Proceedings of the 2015
IEEE Conference on Robotics and Biomimetics
Zhuhai, China, December 6-9, 2015

A Hybrid Convolutional Neural Networks with Extreme Learning Machine for WCE Image Classification

Jia-sheng Yu, Jin Chen, Z.Q. Xiang, Yue-Xian Zou*

Abstract—**Wireless Capsule Endoscopy (WCE) is considered as a promising technology for non-invasive gastrointestinal disease examination. This paper studies the classification problem of the digestive organs for wireless capsule endoscopy (WCE) images aiming at saving the review time of doctors. Our previous study has proved the Convolutional Neural Networks (CNN)-based WCE classification system is able to achieve 95% classification accuracy in average, but it is difficult to further improve the classification accuracy owing to the variations of individuals and the complex digestive tract circumstance. Research shows that there are two possible approaches to improve classification accuracy: to extract more discriminative image features and to employ a more powerful classifier. In this paper, we propose to design a WCE classification system by a hybrid CNN with Extreme Learning Machine (ELM). In our approach, we construct the CNN as a data-driven feature extractor and the cascaded ELM as a strong classifier instead of the conventional used full-connection classifier in deep CNN classification system. Moreover, to improve the convergence and classification capability of ELM under supervision manner, a new initialization is employed. Our developed WCE image classification system is named as HCNN-NELM. With about 1 million real WCE images (25 examinations), intensive experiments are conducted to evaluate its performance. Results illustrate its superior performance compared to traditional classification methods and conventional CNN-based method, where about 97.25% classification accuracy can be achieved in average.**

I. Introduction

Wireless Capsule Endoscopy (WCE) is a novel technology for gastrointestinal disease detection, which was first introduced in [1] and put in use by Given Imaging Ltd. Israel in 2001. Compared with traditional endoscopy, the main advantage of WCE is that the patients can avoid cross infection and suffer no pain.

Commonly, one whole examination process of WCE will last for about 8 hours and produce about 50,000 to 100,000 images per person. Hence, the analysis of the WCE images is a time-consuming work. Digestive organs classification is able to greatly reduce the workload for doctors when they want to quickly review the images of a specific organ. Usually,

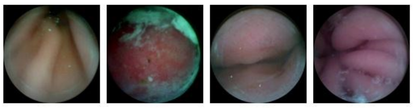

Fig. 1 Images of stomach from different patients

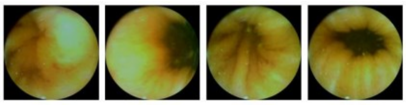

Fig. 2 Images of small intestine of the same patient

WCE images consist of four types of organs, including esophagus, stomach, small intestine and colon. In most cases, there are much fewer images of esophagus than the other three organs, and the majority of gastrointestinal diseases do not happen in the esophagus. Based on these facts, we just focus on classifying WCE images into other three types than esophagus by ignoring the images of esophagus. From the observation of WCE images, there are huge variances between the same organ of the different patients or even the same patient. As an example, Fig. 1 shows four WCE images of stomach from different patients, where both color and texture differs a lot. Fig. 2 shows four WCE images of small intestine of the same patient. The hue is almost the same, but the texture is varied.

Vision-based automatic classification of digestive organs is a typical pattern recognition problem. Most previous works follow the general framework including image feature extraction and classifier design. Berens *et al.* proposed a method to automatically discriminate stomach, intestine and colon by using hue saturation chromaticity histograms which are compressed using a hybrid transform, incorporating the Discrete Cosine Transform (DCT) and Principal Component Analysis (PCA)[2]. Cunha *et al.* [3] extracted MPEG-7 descriptors as low-level image features, and then employed SVM classifier and Bayesian classifier to implement WCE organ classification. It is noted that the methods introduced above all used handcrafted features. The gastric juice and the food debris make the digestive tract circumstance very complicated. The handcrafted features have proved lack of sufficient discriminating power for WCE organs classification [9]. In [4], Ma *et al.* proposed a locality constraint based vector sparse coding algorithm to improve the discriminative capacity of SIFT for WCE organs classification. This method maps the SIFT feature to higher-dimension to gain better discriminating ability.

In 2012, Krizhevsky *et al.* gave the state-of-the-art classification accuracy in ImageNet2012 using Convolutional Neural Networks (CNN) [7]. Afterwards, CNN has brought in revolutions to the computer vision area. A huge amount of researches showed that Deep CNN have been continuously

*Resrach supported by the Shenzhen Science & Technology Fundamental Research Program (No: JCYJ20130329175141512).

Jia-sheng Yu is with ADSPLAB, School of ECE, Peking University, Shenzhen 518055, China (email: 1301213735@pku.edu.cn).

Jin Chen is with ADSPLAB, School of ECE, Peking University, Shenzhen 518055, China (email: 1401213843@pku.edu.cn).

Z.Q. Xiang is with ADSPLAB, School of ECE, Peking University, Shenzhen 518055, China (email: x-lion@pku.edu.cn).

Yue-Xian Zou (correspondence author) is with ADSPLAB/ELIP, School of ECE, Peking University, Shenzhen 518055, China (email: zouyx@pkusz.edu.cn)

978-1-4673-9674-5/15/$31.00 © 2015 IEEE

advancing the image classification accuracy [5], meanwhile it also plays as generic feature extractors for various recognition tasks such as object detection [7], semantic segmentation [6], and image retrieval [8]. Our previous study has proved the Convolutional Neural Networks (CNN)-based WCE classification system is able to achieve 95% classification accuracy in average, but it is difficult to further improve the classification accuracy [9].

In this paper, we aim at improving the WCE organ classification accuracy. The efforts come from two parts: extracting more discriminative WCE image features and designing a powerful classifier. It found that the classification capacity of the fully-connect layer of the CNN is inferior to that of SVM [10]. Moreover, the Extreme Learning Machine (ELM) classifier proposed by Huang *et al.* showed better performance compared with SVM classifier [11][16]. Motivated by above facts, we propose to design a WCE classification system by a hybrid CNN with Extreme Learning Machine (ELM). In our approach, we construct the CNN as a data-driven feature extractor and the cascaded ELM as a strong classifier instead of the conventional used full-connection classifier in deep CNN classification system. Moreover, to improve the convergence and classification capability of ELM under supervision manner, an initialization method to avoid the saturation of the output of the hidden neurons is adopted. As a result, a hybrid CNN-ELM WCE image classification system is developed (named as HCNN-NELM). The proposed method is evaluated with 25 real recording WCE samples (about 1 million WCE images) and get about 97.23% classification accuracy on average.

The rest part of the paper is organized as follows. In section 2, the proposed HCNN-NELM WCE image classification system is presented. The properties of the extracted WCE image features are analyzed, and the initialization method to ELM is employed. The experimental results are given in section 3 and conclusions are drawn in section 4.

II. The proposed HCNN-NELM WCE image classification system

Our proposed HCNN-NELM WCE image classification system is shown in Fig. 3, which includes 2 training stages and a testing stage. Firstly, in training stage 1, an end-to-end CNN classifier is trained, which contains two parts: a CNN feature extractor (characterized by the network weights $\boldsymbol{\theta}_{FE}$) and a softmax classifier. The network weights $\boldsymbol{\theta}_{FE}$, are used to extract WCE image features. In training stage 2, the CNN feature extractor trained in the training stage 1 is cascaded with the ELM classifier (characterized by the network weights $\boldsymbol{W}$, $\boldsymbol{b}$, $\boldsymbol{\beta}$) and then an end-to-end CNN-ELM classifier is trained. The trained network weights $\boldsymbol{W}$, $\boldsymbol{b}$, $\boldsymbol{\beta}$ are used to generate the classification labels. Moreover, in order to further improve the classification capacity of ELM classifier, a new initialization is adopted to avoid the saturation of the output of the hidden neurons. In testing stage, the trained $\boldsymbol{\theta}_{FE}$, $\boldsymbol{W}$, $\boldsymbol{b}$, $\boldsymbol{\beta}$ are used to predict the labels of testing WCE images. All parameters mentioned above are marked in Fig 3. The details of proposed method will be addressed in the following sections.

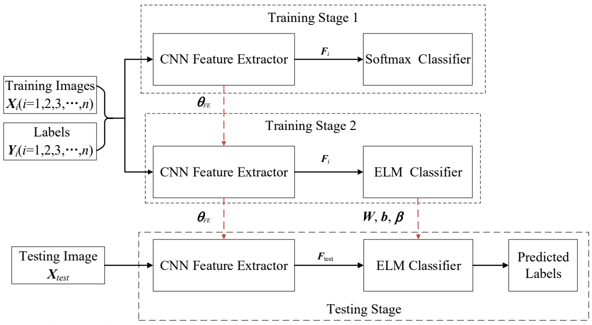

Fig. 3 The procedure of our proposed HCNN-NELM method

A. Training Stage 1

Motivated by the excellent performance of CNN classifier proposed in [9], an end-to-end CNN classifier shown in Fig. 4 is constructed where the feature extractor shown in the red dash line box containing layer 1 to layer 6, and the softmax classifier shown in the blue dash line box. The output of the layer 6 is denoted as the feature vector $\boldsymbol{F}$. The parameter settings of each layer are shown in Table I. The training data is denoted as $\boldsymbol{D}=\{(\boldsymbol{X}_1,Y_1),\ (\boldsymbol{X}_2, Y_2),\ldots,(\boldsymbol{X}_n, Y_n)\}$, where $\boldsymbol{X}_i\in\boldsymbol{R}^{w\times h}$ (w and h are the width and height of the image, respectively) is the i-th training image with its category label Y_i, and n is the total number of WCE training images. It is noted that we only have three categories, therefore we have $Y_i\in\{1,2,3\}$.

We denote $\boldsymbol{\theta}$, $\boldsymbol{\theta}_{FE}$ and $\boldsymbol{\theta}_{SC}$ as the parameters of the full CNN, the CNN feature extractor, and the weight matrix of the softmax classifier, respectively. Therefore $\boldsymbol{\theta}$ is expressed as

$$\boldsymbol{\theta}=\{\boldsymbol{\theta}_{FE},\boldsymbol{\theta}_{SC}\} \tag{1}$$

As shown in Fig 4, for the i-th WCE training image, the output of the CNN feature extractor is $\boldsymbol{F}_i\in\mathbf{R}^{1024}$ which is denoted as

$$\boldsymbol{F}_i=m_{\theta_{FE}}(\boldsymbol{X}_i) \tag{2}$$

where $m_{\theta_{FE}}(\cdot)$ denotes the mapping function to extract the feature vector $\boldsymbol{F}_i$ from $\boldsymbol{X}_i$.

In Fig. 4, it can be seen that $\boldsymbol{F}_i$ is taken as the input of the softmax classifier. To compute the output of the softmax classifier, let's define a middle variable as

$$h_\theta(\boldsymbol{X}_i)=\boldsymbol{\theta}_{SC}\boldsymbol{F}_i \tag{3}$$

Then the output of the softmax classifier is computed as

$$f_\theta(\boldsymbol{X}_i)=\exp(h_\theta(\boldsymbol{X}_i))\Big/\left\|\exp(h_\theta(\boldsymbol{X}_i))\right\|_1 \tag{4}$$

To determine the training loss function, let's firstly define the loss of the i-th image as $\text{loss}_\theta(Y_i,\ \boldsymbol{X}_i)$, which is the cross-entropy between the estimated class probabilities $f_\theta(\boldsymbol{X}_i)$ and the ground truth target vector $\boldsymbol{e}_{Y_i}$. Here, the vector $\boldsymbol{e}_{Y_i}$ is a 3-element standard basis vector corresponding to 1 at the Y_i-th entry and 0 at other entries. For example, if Y_i=1, then $\boldsymbol{e}_{Y_i}=[1,0,0]^{\mathrm{T}}$. Therefore, we have

$$\text{loss}_\theta(Y_i,\boldsymbol{X}_i)=-<\boldsymbol{e}_{Y_i},\log(f_\theta(\boldsymbol{X}_i))>=-\sum_{k=1}^{n_{class}}\boldsymbol{e}_{Y_i}{}^{(k)}\log(f_\theta(\boldsymbol{X}_i)^{(k)}) \tag{5}$$

where n_{class}=3 is the number of classes. $\boldsymbol{e}_{Yi}^{(k)}$ and $f_\theta(\boldsymbol{X}_i)^{(k)}$ are the k-th element of $\boldsymbol{e}_{Yi}$, and $f_\theta(\boldsymbol{X}_i)$, respectively.

The total loss for all training data in $\boldsymbol{D}$ can be computed as

$$L_\theta(\boldsymbol{D}) = 1/n \sum_{(\boldsymbol{X}_i, Y_i)\in \boldsymbol{D}} \text{loss}_\theta(Y_i, \boldsymbol{X}_i) \tag{6}$$

The optimal full CNN classifier characterized by the parameter $\boldsymbol{\theta}$ using the training data set $\boldsymbol{D}$ can be achieved by minimizing the following loss function:

$$\theta^* = \arg\min_\theta L_\theta(\boldsymbol{D}) \tag{7}$$

Research shows that the back propagation algorithm is commonly used to get the optimal solution of (7) .

Intuitively, the supervised training of the full CNN classifier described above serves to ensure that the input WCE images of different organs can be distinguished, meanwhile the WCE images from the same organ can be clustered. In the following section, the properties of the extracted features by the full CNN classifier are analyzed for better understanding of its class discriminating capability.

B. Analysis of the properties of the extracted WCE image features

According to the procedure of training stage 1, the parameter $\boldsymbol{\theta}_{FE}$ of the CNN feature extractor is obtained by minimizing the loss function in (6). In order to get more understanding of classification problem of CNN, we break the loss function into two parts: a well-defined classification problem [13] to maximize the between-class variations and a regularizer to minimize the within-class variations. It confirms that the CNN feature extractor is able to get the discriminant features. The analysis is described as follows.

Let $\boldsymbol{D}^{(j)}$ denotes the image set whose label is j (j =1,2,3):

$$\boldsymbol{D}^{(j)} = \{(\boldsymbol{X}_i, Y_i) \mid (\boldsymbol{X}_i, Y_i) \in \boldsymbol{D}, Y_i = j, i = 1,...,n\} \tag{8}$$

So the loss function for $\boldsymbol{D}^{(j)}$ can be computed as

$$L_\theta(\boldsymbol{D}^{(j)}) = 1/n^{(j)} \sum_{(\boldsymbol{X}_i, Y_i)\in \boldsymbol{D}^{(j)}} \text{loss}_\theta(Y_i, \boldsymbol{X}_i) \tag{9}$$

where $n^{(j)}$ is the number of images whose labels are j. And then the loss function for $\boldsymbol{D}$, $L_\theta(\boldsymbol{D})$, can be rewritten as:

$$L_\theta(\boldsymbol{D}) = 1/n \sum_{j=1}^{n_{class}} n^{(j)} L_\theta(\boldsymbol{D}^{(j)}) \tag{10}$$

By applying formula (3), (4) and (5), we can unfold formula (9) as

$$\begin{aligned} L_\theta(\boldsymbol{D}^{(j)}) &= 1/n^{(j)} \sum_{(\boldsymbol{X}_i, Y_i)\in \boldsymbol{D}^{(j)}} \Big(-\sum_{k=1}^{n_{class}} \boldsymbol{e}_{Y_i}{}^{(k)} \log f_\theta(\boldsymbol{X}_i)^{(k)}\Big) \\ &= 1/n^{(j)} \sum_{(\boldsymbol{X}_i, Y_i)\in \boldsymbol{D}^{(j)}} \Big\{\Big[-\sum_{k=1}^{n_{class}} \boldsymbol{e}_{Y_i}{}^{(k)} h_\theta(\boldsymbol{X}_i)^{(k)}\Big] \\ &\quad + \log(\|\exp(h_\theta(\boldsymbol{X}_i))\|_1)\Big\} \\ &= 1/n^{(j)} \sum_{(\boldsymbol{X}_i, Y_i)\in \boldsymbol{D}^{(j)}} (-< \boldsymbol{e}_{Y_i}, \boldsymbol{\theta}_{SC}\boldsymbol{F}_i >) \\ &\quad + 1/n^{(j)} \sum_{(\boldsymbol{X}_i, Y_i)\in \boldsymbol{D}^{(j)}} [\log(\|\exp(\boldsymbol{\theta}_{SC}\boldsymbol{F}_i)\|_1)] \end{aligned} \tag{11}$$

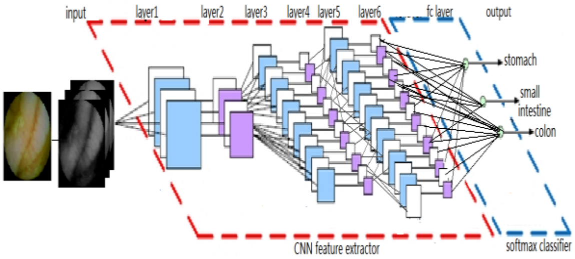

Fig. 4 The full architecture of CNN (Best view in e-print)

Table I. The architecture parameters of CNN

Layer	Type	Number of maps or neurons	Kernel size	Stride
1	convolutional	32	5×5	1
2	max pooling	32	3×3	3
3	convolutional	32	5×5	1
4	max pooling	32	3×3	3
5	convolutional	64	5×5	1
6	max pooling	64	3×3	3
7	fully-connected	3	-	-

We define $\boldsymbol{F}_{\text{avg}}^{(j)}$ as the average feature vector of all images whose label is j.

$$\begin{aligned} \boldsymbol{F}_{avg}^{(j)} &= 1/n^{(j)} \sum_{(\boldsymbol{X}_i, Y_i)\in \boldsymbol{D}^{(j)}} \boldsymbol{F}_i \\ Y_i &= j \end{aligned} \tag{12}$$

Substituting (12) into (11), we get

$$\begin{aligned} L_\theta(\boldsymbol{D}^{(j)}) &= -< \boldsymbol{e}_j, \boldsymbol{\theta}_{SC}(\frac{1}{n^{(j)}}) \sum_{(\boldsymbol{X}_i, Y_i)\in \boldsymbol{D}^{(j)}} F_i) > \\ &\quad + 1/n^{(j)} \sum_{(\boldsymbol{X}_i, Y_i)\in \boldsymbol{D}^{(j)}} [\log(\|\exp(\boldsymbol{\theta}_{SC}\boldsymbol{F}_i)\|_1)] \\ &= [-< \boldsymbol{e}_j, \boldsymbol{\theta}_{SC}\boldsymbol{F}_{avg}^{(j)} > + \log(\|\exp(\boldsymbol{\theta}_{SC}\boldsymbol{F}_{avg}^{(j)})\|_1)] \\ &\quad + \{1/n^{(j)} \sum_{(\boldsymbol{X}_i, Y_i)\in \boldsymbol{D}^{(j)}} [\log(\|\exp(\boldsymbol{\theta}_{SC}\boldsymbol{F}_i)\|_1)] \\ &\quad - \log(\|\exp(\boldsymbol{\theta}_{SC}\boldsymbol{F}_{avg}^{(j)})\|_1)\} \end{aligned} \tag{13}$$

In the study, we suppose $n^{(1)}=n^{(2)}=n^{(3)}$, and substitute (13) into (10), then $L_\theta(\boldsymbol{D})$ can be rewritten as

$$\begin{aligned} \mathrm{L}_\theta(\boldsymbol{D}) &= 1/n_{class} \sum_{j=1}^{n_{class}} [-< \boldsymbol{e}_j, \theta_{SC}\boldsymbol{F}_{avg}^{(j)} > + \log(\|\exp(\theta_{SC}\boldsymbol{F}_{avg}^{(j)})\|_1)] \\ &\quad + 1/n_{class} \sum_{j=1}^{n_{calss}} \{\frac{1}{n^{(j)}} \sum_{(\boldsymbol{X}_i, Y_i)\in \boldsymbol{D}^{(j)}} [\log(\|\exp(\theta_{SC}\boldsymbol{F}_i)\|_1)] \\ &\quad - \log(\|\exp(\theta_{SC}\boldsymbol{F}_{avg}^{(j)})\|_1)\} \end{aligned} \tag{14}$$

We break (14) into two parts for analysis. The first part is

$$1/n_{class} \sum_{j=1}^{n_{class}} [-< \boldsymbol{e}_j, \theta_{SC}\boldsymbol{F}_{avg}^{(j)} > + \log(\|\exp(\theta_{SC}\boldsymbol{F}_{avg}^{(j)})\|_1)] \tag{15}$$

From [13], it is clear that (15) shows the loss function of a multinomial logistic regression problem with input-target ($\boldsymbol{F}_{\text{avg}}^{(j)}$, $\boldsymbol{e}_j$). If the average feature vectors for three categories are classified correctly, the value of (15) would be zero. If there are some average feature vectors misclassified, this term would be greater than zero. Therefore, this term emphasizes the correct classification of the average feature vector.

Then, we analysis the second part of (14)

$$\frac{1}{n^{(j)}} \sum_{(\boldsymbol{X}_i, Y_i)\in \boldsymbol{D}^{(j)}} [\log(\|\exp(\theta_{SC}\boldsymbol{F}_i)\|_1)] - \log(\|\exp(\theta_{SC}\boldsymbol{F}_{avg}^{(j)})\|_1) \quad (16)$$

According to the convexity of the function $\log\|\exp[(\cdot)\|_1$ and Jensen's inequality, the value of (16) is greater than or equal to zero. Only if the feature representation is perfectly invariant, which means $\boldsymbol{\theta}_{SC}\boldsymbol{F}_i = \boldsymbol{\theta}_{SC}\boldsymbol{F}_{avg}{}^{(j)}$, the value of (16) will be equal to zero. Therefore, (16) can be viewed as a regularizer which enforcing all $\boldsymbol{F}_i$ from the same class to close to their average value of that class.

In conclusion, the proposed approach not only provides the ability to classify the WCE images from different categories, but also provides the ability to map the different images of the same category to a similar CNN feature vectors. We are confident that the extracted features by the CNN feature extractor have the discriminating power for different classes.

C. Training stage 2

As mentioned before, Huang *et al.* [12] proposed the extreme learning machine (ELM) classifier as shown in Fig. 5. The input of the ELM is feature vector $\boldsymbol{F}\in\mathbf{R}^{n_{input}}$. $n_{input} = 1024$ is the number of the input neurons. n_{hidden} is the number of the hidden neurons. $n_{output}=3$ is the number of the output hidden neurons. $\boldsymbol{W}\in\mathbf{R}^{n_{hidden}\times n_{input}}$ is the input weight matrix, $\boldsymbol{b}\in\mathbf{R}^{n_{hiddent}}$ is the bias vector of the hidden layer while $\beta \in\mathbf{R}^{n_{hidden}\times n_{outputt}}$ is the output weight matrix. $\boldsymbol{O}\in\mathbf{R}^{n_{output}}$ is the output vector of the ELM classifier.

According to [11], while ELM initializes $\boldsymbol{W}$ and $\boldsymbol{b}$ randomly and keeps them fixed, the training procedure of ELM classifier aims at computing β analytically. Their theory proves that the ELM has a good generalization performance and the learning speed is much faster than that of the single hidden layer neural network. To take the advantages of the ELM, in training stage 2, we replace the softmax classifier by the ELM classifier, and the details about training the ELM classifier are introduced as follows.

As described in Section A, the parameter $\boldsymbol{\theta}_{FE}$ of the CNN feature extractor is obtained, then the feature vectors, which are the inputs of the ELM classifier, can be calculated directly. Specifically, a new training data pairs can be formed to train the ELM classifier, which is denoted as $\boldsymbol{D}_F$:

$$\boldsymbol{D}_F = \{(\boldsymbol{F}_i, Y_i) \mid \boldsymbol{F}_i \in \boldsymbol{R}^{n_{input}}, Y_i \in \boldsymbol{R}, i = 1,2,...,n\} \quad (17)$$

where $\boldsymbol{F}_i$ is the feature vector of input image $\boldsymbol{X}_i$, n denotes the number of the training samples.

As shown in Fig. 5, the output of the hidden layer $h_o(\boldsymbol{F}_i)\in\mathbf{R}^{n_{hiddent}}$ and the ELM classifier $\boldsymbol{O}_i \in\mathbf{R}^{n_{outpu}}$ can be calculated respectively as:

$$\begin{aligned} h_o(\boldsymbol{F}_i) &= g(\boldsymbol{W}\boldsymbol{F}_i + \boldsymbol{b}) \\ \boldsymbol{O}_i &= h_o(\boldsymbol{F}_i)\boldsymbol{\beta} \end{aligned} \quad (18)$$

where g(·) is the activation function in hidden layer, which is usually sigmoid or hyperbolic tangent

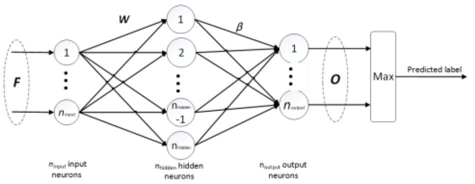
b
W
β
F
1
2
n_{input}
n_{hidden} -1
n_{hidden}
1
n_{output}
O
Max
Predicted label
n_{input} input neurons
n_{hidden} hidden neurons
n_{output} output neurons

Fig. 5 ELM classifier architecture

The input weight matrix $\boldsymbol{W}$ and the bias vector $\boldsymbol{b}$ are initialized by using the numbers that selected randomly from a uniform range [-*a*, *a*] and [-1, 1], respectively. Obviously, selecting different *a* will lead to different initialization and different performance of the ELM classifier.

With the whole training image data set, the output of hidden layer $\boldsymbol{H}\in\mathbf{R}^{n\times n_{hidden}}$ and the corresponding ground truth label matrix $\boldsymbol{K} \in\mathbf{R}^{n\times n_{output}}$ is denoted as follows:

$$\begin{aligned} \boldsymbol{H} &= [h_o(\boldsymbol{F}_1), h_o(\boldsymbol{F}_2), ..., h_o(\boldsymbol{F}_n)]^T \\ \boldsymbol{K} &= [e_{Y_1}, e_{Y_2}, ..., e_{Y_n}]^T \end{aligned} \quad (19)$$

A standard ELM classifier is able to approximate arbitrary samples with zero error [12]. It means that given the training image set $\boldsymbol{D}_F$, there exit $\boldsymbol{W}$***,*** $\boldsymbol{b}$ and β that make (20) hold true

$$O_i = e_{Y_i} \quad (20)$$

where i =1,2,3,…,n.

As a result, the parameter $\boldsymbol{\beta}$ can be obtained by minimum the mean square error. According to (19) and (20), the optimal β can be computed by

$$\boldsymbol{\beta} = \boldsymbol{H}^{\dagger}\boldsymbol{L} \quad (21)$$

where $\boldsymbol{H}^{\dagger}$ is the pseudo inverse of $\boldsymbol{H}$.

As described above, n_{hidden} is the only one super-parameter in the ELM classifier and β is calculated by minimum square error. Compared with the single hidden layer neural network, the ELM classifier has fewer super-parameters and can be trained in faster speed. Meanwhile, it has better generalization performance than that of the single hidden layer neural network [11].

D. Improvement of the classification capacity of the ELM classifier

From the training procedure of the ELM classifier, it can be seen that $\boldsymbol{W}$ and $\boldsymbol{b}$ are initialized randomly and fixed. In this section, we will discuss the impact of the parameter $\boldsymbol{W}$ and $\boldsymbol{b}$ on the classification performance of the ELM classifier.

As discussed before, the sigmoid or hyperbolic tangent activation function g(·) are commonly used in hidden layer of the ELM classifier. However, it is clear that the activation function would be saturated at the tails where the gradient is almost zero. So, when the input of the activation function is with zero mean and small variance, activation function will work properly. However, when the input is with large variance, the activation function for many inputs will work in the saturation condition. This will cause the degradation of the performance of the ELM classifier.

Let $\boldsymbol{F}^{(k)}$ denotes the k-th element of the feature vector $\boldsymbol{F}$ and $h^{(j)}{}_{input}$ as the input of the j-th hidden neuron. As shown in Fig. 5, we can get

$$h_{input}^{(j)} = \sum_{k=1}^{n} \boldsymbol{W}_{j,k} \boldsymbol{F}^{(k)} + \boldsymbol{b}_j \tag{22}$$

where $\boldsymbol{W}_{j,k}$ is the element in j-th row and k-th column, $\boldsymbol{b}_j$ is the bias vector of the j-th hidden neuron.

Suppose that $\boldsymbol{F}^{(k)}$ and $\boldsymbol{W}_{j,k}$ are independent and identically distributed with zero mean, the expectation and variance of $h^{(j)}{}_{input}$ can be expressed as

$$\begin{aligned} &E(\boldsymbol{h}_{input}^{(j)}) = 0 \\ &Var(\boldsymbol{h}_{input}^{(j)}) = Var(\sum_{k=1}^{n} W_{j,k} \boldsymbol{F}^{(k)} + \boldsymbol{b}_j) \\ &\qquad = n_{input} Var(W_{j,k} \boldsymbol{F}^{(k)}) + Var(\boldsymbol{b}_j) \end{aligned} \tag{23}$$

According to the probability theory [14], we can get

$$\begin{aligned} Var(\boldsymbol{W}_{j,k} \boldsymbol{F}^{(k)}) &= E[\boldsymbol{W}_{j,k}]^2 Var(\boldsymbol{F}^{(k)}) + E[\boldsymbol{F}^{(k)}]^2 Var(\boldsymbol{W}_{j,k}) \\ &\quad + Var(\boldsymbol{W}_{j,k}) Var(\boldsymbol{F}^{(k)}) \end{aligned} \tag{24}$$

Substituting (24) into(23), we get

$$Var(\boldsymbol{h}_{input}^{(j)}) = n_{input} Var(W_{j,k}) Var(\boldsymbol{F}^{(k)}) + Var(\boldsymbol{b}_j) \tag{25}$$

When $\boldsymbol{W}$ and $\boldsymbol{b}$ are initialized respectively by using the numbers that selected randomly from an uniform range [-a, a] and [-1, 1], we can rewrite (25) as

$$Var(\boldsymbol{h}_{input}^{(j)}) = {}^{n_{input} a^2}\!/_{3}\, Var(\boldsymbol{F}^{(k)}) + {}^{1}\!/_{3} \tag{26}$$

where a is the boundary value of $\boldsymbol{W}$.

The original initialization method proposed for the ELM classifier [11] sets a as 1, so $Var(h^{(j)}{}_{input})$ in (26) becomes

$$Var_{original}(\boldsymbol{h}_{input}^{(j)}) = {}^{n_{input}}\!/_{3}\, Var(\boldsymbol{F}^{(k)}) + {}^{1}\!/_{3} \tag{27}$$

According to (27), it is clear that $Var_{original}(h^{(j)}{}_{input})$ is larger than $Var(\boldsymbol{F}^{(k)})$ because of n_{input} >>3, which causes the degradation of the performance of the ELM classifier. In order to reduce $Var(h^{(j)}{}_{input})$, we need to make (28) hold true.

$$Var(\boldsymbol{h}_{input}^{(j)}) < Var(\boldsymbol{F}^{(k)}) \tag{28}$$

By examining the first item in (26), it is possible for us to choose parameter a relevant to n_{input} and to make the weight of $Var(\boldsymbol{F}^{(k)})$ smaller than 1. In this case, we can achieve (28). Specifically, we want to achieve

$${}^{n_{input} a^2}\!/_{3} < 1 \tag{29}$$

From (29), after simple manipulation, we get

$$a < \sqrt{{}^{3}\!/_{n_{input}}} \tag{30}$$

Furthermore, the performance of the ELM classifier is also related to the choice of the hidden neurons, it is reasonable to connect to n_{hidden} and we noted n_{hidden} is always larger than n_{input}. As a result, from (30), we get

$$a < \sqrt{{}^{6}\!/_{\left(n_{input} + n_{hidden}\right)}} < \sqrt{{}^{6}\!/_{2n_{input}}} = \sqrt{{}^{3}\!/_{n_{input}}} \tag{31}$$

Selecting a value according to (31) will guarantee the inequality shown in (28) and the method described in (31) is termed as the normalization initialization method. Therefore, (26) can be reformulated as

$$Var_{normalized}(\boldsymbol{h}_{input}^{(j)}) = {}^{2}\!/_{\left(1 + {}^{n_{hidden}}\!/_{n_{input}}\right)}\, Var(\boldsymbol{F}^{(k)}) + {}^{1}\!/_{3} \tag{32}$$

From (32), it is straightforward to get

$$Var_{normalized}(\boldsymbol{h}_{input}^{(j)}) \le Var(\boldsymbol{F}^{(k)}) + {}^{1}\!/_{3} \tag{33}$$

Comparing (27) and (33), it is clear that $Var_{original}(h^{(j)}{}_{input})$ is always larger than $Var_{normalized}$ $(h^{(j)}{}_{input})$. As discussed above, theoretically, the proposed normalization initialization method reduces the variance of the input, then the possibility of the saturation of the output of the hidden neurons is also reduced.

The experiment results in section 3 verify the effectiveness of the normalized initialization method.

E. Testing stage

As shown in Fig 4, the parameter $\boldsymbol{\theta}_{FE}$ of the CNN feature extractor and the parameter $\boldsymbol{W}$, $\boldsymbol{b}$, $\boldsymbol{\beta}$ of the ELM classifier are obtained in training stages, The testing procedure is summarized as follows with the testing image $\boldsymbol{X}_{test}$:

1). Compute the feature vector $\boldsymbol{F}_{test}$ by

$$\boldsymbol{F}_{test} = m_{\theta_{FE}}(\boldsymbol{X}_{test}) \tag{34}$$

where $m_{\theta_{FE}}(\cdot)$ is the mapping function introduced in (2)..

2). According to (18), compute the output of the ELM classifier $\boldsymbol{O}_{test}$ as below:

$$\boldsymbol{O}_{test} = \mathrm{g}(\boldsymbol{W}\boldsymbol{F}_{tet} + \boldsymbol{b})\boldsymbol{\beta} \tag{35}$$

3). The predicted label Y_{test} is calculated as below:

$$Y_{test} = \arg\max_{k} \boldsymbol{O}_{test}^{(k)} \tag{36}$$

where $\boldsymbol{O}_{\text{test}}^{(k)}$ is the k-th element of the $\boldsymbol{O}_{\text{test}}$.

III. Experimental results and analysis

A. Experiment settings

There are 25 whole WCE image sets generated for 25 individuals by WCE examinations. Moreover, there are about 1 million WCE images which have been labeled by medical professionals. All WCE images for experiments are RGB color images. The size of original WCE image is 480 × 480, and we scale them down to 96 × 96 for saving computation cost.

Limited by the memory of the computer, 60,000 images are randomly selected for training and 15,000 images are randomly selected for testing in each experiment.

The experimental results are given by averaging over 5 independent trials.

B. *Experiment results*

The results of WCE organs classification by our proposed algorithm and the compared algorithms are listed in Table II. The SIFT-LCVSC-SVM is proposed by Ma *et al.* in [4]. The CNN$_{\text{plain}}$ is an end-to-end CNN classifier which contains a CNN feature extractor and a softmax classifier [9].The CNN-SVM indicates the classifier consists of a CNN feature extractor and a SVM classifier. The HCNN-NELM is our proposed algorithm where a CNN feature extractor, and a normalized initialized ELM classifier are cascaded. For this experiment, the hidden layer parameter n_{hidden} is set as 9000. From Table II, we can draw the following observations: 1) Comparing the results of the SIFT-LCVSC-SVM with that of the CNN-SVM, the CNN feature extractor is able to extract more discriminate features than the handcrafted feature SIFT with the local constraint sparse coding method. 2) Comparing the results of the CNN$_{\text{plain}}$, the CNN-SVM, and the HCNN-NELM, it can be seen that the classification ability of using softmax classifier is inferior to SVM and ELM classifiers. Moreover, the proposed normalized initialization ELM outperforms to that of the linear SVM. 3) For WCE organ classification application, our proposed HCNN-NELM method offers the best classification performance with the experimental conditions.

C. *Impact of the normalization initialization*

This experiment aims at evaluating the impact of our proposed normalization initialization for the ELM classifier. In this experiment, the number of hidden neurons varies from 2000 to 10000. For comparison, the results using the random initialization (called original initialization method in Fig. 6), with the same experimental settings, are also given. As shown in Fig. 6, the normalized initialization method outperforms to the random initialization method for all values of n_{hidden}. Besides, we can see that the performance of the HCNN-NELM is not sensitive to n_{hidden}. When n_{hidden} increases from 2000 to 10000, the classification accuracy increases about 1%.

IV. CONCLUSION

In this paper, motivated by the excellent performance of the CNN for image classification problems. We firstly analyzed the discriminative capability of the features learned by the CNN with full connection softmax classifier. Then a novel HCNN-NELM method is proposed for WCE organs classification, which is a hybrid system by cascading a CNN feature extractor and an ELM classifier. To reduce the saturation of the output of hidden neurons in the ELM classifier, a normalization initialization is employed. Experiment results validate that the effectiveness of our proposed method. With 1 million WCE images, we achieve 97.23% classification accuracy in average which outperforms the traditional classification methods and the conventional CNN-based method.

ACKNOWLEDGMENT

This work was partially supported by the Shenzhen Science & Technology Fundamental Research Program (No: JCYJ20130329175141512) and the collaboration research project funded by Shenzhen JiFu Technology Ltd.. And the authors would thank Shenzhen JiFu Technology Ltd. for providing us the WCE images.

Table II. The classification accuracy of different methods

Method	Classification Accuracy (%)
SIFT-LSVSC-SVM [4]	85.79
CNN$_{\text{plain}}$ [9]	95.00
CNN-SVM	97.05
HCNN-NELM	**97.23**

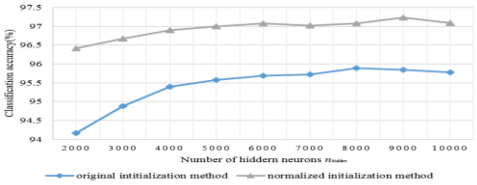

Fig. 6 The classification accuracy of different initialization method

REFERENCES

[1] G. Iddan, G. Meron, A. Glukhovsky, and P. Swain, "Wireless capsule endoscopy," Nature, vol. 405, p. 417, 2000.

[2] J. Berens, M. Mackiewicz, and D. Bell, "Stomach, intestine, and colon tissue discriminators for wireless capsule endoscopy images," in Medical Imaging, 2005, pp. 283-290.

[3] J. S. Cunha, M. Coimbra, P. Campos, and J. M. Soares, "Automated topographic segmentation and transit time estimation in endoscopic capsule exams," Medical Imaging, IEEE Transactions on, vol. 27, pp. 19-27, 2008.

[4] T. Ma, Y. Zou, Z. Xiang, L. Li, and Y. Li, "Wireless capsule endoscopy image classification based on vector sparse coding," in Signal and Information Processing (ChinaSIP), 2014 IEEE China Summit & International Conference on, 2014, pp. 582-586.

[5] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in Advances in neural information processing systems, 2012, pp. 1097-1105.

[6] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, "Overfeat: Integrated recognition, localization and detection using convolutional networks," arXiv preprint arXiv:1312.6229, 2013.

[7] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," in Computer Vision–ECCV 2014, ed: Springer, 2014, pp. 346-361.

[8] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "CNN features off-the-shelf: an astounding baseline for recognition," in Computer Vision and Pattern Recognition Workshops (CVPRW), 2014 IEEE Conference on, 2014, pp. 512-519.

[9] Zou, Y.X., Li, L., Wang, Y., Yu, J.S., Li, Y., Deng, W.J., “Classifying Digestive Organs in Wireless Capsule Endoscopy Images Based on Deep Convolutional Neural Network”, the 20th IEEE International Conference on Digital Signal Processing (DSP), Singapore, July 21-24, 2015.

[10] F. J. Huang and Y. LeCun, "Large-scale learning with svm and convolutional for generic object categorization," in Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on, 2006, pp. 284-291.

[11] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: a new learning scheme of feedforward neural networks," in Neural Networks, 2004. Proceedings. 2004 IEEE International Joint Conference on, 2004, pp. 985-990.

[12] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: theory and applications," Neurocomputing, vol. 70, pp. 489-501, 2006.

[13] A. Dosovitskiy, J. T. Springenberg, M. Riedmiller, and T. Brox, "Discriminative unsupervised feature learning with convolutional neural networks," in Advances in Neural Information Processing Systems, 2014, pp. 766-774.

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