ICIC Express Letters
Volume **13**, Number **7**, July **2019**

ICIC International ©2019 ISSN 1881-803X
pp. **617–624**

WEED SEEDS CLASSIFICATION BASED ON SAAK TRANSFORM

Ruifei Zhang[1] and Cheng Cai[2,*]

[1]College of Information Engineering
Northwest A&F University
No. 3, Taicheng Road, Yangling District, Xianyang 712100, P. R. China

[2]School of Electronic and Information Engineering
Shanghai Dianji University
No. 300, Shuihua Road, Pudong New Area District, Shanghai 201306, P. R. China
*Corresponding author: cheneychengcai@163.com

Received December 2018; accepted March 2019

Abstract. *Agricultural production is closely related to people's lives. In this paper, we focus on the impact of weeds and take corresponding measures. The weeds mainly rely on the seeds to spread and propagate. To facilitate the researchers' work and promote agricultural development, we want to find an efficient method to classify the weed seeds accurately. Saak (Subspace approximation with augmented kernels) transform is a new data-driven way to extract image features, which is improved based on the PCA (Principal Component Analysis) Network. It is similar to PCA Network that using multi-stage PCA for the image to build the orthonormal bases which is called the optimal linear subspace approximation. And the creative innovation is augmenting the negative value of each transform kernel. After transform, there are lots of Saak coefficients in each stage. The F-test method is used to select some coefficients with higher discrimination for classification. In our experiment, we have 211 kinds of weed seeds. After getting image features, we use the LMC (Large Margin Classifier) based on the affine hull to implement the classification task. The result shows that our method has a good performance on classifying weed seeds.*

Keywords: Saak transform, PCA, F-test, LMC, Affine hull

1. **Introduction.** Weed management has always been a difficult problem for farmers and researchers. Weeds compete for soil nutrients, lighting and other resources with crops, hindering their normal growth. Generally, different kinds of weed seeds are often mixed with crop seeds and make use of the power of water, wind, animals or human activities to spread and propagate [1]. Especially in our country, the rotation system makes the situation more complicated [2]. Thus, it is of significance for us to find a machine learning or deep learning method to achieve the classification of weed seeds accurately and efficiently, instead of manual operation, to help the researchers identify the weed seeds and cultivate the crops.

With the development of CNN (Convolutional Neural Network), deep learning has drawn people's attention. Even more surprising to us, it has been applied to the agricultural field in [3-5]. After CNN and PCANet (Principal Component Analysis Network) [6] which is a data-driver method to obtain the filters by PCA, Saak (Subspace approximation with augmented kernels) transform was proposed in [7,8] recently. We apply it to our weed seeds classification task creatively and provide some inspirations for future related research. Saak transform takes the advantage of CNN and PCANet, which adopts multi-layer network structure and gets kernels of each layer by doing the method of PCA saving lots of training time. Besides, augmenting the negative of the kernels is the creative innovation in Saak transform which effectively reduces the loss caused by the

DOI: 10.24507/icicel.13.07.617

ReLU (Rectified Linear Unit). We take the output of each stage in Saak transform as the Saak coefficients. After multi-layer convolution, F-test is a good way to select more discriminative Saak coefficients to form a feature vector for our weed seeds classification. Through experiments, we can see that this method is simpler and more efficient compared with the previous work [9-11]. At the same time, our data set has more kinds of weed seeds, which has practical application value.

In this paper, the whole process of feature extraction by Saak transform and F-test is described in Section 2. The large margin classifier based on affine hulls is introduced in Section 3, which we use in our experiment to accomplish weed seeds classification. In Section 4, we introduce relevant experiments and analyze the results. Finally, the conclusions are presented in Section 5.

2. Feature Extraction by Saak Transform.

2.1. RECOS (REctified-COrrelations on a Sphere) transform.

Before discussing Saak transform, we talk about CNN at first. Recently, the RECOS transform [12,13] has been proposed. In this theory, they try to explain the process in CNN with the mathematical methods. The convolution process is regarded as a projection in a subspace where the filters are the bases of the subspace. Thus, the three terms, "bases", "filters" and "kernels" can be used interchangeably. The vector set E is composed of all bases in one stage.

$$E = \{e_0, e_1, \ldots, e_k, \ldots, e_K\} \tag{1}$$

where e_1 to e_K vectors are usually obtained by training the CNN with labeled data. And

$$e_0 = \frac{1}{\sqrt{N}}(1, 1, \ldots, 1)^T \tag{2}$$

On this condition, the convolution is viewed as projecting input $f \in R^N$ onto the subspace which has $K + 1$ bases:

$$p_k = e_k^T f, \quad k = 0, 1, \ldots, K \tag{3}$$

P is the set of projected values:

$$P = (p_0, p_1, \ldots, p_K)^T \in R^{K+1} \tag{4}$$

And then we pass the set P except p_0 through the ReLU. Y is the set of final outputs:

$$Y = (y_0, y_1, \ldots, y_K)^T \tag{5}$$

where

$$y_0 = p_0 \tag{6}$$

and

$$y_k = \begin{cases} p_k, & \text{if } p_k > 0, \\ 0, & \text{if } p_k \leq 0 \end{cases} \quad k = 1, 2, \ldots, K \tag{7}$$

2.2. Saak transform.

The bases in RECOS model need lots of time to train and correct. In addition, the loss caused by ReLU is also the non-ignorable shortcoming of CNN. To solve these problems, Saak transform has been proposed.

In Saak transform, for input samples $f \in R^N$, we make use of PCA to determine the orthonormal transform kernels as the bases of the subspace. And actually the kernels are the eigenvectors obtained by PCA.

$$V = \{v_1, \ldots, v_k, \ldots, v_N\} \tag{8}$$

V is the set of eigenvectors which are orthogonal to each other.

Then, we augment the negative value of set V by the following rule to form set E, and this step can efficiently reduce the loss caused by ReLU:

$$e_{2k-1} = v_k, \quad e_{2k} = -v_k, \quad k = 1, \ldots, N \tag{9}$$

e_0 is also computed by Equation (2).

After above steps, the final bases used in Saak transform are obtained. Next, it is the same as CNN that computes the projection onto the set E and gets the final output through ReLU:

$$Y = (y_0, y_1, \ldots, y_{2N})^T \in R^{2N+1} \tag{10}$$

where $y_0 = p_0$, and others can be obtained by the following formulas:

$$y_{2k-1} = p_{2k-1} \text{ and } y_{2k} = 0 \text{ if } p_{2k-1} > 0 \tag{11}$$

$$y_{2k-1} = 0 \text{ and } y_{2k} = p_{2k} \text{ if } p_{2k} > 0 \tag{12}$$

Until now, Saak transform is completed. The final outputs are the Saak coefficients. In fact, the Saak transform is just a convolution where the filters are obtained by PCA and augment their negative values as the filters, too. The ReLU here is still the traditional ReLU. We do not make any additional changes.

2.3. **Multi-stage Saak transforms.** Next, we apply the mathematics theory to the practice. For a large size image, multi-stages Saak transforms are needed to accomplish feature extraction. The recursive process is shown as below.

Stage 1:

We decompose our original image into lots of non-overlapping local cuboids of size $2 \times 2 \times k_0$. k_0 is 3 here because our weed seed image is the color image. Then PCA is applied to the matrix which is composed of these local cuboid vectors to get eigenvectors, and then augment the negative values of them to form the orthogonal bases in the first Saak transform stage. Due to the size of local cuboid, the base in subspace or the filter in convolution is 2×2, too. Through the convolution, we get the first set of Saak coefficients. The width and height of the input image are reduced by one half while the channel increases as the number of filters.

Stages 2, 3, ...:

Just do what mentioned above in Stage 1 to the previous stage Saak coefficients, and then we can get the new stage bases and new Saak coefficients.

The number of bases in each stage depends on this formula:

$$K_P = 2 \times 4 \times K_{P-1} \tag{13}$$

where the first one is because we add their negative values of eigenvectors, which is equivalent to double the number. And the product of the second and third terms is the number of eigenvectors obtained through PCA.

As what mentioned above, in practice, we just need to pass the convolution with augmenting the negative values of filters obtained by PCA and the traditional ReLU to accomplish these steps and get Saak coefficients in each stage, instead of doing anything additional. The dimension of filters in each stage is $2 \times 2 \times K_P$.

In convolution, the dimension of Saak coefficients can be computed as

$$N_p = 2^{-1} N_{p-1} \tag{14}$$

$$K_P = 2 \times 4 \times K_{P-1}$$

N_p stands for the width and height of image, and K_P is the number of channels. When the Saak coefficient has the dimension $1 \times 1 \times k_f$, the whole process is over. To accomplish this dimension, we usually resize or correct the image size to $2^p \times 2^p$. On this condition, $k_f = 2^{3p}$.

The relevant process with our experimental image is shown in Figure 1.

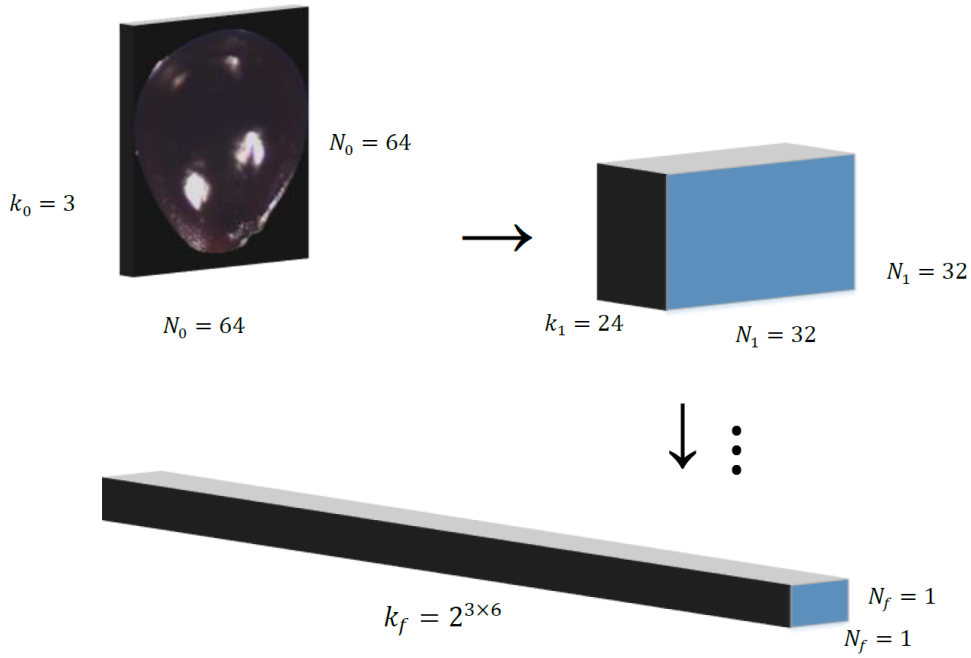


FIGURE 1. Saak transform process with our experimental image

2.4. **Feature selection by F-test.** Through multi-stage Saak transforms, the number of Saak coefficients is too large to use as the features of image. Therefore, we adopt the F-test method to solve this problem, which can be written as:

$$F = \frac{\text{between-group variability (BGV)}}{\text{within-group variability (WGV)}} \tag{15}$$

BGV reflects the feature differences of samples belonging to different classes, and WGV reflects the feature differences of samples belonging to the same class, which can be calculated by these formulas:

$$\text{BGV} = \sum_{c=1}^{C} n_c \left(\bar{S}_c - \bar{S}\right)^2 \Big/ (C-1) \tag{16}$$

and

$$\text{WGV} = \sum_{c=1}^{C} \sum_{f=1}^{n_c} \left(S_{c,f} - \bar{S}_c\right)^2 \Big/ (N-C) \tag{17}$$

In these two formulas, C is the number of classes. N is the total number of all samples in all classes. n_c is the cth class samples number. $\bar{S}_c$, $\bar{S}$ are respectively the mean of the cth class samples and all samples. And $S_{c,f}$ is the specific sample of the specific class determined by the subscripts c and f. We select features used in our weed seeds classification by the scores of F-test, because the scores reflect the features' discriminant power.

3. Classification Based on LMC Classifier.

3.1. **The construction of the binary classifier.** Like SVM (Support Vector Machine) classifier [14-16], LMC [17] also accomplishes classification through determining the separating hyperplane which maximizes the distance between the closest points in two classes and the hyperplane. In fact, that is the perpendicular bisector plane of the line connected by the nearest points. The difference with SVM is that SVM represents class with the convex hull, and in high-dimensional space, the actual class will extend beyond, resulting

in bad classification performance. However, the LMC represents the class by the affine hull composed of its training samples.

After Saak transform and F-test, the extracted feature of each class can form the set: $\{z_i\}_{i=1,2,\ldots,n}$, n is the samples number. Using these samples, we can describe the class through forming the affine hull:

$$H^{aff} = \left\{ z = \sum_{i=1}^{n} \alpha_i z_i \middle| \sum_{i=1}^{n} \alpha_i = 1 \right\} \tag{18}$$

And it is more important that we can further simplify the expression of the affine hull for two linearly separable classes:

$$H^{aff} = \{z = Uv + \mu\} \tag{19}$$

This formula can be seen as the reconstruction of PCA, where μ represents the mean of samples, satisfying $\mu = \left(1/n \sum_{i=1}^{n} z_i\right)$. And U represents the orthonormal bases of the affine subspace. Normally, it can be obtained by SVD (Singular Value Decomposition) of matrix $[z_1 - \mu, \ldots, z_n - \mu]$. Discarding the eigenvector columns corresponding to the singular values that are close to zero in the U-matrix of SVD, we get the U. v is the data value under the basis U in the subspace.

Next, what should we do is determining the optimal separating hyperplane which can be written as $\langle w, z_i \rangle + b = 0$, where z_i represents the points lying on the hyperplane, w is the hyperplane's normal, and $|b|/||w||$ is the vertical distance between the origin and the hyperplane. The points on both sides of the hyperplane belong to two classes, and one is positive class satisfying $\langle w, z_i \rangle + b > 0$ and the other is negative class satisfying $\langle w, z_i \rangle + b < 0$. Based on this condition, we can obtain the class which a new test sample belongs to by using the function $f(z) = \langle w, z \rangle + b$. If the result > 0, it belongs to the positive; in contrast, it belongs to the negative.

Now, suppose that we have two classes to be classified, one of them is positive and the other is negative. These are their affine hull expressions respectively.

$$\begin{cases} \text{positive class } z_+ : \{U_+ v_+ + \mu_+\} \\ \text{negative class } z_- : \{U_- v_- + \mu_-\} \end{cases} \tag{20}$$

Because the optimal separating hyperplane is the perpendicular bisector plane of the line connecting the closest points in two classes, we want to find the line just solving:

$$\min_{v_+ v_-} ||(U_+ v_+ + \mu_+) - (U_- v_- + \mu_-)||^2 \tag{21}$$

Suppose that $\mathrm{U} \equiv [\ U_+ \quad -U_-\ ]$, $\mathrm{v} \equiv [\ v_+ \quad v_-\ ]^T$, the above formula can be simplified as:

$$\min_{\mathrm{v}} ||\mathrm{Uv} - (\mu_- - \mu_+)||^2 \tag{22}$$

Through deriving, we can get the solution $\mathrm{v} = \left(\mathrm{U}^T\mathrm{U}\right)^{-1} \mathrm{U}^T(\mu_- - \mu_+)$, and then put it back into the above formula to obtain the shortest distance of the closest pair of points. Seeking the perpendicular bisector plane, we have

$$w = \frac{1}{2}(z_+ - z_-) = \frac{1}{2}(I - P)(\mu_+ - \mu_-) \tag{23}$$

$P = \mathrm{U}\left(\mathrm{U}^T\mathrm{U}\right)^{-1}\mathrm{U}^T$. I represents the identity matrix. $(z_+ - z_-)$ means the closest points distances and it is replaced by the above formula. Besides, the points lying in the hyperplane satisfy $\langle w, z_i \rangle + b = 0$, so b can be obtained by

$$b = -\frac{w^T(z_+ + z_-)}{2} \tag{24}$$

At this point, we complete the training of the LMC.

3.2. **The construction of multi-class classifiers.** There are two common ways building multi-class classifiers based on binary classifiers: one-vs-one [18] and one-vs-all [15]. Compared with one-vs-all algorithm, one-vs-one algorithm consumes more time but has better fault tolerance and recognition effects. The reason is presented in the next part. So, in this paper, we adopt one-vs-one algorithm to build our multi-class classifiers.

Every two classes of our weed seeds need to build a binary classifier according to the method mentioned above. Suppose we have m classes in this experiment in all, $m(m-1)/2$ binary classifiers are needed, which make up our multi-class classifiers.

3.3. **The prediction of test samples.** For an unknown test sample, we classify the sample through our multi-class classifiers, which means a sample is classified $m(m-1)/2$ times. And in each classification, we add a vote for the class which is determined by the classifier as the sample belongs to. After finishing all classifications, we only need to count which class has the highest votes, that is our final classification result.

It is obvious that all of classifiers come together to obtain the final result. The role of each classifier is equal, and no single classifier plays a decisive role. Therefore, this method reduces the impact of accidental factors, and has better fault tolerance and recognition effect.

4. Experiment.

4.1. **Feature extraction through Saak transform and F-test.** In our experiment, we have 211 classes of weed seeds at all which the number of each kind is mostly about 40. The images' dimensions are $64 \times 64 \times 3$. For each image, after multi-stage Saak transform, we can get lots of Saak coefficients in each stage.

Next, we use F-test method to calculate the scores of the coefficients in the last stage and sort them in descending order. The first n coefficients are selected to form the feature vector for each image. The recognition effect is closely related to the value of n that we discuss in detail in next part 4.3. Now, we set $n = 1000$.

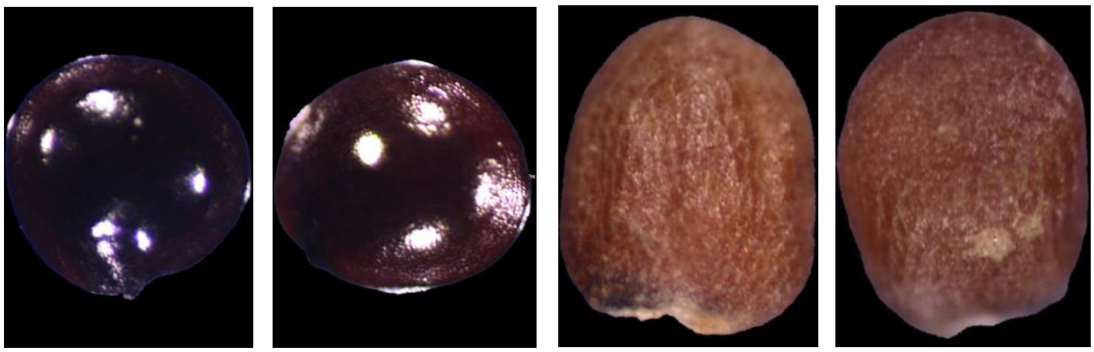
4.2. **Classification through the LMC.** Due to the fact that the image number of each kind of weed seeds is too simple, we adopt 10-fold cross-validation to build the multi-class classifiers and test the classification result. For each class, we divide the extracted features into 10 sets randomly. One of them is used for testing in turn, and the other nine are used to build the classifiers. Through this process, we finally complete the testing task for all image features of all classes. The samples used for training and testing and the result of test is shown in Table 1 below.

Table 1. Classification result

Total number of samples	Training samples number in one cross-validation	Testing samples number in one cross-validation	Maximum recognition rate	Average recognition rate
8910	8019	891	88.10%	86.37%

Misclassified images usually have a high degree of similarity. Some examples of misclassification are shown in Figure 2. How to further improve the recognition rate of these images is our main research topic in the future.

4.3. **Influence of the feature selection number on the classification results.** F-test helps us to understand the contribution of different features to the classification. After getting the feature sequence sorted by the score of F-test, how many features should we choose to increase the speed of calculation in the case of ensuring the recognition rate? To get the answer, we test the classification result of one set of samples which the feature number increases from 100 to 18000. The result is shown in Table 2.



(a) Amarla is classified to amarqa (b) Hrisca is classified to brasda

FIGURE 2. Error (left) and correct (right) classes images analysis

TABLE 2. Recognition rate based on different feature numbers

Feature number	100	500	1000	3000	5000	10000	18000
Recognition rate (%)	68.91	86.42	88.10	89.00	89.56	89.89	90.01

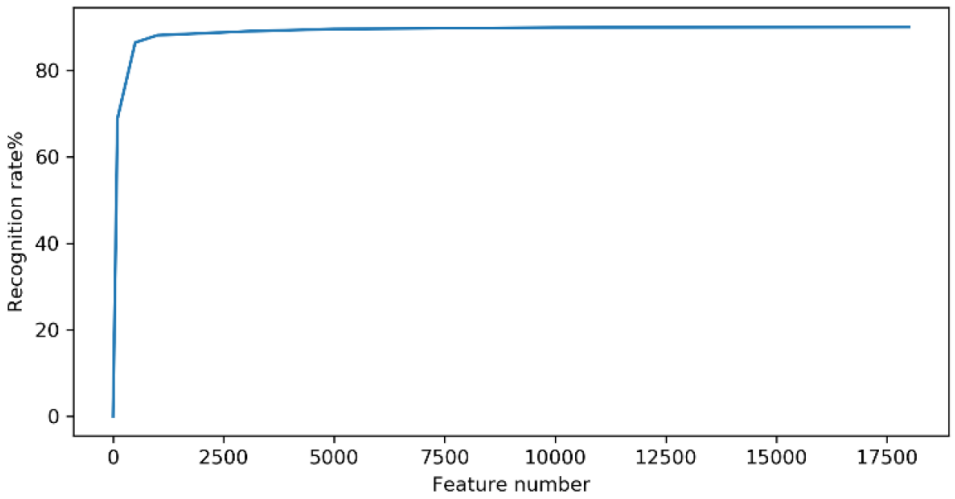


FIGURE 3. Recognition rate based on different feature numbers

And from Figure 3, we can see that the changing curve of recognition rate appears a turning point when the feature number is about 1000. The recognition rate rises sharply in the first 1000 range which proves that F-test is effective. That is also the reason why we set the features number $n = 1000$ in our experiment. If you want a higher recognition rate, you only need to use more features.

4.4. **Classification of weed seeds without feature extraction.** Finally, to verify the efficiency of Saak transform and F-test, in this part, we directly use the LMC to test the classification result of the original seed images instead of the extracted features. We keep other relevant conditions consistent. The average recognition rate is 79.53%.

In this case, we use all the original image data which the dimensions are $64 \times 64 \times 3$. After Saak transform, only 1000-dimensional data is used but the average recognition rate reaches 86.37%, which means our method is very effective.

5. **Conclusions.** In this paper, we introduce weed seeds classification based on Saak transform. It is a data-driven method, which is simple and quick to extract features and does not require a great many training samples, reducing the training time. F-test

is a good way to help us select more discriminative features and large margin classifier based on affine hulls is adopted to accomplish classification. In the experiment, this method completes the classification task of our 211 kinds of weeds seeds very well, which proves that it can provide technical support for the identification of weed seeds and the cultivation of crops in the agricultural field. This is also the significance of our research.

Acknowledgement. This research is supported by National Natural Science Foundation of China (Project 61202188).

REFERENCES

[1] R. H. Li and S. Qiang, Progresses and prospects in research of weed seed dispersal, *Acta Ecologica Sinica*, vol.27, no.12, pp.5361-5370, 2007.
[2] C. P. Cheng, J. F. Pan, K. Y. Wan et al., Research advances in the effects of rotation on cropland weeds, *Chinese Agricultural Science Bulletin*, vol.29, no.30, pp.1-9, 2013.
[3] N. Y. Zhu, X. Liu, Z. Q. Liu et al., Deep learning for smart agriculture: Concepts, tools, applications, and opportunities, *International Journal of Agricultural and Biological Engineering*, vol.11, no.4, pp.32-44, 2018.
[4] H. Lu, Z. G. Cao, Y. Xiao et al., Toward good practices for fine-grained maize cultivar identification with filter-specific convolutional activations, *IEEE Trans. Automation Science and Engineering*, vol.15, no.2, pp.430-442, 2018.
[5] C. DeChant, T. Wiesner-Hanks, S. Y. Chen et al., Automated identification of northern leaf blight-infected maize plants from field imagery using deep learning, *Phytopathology*, vol.107, no.11, pp.1426-1432, 2017.
[6] T. H. Chan, K. Jia, S. Gao et al., PCANet: A simple deep learning baseline for image classification?, *IEEE Trans. Image Processing*, vol.24, no.12, pp.5017-5032, 2015.
[7] Y. Chen, Z. Xu, S. Cai et al., A Saak transform approach to efficient, scalable and robust handwritten digits recognition, *arXiv:1710.10714*, 2017.
[8] C. C. J. Kuo and Y. Chen, On data-driven Saak transform, *arXiv:1710.04176*, 2017.
[9] P. M. Granitto, H. D. Navone, P. F. Verdes et al., Weed seeds identification by machine vision, *Computers & Electronics in Agriculture*, vol.33, no.2, pp.91-103, 2002.
[10] P. M. Granitto, P. A. Garralda, P. F. Verdes et al., Boosting classifiers for weed seeds identification, *Journal of Computer Science & Technology*, vol.3, no.1, pp.34-39, 2003.
[11] X. S. Wang and C. Cai, Weed seeds classification based on PCANet deep learning baseline, *Proc. of the 7th Asia-Pacific Signal and Information Processing Association Summit and Conference*, Hong Kong, China, pp.408-415, 2015.
[12] C. C. J. Kuo, The CNN as a guided multilayer RECOS transform, *IEEE Signal Processing Magazine*, vol.34, no.3, pp.81-89, 2017.
[13] C. C. J. Kuo, Understanding convolutional neural networks with a mathematical model, *Journal of Visual Communication and Image Representation*, vol.41, pp.406-413, 2016.
[14] K. P. Bennett and E. J. Bredensteiner, Duality and geometry in SVM classifiers, *Proc. of the 17th International Conference on Machine Learning*, Stanford, CA, pp.57-64, 2000.
[15] C. J. C. Burges, A tutorial on support vector machines for pattern recognition, *Data Mining and Knowledge Discovery*, vol.2, no.2, pp.121-167, 1998.
[16] C. Cortes and V. Vapnik, Support vector networks, *Machine Learning*, vol.20, no.3, pp.273-297, 1995.
[17] H. Cevikalp, B. Triggs, H. S. Yavuz et al., Large margin classifiers based on affine hulls, *Neurocomputing*, vol.73, no.16, pp.3160-3168, 2010.
[18] C. C. Chang and C. J. Lin, LIBSVM: A library for support vector machines, *ACM Trans. Intelligent Systems and Technology*, vol.2, no.27, pp.1-27, 2011.