;

$$u_{i,G}(j) = \begin{cases} v_{i,G}(j), if rand_{i,j}(0,1) \leq CR \vee j = j_{rand} \\ X_{i,G}(j), otherwise \end{cases} \quad (8)$$

where CR is a crossover probability and the CR value is between 0, 1. In our experiments, we employed a crossover probability as 0.7.

3.1.4 Selection

The selection operation is then compared the trial individual vector $u_{i,G}$ to the target vector $X_{i,G}$ and measured according to the fitness value. The best optimized is selected for the next generation. The selection operation is defined as follows;

$$X_{i,G+1} = \begin{cases} u_{i,G}, iff(u_{i,G}) \leq f(X_{i,G}) \\ X_{i,G}, otherwise \end{cases} \quad (9)$$

where $f(\,)$ is the objective function to be optimized.

The mutation, crossover, and selection operations are reproduced until the termination condition is satisfied.

3.2 Ensemble learning method

In this approach, the weighted average method [24] is proposed as the ensemble learning method. First, the softmax function [15] is proposed to represent the output probability. The equation of the softmax function can be computed as;

$$\sigma(\overrightarrow{w_i}) = \frac{e^{w_i}}{\sum_{j=1}^{k} e^{z_j}}, i = 1,2,...,N \quad (10)$$

where $\sigma(\overrightarrow{w_i})$ is the output probability that belongs to the class k in the multi-class classifier, $\overrightarrow{z_i}$ is input vector, e is standard exponential function, and i is input values of the softmax layer.

Second, the output probabilities of CNN models are computed with a different weighting that is optimized using the differential evolution algorithm. Each ensemble member provides the output probabilities depending on the optimized weight [25] and uses them as the final prediction. The weighted average method can be defined as follows;

$$\hat{y}_i = \frac{1}{n} \sum_{j=1}^{n} \alpha_j \, \sigma(\vec{w})_j \quad (11)$$

where $\hat{y}_i$ is the output of the ensemble method, $\sigma(\vec{w})_j$ is the output probabilities of the model (j), n is the number of ensemble models, α is an optimized weight that computed using differential evolution algorithm.

Finally, the largest element in $\hat{y}_i$ is considered by applying the argmax function [26].

3.3 Stacked ensemble learning method

A stacked ensemble learning method is an alternative approach to the ensemble learning method [27]. The outputs of the first-level classifiers (i.e., output probability and the feature vector of the CNN models) are used as the input data of the second-level classifier [28]. In our experiments, we considered the differential evolution algorithm to optimize the output probability $(w_1, \ldots, w_n)$. The new weighted parameters $(\Delta w_1, \ldots, \Delta w_n)$ were then concatenated and assigned to the second-level classifier. For the second-level classifier, the SVM classifier [21] was examined. The SVM algorithm was first proposed to address the problem of binary classification. The optimal hyperplane that separates the two classes of training points with the maximum distance between two-class was selected. The training points on the maximum-margin of the hyperplane are called the support vectors. The SVM equation is given as follows;

$$g(x) = w^T x + b \tag{12}$$

where w is the weight vector of the linear SVM and b is the bias value. The SVM output is the binary label of pattern x, $y_i \in \{1, -1\}$. The SVM output is equal or larger than the target output if the output is 1 and the SVM output is equal or lower than -1 if the output is -1. As a result, the optimal hyperplane obtains the maximum distance to the closest positives $w^T x + b = +1$ and negatives $w^T x + b = -1$.

For the non-linear similarity function, in this paper, we selected the radial basis function (RBF) kernel, which allows for obtaining more robust performance [29]. The formula to calculate the RBF kernel is defined by the following equation;

$$K(x_i, x_j) = exp\left(-\gamma \|x_i - x_j\|^2\right) \tag{13}$$

where K stands for the RBF kernel, x_i and x_j are input patterns of two-sample data, γ is a kernel parameter that determines the width of the kernel when $\gamma > 0$. The γ parameter requires to be adjusted on a validation set.

4. Experimental results and discussion

4.1 Experimental setup

In order to evaluate the performance of the ensemble convolutional neural networks, we examined pornographic image classification experiments using the TI-UNRAM pornographic image dataset.

The TI-UNRAM dataset was introducted by Wijaya et al. [30]. This dataset contains 1,400 images of two categories; non-pornographic (715 images) and pornographic images (685 images). This dataset entirely comprised images collected from the Internet. The pornographic images comprise diverse image characters, for example, sexual activity, naked, topless, nude, and sexual art. Consequently, non-pornographic images contain portraits, cartoons, underwear, scene, and men topless.

For the experiment, we provided a fair comparison using the same size of the training set and test set as Wijaya et al. [30]. The TI-UNRAM dataset was divided into training (50%) and test set (50%). It contained 700 images for each set. Additionally, the training set was separated by 90% for training (630 images) and 10% for validation (70 images). We trained the CNN model with only 100 epochs to avoid overfitting. The CNN architectures applied in the experiments were implemented using Keras API based on the Tensorflow framework. The specification of the hardware and software used in the experiments are given in Table 1.

Table 1 Hardware and software specifications were used in these experiments

Hardware	Software
Processor: Intel(R) Core(TM) i5-7400 @ 3.00 GHz	Operating System: Ubuntu 18.04.3 LTS
Memory: RAM 32GB DDR4, Speed 2667 MHz	Language: Python 3.6
Graphics Processing Unit: GeForce GTX 1080Ti 11GB GDDR5x	Deep Learning Framework: Tensorflow
	Deep Learning API: Keras

Table 2 Configurations of convolutional neural network architectures

Architecture	Input size	Batch size	Optimization
MobileNetV2	224x224	16	SGD
InceptionResNetV2	224x224	8	SGD
NASNetLarge	331x331	8	SGD
EfficientNetB1	331x331	8	SGD

Table 2 presented the configurations of the CNN architectures. The size of the input layer in CNN architectures was set depending on each CNN architecture and the output layer was set as two units. We decided to train the CNN models using the stochastic gradient descent (SGD) optimization algorithm. The hyper-parameters of the CNNs were set as follows; learning rate = 0.01, momentum = 0.9, weight decay = 0.01, and training with 100 epochs. In the experiment, the learning rate schedule strategy was employed as follows;

$$\Delta\alpha = \frac{\alpha}{1+\lambda * i} \tag{14}$$

where $\Delta\alpha$ is updated learning rate, α is initial learning rate, λ is weight decay, and i is the number of training epochs.

4.2 Performance comparison of different CNN architectures

In this experiment, four pre-trained state-of-the-art CNN models, including MobileNetV2, InceptionResNetV2, NASNetLarge, and EfficientNetB1, were proposed according to discover the base CNN models. We evaluated the CNN models on the TI-UNRAM pornographic image dataset. For the statistical test, the validity was evaluated on the test set. The evaluation metric used in the experiment, include average accuracy (%) and standard deviation. The experimental results are presented in Table 3

Table 3 Experimental results for different convolutional neural network architectures. All the results were reported in terms of average accuracy, standard deviation, and training time.

CNN architectures	Accuracy	Training time (hh:mm:ss)
MobileNetV2	92.98±0.82	00:13:42
InceptionResNetV2	94.29±1.69	00:32:07
EfficientNetB1	96.17±0.18	00:25:21
NASNetLarge	96.36±0.07	01:34:40

From Table 3, the experimental results of comparing different CNN architectures showed that all CNN models provided an accuracy above 92%, which was higher than in previous studies [30, 31]. The highest accuracies of 96.36% and 96.17% were obtained from the NASNetLarge and EfficientNetB1. The confusion matrix of the NASNetLarge model is shown in Figure 2a. However, when considering the computation time, the EfficientNetB1 model (25 mins) required 3.76 times fewer computation times than the NASNetLarge model (1h:34min). Figure 2 presents models of MobileNetV2, InceptionResNetV2, EfficientNetB1, and NASNetLarge.

Our results indicate that the best baseline CNN model is NASNetLarge and EfficientNetB1, which can be offered to solve the pornographic image classification problem. In the next experiments, we have evaluated the ensemble CNN which is based on four CNN models.

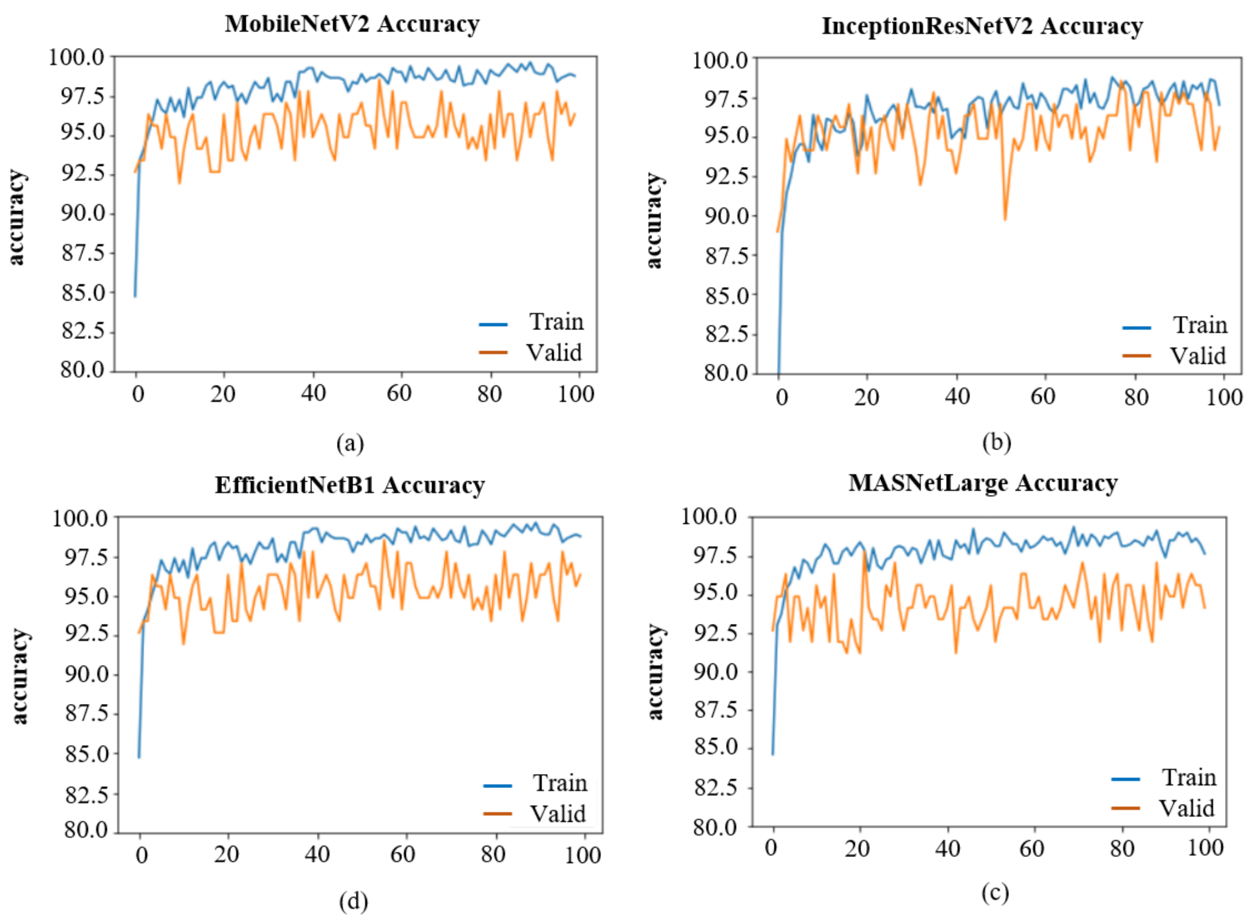

Figure 2 Illustration the CNN training and validation accuracy model of (a) mobileNetV2, (b) inceptionResNetV2, (c) efficientNetB1, and (d) NASNetLarge.

4.3 Experimental results between ensemble CNN and stacked ensemble CNN

In these experiments, we investigated the multiple CNN models, called ensemble CNNs, to enhance the accuracy of the pornographic classification. We evaluated two ensemble CNN architectures; ensemble learning (Ensemble-CNN) and stacked ensemble learning (stackedEnsemble-CNN). The results are showed in Table 4.

Table 4 Experimental results between ensemble CNNs and stacked ensemble CNNs.

Methods	Architectures	Accuracy	Testing time (second)
Ensemble-2CNN	MobileNetV2 NASNetLarge	96.74±0.08	0.001s
Ensemble-3CNN	MobileNetV2 EfficientNetB1 NASNetLarge	96.76±0.08	0.001s
Ensemble-4CNN	All CNN Models	96.62±0.42	0.001s
StackedEnsemble-2CNN	InceptionResNetV2 EfficientNetB1	96.76±0.11	0.44s
StackedEnsemble-3CNN	MobileNetV2 EfficientNetB1 NASNetLarge	96.76±0.18	0.52s
StackedEnsemble-4CNN	All CNN Models	96.67±0.29	0.58s

Table 4 reported that the Ensemble-3CNN (including MobileNetV2, EfficientNetB1, and NASNetLarge) obtained an accuracy of 96.76% and surpassed Ensemble-2CNN (MobileNetV2 and NASNetLarge) with only 0.02%. As a result, both Ensemble-CNN and StackedEnsemble-CNN performed an accuracy above 96%. However, when we compared the testing time, the Stacked Ensemble-CNN required more testing time because it spent more exceeding time (approximately 50 seconds) on the second-level classifier. While the Ensemble-CNN, which used the unweighted average method, spent only 0.001 seconds. From this experiment, we concluded that the Ensemble-2CNN method could be decreased the training time and the accuracy also above 96%. Further, we recommend applying the Ensemble-2CNN to address the problem of pornographic image classification.

As shown in Table 4, we then use the DE algorithm to optimize the weighted parameters using various mutation strategies. The detail of these experiments is shown as follows.

4.4 Comparison of the differential evolution algorithm using different mutation strategies

This study aimed to use the differential evolution (DE) algorithm, which is the optimization algorithm to optimize the weighted parameter. Further, we multiplied the weighted parameter (which was provided from the DE algorithm) with the output probabilities (obtained from the CNN models), called the weighted average ensemble method. The maximum output probabilities were selected and used as the final prediction. To perform the accuracy of the DE algorithm, we compared six mutation strategies; rand/1, rand/2, best/1, best/2, current to best/1, and random to best/1. To prove that the DE algorithm could enhance the accuracy performance of the image classification, we evaluated the DE algorithm on two main methods, including Ensemble-CNN and StackedEnsemble-CNN. For the StackedEnsemble-CNN, we examined the SVM classifier with the radial basis function (RBF) kernel to train the optimal output probabilities of the CNN models. The RBF kernel parameters were defined as; regularization term (C) = 1 and gamma (γ) = 0.001. The results of the DE with different mutation strategies applied to the Ensemble CNN methods are shown in Table 5.

Table 5 Experimental results (average accuracy and computation time) of the differential evolution algorithm with various mutation strategies.

Methods	Mutation Strategies / Testing Time (second)					
	rand/1	**rand/2**	**best/1**	**best/2**	**current to best/1**	**random to best/1**
Ensemble-3CNN	96.69±0.15 1.13s	96.83±0.30 1.27s	96.74±0.11 0.72s	96.83±0.25 1.33s	96.67±0.07 1.09s	96.69±0.21 0.79s
Ensemble-4CNN	96.52±0.04 2.07s	96.69±0.18 2.27s	96.71±0.19 1.63s	96.67±0.11 2.21s	96.67±0.04 1.46s	96.74±0.15 1.59s
StackedEnsemble-3CNN	96.52±0.61 118.36s	96.62±0.39 171.23s	96.55±0.48 71.03s	96.83±0.25 117.62s	96.57±0.40 82.71s	96.43±0.62 145.68s
StackedEnsemble-4CNN	96.67±0.32 166.14s	96.74±0.46 230.71s	96.83±0.29 197.84s	96.62±0.58 159.32s	96.62±0.53 111.22s	96.74±0.53 116.34s

The experimental results in Table 5 showed that the differential evolution algorithm using the best/2 mutation strategy, called DE/best/2, increased the accuracy performance of the Ensemble-CNN and StackedEnsemble-CNN methods on the TI-UNRAM pornographic image dataset. When optimizing the weighted parameters using with DE/best/2 algorithm, the Ensemble-3CNN and StackedEnsemble-3CNN performed with an equal accuracy of 96.83%. However, we compared the testing time and found that the Ensemble-3CNN computed only 1.33 seconds, while the StackedEnsemble-3CNN required 117.62 seconds. Because of the StackedEnsemble-3CNN method needed to assign the optimal output probabilities to the support vector machine (SVM) algorithm to classify as the final prediction.

Figure 3b and Figure 3c illustrated the confusion matrix of the Ensemble-3CNN method and Ensemble-3CNN method with DE/best/2 algorithm. The confusion matrix showed that the Ensemble-3CNN method with DE/best/2 algorithm (see Figure 3c) reduced the misclassified images from 12 images (see Figure 3b) to only 8 images.

Moreover, if the testing time is the main concern, we will consider the Ensemble-3CNN and Ensemble-3CNN using DE/best/2 algorithm. The Ensemble-3CNN computed in 0.001 seconds and Ensemble-3CNN using DE/best/2 algorithm computed in approximately 2 seconds on 700 images. While the StackedEnsemble-3CNN spent around 0.52 seconds and the StackedEnsemble-3CNN using DE/best/2 algorithm spent between 71-197 seconds.

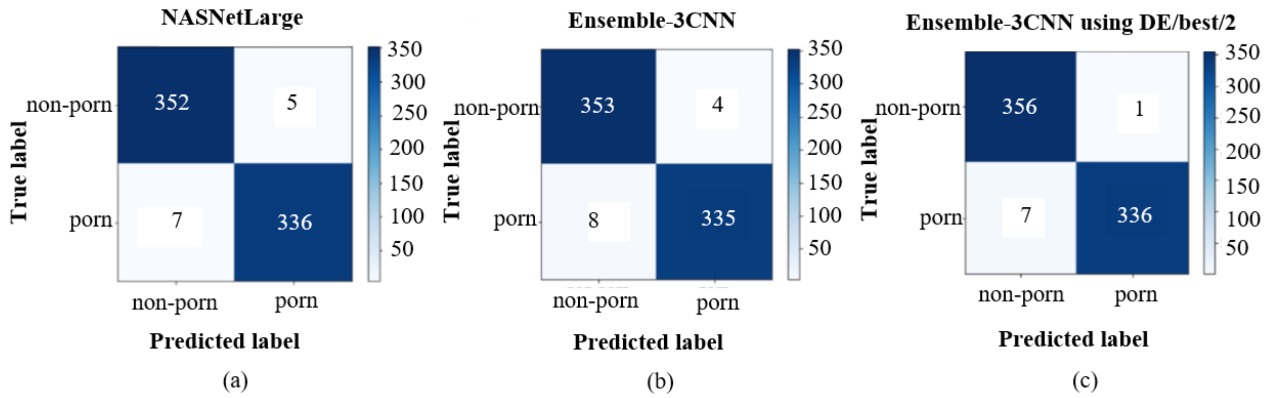

Figure 3 Confusion matrix for two classes of the pornographic image classification with (a) NASNetLarge model, (b) ensemble-3CNN method, and (c) ensemble-3CNN method with DE/best/2 algorithm.

4.5 Comparison of the proposed ensemble CNN with the previous studies

The experimental results in Table 5 presented that Ensemble-3CNN and StackedEnsemble-3CNN obtained the highest performance when using the DE algorithm with the best/2 mutation strategy, called DE/best/2 algorithm. In this section, we then compared the proposed ensemble CNN methods using the DE algorithm consisting of Ensemble-CNN and StackedEnsemble with the previous studies. The results are compared and the weighted parameters are shown in Tables 6 and 7.

Table 6 The performance comparison of different methods on the TI-UNRAM dataset

Methods	Testing time	Accuracy
FD+YCbCr [31]	-	83.97
SP+EP (SEP) [30]	-	90.13
ResNet50 [32]	-	88.00±0.37
Ensemble-3CNN	0.001s	96.76±0.08
StackedEnsemble-3CNN	0.52s	96.76±0.18
StackedEnsemble-3CNN (with DE/best/2)	117.62s	96.83±0.25
Ensemble-3CNN (with DE/best/2)	1.33s	96.83±0.25

Table 7 The optimal weighted parameters of the ensemble CNNs.

Methods	Optimal weighted parameters		
	MobileNetV2	**NASNetLarge**	**EfficientNetB1**
StackedEnsemble-3CNN (with DE/best/2)	0.0537	0.5491	0.3972
Ensemble-3CNN (with DE/best/2)	0.0567	0.9380	0.0053

5. Conclusion and future work

We proposed two ensemble convolutional neural network (CNN) frameworks, called Ensemble-CNN and StackedEnsemble-CNN, to enhance the classification accuracy on the TI-UNRAM pornographic image dataset. Further, we aimed to discover the optimal weighted parameters of the CNNs automatically using the differential evolution (DE) algorithm. Several experiments on the mutation strategies, including rand/1, rand/2, best/1, best/2, current to best/1, and random to best/1, were examined to evaluate our ensemble CNN frameworks.

In the first experiment, we discovered the CNN models based on experimenting with several state-of-the-art CNN models; MobileNetV2, InceptionResNetV2, NASNetLarge, and EfficientNetB1. We found that NASNetLarge, EfficientNetB1, and InceptionResNetV2 models obtained accuracies above 94%. While the MobileNetV2 gave an accuracy of 92.98%. Moreover, we created two ensemble CNN frameworks based on the experimental results shown in Table 3.

In the second experiment, we evaluated two ensemble CNNs; Ensemble-CNN and StackedEnsemble-CNN methods. First, we created the ensemble CNNs with 2 CNN models (consisting of MobileNetV2 and NASNetLarge), called"Ensemble-2CNN", 3 CNN models (consisting of NASNetLarge, EfficientNetB1, and InceptionResNetV2), called"Ensemble-3CNN", and four CNN models, called"Ensemble-4CNN." Second, we constructed the stacked ensemble CNNs, called "StackedEnsemble-CNN." In StackedEnsemble-CNN, the support vector machine (SVM) with the radial basis function (RBF) kernel was proposed for the second-level classifier. We observed that the Ensemble-3CNN, StackedEnsemble-2CNN, and StackedEnsemble-3CNN obtained the highest accuracy of 96.76%. However, for the testing time, the Ensemble-CNN performed much faster than the StackedEnsemble-CNN. The Ensemble-CNN method needed only 0.001 seconds, while the StackedEnsemble-CNN required approximately 45-60 seconds on 700 images. From the experimental results shown in Table 4, when creating the ensemble CNN framework, it can be seen that only 2 CNNs and 3 CNNs perform better than 4 CNN models. We conclude that the ensemble CNN using only 2 CNN and 3 CNN can improve the ensemble CNN performance.

The third experiment, which was mainly focused on this paper. We intended to increase the accuracy performance of the Ensemble-CNN and StackedEnsemble-CNN methods using the DE algorithm. We then proposed using the DE algorithm with diverse mutation strategies to find the optimal weighted parameters. Thus, the output probabilities were computed with the optimal weighted parameters and used as the input of the second-level classifier. The experiment showed that the best optimization algorithm was the DE algorithm with the best/2 mutation strategy, called DE/best/2 algorithm, with an accuracy of 96.83%. Furthermore, we compared the proposed ensemble CNN methods combined with the DE/best/2 algorithm with previous studies. The proposed method showed significantly outperformed all previous methods.

In future work, we will concentrate on improving the performance of pornographic image classification by applying bio-inspired optimization algorithms such as particle swarm optimization (PSO), fish swarm optimization (FSO), cat swarm optimization (CSO), bee colony optimization (BCO), and ant colony optimization (ACO) [33-35]. We will experiment with other pornographic image datasets and other image classification datasets. In another direction, we will consider snapshot ensemble method. In another direction, we will consider using the snapshot ensemble method.

6. Acknowledgements

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was supported by Mahasarakham University [grant number 6308074/2563].

7. References

[1] Lu D, Weng Q. A survey of image classification methods and techniques for improving classification performance. Int J Rem Sens. 2007;28:823-70.
[2] Wang W, Yang Y, Wang X, Wang W, Li J. Development of convolutional neural network and its application in image classification: a survey. Opt Eng. 2019;58(4):1-19.
[3] Majidifard H, Jin P, Adu-Gyamfi Y, Buttlar WG. Pavement image datasets: A new benchmark dataset to classify and densify pavement distresses. arXiv:1910.11123. 2019:1-12.
[4] Han Y, Chen C, Tang L, Lin M, Jaiswal A, Ding Y, et al. Using radiomics as prior knowledge for abnormality classification and localization in chest x-rays. Transport Res Rec J Transport Res Board. 2020;2674(2):1-11.
[5] LeCun Y, Matan O, Boser BE, Denker J, Henderson D, Howard R, et al. Handwritten zip code recognition with multilayer networks. 10^{th} International conference on pattern recognition (ICPR); 1990 Aug 16-21; Atlantic, USA. New York: IEEE; 1990. p. 35-40.
[6] Krizhevsky A, Sutskever I, Hinton GE. ImageNet classification with deep convolutional neural networks. Adv Neural Inform Process Syst. 2012;25(2):1-9.
[7] Howard AG, Zhu M, Chen B, Kalenichenko D, Wang W, Weyand T, et al. MobileNets: efficient convolutional neural networks for mobile vision applications. arXiv:1704.04861. 2017:1-9.
[8] Huang G, Liu Z, Van Der Maaten L, Weinberger KQ. Densely connected convolutional networks. 2017 IEEE Conference on computer vision and pattern recognition (CVPR); 2017 Jul 21-26; Honolulu, USA. New York: IEEE; 2017. p. 2261-9.
[9] Tan M, Le QV. EfficientNet: rethinking model scaling for convolutional neural networks. arXiv:1905.11946. 2019:1-11.
[10] Wu J, Tsai M-H, Huang Y, Islam SH, Hassan M, Alelaiwi A, et al. Applying an ensemble convolutional neural network with Savitzky-Golay filter to construct a phonocardiogram prediction model. Appl Soft Comput J. 2019;78:29-40.
[11] Vasan D, Alazab M, Wassan S, Safaei B, Zheng Q. Image-based malware classification using ensemble of CNN architectures (IMCEC). Comput Secur. 2020;92:1-13.
[12] Hospedales TM, Antoniou A, Micaelli P, Storkey A. Meta-learning in neural networks: a survey. arXiv:2004.05439. 2020:1-20.
[13] Shen S, Gu K, Chen XR, Yang M, Wang R. Movements classification of multi-channel sEMG based on CNN and stacking ensemble learning. IEEE Access. 2019;7:137489-500.
[14] Zhang Y, Zhang H, Cai J, Yang B. A weighted voting classifier based on differential evolution. Abstr Appl Anal. 2014;2014: 1-6.
[15] Wang M, Lu S, Zhu D, Lin J, Wang Z. A high-speed and low-complexity architecture for softmax function in deep learning. 2018 IEEE asia pacific conference on circuits and systems (APCCAS); 2018 Oct 26-30; Chengdu, China. New York: IEEE; 2018. p. 223-6.
[16] Dixit A, Mani A, Bansal R. Feature selection for text and image data using differential evolution with SVM and Naive bayes classifiers. Eng J. 2020;24(5):161-72.
[17] Gour M, Jain S. Stacked convolutional neural network for diagnosis of COVID-19 disease from x-ray images. arXiv:2006.13817. 2020:1-26.
[18] Omran MGH, Engelbrecht A, Salman A. Differential evolution methods for unsupervised image classification. 2005 IEEE congress on evolutionary computation; 2005 Sep 2-5; Edinburgh, UK. New York: IEEE; 2005. p. 966-97.
[19] Wang B, Sun Y, Xue B, Zhang M. A hybrid differential evolution approach to designing deep convolutional neural networks for image classification. 31^{st} Australasian joint conference; 2018 Dec 11-14; Wellington, New Zealand. Berlin: Springer; 2018. p. 237-50.
[20] Yazdizadeh A, Patterson Z, Farooq B. Ensemble convolutional neural networks for mode inference in smartphone travel survey. IEEE Trans Intell Transport Syst. 2020;21(6):2232-9.
[21] Vapnik VN. Statistical learning theory. New Jersey: Wiley; 1998.
[22] Storn R, Price K. Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces. J Global Optim. 1997;11(4):341-59.
[23] Penunuri F, Peon-Escalante R, Solis-Perales G, Villanueva C, Acosta C. A differential evolutionary method for solving a class of differential equations numerically. 2012 IFAC conference on analysis and control of chaotic system; 2012 Jun 20-22; Cancun, Mexico. Switzerland: IFAC; 2012. p. 45-50.
[24] Harangi B. Skin lesion classification with ensembles of deep convolutional neural networks. J Biomed Informat. 2018;86:25-32.
[25] Frazao X, Alexandre LA. Weighted convolutional neural network ensemble. In: Bayro-Corrochano E, Hancock E, Editors. Iberoamerican congress on pattern recognition (ICARP); 2014 Nov 2-5; Mexico. Berlin: Springer; 2014. p. 674-81.
[26] Gao B, Pavel L. On the properties of the softmax function with application in game theory and reinforcement learning. arXiv:1704.00805. 2017:1-10.
[27] Deepika S, Geetha TV. A meta-learning framework using representation learning to predict drug-drug interaction. J Biomed Informat. 2018;84:136-47.

[28] Liu X, Wang X, Matwin S. Interpretable deep convolutional neural networks via meta-learning. arXiv:1802.00560. 2018:1-9.
[29] Codreanu V, Droge B, Williams D, Yasar B, Yang P, Liu B, et al. Evaluating automatically parallelized versions of the support vector machine. Concurrency Comput Pract Ex. 2016;28:2274-94.
[30] Wijaya IS, Widiartha I, Arjarwani SE. Pornographic image recognition based on skin probability and eigenporn of skin ROIs images. TELKOMNIKA (Telecomm Comput Electron Contr). 2015;13(3):985-95.
[31] Wijaya IS, Widiartha I, Uchimura K, Koutaki G. Phonographic image recognition using fusion of scale invariant descriptor. 21st Korea-Japan Joint Workshop on Forntiers of Computer Vision (FCV); 2015 Jan 28-30; Mokpo, South Korea. New York: IEEE; 2015. p. 1-5.
[32] Surinta O, Khamket T. Recognizing pornographic images using deep convolutional neural networks. Joint international conference on digital arts media and technology with ECTI northern section conference on electrical electronics computer and telecommunications engineering (ECTI DAMT-NCON); 2019 Jan 30-Feb 2; Nan, Thailand. New York: IEEE; 2019. p. 150-4.
[33] Darwish A. Bio-inspired computing: algorithms review, deep analysis, and the scope of applications. Future Comput Informat J. 2018;3(2):231-46.
[34] Ma T, Liu S, Xiao H. Location of natural gas leakage sources on offshore platform by a multi-robot system using particle swarm optimization algorithm. J Nat Gas Sci Eng. 2020;84(2020):1-15.
[35] Fan X, Sayers W, Zhang S, Han Z, Ren L, Chizari H. Review and classification of bio-inspired algorithms and their applications. J Bionic Eng. 2020;17:611-31.