Acoustic Classification of Mosquitoes using Convolutional Neural Networks Combined with Activity Circadian Rhythm Information

Jaehoon Kim[1], Jeongkyu Oh[2], Tae-Young Heo[1]*

[1] Department of Information & Statistics, Chungbuk National University, Chungbuk (Republic of Korea)
[2] Data Scientist Team, BEGAS Inc., Seoul (Republic of Korea)

Received 1 June 2020 | Accepted 19 May 2021 | Published 9 August 2021

unir
LA UNIVERSIDAD EN INTERNET

Abstract

Many researchers have used sound sensors to record audio data from insects, and used these data as inputs of machine learning algorithms to classify insect species. In image classification, the convolutional neural network (CNN), a well-known deep learning algorithm, achieves better performance than any other machine learning algorithm. This performance is affected by the characteristics of the convolution filter (ConvFilter) learned inside the network. Furthermore, CNN performs well in sound classification. Unlike image classification, however, there is little research on suitable ConvFilters for sound classification. Therefore, we compare the performances of three convolution filters, 1D-ConvFilter, 3×1 2D-ConvFilter, and 3×3 2D-ConvFilter, in two different network configurations, when classifying mosquitoes using audio data. In insect sound classification, most machine learning researchers use only audio data as input. However, a classification model, which combines other information such as activity circadian rhythm, should intuitively yield improved classification results. To utilize such relevant additional information, we propose a method that defines this information as a priori probabilities and combines them with CNN outputs. Of the networks, VGG13 with 3×3 2D-ConvFilter showed the best performance in classifying mosquito species, with an accuracy of 80.8%. Moreover, adding activity circadian rhythm information to the networks showed an average performance improvement of 5.5%. The VGG13 network with 1D-ConvFilter achieved the highest accuracy of 85.7% with the additional activity circadian rhythm information.

Keywords

Artificial Intelligence, Bayes' Rule, Convolutional Neural Network, Mosquitoes Classification, A Priori Probability.

DOI: 10.9781/ijimai.2021.08.009

I. Introduction

Mosquitoes are amongst the deadliest insects in the world and they have a direct impact on human lives. From malaria alone, 438,000 people died in 2015 [1]. In addition, Zika virus, Dengue, Chikungunya, and Yellow fever are all carried by *Aedes aegypti*, one of the most dangerous mosquito species. It is therefore not surprising that computational entomology, which records insect information and automatically classifies or detects pests, is studied more intensively than ever. Most computational entomology studies use insect image and sound data as important inputs to an algorithm. In an image classification study, Okayasu, Yoshida, Fuchida, and Nakamura. [2] photographed mosquitoes using a single-lens reflex (SLR) camera and mobile phone. The SLR camera images were used for learning in both conventional machine learning and deep learning algorithms. The performance of each algorithm was then tested using mobile phone images. Park, Kim, Choi, Kang, and Kwon [3] caught mosquitoes native to Korea and used their images as inputs for commercial deep learning algorithms such as visual geometry group (VGG), ResNet and SqueezeNet. The authors of [4] developed an inexpensive audio sensor and used it to classify *Bombus impatiens*, *Culex quinquefasciatus*, and *Aedes aegypti*. In [5], the authors obtained audio data from eight mosquitoes and two flies, and classified them.

Traditionally, machine learning algorithms for classifying an audio signal consist of three successive processes. First, the audio signal data for a certain period are converted into spectral-temporal parameters, including frequency and amplitude, which enables decomposition into components. In sound recognition, this spectral-temporal representation is generally used as input to a network. The performance of the network algorithms depends considerably on the type of spectral-temporal representation applied. Because Mel-frequency cepstral coefficient (MFCC) extracts information from human-recognized low frequencies, classifiers trained with this algorithm emulate human hearing. For this reason, most researchers use MFCC as a basic spectral-temporal representation [6]-[9]. In this study, we also utilize MFCC to provide input to classifiers. Second, to determine the signal classes, feature extraction transforms the MFCC data into descriptors representing each audio signal. Typical feature

* Corresponding author.
E-mail address: theo@cbnu.ac.kr

extraction methods include calculation of the average and standard deviation of spectral-temporal features, principal component analysis (PCA) [10], and Autoencoder [11]. Finally, conventional machine learning classifiers such as k-nearest neighbor (kNN), support vector machine (SVM), and random forest (RF) define data classes using the extracted descriptors as input.

However, a convolutional neural network (CNN) simultaneously performs feature extraction in the network to obtain a description. Unlike traditional feature extraction methods, such as PCA and Autoencoder, the convolution filter (ConvFilter) used in CNN learns with the goal of finding a precise description for the distinction of classes. In many studies, CNN is one of the highest performing algorithms in speech recognition as well as image classification. In the Rare Sound Event Detection Task of the IEEE Audio and Acoustic Signal Processing challenge on Detection and Classification of Acoustic Scenes and Events (DCASE) 2017, the 1D convolutional recurrent neural network model recorded an F-score of 93.1 and error rate of 0.13 [12]. In the Acoustic Scene Classification challenge of DCASE 2019, various audio signals such as park, metro, and airport were used as inputs for machine learning algorithms. In this challenge, Naranjo-Alcazar, Perez-Castanos, Zuccarello, and Cobos. [13] compared prediction performances in relation to the number of layers used in a VGG-based network.

Since ConvFilter is responsible for feature extraction in CNN, the performance of CNN greatly depends on the type of filter and the method of layer stacking. For example, VGG, which won second place in the ImageNet Large Scale Visual Recognition challenge (ILSVRC) 2014, one of the most famous image classification challenges, used only a 3×3 ConvFilter. This network showed better performance than CNN models using different size filters [14]. In addition, GoogleNet, which won first place in ILSVRC 2014, was configured to reduce the computational demands of the model and calculate the correlation between channels using a 1×1 filter [15]. ResNet, which won first place in ILSVRC 2015, succeeded in building up to 152 layers using a skip connection that directly connected the layer input to its output [16]. In image classification, a CNN usually stacks layers using the 2D-ConvFilter. The 2D-ConvFilter is an intuitive filter in the image classification field, because it can create feature maps that detect high-level descriptions such as face, nose, and body from edge detection of the image. However, the spectral-temporal representation used as input in sound recognition requires a different approach in the use of ConvFilters, because this representation includes frequencies and amplitudes distributed over time, unlike images. Therefore, comparing the performances of ConvFilters for sound data is very important. Research has been conducted on the sound classification performance of commercial networks such as AlexNet, VGG, Inception, and ResNet [17]. Performance in electrocardiogram classification, which has the same form as audio signals, was studied using 1D-ConvFilter and 2D-ConvFilter [18]. In [19], the bulbul network using 3×1 2D-ConvFilter, and the sparrow network using 3×3 2D-ConvFilter were constructed for bird detection in audio signals. Sharma, Granmo, and Goodwin [20] used various spectral-temporal representations as inputs for their network. The proposed network obtains separate information for each representation, by stacking 3x1 2D-ConvFilter and 1×5 2D-ConvFilter. The accuracy of this network on the environmental sound classification dataset ESC-50 is 88.50%.

Biologically, a mosquito species' activity circadian rhythm refers to the probability of that species being active as a specific of time of day. In computational entomology, the activity circadian rhythm is significant information that can improve the performance of algorithms because different species have different activity cycles [5]. Given these activity cycles, if a trained classifier such as CNN, SVM, or RF is combined with activity cycle information, it will outperform uncombined classifiers. However, there are two major problems. The first problem is that most machine learning methods should retrain when significant new information is added. Second, the method of combination is not well established. Thus, when there is significant additional information, such as geographic distribution or activity circadian rhythm, we propose a simple method to define information as a *priori* probability, and combine it with a trained model using Bayes' rule [21]-[22]. Using this method, we avoid unnecessary relearning when new information is combined with a classification model. In addition, we can contribute to simplifying network learning using variables with different characteristics, such as activity circadian rhythms and audio signals.

The two main purposes of this paper are as follows: first, comparing the performances of 1D-ConvFilter, 3×1 2D-ConvFilter and 3×3 2D-ConvFilter, based on the VGG and a Simple CNN, to find a suitable network for mosquito classification using audio data. To train the networks, we use audio data, which include the signals of eight mosquitoes and two flies from [5]. The second purpose is to propose a simple method of combining the classification from audio signals with appropriate information of a different type. We demonstrate our proposed method by combining activity circadian rhythm information with our network classification. This simply requires the time of day to be recorded in the process of audio data collection.

II. Methods

A. Data

The data [23] provided by [5] are the wingbeat signals of eight mosquitoes and two flies obtained using audio sensors that are able to detect insects' wingbeats. The classes of the mosquitoes and the abbreviations of each class are listed in Table I. All the results of this study use these class abbreviations. According to [5], most of these insects were imported from different regions, such as California, Texas, and Taiwan, and were raised under specific conditions. A dataset of 50,000 samples, with 5,000 samples for each species, was built up.

A noise filter was applied to remove background noise from the wingbeat audio signal detected by the sensor. The audio signal was recorded at a sampling rate of 16 kHz and the duration of the signal was set to 1 s. In addition, where noise was removed, the position of the wingbeat signal was fixed to the center by the centering method, and zero-padding fixed the values in the remaining interval at 0.

The audio signal was converted into an MFCC with a shape of 40 × 43 × 1 to use as input for the models. The values 40, 43, and 1 denote the numbers of time, frequency, and amplitude intervals, respectively.

TABLE I. Abbreviations of Species

Abbreviation	Class
Fruit_Flies (FF)	Drosophila simulans
House_Flies (HF)	Musca domestica
Aedes_Female (AF)	Ae. aegypti (female)
Aedes_Male (AM)	Ae. aegypti (male)
Quinx_Female (QF)	Cx. quinquefasciatus (female)
Quinx_Male (QM)	Cx. quinquefasciatus (male)
Stigma_Female (SF)	Cx. stigmatosoma (female)
Stigma_Male (SM)	Cx. stigmatosoma (male)
Tarsalies_Female (TF)	Cx. tarsalis (female)
Tarsalies_Male (TM)	Cx. tarsalis (male)

B. Convolutional Neural Networks

The first purpose of this study was to compare the performances of three ConvFilters to determine the filter with the highest performance in classifying mosquitoes using audio data. For this comparison, we shared the same network structure with each of the filters. The ConvFilters used for classification were the 1D-ConvFilter, 3×1 2D-ConvFilter and 3×3 2D-ConvFilter, of size 3. Fig. 1 shows the feature extraction process for these three filters. 1D-ConvFilter extracts the description for a time domain of size 3 and the entire frequency domain. Additionally, 3×3 2D-ConvFilter extracts the local description of time × frequency as 3×3. However, 3×1 2D-ConvFilter obtains a separable description of a time domain for each frequency.

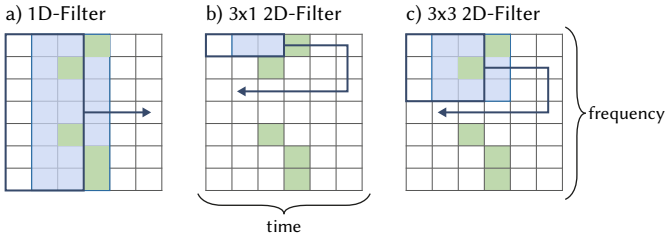

Fig. 1. Feature extraction process for three different filters.

Table II summarizes the simple CNN and VGG networks configured for filter comparison. Simple CNN is a model that measures the performance of each filter in a shallow network. This network has 6 layers, including the fully connected (FC) layer. The 1D-ConvFilter, 3×1 2D-ConvFilter, and 3×3 2D-ConvFilter have 0.6 million, 4.7 million, and 3.5 million parameters, respectively. In Simple CNN, the shapes of the final feature maps of the filters are 7×128, 7×10×128, and 7×7×128. Thus, the total number of parameters differs dramatically depending on the size of the feature map entering the FC layer.

TABLE II. Configuration Summaries of Convolutional Neural Networks

CNN	Filter	Input Shape	Number of Layers	Number of Parameters (M : million)
Simple CNN	1D-ConvFilter	40×43	6	0.6M
	3×1 2D-ConvFilter	40×43×1	6	4.7M
	3×3 2D-ConvFilter		6	3.5M
VGG13	1D-ConvFilter	40×43	13	22M
	3×1 2D-ConvFilter	40×43×1	13	22M
	3×3 2D-ConvFilter		13	28.3M

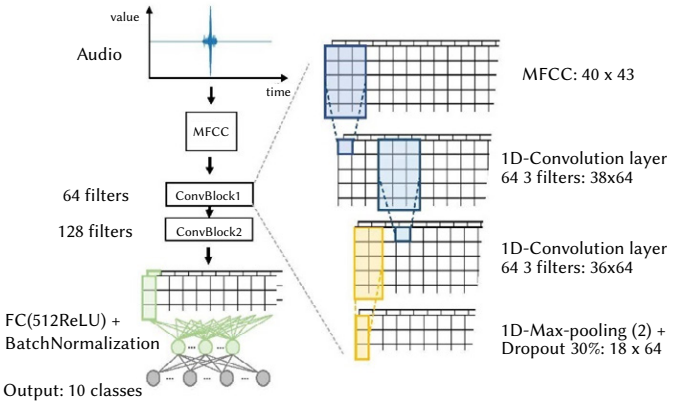

Fig. 2. Overall structure of the Simple CNN with 1D-ConvFilter (Simple 1D-CNN).

For VGG, we use VGG13 (configuration B of [14]) to adjust the shape of the feature map. To compare the performance of the ConvFilters, the filters of VGG13 were designated as 3×3 2D-ConvFilter, 3×1 2D-ConvFilter and 3 1D-ConvFilter. The numbers of parameters were approximately 22 million, 22 million, and 28.3 million. The shapes of the final feature maps of the filters were 1×512, 1×1×512, and 1×1×512. In VGG, the total number of parameters is determined by the different numbers of parameters of each filter, regardless of the size of the feature map entering the FC layer.

1. Simple CNN

We proposed three ConvFilters, with two shared CNN structures for each. The first shared network is Simple CNN with a shallow layer. To aid understanding, we describe a Simple CNN trained with 1D-ConvFilter.

Fig. 2 shows the overall structure of this Simple 1D-CNN. The layers consist of 1D-Convolution, 1D-Max-pooling, Dropout, BatchNormalization and FC. Each ConvBlock consists of two 1D-convolution layers of the same size, a 1D max-pooling layer with a kernel size of 2, and a dropout layer with a ratio of 30%. The kernel size of all ConvFilters is 3, and the 1D-convolution filter sizes of the two ConvBlocks are 64 and 128. For feature extraction, the first convolutional layer (ConvLayer) extracts a 38×64 feature map from a 40×43 shaped MFCC. The feature map provides descriptors of the entire frequency domain in a specific time domain MFCC, as previously described. The feature map channel is determined according to the number of filters. In total, the MFCC shrinks from 40×43 to 7×128 as it proceeds through the feature extraction. The FC layer, of size 512, uses descriptors obtained through the filters as inputs to classify the mosquito species. The activation function of the final FC layer, of size 10, uses softmax.

2. VGGNet

VGGNet is a commonly used CNN structure in many fields, because of the intuitiveness of its model structure. We now describe VGG13 with 3×3 2D-ConvFilter to illustrate how 2D-ConvFilter is learned inside a CNN. Essentially, 2D-ConvFilter moves in two dimensions in the MFCC and learns features locally. The deeper the layer, the more effective it is in creating high-level descriptors by combining local low-level descriptors.

Fig. 3 shows the overall structure of VGG13 with 3×3 2D-ConvFilter. VGG13 consists of 2D-Convolution, 2D max-pooling, dropout and FC layers. Each ConvBlock consists of two 2D-convolution layers with the same filter size, and a 2D max-pooling layer. The kernel size of all

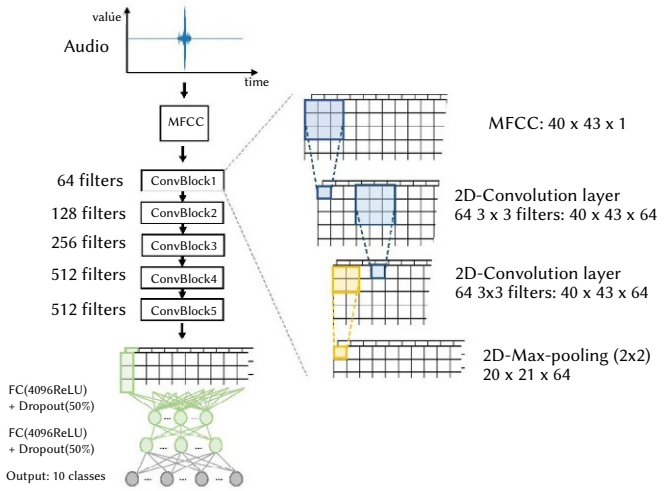

Fig. 3. Overall structure of VGG13 with 3×3 2D-ConvFilter.

filters is 3×3, and the filter sizes of the successive 2D-ConvBlocks are 64, 128, 256, 512 and 512. The shape of a feature map is interpreted as time × frequency × channels. In feature extraction, since all ConvLayers use padding, the first ConvLayer extracts a feature map of 40×43×64 from the 40×43×1 MFCC, which is the same size. Throughout feature extraction, MFCC is reduced from 40×43×1 to 1×1×512. The descriptors obtained through the filters classify the mosquito species after passing through two FC layers of size 4096 and a final FC layer of size 10. The activation function of the final layer uses softmax, as with Simple CNN.

C. Bayes' Rule-based Method for Adjusting Classification Output

In [5], a Naïve Bayes classifier is combined with activity cycle information, and it will outperform uncombined classifiers. However, most classifiers except the Naïve Bayes classifier face two major problems in combining activity cycle information. The first problem is that most machine learning methods should retrain when significant new information is added. Second, the method of the combination is not well established.

The appearance rate of insects differs according to information such as geographic distribution and activity circadian rhythm. Intuitively, if we have previously obtained information affecting the appearance rate, a trained classifier could use this information to obtain better performance. However, in most computational entomology studies, although such information regarding appearance rate is known, there is insufficient discussion about how to use it. In this study, we propose a method that defines prior information about appearance rate as a *priori* probability, and combines it with trained classifiers.

The prior information obtained by [5] was the activity cycle for each species. This was based on the time of observation of individual insects over one month. Fig. 4 shows the diurnal activity cycles, or activity circadian rhythms, of the ten species, identified by the abbreviations in Table I. In Fig. 4, there are two moments in the day in which there is a more notorious activity, of all the species in general. QM showed the most activity between 9 p.m. and 11 a.m., and TM showed the most activity between 5 a.m. and 7 a.m.

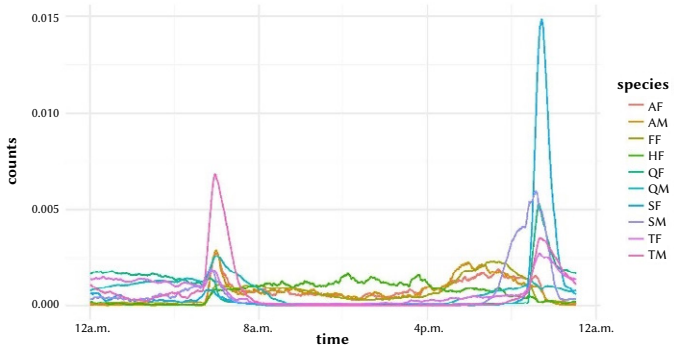

Fig. 4. Activity circadian rhythms for 10 species (see Table I for abbreviations).

The ultimate purpose of the adjusted classifier is to predict the new rate at which the i-th species c_i will appear when the independent variable, $\tilde{x}$, (here MFCC) exists at a certain time, t. We define an activity time rate as a *priori* probability $\hat{p}_{c_i}(t)$ for each species c_i and apply it to the trained CNN. Before calculating the predicted appearance rate of insect species, we assume that the scores of the training data $p(\tilde{x}|c_i)$ and the scores of the new data $\hat{p}_t(\tilde{x}|c_i)$ are the same.

Suppose we wish to predict the new appearance rate of a certain species at time. Bayes' rule provides

$$\hat{p}_t(c_i|\tilde{x}) = \frac{\hat{p}_t(\tilde{x}|c_i)\hat{p}_t(c_i)}{\hat{p}_t(\tilde{x})} \tag{1}$$

where a *priori* probability $\hat{p}_t(c_i)$ denotes the appearance rate of i-th species c_i at time t in the new data, and $\hat{p}_t(\tilde{x})$ denotes the marginal probability of $\hat{p}_t(\tilde{x}|c_i)$.

The estimated probability of the classifier for the training data $\hat{p}(c_i|\tilde{x})$ is as follows:

$$\hat{p}(c_i|\tilde{x}) = \frac{\hat{p}(\tilde{x}|c_i)\hat{p}(c_i)}{\hat{p}(\tilde{x})} \tag{2}$$

where a *priori* probability $\hat{p}(c_i)$ denotes the appearance rate of i-th species c_i in the training data, and $\hat{p}(\tilde{x})$ denotes the marginal probability of $\hat{p}(\tilde{x}|c_i)$.

Since the scores of the training data $\hat{p}(\tilde{x}|c_i)$ and the new data $\hat{p}_t(\tilde{x}|c_i)$ are the same, by equating equation (1) to (2) and defining $g(\tilde{x}) = \hat{p}(\tilde{x})/\hat{p}_t(\tilde{x})$, we obtain

$$\hat{p}_t(c_i|\tilde{x}) = g(\tilde{x})\frac{\hat{p}_t(c_i)}{\hat{p}(c_i)}\hat{p}(c_i|\tilde{x}) \tag{3}$$

Since $\sum_{i=1}^{n}\hat{p}_t(c_i|\tilde{x})=1$, we obtain $g(\tilde{x})=[\sum_{i=1}^{n}\hat{p}_t(c_i)/\hat{p}(c_i)\cdot\hat{p}(c_i|\tilde{x})]^{-1}$. This also means that the term is statistically normalized.

Finally, by Bayes' rule, the relationship between the a *priori* probability $\hat{p}_t(c_i)$ in the above equation and the a *priori* probability for activity cycle $\hat{p}_{c_i}(t)$ is

$$\hat{p}_t(c_i) = \frac{\hat{p}_{c_i}(t)\hat{p}(c_i)}{\hat{p}(t)} \tag{4}$$

where the probability $\hat{p}(t)$ denotes the appearance rate at time t in the activity cycle. Since $\hat{p}_t(c_i|\tilde{x}) = g(\tilde{x})\frac{\hat{p}_t(c_i)}{\hat{p}(c_i)}\hat{p}(c_i|\tilde{x})$, and $\hat{p}(t)$ is included in normalizing term $g(\tilde{x})$, we can easily obtain $\hat{p}_t(c_i|\tilde{x}) \propto \hat{p}_{c_i}(t)\hat{p}(c_i|\tilde{x})$. The process of obtaining the *posteriori* probability above is illustrated in Fig. 5.

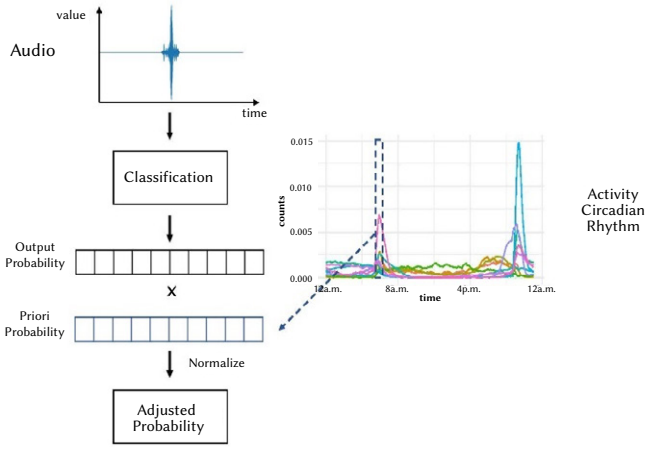

Fig. 5. Process of adjusting output probabilities using Bayes' rule method.

Suppose we have additional information such as geographic distribution as well as activity circadian rhythm, which together is expressed as m variables F_j, $j = 1 \ldots m$. If we assume that these variables are independent, and $\hat{p}(\tilde{x}|c_i) = \hat{p}_{Fs}(\tilde{x}|c_i)$, the a *posteriori* probability can be generalized as

$$\hat{p}_t(c_i|\tilde{x}) \propto \prod_{j=1}^{m}\hat{p}_{c_i}(F_j)\,\hat{p}(c_i|\tilde{x}) \tag{5}$$

D. Training the Convolutional Neural Networks

Training of neural networks involves the process of repeatedly adjusting weights to reduce differences between network-predicted and actual values to below a threshold. The tuning parameters required for network training are initializer, optimizer, epoch, and batch size. Simple CNN uses the Xavier initializer [24], which depends on the number of previous and next nodes. The VGG network uses

the uniform initializer. Most sound classification models use the Adam optimizer [19]-[20]. We also use the Adam optimizer [25], in which decay rates $\beta_1\beta_1$ and $\beta_2\beta_2$ are set to 0.9 and 0.999, respectively, and the learning rates are set to 1e-5 and 1e-4 for optimal network selection. The epoch of the network is fixed at 100, and the batch sizes are set at 128 and 256. To train the networks, we used the Keras framework in a Ryzen 2700x @3.70 GHz with 32 GB RAM and RTX 2080ti.

E. K-fold Cross-Validation

The k-fold cross-validation method divides the training dataset into k data subsets. Next, one of the k subsets is used for model evaluation, and the remaining k-1 subsets are used as training data. By repeating this process k times, k-fold cross-validation uses all the subsets as validation data. The final accuracy of the classifier is the average of the k-fold accuracies.

In this study, we used 5-fold cross-validation to compare the performances of the networks. In addition, we separated the data into 80% training-set and 20% validation-set, for comparison of networks that had been trained only with MFCC and those that were combined with activity circadian rhythm information.

III. Results

A. Classification Performance

Our metric for evaluating classification performance is the accuracy of each network obtained by 5-fold cross-validation. For each model, learning rates of 1e-5 and 1e-4 were applied, and batch sizes of 256 and 128 were used. Table III shows the average accuracies of Simple CNNs using 5-fold cross-validation. For Simple CNNs with 4 ConvLayers, the 1D-ConvFilter with learning rate 1e-4 and batch size 128 has an accuracy of 80.0%, higher than any of the 2D-ConvFilter configurations.

Table IV shows the average accuracies of VGG13 networks, with the same layout as Table III. Here, the highest accuracy of 80.8% is obtained for the VGG13 with 3×3 2D-ConvFilter, a learning rate of 1e-5 and a batch size of 256.

TABLE III. Comparison of Networks for Simple Cnns with Different Convolution Filters Using 5-Fold Cross Validation

CNN	Filter	Learning Rate	Batch Size	Accuracy
Simple CNN	1D-ConvFilter	1e-5	256	78.4%
		1e-5	128	79.3%
		1e-4	128	**80.0%**
	3×1 2D-ConvFilter	1e-5	256	77.4%
		1e-5	128	78.3%
		1e-4	128	79.1%
	3×3 2D-ConvFilter	1e-5	256	78.9%
		1e-5	128	79.8%
		1e-4	128	79.8%

TABLE IV. Comparison of Networks for Vgg13 with Different Convolution Filters Using 5-Fold Cross Validation

CNN	Filter	Learning Rate	Batch Size	Accuracy
Simple CNN	1D-ConvFilter	1e-5	256	80.5%
		1e-5	128	80.1%
		1e-4	128	79.0%
	3×1 2D-ConvFilter	1e-5	256	80.3%
		1e-5	128	80.4%
		1e-4	128	79.9%
	3×3 2D-ConvFilter	1e-5	256	**80.8%**
		1e-5	128	80.6%
		1e-4	128	80.1%

We conclude from Table III and Table IV that the 1D-ConvFilter shows the highest performance when the number of network layers is small and the 2D-ConvFilter shows the highest performance when the network is deeper.

B. Effect of Activity Circadian Rhythms on A Priori Probabilities

In Section II, we described CNNs with different ConvFilters, and explained how combining a trained network with significant a *priori* information could be used to obtain improved predictions. In this section, we discuss the effect of using activity circadian rhythms as a *priori* information to aid mosquito species classification. To this end, 50,000 audio datasets were divided into an 80% training-set and a 20% test-set. The learning rate and epoch for network training were set to 1e-5 and 256, respectively. The values of the remaining tuning parameters were the same as in the previous results.

Before discussing the results, we first describe the nature of the Naive Bayes classifier. Because we use Bayes' rule to train the classifier, a Naive Bayes classifier is more flexible than other classifiers for the problem of applying additional information. In [5], the Naive Bayes classifier was trained to use insect sound data and activity cycle information as input to the classifier. Table V shows the accuracy of each network according to whether or not activity cycle information was added. The average difference between networks with and without activity cycle information is approximately 5.5%. In addition, all VGG13 networks have higher accuracy than the reference accuracy of [5]. Generally, the networks without activity cycles have similar results to Table III and Table IV. Moreover, when applying activity cycles as additional information, the accuracy of the 1D-ConvFilter in Simple CNN is highest at 84.68%. However, unlike the results in Table IV, the accuracy of the 1D-ConvFilter in VGG13 is highest at 85.72%.

Fig. 6 shows the change in recall of VGG13 when activity circadian rhythm is added. Recall represents the ratio of the predicted number in the *i*-th class to the actual number in the *i*-th class. In other words, this measure indicates how well the classifier predicts the mosquito species for each sound signal. In Fig. 6, the overall recall of QF is noticeably lower than other classes. Conversely, AM has the highest average recall. We see that a network with activity cycle information (circle in Fig. 6) has a higher recall of all classes by about 1% difference than a network without this information (triangle). This improvement in recall is largest for AF, and smallest for FF. In addition, in the result of 1D-ConvFilter VGG13 with activity circadian rhythm, the average difference in recall for each class is significantly higher than the differences for other networks. This result causes the highest accuracy 85.72% for 1D-ConvFilter in Table IV results for VGG13.

TABLE V. Comparison of Networks with or without Addition of Activity Circadian Rhythms

Adding Activity Circadian Rhythms	CNN	Filter	Accuracy
No	Simple CNN	1D-ConvFilter	78.76%
		3×1 2D-ConvFilter	76.46%
		3×3 2D-ConvFilter	74.90%
	VGG13	1D-ConvFilter	80.36%
		3×1 2D-ConvFilter	80.41%
		3×3 2D-ConvFilter	80.47%
Yes	Simple CNN	1D-ConvFilter	**84.68%**
		3×1 2D-ConvFilter	83.40%
		3×3 2D-ConvFilter	82.28%
	VGG13	1D-ConvFilter	**85.72%**
		3×1 2D-ConvFilter	85.66%
		3×3 2D-ConvFilter	82.91%
	Naive Bayes Method [5]		79.44%

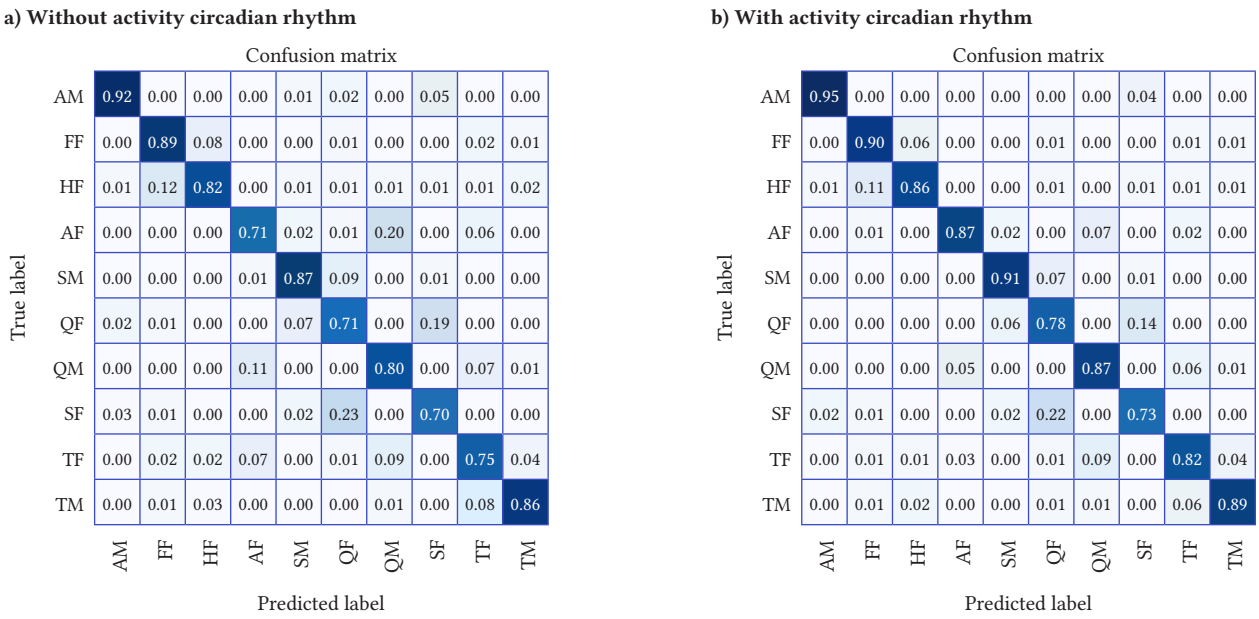

Fig. 7. Confusion matrix of VGG13 with 3×3 2D-ConvFilter (a, left) without and (b, right) with activity circadian rhythm information added.

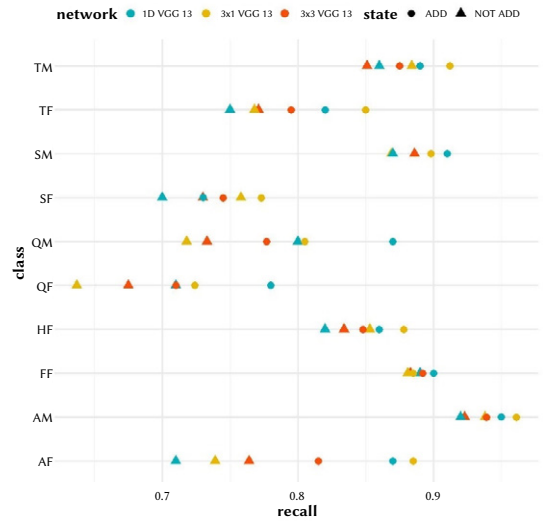

Fig. 6. Scatter plot of recall of different VGG13 networks without (triangles) and with (circles) activity cycle information.

Fig. 7 shows the change in the confusion matrix of VGG13 using 1D-ConvFilter from (a) without to (b) with activity circadian rhythm information. This network gave the highest accuracy with activity information (Table V). Without activity circadian rhythm, the AF to QM and QM to AF prediction rates account for 20% and 11% of misclassifications, respectively. However, with activity circadian rhythm, these rates fall to 7% and 5%. Other cases similarly show that adding activity circadian rhythm information reduces misclassification errors.

IV. Discussion and Limitation

In this section, we discuss some contributions and limitations of our study. Traditionally, CNN is a method in which research on image data is becoming active. And ConvFilter, which determines the performance of this method, is also being studied a lot on image data. However, research on ConvFilter is not active on sound data. So we introduced ConvFilters in section II and analyzed their results in section III. Simple CNN with the 1D-ConvFilter has an accuracy of 80.0%, higher than any of the Simple CNNs with 2D-ConvFilter. In VGG13, the highest accuracy of 80.8% is obtained for the VGG13 with 3×3 2D-ConvFilter.

When applying activity cycles as additional information, the accuracy of Naive Bayes classifier [5] is 79.44%. On the other hand, the accuracies of our proposed methods are 84.68% and 85.72% at the 1D-ConvFilter in Simple CNN and the 1D-ConvFilter in VGG13, respectively. Moreover, adding activity circadian rhythm information to each of the VGG 13 and Simple CNNs results in a 5.5% and 6.7% difference in average performance improvement, respectively. In Fig. 6, the recall values and misclassification rates of each class show better results by about 10% difference with adding activity circadian rhythm information. Thus, it is explained that additional information such as activity rhythm information improves the performance of the network.

While our evaluations are encouraging, there are certain limitations to our method. We proceed with the analysis using limited data. If the data containing other information such as location and seasonality as well as activity circadian rhythm information are used, the analysis results are more reliable. In order to extract features of sound data, feature extraction methods other than MFCC may be used. Furthermore, in order to compare with CNN models, we can apply end-to-end neural network models that take sound data of mosquitoes as input.

V. Conclusion

The first objective of this study was to find a network filter configuration that could use audio signals to classify mosquitoes and flies. We selected three different filters, 1D-ConvFilter, 3×1 2D-ConvFilter, and 3×3 2D-ConvFilter, and applied them to Simple CNN and VGG13 networks to classify mosquito and fly species. The accuracy of each network was calculated using 5-fold cross-validation. Comparing the results, VGG13 with 3×3 2D-ConvFilter showed the highest accuracy of 80.8%. Also, all the accuracies of VGG13 networks are greater than that of Simple CNN.

Second, because different species have different activity cycles, we proposed a method using Bayes' rule to combine activity cycle information with trained networks. The activity circadian rhythm for each species was defined as an a *priori* probability to use Bayes' rule.

The adjusted probability for each species was obtained by multiplying the defined a *priori* probability by the probability obtained from the trained network. Combining networks with activity cycles in this way showed an average improvement in accuracy of 5.5%, with VGG13 using 1D-ConvFilter showing the highest accuracy of 85.72%. Furthermore, by incorporating activity cycle information, misclassifications, such as AF to QM, can be reduced.

In conclusion, when performing classification, we can use not only audio data or image data, but also other types of information, such as activity cycle and geographical distribution. Thus, if location and time information are also collected in the process of collecting audio data, we believe that this relatively simple method can obtain even better results.

References

[1] World Health Organization: Mosquito-Borne Diseases. Available online: http://www.who.int/neglected_diseases/vector_ecology/mosquito-borne-diseases/en/ (accessed on 1 April 2017).

[2] K. Okayasu, K. Yoshida, M. Fuchida, and A. Nakamura, "Vision-Based Classification of Mosquito Species: Comparison of Conventional and Deep Learning Methods," *Applied Sciences*, vol. 10, no. 1, pp. 3935, 2019, doi:10.3390/app9183935.

[3] J. Park, D. I. Kim, B. Choi, W. Kang, and H. W. Kwon, "Classification and Morphological Analysis of Vector Mosquitoes using Deep Convolutional Neural Networks," *Scientific Reports*, vol. 10, no. 1, pp. 1-12, 2020, doi: 10.1038/s41598-020-57875-1.

[4] G. E. Batista, E. J. Keogh, A. Mafra-Neto, and E. Rowton, "SIGKDD demo: sensors and software to allow computational entomology, an emerging application of data mining," in *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2011, pp. 761-764, doi: 10.1145/2020408.2020530.

[5] Y. Chen, A. Why, G. E. Batista, A. Mafra-Neto, and E. J. Keogh, "Flying insect classification with inexpensive sensors," *Journal of insect behavior*, vol. 27, no.5, pp. 657-677, 2014, doi: 10.1007/s10905-014-9454-4.

[6] J. J. Noda, C. M. Travieso-González, D. Sánchez-Rodríguez, and J. B. Alonso-Hernández, "Acoustic Classification of Singing Insects Based on MFCC/LFCC Fusion," *Applied Sciences*, vol. 9, no. 19, pp. 4097, 2019, doi: 10.3390/app9194097.

[7] Z. Le-Qing, "Insect sound recognition based on MFCC and PNN," in *2011 International Conference on Multimedia and Signal Processing*, 2011, pp. 42-46, doi: 10.1109/CMSP.2011.100.

[8] N. Saleem, and T. G. Tareen, "Spectral Restoration based speech enhancement for robust speaker identification," *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 5, no. 1, pp. 34-39, 2018, doi: 10.9781/ijimai.2018.01.002.

[9] N. Saleem, and M. I. Khattak, "Deep Neural Networks for Speech Enhancement in Complex-Noisy Environments," *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 6, no. 1, pp. 84-90, 2020, doi: 10.9781/ijimai.2019.06.001.

[10] X. Fan, H. Feng, and M. Yuan, "PCA based on mutual information for acoustic environment classification," In *2012 International Conference on Audio, Language and Image Processing*, 2012, pp. 270-275, doi: 10.1109/ICALIP.2012.6376624.

[11] S. Ghosh, E. Laksana, L. P. Morency, and S. Scherer, "Learning representations of affect from speech," 2015, arXiv:1511.04747.

[12] H. Lim, J. Park, and Y. Han, "Rare sound event detection using 1D convolutional recurrent neural networks," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2017 Workshop*, 2017, pp. 80-84.

[13] J. Naranjo-Alcazar, S. Perez-Castanos, P. Zuccarello, and M. Cobos, "DCASE 2019: CNN depth analysis with different channel inputs for Acoustic Scene Classification," 2019, *arXiv:1906.04591*.

[14] K. Simonyan, and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.

[15] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1-9, doi: 10.1109/CVPR.2015.7298594.

[16] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770-778, doi: 10.1109/CVPR.2016.90.

[17] S. Hershey, S. Chaudhuri, D. P. Ellis, J. F. Gemmeke, A. Jansen, R. C. Moore, M. Plakal, D. Platt, R. A. Saurous, B. Seybold, M. Slaney, R. J. Weiss and K. Wilson, "CNN architectures for large-scale audio classification," in *2017 IEEE international conference on acoustics, speech and signal processing*, 2017, pp. 131-135, doi: 10.1109/ICASSP.2017.7952132.

[18] Y. Wu, F. Yang, Y. Liu, X. Zha, and S. Yuan. "A comparison of 1-D and 2-D deep convolutional neural networks in ECG classification," 2018, *arXiv:1810.07088*.

[19] T. Grill, and J. Schlüter, "Two convolutional neural networks for bird detection in audio signals," in *2017 25th European Signal Processing Conference (EUSIPCO)*, 2017, pp. 1764-1768, doi: 10.23919/EUSIPCO.2017.8081512.

[20] J. Sharma, O. C. Granmo, and M. Goodwin, "Environment Sound Classification using Multiple Feature Channels and Attention based Deep Convolutional Neural Network," 2019, *arXiv:1908.11219*.

[21] M. Saerens, P. Latinne, and C. Decaestecker, "Adjusting the outputs of a classifier to new a priori probabilities: a simple procedure," *Neural computation*, vol. 14, no. 1, pp. 21-41, 2002, doi: 10.1162/089976602753284446.

[22] P. Latinne, M. Saerens, and C. Decaestecker, "Adjusting the outputs of a classifier to new a priori probabilities may significantly improve classification accuracy: evidence from a multi-class problem in remote sensing," in *Proceedings of the Eighteenth International Conference on Machine Learning*, 2001, Vol. 1, pp. 298-305.

[23] Y. Chen, A. Why, G. E. Batista, A. Mafra-Neto, and E. J. Keogh. "Flying Insect Classification with Inexpensive Sensors." Distributed by Y. Chen. https://sites.google.com/site/insectclassification/.

[24] X. Glorot, and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, 2010, pp. 249-256.

[25] D. P. Kingma, and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.

Jaehoon Kim

Jaehoon Kim received the B.S. and M.S. in Information & Statistic from Chungbuk National University, South Korea. Now he is Ph. D course in statistics from Chungbuk National University. His research interests are hyperparameter optimization problem, computer vision, digital signal processing, data mining, statistical learning, and deep learning.

Jeonkyu Oh

He obtained his B.S and M.S. in Statistic from Chungbuk National University, South Korea. Now he is working as data scientist in BEGAS Inc. in South Korea. His area of interest includes machine learning, data mining and statistical learning.

Tae-Young Heo

He received the B. S. in Statistics from the Chungbuk National University and the M. S. and Ph. D. in Statistics from the North Carolina State University. Now he is a professor in department of Information and Statistics, Chungbuk National University. His current research interests include statistical learning and modeling.