INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
Online ISSN 1841-9844, ISSN-L 1841-9836, Volume: 15, Issue: 1, Month: February, Year: 2020
Article Number: 1007, https://doi.org/10.15837/ijccc.2020.1.3712

CCC Publications
AGORA UNIVERSITY PRESS

A Stochastic Max Pooling Strategy for Convolutional Neural Network Trained by Noisy Samples

S. Sun, B. Hu, Z. Yu, X.N. Song

Shuai Sun
Department of Electrical Engineering,
Zhengzhou Electric Power College,
Zhengzhou 450000, China
sunshuai@126.com

Bin Hu
Department of Electrical Engineering,
Zhengzhou Electric Power College,
Zhengzhou 450000, China
13949101656@163.com

Zhou Yu
Department of Electrical Engineering,
Zhengzhou Electric Power College,
Zhengzhou 450000, China
232689536@163.com

Xiaona Song*
College of Mechanical Engineering,
North China University of Water Resources and Electric Power,
Zhengzhou 450000, China
*Corresponding author: songxiaona1@126.com

Abstract

The deep convolutional neural network (CNN) has made remarkable progress in image classification. However, this network performs poorly and even cannot converge in many actual applications, where the training and test samples contain lots of noises. To solve the problems, this paper puts forward a network training strategy based on stochastic max pooling. Unlike the traditional max pooling, the proposed strategy first ranks all the values in each receptive field, and then selects a random value from the top-n values as the pooling result. Compared with common pooling methods, stochastic max pooling can limit the pooling selection to a larger value that represents the main information of the pooling area which reduces the chance of introducing noises into the network, and enhances the robustness of extracting noisy image features. Experimental results show that the CNN used stochastic max pooling Strategy can converge better than traditional CNN and classified noisy images much more accurately than traditional pooling methods.

Keywords: image classification, deep learning, convolutional neural network (CNN), stochastic max pooling.

1 Introduction

In recent years, deep learning has become a research hotspot of artificial intelligence (AI). This technology has been successfully applied in various field, such as image classification [1, 2, 9], object detection [3, 7, 10, 11], and image segmentation [4, 5]. One of the most important deep learning models is the deep convolutional neural network (CNN) [6]. With special structural elements (e.g. pooling, convolution and receptive field), the deep CNN enjoys immense popularity in image processing. Many scholars have attempted to further improve the excellent performance of the deep CNN in terms of function and structure.

The functional improvement either modifies or increases network functions, enhancing the network ability to achieve specific functions. For instance, Nair [8] replaced the traditional sigmoid function with rectified linear unit (ReLU) function, leading to better network convergence. Srivastava [13] put forward the dropout function, which greatly suppresses network overfitting than traditional methods. Wen [14] designed and implemented the intermediate loss function, and realized more accurate classification based on sample correlations than the traditional softmax loss function.

The structural improvement aims to express features in a better way through structural adjustment to the network. Currently, many scholars strive to obtain richer feature information by increasing the depth and width of the network. For example, Krizhevsky [6] proposed an eight-layer CNN called the AlexNet, which won the Large Scale Visual Recognition Challenge (ILSVRC) 2012. Simonyan [12]further increased the number of network layers to 19, and achieved 92.7% top-5 test accuracy in ImageNet, 55% higher than that of AlexNet. In ILSVRC 2015, the 152-layer ResNet [5] further improved the top–5 test accuracy to 95.7%.

However, the deep CNN provides no specific solution to problems like noisy images. In actual engineering, both the training and test samples contain lots of noises under the effects of lighting or imaging device. Despite being capable of processing images, the special structure of the CNN cannot eliminate the effects of noises on the network. If the deep CNN is directly applied to feature extraction, it will be inevitable for the extracted features to contain noises.

In the deep CNN, the key to feature extraction is the pooling layer. This layer extracts features for the second time, making them invariant to displacement, scaling and deformation. The most popular pooling method is max pooling, which takes the maximum value in each receptive field. Many experiments have found that this method can easily retain the noise (usually the maximum value) of the original image, and amplify the noise effects. This phenomenon is particularly obvious in max pooling of a highly noisy image. Unable to differentiate between noises and normal signals, max pooling does not apply to processing noisy images.

To solve the above problem, this paper puts forward a deep CNN based on stochastic max pooling. Unlike max pooling, the stochastic max pooling ranks the values in each receptive field and selects one from the top-n values as the pooling result. This approach reduces the chance of taking a noise as the feature. Like dropout structure, the network structure based on stochastic max pooling can be viewed as a set of n sub-models, and thus have a high accuracy.

The rest of the paper is organized as follows. Section 2 reviews recent pooling methods. Section 3 introduces our method of stochastic max pooling. Experiments are discussed and evaluated in Section 4. A summary in Section 5 concludes this paper.

2 Literature review

In the deep CNN, the pooling layer plays a critical role in feature extraction. In this layer, the convoluted features are further extracted and reduced dimensionally. Meanwhile, their invariance to displacement, scaling and deformation is enhanced.

There are three commonly used pooling methods, including average pooling, max pooling and stochastic pooling [15]. The average pooling outputs the average of all values in each receptive field. This pooling method reflects the overall features of the receptive field.

The max pooling takes the maximum value in the respective field as its result, highlighting the most salient feature of each receptive field. Many experiments have confirmed the superiority of max pooling over average pooling. However, max pooling only considers the value with the strongest response, ignoring all the other values. Thus, the effects of the low-response values on the network are neglected in max pooling.

To solve the problem, Zeiler [15] presented the stochastic pooling on International Conference on Learning Representations (ICLR) 2013. This pooling method randomly selects a value from each receptive field at a certain probability. With such randomness, the network structure based on stochastic pooling is equivalent to a set of multiple sub-models, similar to that of dropout [13]. Each sub-model is a network that outputs a value of the corresponding receptive field. Nevertheless, a stochastic pooling network may treat a small value in the receptive field as the pooling result. Such a small value cannot reflect the typical features of the receptive field. What is worse, the small value could become a noise in feature transmission, and suppress the network performance.

In practical applications, the images often contain a lot of noises. When the image contains a lot of noise, noise pixels may appear in the pooled area, and these pixels are often very large. Therefore, noise may be introduced into the network through maximum pooling, thus reducing the performance of the network. In this case, the stochastic pooling result might be a small value, which is not a good reflection of the essential characteristics of this region. To solve the problem, this paper combines the max pooling and stochastic pooling into a hybrid strategy called the stochastic max pooling. If applied to a noisy image, the proposed strategy can extract the primary features by selecting the value with high response and filtering out the small values, and enhance the stochasticity of the network, turning the network into a set of sub-models. In this way, the network performance can be enhanced despite the noises in the training samples.

3 Design of stochastic max pooling

3.1 Deep CNN

The CNN is a deep learning model that can automatically learn and extract features from data. The model greatly outshines the traditional methods in generalization. It has been successfully applied in the fields of pattern classification, object detection and object recognition. Being a multi-layer supervised learning network, the CNN mainly consists of input layer, hidden layers and output layer. The network structure is optimized through error backpropagation to solve unknown parameters. The hidden layers, including alternating layers of convolution and down-sampling modules, are the cornerstone of the CNN. During the operation, the original image is imported to the CNN via the input layer, subjected to convolution and pooling, and then outputted via the output layer.

In each convolution layer, every neuron in the feature map is connected to a local receptive field on the previous layer, and the local features are extracted through convolution. In general, the convolution operation can be expressed as:

$$x_j^l = f(\sum_{i \in M_j} x_i^{l-1} \times k_{ij}^l + b_j^l) \tag{1}$$

where, l is the serial number of the layer; k is the convolutional kernel; M_j is a receptive field of the input layer; b is bias.

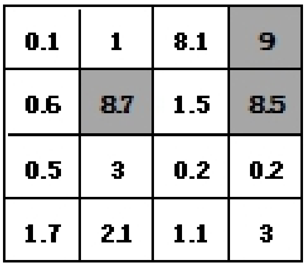
In each pooling layer, the pooled feature map has the same number of neurons as the inputted feature map, but its size is only 1/n of the former (n is the pooling size). The pooling operation mainly reduces the resolution and feature dimensions of the feature map, while enhancing the network robustness to displacement, scaling and distortion. The pooling operation can be defined as:

$$x_j^l = f(\beta_j^l \text{down}(x_j^{l-1}) + b_j^l) \tag{2}$$

where down()is the pooling function, β is the weight coefficient.

3.2 Deep CNN based on stochastic max pooling

The proposed stochastic max pooling first ranks all the values in each receptive field, and then randomly selects one of the top-n values as the pooling result. This strategy is different from average



0.1	1	8.1	9
0.6	8.7	1.5	8.5
0.5	3	0.2	0.2
1.7	2.1	1.1	3

Figure 1: A pooling example

pooling, max pooling and stochastic pooling. However, it can be viewed as the integration between max pooling and stochastic pooling. To describe our strategy clearly, the first step is to define the three common pooling methods.

The max pooling, which outputs the maximum value in each receptive field, can be defined as:

$$y_{max} = max(r) \tag{3}$$

where y_{max} is the pooled maximum value, r is all the values in each receptive field.

In the pooling area (i.e. receptive field) of Figure 1, the maximum value in that area is 9, that is, $y_{\max} = 9$.

The average pooling, which takes the average of all the values in each receptive field, can be defined as:

$$y_{mean} = mean(r) \tag{4}$$

where y_{mean} is the result of average pooling. In Figure 1, the average of all the values in the pooling area is 3.1. In this case, $y_{mean} = 3.1$.

The stochastic pooling selects a value from each receptive field at a certain probability. The probability p_i for the value a_i at position i in the receptive field can be defined as:

$$p_i = \frac{a_i}{\sum_{k \in r} a_k} \tag{5}$$

Then, the stochastic pooling operation can be defined as:

$$y_{stochastic} = a_l \ where \ l \sim P(p_1, ..., p_l) \tag{6}$$

where $y_{stochastic}$ is the result of stochastic pooling.

In the proposed strategy, i.e. the stochastic max pooling, the pooling result is selected from the top-n values of each receptive field. Thus, all the values in the receptive field should be ranked to determine the top-n values. Then, a value is randomly taken from the n values. Hence, the stochastic max pooling can be defined as:

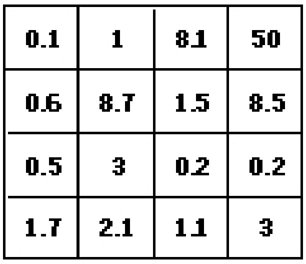
$$y_{restricted} = random(s_n) \tag{7}$$

where random is a function that randomly selects a value; s_n is the top-n values in the receptive field. If n=3, then the gray values in Figure 1 are the top-3 values in the receptive field. Then, one of the three values will be selected as the result of stochastic max pooling.

3.3 Discussion

The pooling layers are of great significance to the deep CNN. These layers both transmit and reduce the dimensions of features, making the network invariant to displacement, scaling and deformation. The value selection from each receptive field directly bears on the effects of feature extraction and transmission in the pooling layers.

Among the three common pooling methods, the max pooling often outperforms the average pooling, and enjoys greater popularity than the other two methods. In general, the maximum value in a



0.1	1	8.1	50
0.6	8.7	1.5	8.5
0.5	3	0.2	0.2
1.7	2.1	1.1	3

Figure 2: A pooling example

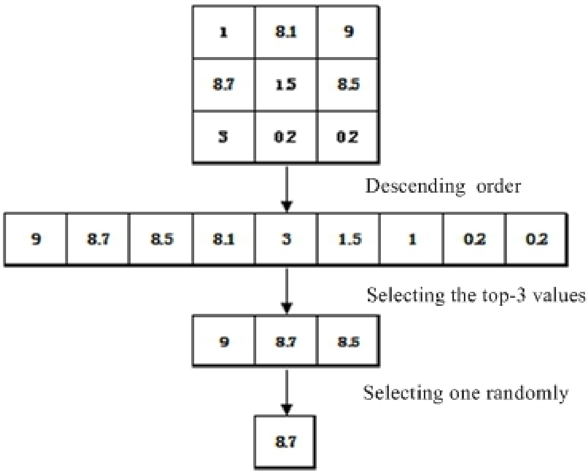


Figure 3: An example of stochastic maximum pooling method process

receptive field can represent the features of that field. However, max pooling was found not suitable for two cases.

The first case is that the original image contains a lot of noises. In this case, the maximum value in a receptive field might be noise. Through max pooling, the noise will be mistaken for feature and introduced to the network. As shown in Figure 2, the value of 50 is possibly a noise, and will be selected by max pooling as the result. However, the pixel value represented by 50 is just noise, which obviously does not conform to the principle of feature extraction.

In the second case, there are many values in the receptive field that are close to the maximum value. As shown in Figure 1, the value 9 will be selected by max pooling. However, the other values (e.g. 8.7 and 8.5) are close to 9 and also reflect the features of the field. The slight differences between these values could be magnified through the layer-by-layer transmission and the processing by thousands of neurons in the network, exerting a huge impact on the network performance. In this case, max pooling eliminates the chance for other values to be selected, which might otherwise enhance network performance.

In stochastic pooling, any value in a receptive field has a chance to be selected. Compared to max pooling, stochastic pooling allows a great diversity in pooling result. However, some small values from pooling areas may be introduced to the network, and disturb the feature extraction. As shown in Figure 1, the value 0.1 might be selected as the result of stochastic pooling. This value does not reflect the typical features of the field and will affect network accuracy.

Our strategy of stochastic max pooling overcomes the above defects. Figure 3 shows the process of stochastic maximum pooling method. We can see that Stochastic max pooling can select a representative value from each receptive field and randomize the pooling result. Similar to stochastic pooling, the randomized network can be considered as a set of multiple sub-models, each of which operates based on a unique pooling result. The set of sub-models outperforms any individual sub-model, which is proved by the research of the dropout mechanism. In addition, stochastic max pooling also enjoys a high robustness in processing highly noisy images.

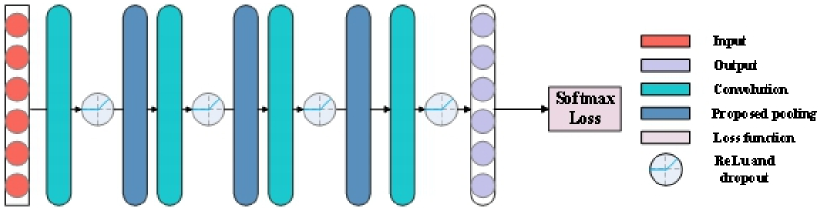
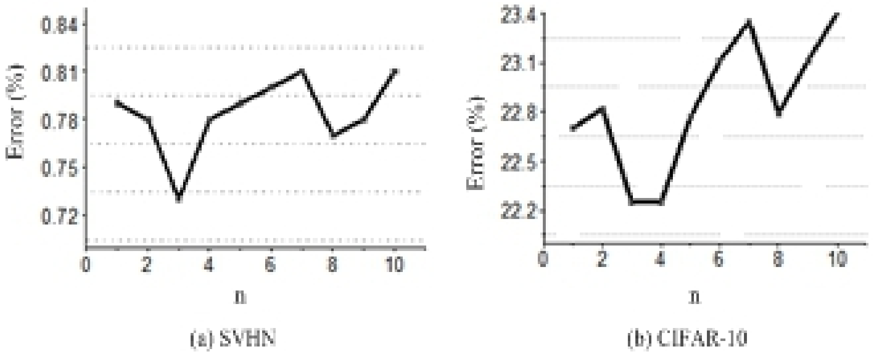


Figure 4: Structure of the deep CNN model



(a) SVHN

(b) CIFAR-10

Figure 5: Errors at different n values.

4 Experiments and results analysis

4.1 Experimental verification of stochastic max pooling

This subsection aims to verify the effectiveness and superiority of the deep CNN based on stochastic max pooling.

The deep CNN model was extended from LightNet [1], a versatile, standalone Matlab-based environment for deep learning. The extension only adjusts the parameters of the pooling layers. As shown in Figure 4, the pooling size is $3x3$. For comparison, four models were developed, respectively based on max pooling (model A), average pooling (model B), stochastic pooling (model C) and stochastic max pooling (model D).

Two datasets were selected for our experiments, including the Street View House Numbers (SVHN) and CIFAR-10 dataset (Canadian Institute for Advanced Research). The SVHN dataset was divided into a training set of 73,257 images and a test set of 26,032 images. The CIFAR-10 dataset contains 60,000 images in 10 categories. Here, 50,000 images are used for training and 10,000 for test. All the images in the two datasets are of the size $32x32$, and not preprocessed.

The four models were applied to the two datasets. The number of top-values n was set to 3 and the number of network trainings to 40. The final results are displayed in Table 1 below.

Table 1: Errors of the four models on the two datasets (%)

Model	**A**	**B**	**C**	**D**
SVHN	6.5	6.4	5.7	4.9
CIFAR-10	23.1	23.5	22.3	21.1

As shown in Table 1, model D (stochastic max pooling) achieved the minimum errors. The accuracies of model D were as high as 95.1% and 78.9%, respectively, on SVHN and CIFAR-10. Model A (max pooling) and model B (average pooling) had basically the same accuracies. Model C (stochastic pooling) realized a lower error than models A and B, but the error was higher than that of model D.

The experiment shows that the n value has a great impact on the final results. To find the best n value, a series of experiments were conducted on model D by adjusting the n value from 1 to 9. According to the experimental results in Figure 5, the error gradually increased with the growth in

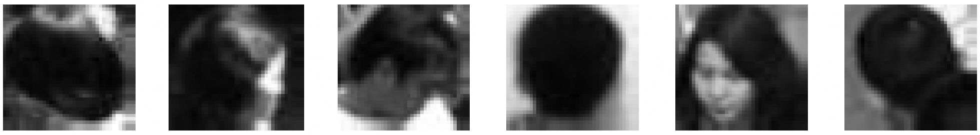


Figure 6: Several images from the head dataset.

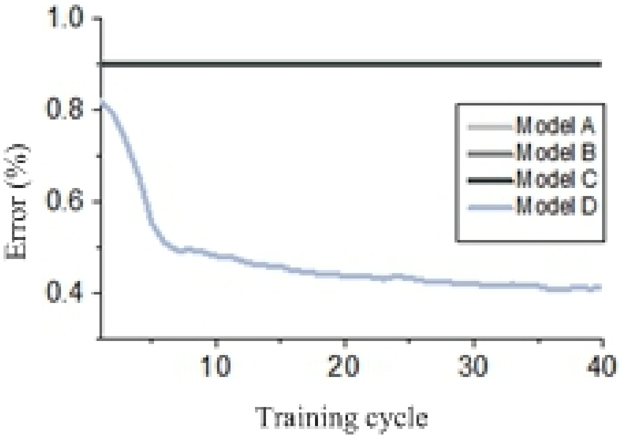


Figure 7: Convergence curves of the four models.

n value. When n increased to 9, the model acted the same as model C (stochastic pooling) and introduced noises into the network, pushing up the error.

4.2 Noisy image experiments based on stochastic max pooling

This subsection attempts to verify if stochastic max pooling is robust in processing noisy images. For this purpose, the heads images were extracted from surveillance videos on public places. These highly noisy images, all of which are $32x32$ in size, were collected into a head dataset, including 10,000 training samples and 2,000 test samples. Several images from the head dataset are given in Figure 6.

For comparison, Gaussian noises were added to SVHN and CIFAR-10. Then, the four models in subsection 4.1 were applied to the head dataset, the noisy SVHN and the noisy CIFAR-10. The results (Table 2) show that model D (stochastic max pooling) outperformed the other models in processing noisy images [17, 18].

Table 2: Errors of the four models on noisy images (%)

Model	**A**	**B**	**C**	**D**
SVHN	12.1	12.5	11.2	9.1
CIFAR-10	29.3	29.5	28.7	26.1
Head dataset	3.4	3.4	3.0	2.1

Next, the intensity of the additive Gaussian noise was increased, and the resulting noisy datasets were processed by the four models again. Figure 7 displays the convergence curves of the four models on the CIFAR-10 dataset added with high-intensity Gaussian noise (mean: 0; variance: 0.1). With the growing intensity of the additive Gaussian noise, models A, B and C could not converge to the optimal solution (the curve A, B, C are coincident in Figure 7), while model D (stochastic max pooling) continued to converge to that solution. It can be seen from the Figure 7 that stochastic max pooling can continue to converge at this time, but the error rate is high. This kind of limit experiment on the other hand shows the processing ability of the stochastic max pooling proposed in this paper. The comparison further verified the robustness of stochastic max pooling to noises.

5 Conclusions

The deep CNN has achieved remarkable results in image processing, thanks to its special structure. To further improve its performance, this paper optimizes the deep CNN both in function and structure. The stochastic max pooling was proposed to solve the problem that many training and test samples contain a lot of noises. Stochastic max pooling limits the selection range of pooling to a larger value that can represent the main information of the pooling area, reduces the possibility of introducing features, increases the randomness of the network, and enhances the generalization ability of the network. The proposed method can extract representative features through the pooling layer, and enhance the network randomness. The randomized network has a better performance in processing noisy images. The proposed method was verified on a self-designed head dataset and two public datasets (SVHN and CIFAR-10), and proved to be effective, superior and robust to noises.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (No.61472444, 61671470). This work is also supported by the Henan province Science and Technology Research Project of China (No.182102210065), the Key scientific research project in Henan colleges and Universities of China (No.17B520019) and the 13th five-year plan for Education Science in Henan province (No.2019-JKGHYB-0461).

References

[1] Benkaddour, M.K.; Bounoua, A. (2017). Feature extraction and classification using deep convolutional neural networks, PCA and SVC for face recognition, *Traitement du Signal*, 34(1-2), 77-91, 2017.

[2] Boureau, Y.L.; Bach, F.; LeCun, Y.; Ponce, J. (2010). Learning mid-level features for recognition, *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2559-2566, 2010.

[3] Gidaris, S.; Komodakis, N. (2015). Object detection via a multi-region and semantic segmentation-aware CNN model, *Proceedings of the IEEE International Conference on Computer Vision*, 1134-1142, 2015.

[4] Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation, *Proceedings of the IEEE conference on computer vision and pattern recognition*, 580-587, 2014.

[5] He, K.; Zhang, X.; Ren, S.; Sun, J. (2016). Deep residual learning for image recognition, *2016 IEEE Conference on Computer Vision and Pattern Recognition*, 770-778, 2016.

[6] Krizhevsky, A.; Sutskever, I.; Hinton, G.E. (2012). Imagenet classification with deep convolutional neural networks, *Advances in Neural Information Processing Systems*, 25(2), 1097-1105, 2012.

[7] Lakshmipathi, A.N.; Battula, B.P. (2018). Deep convolutional neural networks for product recommendation, *Ingénierie des Systèmes d'Information*, 23(6), 161-172, 2018.

[8] Nair, V.; Hinton, G.E. (2010). Rectified linear units improve restricted boltzmann machines, *Proceedings of the 27th international conference on machine learning (ICML-10)*, 807-814, 2010.

[9] Neelapu, R.; Devi, G.L.; Rao, K.S. (2018). Deep learning based conventional neural network architecture for medical image classification, *Traitement du Signal*, 35(2), 169-182, 2018.

[10] Raguram, L.S.B.; Shanmugam, V.M. (2017). Deep belief networks for phoneme recognition in continuous Tamil speech–an analysis, *Traitement du Signal*, 34(3-4), 137-151, 2017.

[11] Ren, S.; He, K.; Girshick, R.; Sun, J. (2017). Faster R-CNN: towards real-time object detection with region proposal networks, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6), 1137-1149, 2017.

[12] Simonyan, K.; Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition, *arXiv*.

[13] Srivastava, N.; Hinton, G.E.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting, *Journal of Machine Learning Research*, 15(1), 1929-1958, 2014.

[14] Wen, Y.; Zhang, K.; Li, Z.; Qiao, Y. (2016). A discriminative feature learning approach for deep face recognition, *European Conference on Computer Vision*, 499-515, 2016.

[15] Zeiler, M.D.; Fergus, R. (2013). Stochastic pooling for regularization of deep convolutional neural networks, *arXiv*.

COPE
Member since 2012
JM08090

This journal is a member of, and subscribes to the principles of,
the Committee on Publication Ethics (COPE).
https://publicationethics.org/members/international-journal-computers-communications-and-control

Cite this paper as:
Sun, S.; Hu, B.; Yu, Z.; Song, X.N. (2020). A Stochastic Max Pooling Strategy for Convolutional Neural Network Trained by Noisy Samples, *International Journal of Computers Communications & Control*, 15(1), 1007, 2020. https://doi.org/10.15837/ijccc.2020.1.3712