Journal of Information Hiding and Multimedia Signal Processing ©2019 ISSN 2073-4212
Ubiquitous International Volume **10**, Number **4**, December **2019**

A Study on Combined CNN-SVM Model for Visual Object Recognition

Fengyu Gao[1,2], Jer-Guang Hsieh[1] and Jyh-Horng Jeng[3*]

[1]Department of Electrical Engineering
I-Shou University
No.1, Sec.1, Syuecheng Road, Kaohsiung City, 84001, Taiwan
isu10702050d@cloud.isu.edu.tw; jghsieh@isu.edu.tw

[2]Key Laboratory of Nondestructive Testing (Fuqing Branch of Fujian Normal University)
Fujian Province University
Fuqing, 350300, China
isu10702050d@cloud.isu.edu.tw

[3*]Department of Information Engineering
I-Shou University
No.1, Sec.1, Syuecheng Road, Kaohsiung City, 84001, Taiwan
Corresponding author: jjeng@isu.edu.tw

Received August 2019; revised Nobember 2019

Abstract. *In this paper, a model combining convolutional neural network (CNN) and support vector machine (SVM) as a classifier is studied. A traditional CNN model is used to train a dataset, which is then regarded as a pre-trained model. We then extract the feature vectors from the outputs of different layers and feed them to SVM to perform the classification task. The main issue of this study is to examine which layers of a traditional CNN are more suitable for SVM in terms of recognition accuracy, generalization ability, and computation complexity. Experimental results, tested on the well-known datasets MNIST, Fashion-MNIST, and CIFAR-10, demonstrate that feature maps extracted from the last convolution layer produce the best performances in both accuracy and generalization.*
Keywords: Convolutional Neural Network (CNN), Support Vector Machine (SVM), MNIST, Fashion-MNIST, CIFAR-10.

1. **Introduction.** Visual object recognition is a challenging work in the field of computer vision for many years, which can be applied to various fields such as automated image organization [1], visual search [2], and image recognition and classification [3, 4]. It is still an interesting and commercially valuable area. Various methods for recognition have been intensively studied, e.g., K-nearest-neighbors (KNN) [5], Deep Neural Network (DNN) [6], Support Vector Machine (SVM) [7], Convolutional Neural Network (CNN) [8], and etc. Among the methods, CNN is one of the most encouraging techniques.

Inspired by the biology of a special part of the retinal cells which are sensitive to light, David H. Hubels team published a study of the visual cortex in cats. In 1980, based on this concept of receiving light areas, Kunihiko Fukushima et al [9] proposed a perceptron model with different architecture which can be regarded as the original model of CNN. The earliest CNN model employed in image recognition is developed by LeCun in 1998 [10]. Until 2012, Alex Krizhevsky designed a CNN model, namely AlexNet [11], to take part in a competition and defeat SVM by great advantage.

SVM can be used as a learning machine based on novel statistical learning techniques, which was developed mainly by Vapnik [12]. It is one of the most robust and reliable algorithm for classification and regression tasks. Real world applications include face detection [13], handwriting recognition [14], image classification [15], bio-information [16], and etc. One of typical CNN applications is classification. The ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [17] evaluates algorithms for object detection and image classification at large scale.

To improve the performance of CNN models, some researchers used CNN as a feature extractor and adopt SVM as the classifier [18]. This model is referred to as CNN-SVM model. Such models do exhibit higher recognition accuracy in digital handwritten images [19, 20, 21]. In their study, the features are all extracted from the dense (fully connected) layer which is located right before the output layer. One example was described in [19] in which experiments on the MNIST datasets with distortions achieved the accuracy of 99.81%. However, the outputs of the dense layer are not a proper candidate for the purpose of visual object recognition.

In this paper, we propose that the CNN-SVM model with features extracted from the last convolution layer performs the best. Experiments are conducted on MNIST, Fashion-MNIST, and CIFAR-10 datasets. The results demonstrate that the proposed model exhibits higher accuracy and better generalization ability.

2. CNN and SVM Models.

2.1. CNN model.

CNN is a multi-layer neuron network which can be used as a supervised learning machine. The key layer type is the convolutional layer (conv layer), usually followed by a pooling layer, which may be regarded as a feature extractor. The convolution algorithm uses the kernel to scan the two-dimensional images or feature maps and extract the features. The operation is exactly the same as filtering process in image processing where the kernel is the filter or mask. The outputs of conv layer are a sequence of feature maps, which are presented in a two-dimensional way. The conv layer is not fully connected to the previous layer, which is intentionally designed to mimic the image filtering operation.

The difference among CNNs is in their structures such as number of layers and number of feature maps. A typical structure of CNN is shown in Figure 1, where 'C' stands for a conv layer, 'P' stands for pooling layer, 'F stands for flatten layer, and 'D' stands for dense layer. In this model, there are 2 conv-pooling pairs. The following flatten layer rearrange the 2-dim feature maps into 1-dim vector. And finally, a fully connected (dense) layer and a output layer are connected to the flatten layer.

2.2. SVM model.

SVM is also a supervised learning machine that usually performs classification (SVC) or regression (SVR) tasks. In this study, we only consider SVC.

In the following development, let $K(x, x')$ be a kernel defined on $X \times X$. In this study, we adopt the following two kernels:

$$\text{Euclidean inner product: } K(x, x') = \langle x, x' \rangle$$
$$\text{Gaussian kernel: } K(x, x') := exp(-\gamma \parallel x - x' \parallel^2),\ \gamma > 0.$$

Consider first the binary classification problems. Let $X \subseteq \Re^n$ and $Y := \{1, -1\}$. Suppose we are given the training dataset

$$S := \{(x_i, y_i)\}_{i=1}^{l} \subseteq X \times Y \subseteq \Re^n \times \{1, -1\}. \tag{1}$$

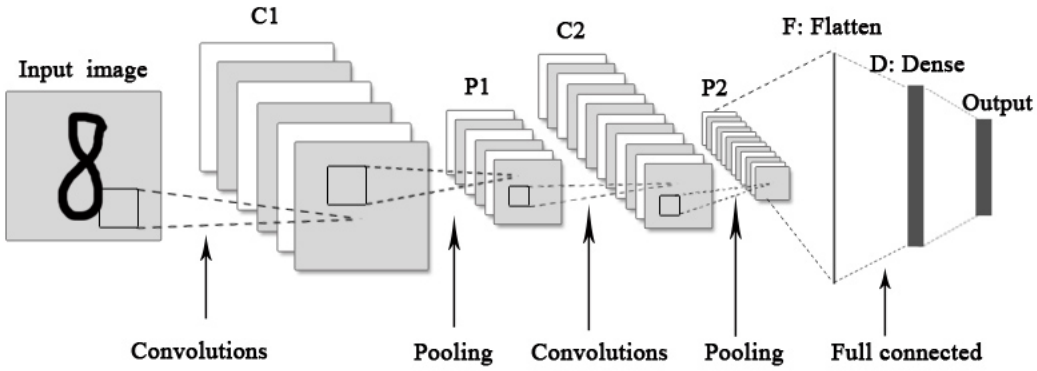

FIGURE 1. Typical structure of CNN model

Given the training dataset (1), the SVC can be obtained by solving the following optimization problem:

$$\text{maximize } \sum_{i=1}^{l} z_i - \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l} z_i z_j y_i y_j K(x_i, x_j) \tag{2a}$$

$$\text{subject to } \sum_{i=1}^{l} z_i y_i = 0 \text{ and } 0 \le z_i \le C \text{ for all } i = 1, 2, ..., l. \tag{2b}$$

In (2), the regularization (or smoothing) parameter $C > 0$ controls the tradeoff between complexity of the machine and the number of non-separable points.

Suppose z_i^* solves the problem (2). Then the optimal discriminant function is given by (3):

$$f^*(x) = \sum_{i=1}^{l} z_i^* y_i K(x_i, x) + b^*, \tag{3}$$

where b^* is the optimal bias term.

The simulation examples to be presented are all multi-class classification problems. A natural approach to a multi-class classification problem is to reduce this multi-class classification problem to several binary classification problems. To this end, there are two popular methods for multi-class classification problems using SVC. These are one-versus-rest (OVR) method and one-versus-one (OVO) method. In this study, we simply choose one of them, namely the OVR method. Similar results can be obtained using the OVO method.

Consider the m-class classification problem with label set $Y := \underline{m} := \{1, 2, ..., m\}$. In the OVR method, it is required to design m binary classifiers. For each $j \in \underline{m}$, we label all training examples having $y_i = j$ with 1 and $y_i \neq j$ with -1 during the training of the jth classifier. The final decision is to assign $x \in X$ with class j if the following equation (4) is satisfied:

$$j(x) = arg \max_{k=1}^{m} f_k(x). \tag{4}$$

2.3. **Combined CNN-SVM model.** In the combined model, we first train a traditional CNN and regard it as a pre-trained model. We then extract features from the outputs of certain layer for the following SVM to perform the classification. Given a pair of training set and testing set, the training and testing steps for the combined model are given as follows:

(1) Train the CNN using the training set and regard it as a pre-trained model;
(2) Feed the training set to the CNN model and take the outputs of some layer with the corresponding labels to form the new training set;
(3) Feed the testing set to the CNN model and take the outputs of same layer with the corresponding to labels form new testing set;
(4) Use the new sets to train and test the SVM.

For the architecture shown in Figure 1, we could take the outputs as features from C1-layer. However, P1-layer is the subsampling of C1-layer, thus it has same visual structures. Therefore we extract the features from P1-layer and identify this model as P1-SVM. Since P2-layer also has the same visual structures as C2-layer and F-layer is just the flattened vector of P2-layer, we extract the features from F-layer and identify this model as F-SVM.

3. **Analyzing the Features from Different Layers.** As mentioned, the features can be extracted from the outputs of P1-layer, F-layer, or D-layer as illustrated in Figure 1. In this section, we will provide a comprehensive comparison among these layers.

In image processing, filtering operation is one of the most commonly used techniques to extract specific features such as edges, corners, low/high frequency information, and etc. The operation is illustrated in Figure 2, where f is the input and g is the output image. The 3×3 (or $m \times m$) block is usually referred to as the mask, kernel or filter.

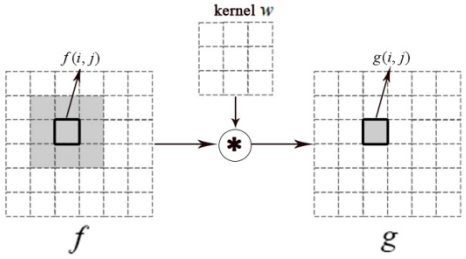

FIGURE 2. Filtering operation (convolution) of an image f and the kernel w

The resulting value of $g(i, j)$ can be obtained by (5):

$$g(i,j) = \sum_{s=-1}^{1} \sum_{t=-1}^{1} f(i+s, j+t) \times w(s,t). \tag{5}$$

As the indices i and j traverse the whole image, the operation is done. This operation is just similar to the traditional convolution operation.

In this convolution operation, there are two very important characteristics. One is that only nearby neighbors corresponding to $f(i, j)$ are considered. This is because neighboring pixels provide spatial correlation and visual effects. The other one characteristic is that the weights (values) in the filter are fixed throughout the whole image. This means same measurements are considered for the whole image. Therefore filtering can extract important information or change the appearance but it still preserves the main structure

and visual effects of the original image. These characteristics are very suitably served as"feature vectors" for visual object recognition.

In terms of the neural network architecture, the filtering operation can be depicted similar to C1-layer in Figure 1. Note that there is one more bias connection which is required for traditional network architecture. Here, the kernel can be regarded as the weights of C1-layer. This is exactly the same as the so-called conv layers in a CNN.

Consider the conv layer C1 in Figure 1. There are 6 feature maps as the outputs of C1-layer where each one is obtained from the filtering operation. Therefore each one preserves the visual structures of the original image. The successive layer to P1 is the subsampling operation which reduces the dimensionality but still preserves the main visual structure. It is important to mention that these 6 maps possess different features or information but still possess similar visual structure of the original image. Therefore they are more suitable to serve as the feature vectors for later classification tasks such as visual object recognition.

Next in the conv layer C2, there are there are 6 maps as the inputs and 12 maps as the outputs. We observe one"pixel" of one output map. For each input map, there is one filter operation, each corresponding to one kernel. Adding up the 6 filtering results, together with one bias, results in that "pixel" value in the observed output map. As shown, the 6 filters operate on the 6 different input maps on the same position with neighbors corresponding to the same "pixe" position in the observed output map. Since the each input map possess similar visual structure to the original image, the observed resulting output map also possesses similar visual structure. And thus, all the 12 output maps all possess similar visual structure. As a consequence, the outputs of conv layers are more suitable to serve as the feature vectors for visual object recognition, although the outputs may exhibit higher level, lower level, or even abstract visual effects.

In contrast to conv layers, features extracted from dense layer (hidden layer) will not be good candidates for visual object recognition. As shown in Figure 1, F-layer rearranges the outputs of P2-layer to a 1-dim vector which is then fed to the fully connected layer D. Such a setup first breaks up the structure of the 12 individual yet highly correlated maps from P2-layer. Second, due to the fully connection mixing the whole features together, all the meaningful visual properties are corrupted. They are just signals rather than meaningful 2-dim features with spatial correlations. So, features extracted from dense layer will not be a good candidate to serve as feature vectors for visual object recognition.

Since the original CNN is already well trained, we assert that, for following classification tasks, features extracted from F-layer should outperform those extracted from D-layer in terms of recognition accuracy and generalization ability. Moreover, the proposed idea also saves a lot of computations due to the lack of the full connection computation.

4. **Experimental Results.** In this study, the simulations are conducted using the platform consisting of Windows10 and Intel(R) Core (TM) i5-8265 @1.6GHz1.8 CPU with 8 GB RAM. The software programming environment is Python 3.5 and the deep learning framework is Keras with TensorFlow as the backend.

The experimental results are evaluated using recognition accuracy and generalization ability. The accuracy is defined as the ratio of the number of examples correctly classified to the total number of examples. The standard deviation of the accuracies from cross validation is strongly related to the generalization ability, which is defined as (6):

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2}, \tag{6}$$

where N is the number of folds for cross validation, x_i is the accuracy of the ith accuracy, and μ is the average of the N accuracies.

In the study, we compare the three models F-SVM, D-SVM, and P1-SVM together with the original CNN model. Three datasets with different style, i.e., MNIST, Fashion-MNIST, and CIFAR-10, are used to compare the performances.

4.1. **Experiments on MNIST.** MNIST is a dataset of handwritten digits consisting of a training set of 60,000 examples and a testing set of 10,000 examples. Each example is a 28×28 grayscale image, associated with a label from 10 classes. It is a commonly used dataset established by Yann LeCun et al in 1998. In the experiments, we adopt 7-fold cross validation. That is the total 70,000 images are divided into 7 groups, 6 for training and the remaining for testing adding up to 7 experiments. The recognition accuracy or accuracy for short used in the experiments is calculated from the testing results.

The network structure of CNN is shown in Figure 1 with following details:

(0) Input image: 28×28×1
(1) C1: conv layer, kernel size 5×5, 6 filters, stride 1
(2) P1: pool layer, size 2×2, max pooling
(3) C2: conv layer, kernel size 5×5, 12 filters, stride 1
(4) P2: pool layer, size 2×2, max pooling
(5) F: flatten to a vector, 192-dim
(6) D: dense layer, 64 hidden neurons
(7) O: output layer, 10 neurons

In the CNN training, we use categorical cross-entropy as the objective function. The training epoch is set to 100. For the first pair of training and testing sets, i.e., the first-fold data, after the CNN is trained we test the CNN and report the accuracy. Then we regard this CNN as a pre-trained model. For the SVM, we select the RBF as the kernel function. The regularization parameter C is set to 10 and the kernel parameter γ is set to the reciprocal of the dimension of the feature vectors.

The results of the first pair of training and testing sets are listed in the first column of Table 1, i.e., cv1 field. As shown, the proposed F-SVM achieves 99.25% accuracy which is much better than that of the original CNN, 98.86%. Also, F-SVM outperforms D-SVM and P1-SVM with accuracy of 90.39% and 99.05%, respectively. It means that among the CNN-SVM models, features extracted from the F-layer performs the best.

The results of all the 7 folds are listed in Table 1 (cv1-cv7). The mean value (mean) and the standard deviation (σ) are also presented. It can be seen that F-SVM produces higher accuracy than the traditional CNN, D-SVM, and P1-SVM in all the 7 folds. Moreover, F-SVM also exhibits lowest standard deviation of the 7-fold cross validation. This reveals that F-SVM tends to have better generalization ability.

The misclassified count of CNN and CNN-SVMs out of 10,000 testing data is shown in Table 2.

Table 1. Accuracy of CNN and CNN-SVMs on MNIST

acc(%)	cv1	cv2	cv3	cv4	cv5	cv6	cv7	mean	σ
CNN	98.86	98.95	98.74	98.70	98.71	98.73	98.91	98.80	0.0962
F-SVM	99.25	99.17	99.17	99.08	99.19	99.26	99.17	99.18	0.0555
D-SVM	90.39	93.34	93.87	92.76	93.48	96.13	92.30	93.18	1.6049
P1-SVM	99.05	99.18	98.87	98.85	98.82	99.00	99.06	98.98	0.1231

TABLE 2. Misclassified count of CNN and CNN-SVMs on MNIST

count	cv1	cv2	cv3	cv4	cv5	cv6	cv7	mean	σ
CNN	114	105	126	130	129	127	109	120.00	114
F-SVM	75	83	83	92	81	74	83	81.57	75
D-SVM	961	666	613	724	652	387	770	681.86	961
P1-SVM	95	82	113	115	118	100	94	102.43	95

Among the 3 CNN-SVM models, D-SVM performs the worst, even worse than the traditional CNN. This is because we set the same parameters for the 3 SVMs. For the sake of fair comparison, we try to seek a set of parameters that is most suitable for D-SVM. Out of many trial and error, we found that linear kernel and C=0.1 is the best combination for D-SVM and it produces the best results which are shown in Table 3, the 2nd record. However, the results are still inferior to those of the proposed F-SVM.

TABLE 3. D-SVM results using different parameter settings

acc(%)	cv1	cv2	cv3	cv4	cv5	cv6	cv7	mean	σ
F-SVM RBF, C=1	99.25	99.17	99.17	99.08	99.19	99.26	99.17	99.18	0.0555
D-SVM RBF, C=10	90.39	93.34	93.87	92.76	93.48	96.13	92.30	93.18	1.6049
D-SVM LINEAR, C=0.1	98.91	98.97	98.77	98.90	98.72	98.91	98.95	98.88	0.0868

4.2. **Experiments on Fashion-MNIST.** Fashion-MNIST, provided by a Germany company fashion Zalando, is a dataset of fashion objects consisting of a training set of 60,000 examples and a testing set of 10,000 examples, same as MNIST. Each example is a 28×28 gray scale image, associated with a label from 10 classes. In this experiment, we adopt 7-fold cross validation. The SVM parameter settings are all the same as in section 4.1. The accuracy and misclassified count out of 10,000 examples are shown in Table 4 and Table 5, respectively. Again, the proposed F-SVM is superior to the other 2 CNN-SVM models.

TABLE 4. Accuracy of CNN and CNN-SVMs on Fashion-MNIST

acc(%)	cv1	cv2	cv3	cv4	cv5	cv6	cv7	mean	σ
CNN	89.99	89.49	90.86	90.02	90.26	89.72	89.81	90.02	0.4104
F-SVM	91.48	91.27	91.89	91.65	91.36	91.31	91.12	91.44	0.2399
D-SVM	90.73	90.06	90.94	90.62	90.57	90.18	90.32	90.49	0.2911
P1-SVM	91.22	90.32	90.83	90.71	90.84	90.32	90.15	90.63	0.3507

TABLE 5. Misclassified count of CNN and CNN-SVMs on Fashion-MNIST

count	cv1	cv2	cv3	cv4	cv5	cv6	cv7	mean	σ
CNN	1001	1051	914	998	974	1028	1019	997.86	1001
F-SVM	852	873	811	835	864	869	888	856.00	852
D-SVM	927	994	906	938	943	982	968	951.14	927
P1-SVM	878	968	917	929	916	968	985	937.29	878

4.3. **Experiments on Fashion-MNIST.** The CIFAR-10 dataset is collected by Alex Krizhevsky et al, consisting of a training set of 50,000 examples and a testing set of 10,000 examples. Each image is of size 32×32×3 (3-channel color) in 10 classes. The total 60,000 sets are partitioned into 6 folds for cross validation. The pre-trained CNN model is described as follows:

(0) Input image: 32×32×3
(1) C1: conv layer, kernel size 5×5, 6 filters, stride 1d
(2) P1: pool layer, size 2×2, max pooling
(3) C2: conv layer, kernel size 5×5, 12 filters, stride 1
(4) P2: pool layer, size 2×2, max pooling
(5) F: flatten to a vector, 300-dim
(6) D: dense layer, 64 hidden neurons
(7) O: output layer, 10 neurons

The SVM parameter settings are all the same as in section 4.1. The accuracy and misclassified count out of 10,000 examples are shown in Table 6 and Table 7, respectively. Again, experimental results on CIFAR-10 further justifies that the proposed F-SVM performs the best in accuracy as well as in generalization ability.

Table 6. Accuracy of CNN and CNN-SVMs on CIFAR-10

acc(%)	cv1	cv2	cv3	cv4	cv5	cv6	mean	σ
CNN	61.80	62.42	64.47	62.82	63.29	61.68	62.75	0.9497
F-SVM	66.52	66.83	68.76	67.03	68.03	67.35	67.42	0.7619
D-SVM	63.40	63.60	65.14	63.90	64.54	63.95	64.09	0.5883
P1-SVM	59.17	59.41	61.66	60.36	60.94	61.37	60.48	0.9373

Table 7. Misclassified count of CNN and CNN-SVMs on CIFAR-10

count	cv1	cv2	cv3	cv4	cv5	cv6	mean	σ
CNN	3820	3758	3553	3718	3671	3832	3725	94.97
F-SVM	3348	3317	3124	3297	3197	3265	3258	76.19
D-SVM	3660	3640	3486	3610	3546	3605	3591	58.83
P1-SVM	4083	4059	3834	3964	3906	3863	3952	93.73

5. **Conclusions.** This paper has proposed a combined model in which a well-trained CNN is used as a pre-trained model and new dataset are extracted from the outputs of some layer. The new dataset is then used for training the following support vector classifier. We proposed the model with features which are extracted from the last convolution layer. This model outperforms the other two models which extract the data from the dense layer and the 1^{st} conv layer, respectively, in terms of accuracy and generalization ability using cross validation. Experiments were conducted on three real-world datasets: MNIST, Fashion-MNIST, and CIFAR-10. This comparative study on the combined model investigates on how the new data extracted from different layers of the CNN affect the performance of SVM. Our results indicated that for combined model on which the features are extracted from the last convolution layer is quite a promising method for image recognition and classification.

Acknowledgment. The research reported here was supported by the Ministry of Science and Technology, Taiwan, under grants MOST 108-2221-E-214-026 and ISU-107-01-03A.

REFERENCES

[1] A. I. Dell, J. A. Bender, K. Branson, I. D. Couzin, G. G. Polavieja, L. P. J. J. Noldus, A. Perez-Escudero, P. Perona, A. D. Straw, M. Wikelski, and U. Brose, Automated image-based tracking and its application in ecology, *Trends in Ecology and Evolution*, vol. 29, no. 7, pp. 417–428, 2014.

[2] J. M. Wolfe and T. S. Horowitz, Five factors that guide attention in visual search, *Nature Human Behaviour*, vol. 1, no. 3, DOI: 10.1038/s41562-017-0058, 2017.

[3] Q. Feng, Q. Zhu, L. Tang, and J. S. Pan, L1 plus L2 sparse parameter for image recognition, *Optik-International Journal for Light and Electron Optics*, vol. 126, issue 23, pp. 4078–4082, 2015.

[4] J. S. Pan, Q. Feng, L. Yan, and J. Yang, Neighborhood feature line segment for image classification, *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 25, no. 3, pp. 387–398, 2015.

[5] M. Potamias, F. Bonchi, A. Gionis, and G. Kollios, K-nearest neighbors in uncertain graphs, *Proceedings of the VLDB Endowment*, vol. 3, issue 1-2, pp. 997–1008, 2010.

[6] D. C. Ciresan, A. Giusti, L. M. Gambardella, and J. Schmidhuber, Mitosis detection in breast cancer histology images with deep neural networks, *Proceedings of the International Conference on Medical Image Computing and Computer-assisted Intervention*, vol. 8150, pp. 411–418, 2013.

[7] D. Decoste and B. Schlkopf, Training invariant support vector machines, *Machine Learning*, vol. 46, issue 1-3, pp. 161–190, 2002.

[8] H. Shin, H. Roth, M. Gao, L. Lu, Z. Xu, I. Nogues, J. Yao, D. Mollura, and R. Summers, Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning, *IEEE Transactions on Medical Imaging*, vol. 35, no. 5, pp. 1285–1298, 2016.

[9] K. Fukushima and S. Miyake, Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position, *Biological Cybernetics*, vol. 36, no. 4, pp. 193–202, 1980.

[10] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, Gradient-based learning applied to document recognition, *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[11] A. Krizhevsky, I. Sutskever, and G. E. Hinton, Imagenet classification with deep convolutional neural networks, *Advances in Neural Information Processing Systems*, vol. 25, no. 2, pp. 1097–1105, 2012.

[12] V. Vapnik, *The Nature of Statistical Learning Theory*, Springer-Verlag, New York, 2013.

[13] E. Osuna, R. Freund, and F. Girosi, Training support vector machines: An application to face detection, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 130–136, 1997.

[14] J. X. Dong, A. Krzyzak, and C. Y. Suen, Fast SVM training algorithm with decomposition on very large data sets, *IEEE Transactions Pattern Analysis Machine Intelligence*, vol. 27, issue 4, pp. 603–618, 2005.

[15] H. Zhang, A. C. Berg, M. Maire, and J. Malik, SVM-KNN: Discriminative nearest neighbor classification for visual category recognition, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 2126–2136, 2006.

[16] G. C. Sahoo, M. R. Dikhit, and P. Das, Functional assignment to JEV proteins using SVM, *Bioinformation*, vol. 3 , no. 3, pp. 1–7, 2008.

[17] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and F. F. Li, Imagenet large scale visual recognition challenge, *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[18] C. J. C. Burges, A tutorial on support vector machines for pattern recognition, *Data Mining and Knowledge Discovery*, vol. 2, issue 2, pp. 121–167, 1998.

[19] X. X. Niu and C. Y. Suen, A novel hybrid CNNSVM classifier for recognizing handwritten digits, *Pattern Recognition*, vol. 45, issue 4, pp. 1318–1325, 2012.

[20] F. Lauer, C. Y. Suen, and G. Bloch, A trainable feature extractor for handwritten digit recognition, *Pattern Recognition*, vol. 40, no. 6, pp. 1816–1824, 2007.

[21] E. Mohamed, R. Maalej, and M. Kherallah, A new design based-SVM of the CNN classifier architecture with dropout for offline Arabic handwritten recognition, *Procedia Computer Science*, vol. 80, pp. 1712–1723, 2016.