dropCyclic: Snapshot Ensemble Convolutional Neural Network Based on a New Learning Rate Schedule for Land Use Classification

SANGDAOW NOPPITAK AND OLARIK SURINTA
Multi-agent Intelligent Simulation Laboratory (MISL), Department of Information Technology, Faculty of Informatics, Mahasarakham University, Mahasarakham 44150, Thailand

Corresponding author: Olarik Surinta (e-mail:olarik.s@msu.ac.th).

This work was supported in part by the Mahasarakham University under Grant 6508011/2565.

ABSTRACT The ensemble learning method is a necessary process that provides robustness and is more accurate than the single model. The snapshot ensemble convolutional neural network (CNN) has been successful and widely used in many domains, such as image classification, fault diagnosis, and plant image classification. The advantage of the snapshot ensemble CNN is that it combines the cyclic learning rate schedule in the algorithm to snap the best model in each cycle. In this research, we proposed the dropCyclic learning rate schedule, which is a step decay to decrease the learning rate value in every learning epoch. The dropCyclic can reduce the learning rate and find the new local minimum in the subsequent cycle. We evaluated the snapshot ensemble CNN method based on three learning rate schedules: cyclic cosine annealing, max-min cyclic cosine learning rate scheduler, and dropCyclic then using three backbone CNN architectures: MobileNetV2, VGG16, and VGG19. The snapshot ensemble CNN methods were tested on three aerial image datasets: UCM, AID, and EcoCropsAID. The proposed dropCyclic learning rate schedule outperformed the other learning rate schedules on the UCM dataset and obtained high accuracy on the AID and EcoCropsAID datasets. We also compared the proposed dropCyclic learning rate schedule with other existing methods. The results show that the dropCyclic method achieved higher classification accuracy compared with other existing methods.

INDEX TERMS snapshot ensemble convolutional neural network, ensemble learning, convolutional neural network, learning rate schedule, land use classification, aerial image.

I. INTRODUCTION

Most remote sensing applications have been proposed for scene understanding, including scene classification, scene retrieval, and scene-driven object detection [1], and computed based on 11 bands of satellite imagery. However, many published datasets have been collected from aerial images that include only three bands (red, green, and blue) and proposed for classification, segmentation, and retrieval tasks [2]. In the last decade, traditional methods, including image processing and machine learning, have been studied extensively [3], [4].

The appearance of deep learning algorithms has brought about an interest in by researchers neural networks. Deep learning algorithms have been applied to solve many problems, such as image and video classification, speech recognition, and natural language processing.

Additionally, the remote sensing community has shifted its attention to deep learning algorithms and succeeded in image analysis, land use and land cover evaluation, scene classification, segmentation, object detection, and many other tasks [5]. The deep learning performance depends on the resolution level of remote sensing images. Although, many researchers have proposed methods to classify land use and land cover from the moderate resolution remote sensing images [6].

The theory of the ensemble learning method is to combine outputs of various machine learning and even deep learning models and then decide from many strategies a final prediction resulting in better performance [7]. It uses many different decision strategies to find the final prediction, such as unweighted average, weighted average,
majority voting, Bayesian optimal, and stacked generalization. Kulkarni and Kelkar [8] used ensemble learning methods to classify multispectral satellite images. Three ensemble learning methods were compared (bagging, boosting, and AdaBoosting) and it was found that the ensemble learning method achieved better classification results than the single model. Cao et al. [9] segmented the building areas from the remote sensing images using a stacking ensemble deep learning model. In their method, images were first segmented using three models: FCN-9s, U-Net, and SegNet followed by, optimizing prediction results using a fully connected conditional random field (CRF). Finally, the multilayer features were extracted using a sparse autoencoder. Then, the final prediction results were computed using the Euclidean distance weighting method.

Additionally, several researchers have proposed ensemble learning methods for classifying satellite images. Minetto et al. [10] proposed an ensemble of convolutional neural networks (CNNs) for geospatial land classification. In their method the geospatial images were first sent to CNNs to predict the output. Hence, the predicted outputs from CNNs were determined as the final output using a majority voting method. Diengdoh et al. [11] used weighted and unweighted ensemble learning for land cover classification from the predicted output of various machine learning methods. Huang et al. [12] proposed an ensemble learning method for urban land use mapping tasks based on satellite images, street-view images, building footprints, points-of-interest, and social sensing data to explain the associations of land cover, socioeconomic activities, and land use categories.

Furthermore, new kinds of ensemble learning require expensive computation with no additional training cost while training neural network models, called snapshot ensemble learning [13]. The snapshot ensemble learning method aims to discover several local minimum values in one training. While training the model, we defined the number of cycles that we desired to snap the best model. For example, defining three cycles will return the three best models from each cycle, called snapshot. Further, the best model was snapped at the minimum loss value. Additionally, the learning rate schedule was used to quickly reduce the training loss value using the cyclic cosine annealing function. In addition, Wen et al. [14] proposed a new max-min cosine cyclic learning rate scheduler invented to find the acceptable ranges of maximum and minimum learning rates used in training.

Contribution. In this research, we focus on proposing the new cosine cyclic learning rate schedule by adding a step decay function to reduce the learning rate that directly decreases the training loss to converge local minimum in each cycle, called dropCyclic. For the dropCyclic learning rate schedule, the learning rate starts at the maximum learning rate. Further, the training loss decreases to converge on a local minimum while training in the first cycle. In the next cycle, the new maximum learning rate, which is a smaller value than the previous learning rate, is defined using the dropCyclic method. Consequently, the dropCyclic method narrows the learning rate range from the start until the last cycle. The snapshot ensemble CNN based on the dropCyclic learning rate schedule is proposed for aerial image classification. The proposed method is evaluated on three aerial image datasets: UCM, AID, and EcoCropsAID, and achieved good performance.

Outline of the paper. This paper is organized into five sections, as follows. Surveys of the related works are presented in Section 2. Section 3 presents the snapshot ensemble CNN for aerial image classification and the new learning rate schedule. In Section 4, three aerial image datasets are briefly described. Section 5 presents the experimental results and discussions. The conclusion and future work are presented in Section 6.

II. RELATED WORK
In this section, we briefly explain the research related to the ensemble learning and snapshot ensemble CNNs, including ensemble learning, snapshot ensemble CNN, and learning rate schedules for snapshot ensemble CNN.

A. ENSEMBLE LEARNING
Ensemble learning methods have been a growing research area in recent years. In this study, we surveyed ensemble learning methods with only two strategies: decision and ensemble.

1) THE DECISION STRATEGY
The outputs of other classifiers are combined and classified to the final output with various strategies, such as unweighted average, weighted average, majority vote, Bayes optimal, and stacked generalization [7]. Kim and Lim [15] proposed the ensemble CNNs method to learn on a large vehicle type dataset. The dataset contained more than 500,000 images and had 11 classes. The bagging method was used to randomly select the training data because the image distribution in each class was imbalanced. In the ensemble CNNs, the training images selected using the bagging method were transferred to the three CNNs. While training the CNN, the data augmentation techniques (flip, rotation, AR-fixed, AR-fixed rotation) were applied. The weighted average method was applied for the final prediction and achieved high performance. Minetto et al. [10] used state-of-the-art CNNs (ResNet50 and DenseNet161) and a majority voting method for geospatial land classification on multispectral images. In the first step, 12 CNN models were created using various settings, such as data augmentation, image crop style, and class weighting. The output of this step was the probabilities obtained from 12 CNN models. In the second step, the output probabilities were classified using the majority voting method. However, the correct prediction was accepted when the outputs from the CNNs were correct in more than five models. Their proposed method achieved an accuracy of 94.51% on the FMOW dataset.

Moreover, Diengdoh et al. [11] classified land cover using the ensemble learning method based on satellite imagery.
Their study classified the land cover images into six classes using the unweighted ensemble prediction method. First, four machine learning techniques: K-nearest neighbor (KNN), naive Bayes (NB), random forest (RF), and support vector machine (SVM), were proposed to predict probability outputs. Second, the probability outputs were classified using the unweighted ensemble learning method for the final output. Sefrin et al. [16] used three voting methods: unison vote, absolute majority, and no majority, to detect a land cover change from time-sequence Sentinel-2 images. The main architecture was the combination between the fully convolutional neural network (FCN) and long short-term memory (LSTM), called FCN+LSTM architecture. The time-sequence images were first classified by the FCN+LSTM model and output as six classification maps and the final output classified using the voting method. The results showed that the final predicted class using unison or the absolute majority method achieved high accuracy.

2) THE ENSEMBLE STRATEGY
The ensemble strategy uses the weak learner to create a stronger learner and minimize errors while training. It also has various ensemble strategies. For instance, the bagging ensemble randomly selects subsets of the independent data of the same size. Then, the first, second, and $N$ subsets are trained using the first, second, and $N$ classifiers, respectively. Finally, fusing the output of the base classifiers with the majority voting method for predicting the final output [7], [15]. The boosting strategy, the original data is given to classify using a weak classifier. The original data that was misclassified from the weak classifier is weighted, and called weighted data, due to a decrease in bias obtained while training the weak classifier. Further, the weighted data is sent to the second weak classifier and again weighted to the misclassified data. It could repeat training with a weak classifier many times until it obtained the best weak classifier [8].

The idea of a combination between ensemble strategy (bagging and boosting learning) and the CNN-based method was proposed to short-term load forecasting [17]. In [17], the CNN model was firstly trained on the existing dataset to create the pre-trained CNN model. Then, the fine-tuned model was created by training the pre-trained CNN model from the first phase with the new dataset. Finally, the weak CNN models from phases one and two were constructed to create a robust model. Consequently, the average weighted method was used to compute the prediction result.

Korzh et al. [18] proposed the bagging ensemble and the stacking of CNN to classify remote sensing imagery. In their method, image processing techniques were first applied to the original images to reduce noise and increase sharpness. Then, the set of the original images was sent to CNN models (AlexNet, GoogLeNet, and VGG19) to extract the first feature. Also, the set of processing images was sent to CNN models to extract the second feature. Hence, the first and second features of more than 16,000 features were concatenated before sending to the machine learning technique. In their experiments, many machine learning techniques were compared, including SVM with different kernels (linear, radial basis function, polynomial), random forest, logistic regression. As a result, the SVM with a linear kernel obtained the highest performance on the Brazilian coffee scenes dataset with an accuracy of 96.11%.

B. SNAPSHOŦ T ENSEMBLE CNN
The snapshot ensemble method was first proposed by Huang et al. [13]. The main purpose of the snapshot ensembles was to train CNN one time and obtain more CNN models. Therefore, while training the original CNN model, the model converged the minimum training loss value at the end. Hence, only one CNN model was obtained from the original CNN model. On the other hand, cyclic cosine annealing was used to converge multiple training loss values. The best CNN model in each cycle was used, called snapshot. Consequently, the output probability of each CNN model was calculated using the softmax function. Additionally, the unweighted average method was used as the final prediction. The output probabilities were averaged and the maximum probability was selected. The snapshot ensemble method was evaluated on various image classification datasets and achieved the best performance compared with a single CNN model.

In 2019, Wen et al. [14] proposed a new snapshot ensemble CNN for fault diagnosis. The max-min cosine cyclic learning rate scheduler (MMCCLR) was proposed instead of cyclic cosine annealing. The log-linear learning rate testing (LogLR) method was invented to search the fitting range of the max-min learning rate when encountering new datasets. The MMCCLR method was evaluated on three datasets (bearing dataset of Case Western Reserve University, self-priming centrifugal pump dataset, and bearing dataset) and achieved very high accuracy on three datasets with 99.9%.

Moreover, Babu and Annavarapu [19] modified the snapshot ensemble method to classify COVID-19 from chest X-ray images. For training the CNN model, the pre-trained model ResNet50 was used and trained on the chest X-ray images. The data augmentation techniques (rotation, zoom, flip, and shift) were also applied while training. Subsequently, the weighted average method was used for ensemble learning instead of the unweighted average method. Hence, the weighted parameter was updated until it did not improve accuracy performance. The modified snapshot ensemble method achieved 95.18% accuracy on the COVID-19 XCR dataset and outperformed existing methods. Puangsawan and Surinta [20] used the snapshot ensemble method to classify plant leaf diseases. Three CNN architectures (VGG16, MobileNetV2, InceptionResNetV2, and DenseNet201) were used as the backbone architecture of the snapshot ensemble method. The rotation method was used as the data augmentation technique while training the CNN models. In the snapshot ensemble method, training the DenseNet201 model using four cosine annealing cycles achieved the highest
accuracy of 69.51% on the PlantDoc dataset compared to other ensemble methods (unweighted ensemble and weighted ensemble).

For the aerial images, Dede et al. [21] studied various ensemble strategies (including homogeneous, heterogeneous, and snapshot ensemble) to classify aerial scene images. In their experiments, two pre-trained CNN models were used: Inception and DenseNet. The snapshot ensemble method with Inception as a backbone architecture achieved an accuracy of 96.01% on the RESISC45 dataset. However, the snapshot ensemble method did not attain the best accuracy on the AID dataset. The best algorithm on the AID dataset was the heterogeneous strategy combining Inception and DenseNet and classified using the multi-layer perceptron (MLP). It achieved an accuracy of 97.15% on the AID dataset.

C. THE CYCLICAL LEARNING RATE FOR SNAPSHOT ENSEMBLE CNN
The popular optimization algorithm used while training the CNN model is stochastic gradient descent (SGD). SGD is used to update parameters of the CNN model until it converges to the local minimum value. In the original snapshot ensemble method, the SGD optimizer and the cyclic cosine annealing were computed to quickly decrease the training loss to converge the local minimum [13]. The training loss decreased very fast compared to the original CNN model. Wen et al. [14] proposed a new snapshot ensemble method that used the MMCCLR method to find the range of learning rates. Petrovska et al. [22] used an adaptive learning rate schedule with a triangular policy to train the snapshot ensemble method. Furthermore, Hung et al. [23] proposed a two-stage cyclical learning rate method using triangular methods. The triangular and triangular2 methods were used in the first and second states to find the best stable model and required few iterations while training.

III. PROPOSED SNAPSHOT ENSEMBLE CNN FOR AERIAL IMAGE CLASSIFICATION
The snapshot ensemble CNN was first proposed by Huang et al. [13] with the simple concept of finding many local minima values and then snapping the best CNN model at the local minima in each cycle. Subsequently, the outputs of CNN models were combined and computed using the ensemble method. The essence of the snapshot ensemble CNN method is the cyclic learning rate schedule, which is the cyclic cosine annealing (CCA) schedule. The CCA schedule allows the learning rate to decrease quickly, stimulating the CNN model to reach local minimum after a few epochs. The snapshot ensemble CNN produces lower errors than a single CNN model. This section briefly describes 1) The cosine cyclic learning rate schedule, including cyclic cosine annealing, max-min cosine cyclic, and the proposed dropCyclic. 2) The snapshot ensemble method.

A. COSINE CYCLIC LEARNING RATE SCHEDULE
1) CYCLIC COSINE ANNEALING
The cyclic cosine annealing (CCA) is the primary learning rate schedule of the snapshot ensemble CNN method used while training the CNN model. CCA allows the CNN to lower the learning rate faster than the traditional CNN model and converge to diverse local minimums [13]. The CCA curve training with 100 epochs and using five cycles (M), when $M_1, M_2, ..., M_5$ denote the CNN model for each local minimum, as shown in Figure 1. The CCA is computed as Equation (1).

$$\eta = \frac{\eta_{\text{init}}}{2} \left( \cos \left( \frac{\pi \mod (t-1)[T/M]}{[T/M]} \right) + 1 \right)$$

(1)

where $\eta$ stands for the learning rate of current iteration, $\eta_{\text{init}}$ is the initial learning rate, $t$ is for the current iteration number, $T$ is the total iterations, and $M$ is the number of cycles.

In the CCA, only the initial learning rate ($\eta_{\text{init}}$) is required to be adjusted. As a result, the wrong learning rate will cause the training process to not converge with the local minimum at the end of each cycle.

2) MAX-MIN COSINE CYCLIC LEARNING RATE SCHEDULER
Wen et al. [14] proposed the max-min cosine cyclic learning rate scheduler (MMCCLR). The upper and lower boundaries of the learning rate were proposed to adjust the boundary of the learning rate. However, in the MMCCLR, the log linear learning rate test (LogLR Test) method was proposed to find the learning rate range, which is the max and min learning rates. The LogLR Test method and the MMCCLR are calculated as Equation (2) and Equation (3).

$$log_{10}\eta = log_{10}\eta_{\text{min}} + (log_{10}\eta_{\text{max}} - log_{10}\eta_{\text{min}}) \times \frac{[L_T]}{[T_M]}$$

(2)

$$\eta = \eta_{\text{min}} + \frac{\eta_{\text{max}} - \eta_{\text{min}}}{2} \times (1 + \cos \left( \frac{\pi \mod (t-1)[T/M]}{[T/M]} \right))$$

(3)
where $\eta_{\text{min}}$ is the minimum learning rate and $\eta_{\text{max}}$ is the maximum learning rate that is tested using Equation (2).

3) PROPOSED DROP CYCLIC COSINE LEARNING RATE SCHEDULE

In this study, we proposed a drop cyclic cosine learning rate schedule, called dropCyclic. The dropCyclic is the systematic reduction of the learning rate over a specific time during training. This research aims to decrease the learning rate that cuts by a constant factor called the drop parameter, every constant number of epochs (see Figure 2), as same as the step decay schedule [24]. The dropCyclic is also efficient in discovering the diversity of local minimums in each cycle using the c parameter.

Moreover, in dropCyclic, the maximum learning rate in each cycle is changed according to the drop parameter. While the learning rate range is limited, the CNN model can faster converge to the local minimum. The equation of the dropCyclic is computed as Equation (4).

$$\eta = \eta_{\text{init}} \times \text{drop}^{\text{floor}\left(\frac{(1+t)/c}{(1+\cos(\pi \text{mod}(t, b)/Mb))}\right)}$$  \hspace{1cm} (4)

where $\eta_{\text{init}}$ is the initial learning rate, drop is the step decay parameter that drops the learning rate in every $n$ epoch, $c$ is a constant number that lets the model change to the new local minimum in the next cycle.

B. SNAPSHOT ENSEMBLE METHODS

The ensemble method is the final step of the snapshot ensemble CNN to enhance accuracy based on diverse CNN models from single training. We trained the CNN model using the proposed dropCyclic method and snapped the best CNN model from each cycle in the previous step. The output probabilities of each CNN computed using the softmax function were combined and classified using the unweighted average ensemble method. Indeed, the last $N$ models are significant to have the lowest error. We then normally ensemble the last $N$ models. The ensemble method is calculated as Equation (5).

$$\hat{y} = \frac{1}{M} \sum_{j=1}^{M} f_j$$  \hspace{1cm} (5)

where $M$ is the number of CNN models and $f_j$ is the output probabilities of CNN model $j$ that is computed using the softmax function.

IV. AERIAL IMAGE DATASETS

A. UC MERCED LAND USE (UCM) DATASET

Yang and Newsam [25] first proposed the UCM dataset for land use classification tasks collected from the USGS national map urban area imagery. The aerial images were extracted from large images and divided into 21 classes, such as agricultural, forest, golf course, beach, harbor, buildings, medium residual, sparse residential, and dense residential. It is stored in the RGB color space image with 256x256x3 pixels. The UCM dataset contains 2,100 images and some examples of the UCM dataset are shown in Figure 3. However, the challenge of the UCM dataset is that classes of medium residual, sparse residential, and dense residential, are similar and difficult to classify, as shown in Figure 4.

B. AERIAL IMAGE DATASET (AID)

The AID [26] was proposed for aerial scene classification tasks. It has 10,000 images and contains 30 different aerial scene classes, for example, dense residential, medium residential, sparse residential, stadium, industrial, bridge, and baseball field. Each class has approximately 200 to 400 images of 600x600 pixels. The AID was collected from the Google Earth application at resolution of 8 to about 0.5 meters. Examples of the AID are shown in Figure 5.
FIGURE 5. Example aerial images of the AID dataset: a) airport, b) bridge, c) desert, d) pond, and e) storage tanks.

C. ECOCROPSAID DATASET

Thailand’s economic crops aerial image dataset (EcoCropsAID) was proposed by Noppitak and Surinta [27] for land use classification. The EcoCropsAID dataset was collected according to the information on the cultivation of economic crops in different regions between 2014 and 2018 obtained from Agri-Map Online. The economic crops aerial images were collected from the Google Earth application at resolution of 30 to 0.2 meters. The images were stored in the RGB format with 600x600 pixels. It has 5,400 aerial images of five classes: rice, sugarcane, cassava, rubber, and longan. Example images of the EcoCropAID dataset are shown in Figure 6. The challenges of the EcoCropsAID dataset are that the pattern of each class is quite similar (see Figure 7), and various patterns occur in the same class (see Figure 8).

FIGURE 6. Examples of the EcoCropsAID dataset: a) cassava, b) sugarcane c) longan d) rubber, and e) rice.

FIGURE 7. Illustration of the similarity patterns between two classes: (a) longan and (b) rubber and (c) cassava and (d) rice, of the EcoCropsAID dataset.

FIGURE 8. Illustration of the diversity in the patterns of the EcoCropsAID dataset. Examples of (a) longan and (b) rubber images.

The details of the three aerial image datasets used in our experiments are summarized in Table I.

TABLE I

| Aerial Image Datasets | Classes | Image Per Class | Total Images | Image Size (pixels) | Resolution (meters) | Year | Training Images | Test Images |
|-----------------------|---------|----------------|--------------|---------------------|---------------------|------|-----------------|-------------|
| UCM [25]              | 21      | 100            | 2,100        | 256 x 256           | 0.3                 | 201  | 1,480           | 420         |
| AID [26]              | 30      | 2 to 420       | 10,000       | 300 x 300           | 8 to 0.5            | 201  | 5,000           | 5,000       |
| EcoCropsAID [27]      | 5       | 100 to 1,000   | 5,400        | 600 x 600           | 30 to 0.2           | 202  | 4,320           | 1,080       |

V. EXPERIMENTAL RESULTS AND DISCUSSION

We demonstrated the effectiveness of the proposed cyclic learning rate (dropCyclic) and compared it with two existing cosine cyclic learning rate methods: cyclic cosine annealing (CCA) and the max-min cyclic cosine learning rate scheduler (MMCCCLR) on three aerial image datasets. For the backbone
of the snapshot ensemble, we compared three CNN architectures: MobileNetV2, VGG16, and VGG19.

All the experiments were trained and evaluated on a Linux operating system using Intel(R) Core-i9-9900K CPU @ 3.60GHz x 16, RAM 32GB, and GPU GeForce GTX 1080Ti with RAM 11GB GDDR5x. We implemented all snapshot ensemble methods based on the TensorFlow deep learning framework with the Keras library.

A. EVALUATION METRICS
In this experiment, we used K-fold cross-validation (cv) with K=5 over the training set to prevent overfitting problems. Hence, the overall accuracy (%) and standard deviation evaluated the training set. Further, test accuracy was used to evaluate the classification performance, and the results were compared with existing snapshot ensemble methods. The accuracy performance was computed as shown in Equation (6).

\[
ACC = \frac{TP + TN}{TP + FN + FP + TN} \times 100
\]  

where \(TP\) is a true positive (indicates the positive samples that are correctly classified), \(TN\) is a true negative (indicates the negative samples that are correctly classified), \(FP\) is a false positive (indicates the negative samples that are misclassified), and \(FN\) is a false negative (indicates the positive samples that are misclassified).

In order to prevent overfitting problems and compare different learning rate methods, we used the loss difference (LD) metric to evaluate snapshot ensemble methods when the difference learning rate policy was performed. The LD is the evaluation metric that indicates the robustness of the model against the overfitting problems [28]. Overfitting problems appear when the low loss value is on the training set, but the high loss value is on the test set. Hence, it results in low accuracy. The smallest LD value shows the robustness of the model, which is computed as Equation (7).

\[
LD = valid_{loss} - training_{loss}
\]  

where \(training_{loss}\) and \(valid_{loss}\) are the loss values obtained while training on the training set and validation set, respectively.

B. TRAINING SETTING
1) DATA RATIO AND NUMBER OF EXPERIMENTS
We reported the data ratio and the number of experiments of each dataset. We divided the dataset into training, validation, and test sets. We split with the ratio of 4:1:5 for the AID dataset and 7:1:2 for the UCM and EcocropsAID datasets. Due to the randomness of the training and validation sets, we computed experiments three times and reported the mean accuracy and standard deviation on the validation set. Further, we trained the model again on the training and validation sets with the best setting and evaluated the test set.

2) BACKBONE CNN ARCHITECTURES
In our previous study [27], several CNN architectures, including InceptionResNetV2, DenseNet201, Xception, ResNet52V2, NASNetLarge, MobileNetV2, VGG16, and VGG19 were experimented with. We found that the VGG16 and VGG19 achieved the highest accuracy. Subsequently, MobileNetV2 showed worse accuracy compared to VGG16 and VGG19. Hence, in this study, we mainly experimented with the snapshot ensemble CNN using three state-of-the-art architectures as a backbone CNN: VGG16, VGG19 [29], and MobileNetV2 [30]; to prove that the snapshot ensemble could manage both the best and the worst CNN architectures and also enhance the classification performance on the land use images.
3) SNAPSHOT ENSEMBLE METHODS
We compared the proposed drop cyclic cosine learning rate schedule (dropCyclic) with two existing learning rate schedules: CCA and MMCCLR. We trained the snapshot ensemble with 100 epochs and the snapshot parameter with $M=5$ cycles.

4) DROPYCYCLIC LEARNING RATE SCHEDULE METHOD
As shown in Figure 9, we illustrated the learning rate curve of the dropCyclic learning rate schedule. The learning rate was computed using Equation 4.

The small $c$ value made the model capable of escaping the local minimum to find a new local minimum, as shown in Figure 9(a). With the large $c$ value, the model could have the energy to discover a new local minimum, as shown in Figures 9(b) and 9(c). In the case of $\text{drop} = 1.0$, at the first cycle, the maximum learning rate was 0.0010 and the learning rate decreased in each epoch until zero. In the second cycle, the learning rate started again at 0.0010. Further, in the case of $\text{drop} = 0.95$, the maximum learning rate in the next cycle was slightly dropped until the last cycle according to the $\text{drop}$ parameter. In our dropCyclic experiments, the $\text{drop}$ and $c$ values were 0.95 and 10, respectively.

The details of the hyperparameter settings on CNN architectures and the learning rate schedule are summarized in Table II.

| Parameter Settings | Learning Rate Schedules |
|--------------------|-------------------------|
| CNN Parameters     | CCA | MMCCLR | dropCyclic |
| Image normalization | 0-1 | 0-1    | 0-1        |
| Mini batch size    | 16  | 16     | 16         |
| Data augmentation  | No  | No     | No         |
| Optimizer          | SGD | SGD    | SGD        |
| Momentum           | 0.9 | 0.9    | 0.9        |
| Learning rate      | 0.00| Min=0  | Max=0.001  |

| Snapshot Ensemble  | Cycle ($M$) | 5 | 5 | 5 |
|--------------------|-------------|---|---|---|
| $\text{drop}$      | 0.95        |   |   |   |
| $c$                | 10          |   |   |   |

C. CLASSIFICATION RESULTS ON THE UCM DATASET
We first observed the optimal learning rate values to confirm that the proposed dropCyclic learning rate schedule performs well when using the optimal learning rate. In this experiment, we trained the snapshot ensemble learning using the MobileNetV2 as a backbone CNN and the proposed dropCyclic learning rate schedule on the UCM dataset. In this study, we illustrated the accuracies of various learning rates, including 0.1, 0.01, 0.001, and 0.0001. The validation accuracies are shown in Figure 10.

![Figure 10](image)

The scatter plot Figure 10, show that the validation accuracies closed to approximately 98% were achieved when the learning rate value reached zero. However, we observed that the validation dropped when the learning rate was high. In this study, the validation accuracy dropped from approximately 80% to 40% when the learning rates were in the range of 0.0008 to 0.0010.

![Figure 11](image)

Figure 11 presents the test error (%) of snapshot ensemble CNN with different learning rate schedule methods: CCA (first column), MMCCLR (second column), dropCyclic (third column) and CNN architectures: (a) MobileNetV2 (b) VGG16, (c) VGG19, on the UCM dataset. Each snapshot ensemble CNN was trained with $M=5$ cycles.
two, three, four, and five models, respectively, were reported. As seen in Figure 11(a) in the third column, the snapshot ensemble CNN using MobileNetV2 with dropCyclic learning rate schedule obtained the lowest test error when ensemble with only three models. In comparison, CCA methods achieved the lowest test error when ensemble with five models (see Figure 11(a) in the first column).

Furthermore, when training the snapshot ensemble CNN using VGG19 with MMCCLR and dropCyclic learning rate schedule methods, as shown in Figure 11(c) in the second and third columns, the lowest error was achieved by using only one model. Therefore, the VGG19 discovered the optimal local minimum at the first cycle according to the number of aerial images in the UCM dataset with only 2,100 images. The results are presented in Table III.

**TABLE III**

**CLASSIFICATION PERFORMANCES (LD, MEAN VALIDATION ACCURACY, STANDARD DEVIATION AND TEST ACCURACY) OF THE SNAPSHOT ENSEMBLE CNN USING DIFFERENT LEARNING RATE SCHEDULES: CCA, MMCCLR, AND DROP CYCLIC AND TRAINING WITH DIFFERENT STATE-OF-THE-ART CNNS: MOBILENETV2, VGG16, VGG19, ON THE UCM DATASET.**

| CNNs         | LR       | LD         | Validation | Test        |
|--------------|----------|------------|------------|-------------|
| MobileNetV2  | CCA      | 0.0422     | 97.30±0.00085 | 97.14       |
|              | MMCCLR   | 0.0626     | 97.10±0.0081 | 96.43       |
| VGG16        | dropCyclic | 0.0560     | 97.38±0.0050 | 97.38       |
|              | CCA      | 0.1650     | 95.47±0.0050 | 93.57       |
|              | MMCCLR   | 0.1371     | 95.83±0.0042 | 93.10       |
| VGG19        | dropCyclic | 0.1965     | 96.51±0.0054 | 92.62       |
|              | CCA      | 0.2285     | 96.98±0.0097 | 93.10       |
|              | MMCCLR   | 0.1475     | 96.03±0.0056 | 93.57       |
|              | dropCyclic | 0.0896     | 96.63±0.0020 | 94.76       |

We evaluated the snapshot ensemble CNN using three evaluation metrics: LD, validation (mean accuracy and standard deviation), and test accuracy. Note that the LD value presented the best method for preventing the overfitting problem. On examining Table III, we discovered that all learning rate schedule methods: CCA, MMCCLR, and proposed dropCyclic, can address the problem with overfitting because all learning rate schedule methods achieved low LD values. Consequently, the accuracies of the validation and test did not show an enormous difference value. As a result, the proposed dropCyclic learning rate schedule outperformed the existing learning rate schedules: CCA and MMCCLR when training the CNN model with MobileNetV2 and VGG19 in terms of test accuracy. The proposed dropCyclic method also outperformed when training with MobileNetV2 and VGG16 on the validation set. In conclusion, the snapshot ensemble CNN using the MobileNetV2 as a backbone CNN and the proposed dropCyclic learning rate schedule (\(d = 0.95\) and \(c = 10\)) achieved the highest test accuracy of 97.38% on the UCM dataset.

We illustrated the confusion matrix to show that the snapshot ensemble CNN method can be proposed to learn from many aerial image patterns and even similar patterns between two or more classes, such as classes of residential, including spare, medium, and dense. The medium residential was misclassified as sparse residential (2 misclassified images) and dense residential (one misclassified image), as shown in Figure 12.

**FIGURE 12.** The confusion matrix of residential classes: sparse, medium, dense that has a similar pattern.

We compared the snapshot ensemble CNN using the proposed dropCyclic learning rate schedule with existing methods. The experimental results in Table IV show that our method achieved 97.38% on the UCM dataset and outperformed other methods, except only the IRELB+SDSAE method that slightly obtained better accuracy with 97.61%.

**D. CLASSIFICATION RESULTS ON THE AID DATASET**

We experimented on a snapshot ensemble CNN using three CNNs: MobileNetV2, VGG16, and VGG19 and three learning rate schedules: CCA, MMCCLR, and dropCyclic. The test errors of each experiment are illustrated in Figure 13. The graphs show that combining more models obtained better performance than using a single model. We obtained the lowest test error when using MobileNetV2 as a backbone CNN, as shown in Figure 13(a). Furthermore, using the learning rate schedule with the CCA method outperformed other learning rate schedules on both validation and test sets. The overall performance is shown in Table V.

**TABLE IV**

**THE PERFORMANCE COMPARISON OF THE SNAPSHOT ENSEMBLE CNN USING THE PROPOSED DROP CYCLIC LEARNING RATE SCHEDULE WITH EXISTING TECHNIQUES ON THE UCM DATASET.**

| Methods         | Accuracy | References |
|-----------------|----------|------------|
| GoogLeNet       | 94.31±0.89 | Xia et al. [26] |
| CaffeNet        | 95.02±0.81 | Xia et al. [26] |
| VGG-VD-16       | 95.21±1.20 | Xia et al. [26] |
| AlexNet-SPP-SS  | 96.67±0.94 | Han et al. [31] |
| GC-DSDM         | 97.05     | Yuan et al. [32] |
| IRELB+SDSAE     | 97.61±0.36 | Zhao et al. [33] |
| dropCyclic      | 97.38     | Our Proposed |

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/
FIGURE 13. Loss error (%) of snapshot ensemble CNN with different learning rate schedule methods: CCA (first column), MMCCLR (second column), dropCyclic (third column) and CNN architectures: (a) MobileNetV2 (b) VGG16, (c) VGG19, on the AID dataset. Each snapshot ensemble CNN was trained with M=5 cycles.

TABLE V

Classification performances of the snapshot ensemble CNN using different learning rate schedules and training with different state-of-the-art CNNs on the AID dataset.

| CNNs   | LR methods | LD       | Validation | Test   |
|--------|-------------|----------|------------|--------|
| MobileNetV2 | CCA | 0.1498 | 93.84±0.0121 | 94.86 |
|         | MMCCLR     | 0.2018 | 93.28±0.0036 | 94.70 |
|         | dropCyclic | 0.2052 | 93.57±0.0065 | 94.58 |
| VGG16   | CCA         | 0.3217 | 91.61±0.0084 | 93.60 |
|         | MMCCLR     | 0.3070 | 91.27±0.0052 | 93.02 |
|         | dropCyclic | 0.2189 | 91.79±0.0072 | 93.38 |
| VGG19   | CCA         | 0.3695 | 92.25±0.0022 | 93.32 |
|         | MMCCLR     | 0.3243 | 91.47±0.0051 | 92.24 |
|         | dropCyclic | 0.3208 | 91.93±0.0070 | 93.54 |

Table V shows the performance of the snapshot ensemble CNN methods on the AID dataset, which is an unbalanced dataset because each class has between 220 to 420 aerial images (see Figure 14). We used the LD value to measure the overfitting problems that can be found when training the CNN model. We found LD values between 0.1-0.3 with all experiments. Hence, the test accuracies of all experiments were higher than the validation accuracies. The snapshot ensemble CNN can address the problem with overfitting problems. As a result, using MobileNetV2 and the CCA learning rate schedule achieved 94.86% accuracy and outperformed other methods on the AID dataset.

FIGURE 14. Class distribution of the AID dataset.

Three models of snapshot ensemble CNN with various learning rate schedules were selected for receiver operating characteristic (ROC) comparison, as shown in Figure 15. The snapshot ensemble CNN using MobileNetV2 and CCA learning rate schedule attained an AUC value of 0.9982. Using VGG16 and the CCA learning rate schedule and VGG19 and the dropCyclic learning rate schedule achieved AUC values of 0.9982 and 0.9981, respectively.

Consequently, the dropCyclic learning rate schedule outperformed other learning rate schedules when training with the VGG19. Moreover, we concluded that the snapshot ensemble CNN using MobileNetV2 could address the unbalanced data better than VGG16 and VGG19 architectures.

FIGURE 15. Illustration of the ROC curve for snapshot ensemble CNN models. The highlighted area is zoomed in at the upper left area of the curve.

TABLE VI

The performance comparison of the snapshot ensemble CNN using the proposed dropCyclic learning rate schedule with existing techniques on the AID dataset.

| Methods       | Accuracy     | References   |
|---------------|--------------|--------------|
| GoogLeNet     | 86.39±0.55   | Xia et al. [26] |
| CaffeNet      | 89.53±0.31   | Xia et al. [26] |
| VGG-VD-16     | 89.64±0.36   | Xia et al. [26] |
| IRELBP+SDSA   | 94.11±0.55   | Zhao et al. [33] |
| dropCyclic    | 94.58        | Our Proposed  |
| BiMobileNet   | 96.87±0.23   | Yu et al. [34] |

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/
Table VI compares the performance of the proposed dropCyclic method with other models for aerial image classification. The proposed method outperformed other methods, except for the BiMobileNet method. The proposed method achieved an accuracy of 94.58%, while the BiMobileNet method obtained an accuracy of 96.87% on the AID dataset. The snapshot ensemble CNN method based on the proposed dropCyclic method did not solve the unbalanced dataset problem.

E. CLASSIFICATION RESULTS ON THE ECOCROPSAID DATASET

This experiment discovered that combining only one or two models could achieve the lowest test error, as shown in Figure 16. Because the EcoCropsAID dataset has only five classes and each class contains ~1,000 aerial images, the CNN models can cope well with many patterns in each class and are well classified. The overall performance is shown in Table VII.

Table VII compares the performance of the snapshot ensemble CNN methods on the EcoCropsAID dataset. All the snapshot ensemble CNN models were trained using five cycles. We found that all the experiments prevented the overfitting problems with the small LD values between approximately 0.006 - 0.01. In this experiment, training the VGG16 using the dropCyclic and the CCA learning rate schedules achieved the highest validation accuracy of 99.60%. However, the MMCCCLR learning rate schedule slightly outperformed with an accuracy of 99.26%. Figure 17 shows the confusion matrix of the snapshot ensemble CNN based on dropCyclic and MobileNetV2 on the EcoCropsAID dataset and illustrates only a few misclassifications.

In addition, we compared the snapshot ensemble CNN based dropCyclic learning rate schedule and MobileNetV2 with the existing method [27]. The result showed that our proposed method achieved an accuracy that was 6% higher than the existing method, which achieved only 92.80%.

F. DISCUSSION

1) LOSS ERROR CURVE OF THE DIFFERENT LEARNING RATE SCHEDULES

We discovered that the learning rate parameter directly affects the accuracy of the CNN model. Hence, we have seen existing research focused on tuning the learning rate parameter [28], [35], [36]. However, the learning rate parameter was not the primary priority in our experiment because the dropCyclic method was proposed to change the maximum of the learning rate in each cycle. Further, the maximum learning rate was decreased in each cycle according to the drop parameter, as presented in Figure 9.
schedule outperformed other learning rate schedules on the UCM dataset. Further, the proposed dropCyclic learning rate schedule achieved high accuracy on the AID and EcoCropsAID aerial image datasets. In conclusion, the proposed dropCyclic learning rate schedule has the advantage that it restricts the maximum learning rate in each cycle by using step decay parameters: drop and c. Hence, the maximum learning rate is not the priority parameter required to adjust.

VI. CONCLUSION AND FUTURE WORK

This research proposed a new learning rate schedule called the dropCyclic. We developed the concept of the step decay schedule that decreases half of the learning rate value in every c epoch, call drop. The drop parameter was contained in the cosine cyclic learning rate schedule. It contained two parameters, the drop and c parameters. The benefit of the dropCyclic learning rate schedule is that the learning rate was dropped in the next cycle according to the drop parameter. The method allows the convolutional neural network (CNN) model to discover the new local minimum in the subsequent cycle using the c parameter. We evaluated the proposed dropCyclic learning rate schedules and the existing methods: cyclic cosine annealing (CCA) and max-min cyclic cosine learning rate scheduler (MMCCLR) on three aerial image datasets, including UCM, AID, and EcoCropsAID datasets. Three CNN architectures were compared for the backbone CNN architectures, consisting of MobileNetV2, VGG16, and VGG19. The proposed dropCyclic learning rate schedule achieved the best results on the UCM dataset. The dropCyclic method obtained very high results on the AID and EcoCropsAID datasets. In comparison with other methods, the proposed dropCyclic learning rate schedule outperformed all methods on the AID and EcoCropsAID datasets, except on the UCM dataset for which the IRELBp+SDSAE slightly outperformed the dropCyclic method.

In future work, we will continue to concentrate on the learning rate schedule, such as adaptive learning rate [37], cyclical learning rate with triangular, triangular2, and exp_range policies [23], [38]. Second, we will consider extracting the spatial and temporal features instead of extracting them using only CNN architectures [38]–[40]. Finally, the unbalanced data is also a big challenge to be addressed and enhance the classification performance [41]–[43].

ACKNOWLEDGMENT

This research project was financially supported by Mahasarakham University (Grant No. 6508011/2565).

REFERENCES

[1] Y. Li, Y. Zhang, and Z. Zhu, “Error-tolerant deep learning for remote sensing image scene classification,” IEEE Transactions on Cybernetics, vol. 51, no. 4, pp. 1756–1768, Apr. 2021, doi: 10.1109/TCYB.2020.2989241.

[2] Y. Gu, Y. Wang, and Y. Li, “A survey on deep learning-driven remote sensing image scene understanding: Scene classification, scene
One neural network for fault diagnosis, L2020, doi: 10.1007/s10499-021-1994-4.

P. S. A. Babu and C. S. R. Annanarapu, “Deep learning-based improved snapshot ensemble technique for COVID-19 chest X-ray classification,” Applied Intelligence, vol. 51, no. 5, pp. 3104–3120, May 2021, doi: 10.1007/s10499-021-02199-4.

T. Pangsawan and O. Suriint, “Enhancement of plant leaf disease classification based on snapshot ensemble convolutional neural network,” ICIC Express Letters, vol. 15, no. 6, 2021, doi: 10.24507/icicel.15.06.669.

M. A. Dedde, E. Aptoula, and Y. Genc, “Deep network ensembles for ariplane scene classification,” IEEE Geoscience and Remote Sensing Letters, vol. 16, no. 5, pp. 732–735, May 2018, doi: 10.1109/LGRS.2018.2880136.

B. Petrovska, E. Zdravevski, P. Lameski, R. Corizzo, I. Stojdahar, and J. Petrovska, “Deep learning for feature extraction in remote sensing: A case-study of aerial scene classification,” Sensors, vol. 20, no. 14, 3906, Jul. 2020, doi: 10.3390/s20143906.

S.-C. Hung, H.-C. Wu, and M.-H. Tseng, “Remote sensing scene classification and explanation using RSSCNNet and LIME,” Applied Sciences, vol. 10, no. 18, 6151, Sep. 2020, doi: 10.3390/app10186151.

R. Ge, S. M. Kakade, R. Kidambi, and P. Netrapalli, “The step decay schedule: A near optimal, geometrically decaying learning rate procedure for least squares,” in Proc. 33rd Neural Information Processing Systems (NeurIPS), Vancouver, Canada, 2019, pp. 32 pp., 14977–14988.

Y. Yang and S. Newsam, “Bag-of-visual-words and spatial extensions for land-use classification,” in Proc. 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems (ACM SIGSPATIAL GIS), San Jose, California, USA, 2010, pp. 270–279, doi: 10.1145/1869790.1869829.

G.-S. Xia et al., “AID: A benchmark data set for performance evaluation of aerial scene classification,” IEEE Transactions on Geoscience and Remote Sensing, vol. 55, no. 7, pp. 3965–3981, Jul. 2017, doi: 10.1109/TGRS.2017.2685945.

S. Noppitak and O. Suriint, “Ensemble convolutional neural network architectures for land use classification in economic crops aerial images,” ICIC Express Letters, vol. 15, no. 6, pp. 531–543, 2021, doi: 10.24507/icicel.15.06.531.

Y. Wu et al., “Demystifying learning rate policies for high accuracy training of deep neural networks,” in Proc. IEEE International Conference on Big Data, Los Angeles, California, USA, 2019, pp. 1971–1980, doi: 10.1109/BigData47090.2019.9006104.

K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” in Proc. 3rd International Conference on Learning Representations (ICLR), San Diego, California, USA, 2014, pp. 1–14, Accessed: Mar. 07, 2022. [Online]. Available: https://arxiv.org/pdf/1409.1556.pdf.

M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, “MobileNetV2: Inverted residuals and linear bottlenecks,” in Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition, (CVPR), Salt Lake City, Utah, USA, Jun. 2018, pp. 4510–4520, doi: 10.1109/CVPR.2018.00474.

X. Han, Y. Zhong, L. Cao, and L. Zhang, “Pre-trained AlexNet architecture with pyramid pooling and supervision for high spatial resolution remote sensing image scene classification,” Remote Sensing, vol. 9, no. 8, p. 848, Aug. 2017, doi: 10.3390/rs9080848.

L.-X. Yuan, S.-J. Li, and Y.-P. Jiang, “Remote sensing scene classification using a preclassification strategy and an improved structural feature,” IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 10, no. 9, pp. 4904–4910, 2017, doi: 10.1109/JSTARS.2017.2707519.

F. Zhao, X. Mu, Z. Yi, and Z. Yang, “High-resolution remote sensing scene classification using improved LBP and SDSAE,” in Proc. 2nd International Conference on Algorithms, Computing and Systems - ICACS ’18, 2018, pp. 155–159, doi: 10.1145/3242840.3242868.

D. Yu, Q. Xu, H. Guo, C. Zhao, Y. Lin, and D. Li, “An efficient and lightweight convolutional neural network for remote sensing image scene classification,” Sensors, vol. 20, no. 7, p. 1999, Apr. 2020, doi: 10.3390/s20071999.

X. Wu, R. Ward, and L. Bottou, “WNGrad: Learn the learning rate in gradient descent,” arXiv, pp. 1–16, Mar. 2018, Accessed: Mar. 11, 2022. [Online]. Available: https://arxiv.org/pdf/1803.02865.pdf.
[36] M. Xue, J. Li, and Q. Luo, “Toward optimal learning rate schedule in scene classification network,” IEEE Geoscience and Remote Sensing Letters, vol. 19, pp. 1–5, 2022, doi: 10.1109/LGRS.2020.3040359.

[37] J. Yang and F. Wang, “Auto-ensemble: An adaptive learning rate scheduling based deep learning model ensembling,” IEEE Access, vol. 8, pp. 217499–217509, 2020, doi: 10.1109/ACCESS.2020.3041525.

[38] B. Petrovska, T. Atanasova-Pacemska, R. Corizzo, P. Mignone, P. Lameski, and E. Zdravevski, “Aerial scene classification through fine-tuning with adaptive learning rates and label smoothing,” Applied Sciences, vol. 10, no. 17, p. 5792, Aug. 2020, doi: 10.3390/app10175792.

[39] S. Chaih, H. Yao, Y. Gu, and M. Amrani, “Deep feature extraction and combination for remote sensing image classification based on pre-trained CNN models,” in Proc. 9th International Conference on Digital Image Processing (ICDIP), Hong Kong, China, Jul. 2017, doi: 10.1117/12.2281755.

[40] S. Phiphitphaisit and O. Surinta, “Deep feature extraction technique based on Conv1D and LSTM network for food image recognition,” Engineering and Applied Science Research, vol. 48, no. 5, pp. 581–592, 2021, doi: 10.14456/easr.2021.60.

[41] L. Jiao and J. Zhao, “A survey on the new generation of deep learning in image processing,” IEEE Access, vol. 7, pp. 172231–172263, 2019, doi: 10.1109/ACCESS.2019.2956508.

[42] A. Özdemir, K. Polat, and A. Alhudaif, “Classification of imbalanced hyperspectral images using SMOTE-based deep learning methods,” Expert Systems with Applications, vol. 178, p. 114986, Sep. 2021, doi: 10.1016/j.eswa.2021.114986.

[43] G. Sambusivam and G. D. Opiyo, “A predictive machine learning application in agriculture: Cassava disease detection and classification with imbalanced dataset using convolutional neural networks,” Egyptian Informatics Journal, vol. 22, no. 1, pp. 27–34, Mar. 2021, doi: 10.1016/j.eij.2020.02.007.

SANGDAOW NOPPITAK received the B.S. degree in Computer Science from Khon Kaen University, Khon Kaen, Thailand, in 2004 and the M.S. degree in Remote Sensing and Geographic Information System from Khon Kaen University, Khon Kaen, Thailand, in 2010. From 2007 to 2010, she was a research assistant with the Geo-Informatics Centre for Development of Northeast Thailand, Khon Kaen University, Khon Kaen, Thailand. She is currently pursuing a Ph.D. degree in the Department of Information Technology, Faculty of Informatics, Mahasarakham University, Mahasarakham, Thailand. Her research interest includes land use and land cover classification and deep learning.

OLARIK SURINTA received a Ph.D. in Artificial Intelligence from the University of Groningen, The Netherlands, in 2016. He works at the Department of Information Technology in the Faculty of Informatics, Mahasarakham University, Thailand and is a research member of the Multi-agent Intelligent Simulation Laboratory (MISL). His research interests include historical document analysis and recognition, deep learning, machine learning, and image and video classifications.