IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received February 5, 2018, accepted March 13, 2018, date of publication March 20, 2018, date of current version May 16, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2817593

Breast Cancer Classification Based on Fully-Connected Layer First Convolutional Neural Networks

KUI LIU[1,2], GUIXIA KANG[1,2], NINGBO ZHANG[1,2], (Member, IEEE) AND BEIBEI HOU[1,2]

[1]Key Laboratory of Universal Wireless Communications, Beijing University of Posts and Telecommunications, Beijing 100876, China
[2]Wuxi BUPT Sensory Technology and Industry Institute Co. Ltd., Wuxi, China

Corresponding author: Kui Liu (liukui_006@126.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61471064 and in part by the National Science and Technology Major Project of China under Grant 2017ZX03001022-005.

ABSTRACT Both Wisconsin diagnostic breast cancer (WDBC) database and the Wisconsin breast cancer database (WBCD) are structured datasets described by cytological features. In this paper, we were seeking to identify ways improve the classification performance for each of the datasets based on convolutional neural networks (CNN). However, CNN is designed for unstructured data, especially for image data, which has been proven to be successful in the field of image recognition. A typical CNN may not keep its performance for structured data. In order to take advantage of CNN to improve the classification performance for structured data, we proposed fully-connected layer first CNN (FCLF-CNN), in which the fully-connected layers are embedded before the first convolutional layer. We used the fully-connected layer as an encoder or an approximator to transfer raw samples into representations with more locality. In order to get a better performance, we trained four kinds of FCLF-CNNs and made an ensemble FCLF-CNN by integrating them. We then applied it to the WDBC and WBCD datasets and obtained the results by a fivefold cross validation. The results showed that the FCLF-CNN can achieve a better classification performance than pure multi-layer perceptrons and pure CNN for both datasets. The ensemble FCLF-CNN can achieve an accuracy of 99.28%, a sensitivity of 98.65%, and a specificity of 99.57% for WDBC, and an accuracy of 98.71%, a sensitivity of 97.60%, and a specificity of 99.43% for WBCD. The results for both datasets are competitive compared to the results of other research.

INDEX TERMS Breast cancer, structured data, classification, fully-connected layer first, convolutional neural networks.

I. INTRODUCTION

It is a significant challenge to identify ways that can successfully fight against breast cancer, which has become a major threat to the health of women. Fine needle aspiration cytology (FNAC) is a diagnostic tool used for breast cancer detection, with a correct classification rate running at about 90%. Improved classification systems are in demand, and machine learning can help to achieve this [1]. Many machine learning methods have been used in classification of the datasets obtained via FNA biopsy [2], [3], including weighted Naive Bayesian [2], multilayer perceptrons [4], radial basis function networks [5], fuzzy classifiers [6], clustering algorithms [7] and kernel-based methods [8].

Convolutional neural networks (CNN) have recently made great success in the field of image recognition [9]–[11], object detection [11], [12] and image segmentation [13], [14]. In this paper, we were seeking ways to improve the breast cancer classification performance based on CNN. The Wisconsin Diagnostic Breast Cancer (WDBC) database and the Wisconsin Breast Cancer Database (WBCD) are two datasets used for the development of breast cancer automated diagnostic systems. However, they are both structured datasets. More specifically, each of these samples is described by the existence of cytological features. CNN has been specifically designed for image data. The local connection and multi-layer architecture in CNN can extract multi-level local features in image data, making the CNN outperform other models in the field of image recognition. By breaking down the local structure of the image data, CNN can still work but the performance is inadequate. We illustrate this by empirically exploring the MNIST dataset in Section 2. In order to get better results for both the WBCD and WDBC datasets, we proposed the fully-connected layer first CNN (FCLF-CNN), in which the fully-connected layers are

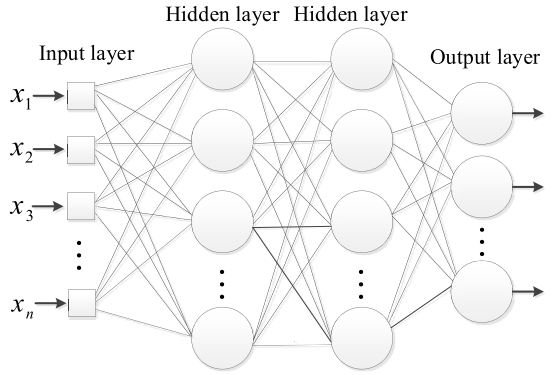


FIGURE 1. The architecture of a MLP with 2 hidden layers.

embedded before the first convolutional layer. We used these layers as an encoder by setting a softmax loss or an approximator by setting a mean square error (MSE) loss, which can transfer raw samples into representations with more locality. Our experiments demonstrate that the FCLF-CNN can achieve a better classification performance than pure multi-layered perceptrons (MLP) and pure CNNs.

The remainder of this paper is organized as follows. In Section II, we briefly discuss MLP and CNNs. In Subsection C of Section II, we focus on the FCLF-CNN. The performance evaluation methods are explained in Section III. Section IV presents the data, the experiments and the results. We analyze some additional data and provide a discussion about these findings in Section V. The conclusion and suggestions for future research are summarized in Section VI.

II. METHOD

A. MLP

A MLP with 2 hidden layers is shown in Fig. 1. As shown in the figure, the layers in a MLP are composed of many neurons. Each neuron in one layer is connected to every neuron in the next layer through a weighted connection. The first layer in the figure is referred to as the input layer. The second and third layers are called hidden layers because they have no connection with the outside world. The last layer is known as the output layer.

1) FULLY-CONNECTED LAYERS

All layers in a MLP are fully-connected. We used $\boldsymbol{y} = fc(\boldsymbol{x}, \boldsymbol{w}, \boldsymbol{b})$ to denote the function operated by the fully-connected layer, where $\boldsymbol{x}$ represents the input to the fully-connected layer, $\boldsymbol{w}$ denotes the weight matrix, $\boldsymbol{b}$ is the bias and $\boldsymbol{y}$ denotes the output. The data $\boldsymbol{x}$ has M features and the element x_i denote feature i. Note that $\boldsymbol{w}$ has dimension $M \times K$, where K represents the dimension of $\boldsymbol{y}$. It operates on a vector $\boldsymbol{x}$, generating the element i' in $\boldsymbol{y}$ as follows

$$y_{i'} = \sum_i w_{ii'} x_i + b_{i'} \tag{1}$$

2) ACTIVATION FUNCTION

We used the Rectified Linear Unit (ReLU) as the activation function in this paper. It works in an element-wise manner as follows

$$y_i = \max\{0, x_i\} \tag{2}$$

3) LOSS FUNCTION

We use $\{(\boldsymbol{x}^{(1)}, y^{(1)}), (\boldsymbol{x}^{(2)}, y^{(2)}), \ldots, (\boldsymbol{x}^{(m)}, y^{(m)})\}$ to represent the m training examples, where $y^{(i)}$ denotes the label for sample i. The neural network can be trained using gradient descent. In this article, cross-entropy loss function is used for supervised training. The cross-entropy loss function for a single example can be defined as

$$J(\boldsymbol{w}, \boldsymbol{b}; \boldsymbol{x}^{(i)}, y^{(i)}) = -\sum_{j=1}^{k} 1\{y^{(i)} = j\} \log \frac{e^{h_j(\boldsymbol{x}, \boldsymbol{w}, \boldsymbol{b})}}{\sum_{l=1}^{k} e^{h_l(\boldsymbol{x}, \boldsymbol{w}, \boldsymbol{b})}} \tag{3}$$

where $h_j(\boldsymbol{x}, \boldsymbol{w}, \boldsymbol{b})$ denotes the jth neurons in the output layer corresponding to class j and $1\{\cdot\}$ is the indicator function. Another loss function used in this article is the MSE loss function, as defined in Equation 6.

B. CNN

A CNN includes at least one convolutional layer. Different from the fully-connected layer in MLP, the convolutional layer is locally connected. Fig. 2 shows a typical 2 dimensional (2D) CNN used to recognize images. In the 2D CNN, the convolutional layers and pooling layers are operated in 2D space.

However, WDBC and WBCD datasets are both structured datasets, which have 30 and 9 features, respectively. Each sample in these datasets is a vector and the position of each feature is insignificant. So, we changed the convolutional layer and the pooling layer to operate in 1D space, which will be explained in the next section. In this case, all of the latent features are organized in the form of vectors. The architecture of a 1D CNN is presented in Fig. 3. We introduce the 1D convolutional layers and the 1D pooling in this subsection. The 2D convolutional layers and 2D pooling layers do not need to be explained because they have been explained extensively in previous research.

1) 1D CONVOLUTIONAL LAYERS

We used $\boldsymbol{y} = conv1d(\boldsymbol{x}, \boldsymbol{w}, \boldsymbol{b})$ to denote the 1D convolutional function operated by the convolutional layer, where $\boldsymbol{x}$ represents the input to the convolutional function, $\boldsymbol{w}$ denotes the filters, and $\boldsymbol{y}$ denotes the output of the convolutional layer. The data $\boldsymbol{x}$ has $M \times K$ dimensions, where M represents the number of latent features, and K is the number of channels. Note that $\boldsymbol{w}$ has the dimension $M_f \times K \times K'$, where M_f is the filter size. It operates on a latent vector $\boldsymbol{x}$ with K channels, generating $\boldsymbol{y}$ including K' feature vectors as follows

$$y_{i'k'} = \sum_{ik} w_{ikk'} x_{i+i',k} + b_{k'} \tag{4}$$

2) 1D POOLING

The pooling layer is another important operator in a CNN to introduce non-linearity. A pooling operator runs on individual feature channels, coalescing nearby feature values

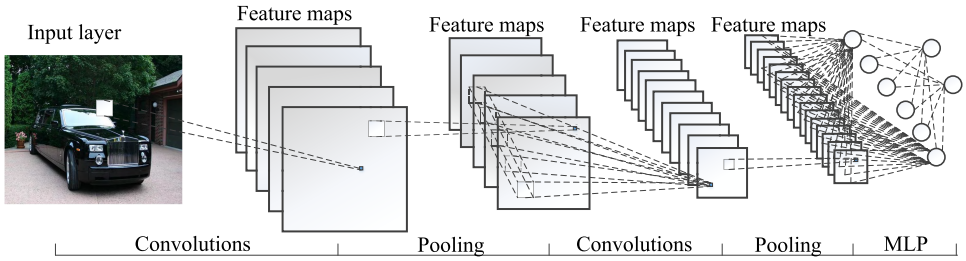


FIGURE 2. A typical 2D CNN for image data.

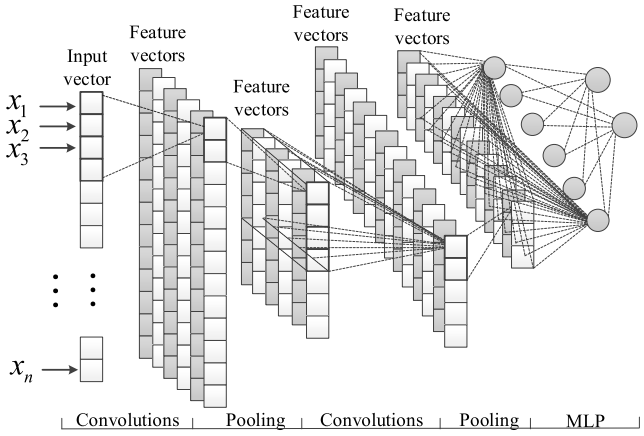


FIGURE 3. A 1D CNN for structured data.

into one, by the application of a suitable operator. Common choices include max-pooling or average-pooling. We used max-pooling in this paper, which is defined as

$$y_{ik} = \max\{x_{i'k} : i \leq i' < i + p\} \tag{5}$$

C. FCLF-CNN

The 2D CNN is more suitable than MLP to extract features of image data. The image data has an explicit local structure (or locality). In more detail, in some local regions, the pixel intensity does not significantly change, with any great change occurring at the boundary between adjacent local regions. Each element of the feature map in a 2D CNN can represent a local region in raw samples, which we call a local feature. Therefore, a 2D CNN can extract local features in the image data, which results in the 2D CNN getting better classification performance than the MLP for image data. To confirm these results, we used the MNIST dataset to perform a comparative experiment. Specifically, we first used a 2D CNN and a MLP with 2 hidden layers to classify the MNIST dataset and achieved an error rate of 0.7% and 1.6%, respectively. It is therefore evident that the 2D CNN performed much better than the MLP. We then randomly disrupted the position of each pixel to destroy the local structure of the image. For the disrupted data, both of the networks that are same to previous ones achieved an error rate of 2.6% and 1.7%, respectively. It was therefore determined that disrupting the local structure in the MNIST data has a negative influence on the performance of 2D CNN, but it has little effect on the performance of the MLP. This is because when you randomly disrupt the pixel position, it does not result in the image data losing its overall pattern.

To improve the classification performance of the disrupted data, we forced them to approximate some image data (the MNIST dataset) by the network shown in Fig.5. In this network, the input layer is first connected to one or several fully-connected layers followed by a typical CNN. Therefore, we call it fully-connected layer first CNN (FCLF-CNN). We used the fully-connected layer as an approximator, and let $\boldsymbol{c}$ represent the data that has an explicit local structure. We then set up a MSE loss function for the fully-connected layer as follows

$$J(\boldsymbol{w}, \boldsymbol{b}; \boldsymbol{c}) = \frac{1}{2} \sum_{j=1}^{n} (h_j(\boldsymbol{x}, \boldsymbol{w}, \boldsymbol{b}) - c_j)^2 \tag{6}$$

where c_j denotes the jth member in $\boldsymbol{c}$, $h_j(\boldsymbol{x}, \boldsymbol{w}, \boldsymbol{b})$ is the output of the fully-connected layer, and n is the number of the member. For the MNIST data, $n = 28 \times 28$. In this case, note that for the FCLF-CNN, fully-connected layers exist both in front and behind the convolutional layers. However, they are slightly different. The fully-connected layer before the convolutional layer, operates on the nodes only in one channel, while the fully-connected layer behind the convolutional layer operates on all nodes, in all channels. For the disrupted MNIST dataset, the FCLF-CNN can achieve an error rate of 0.9%, which is better than the typical 2D CNN and MLP. It reveals that the FCLF-CNN has a positive effect for the disrupted image data. Structured data is similar to disrupted image data, which seems to have no local structure at all. Therefore, we believe that the FCLF-CNN can also improve the classification performance for structured data.

We also adopted an additional method, to use the fully-connected layer as an encoder. We expect that the encoder can transfer the raw data into representations with better local structure, which we found can be achieved by adding a softmax loss to the fully-connected layer (see Fig. 10 and Fig. 11). Therefore, we think that the FCLF-CNN that uses the fully-connected layer as an encoder, should also have better performance outcomes than MLP and 1D CNN.

Therefore, we used two architectures for FCLF-CNN, including 1D FCLF-CNN and 2D FCLF-CNN. Note that,

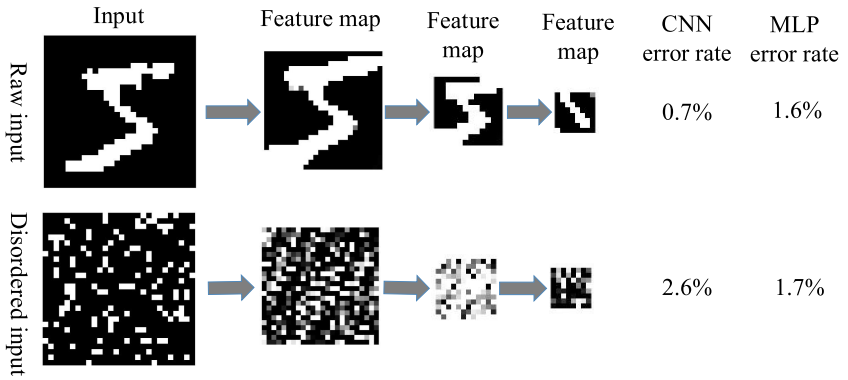


FIGURE 4. The classification error rate of a CNN and a MLP for the raw input and the disrupted input.

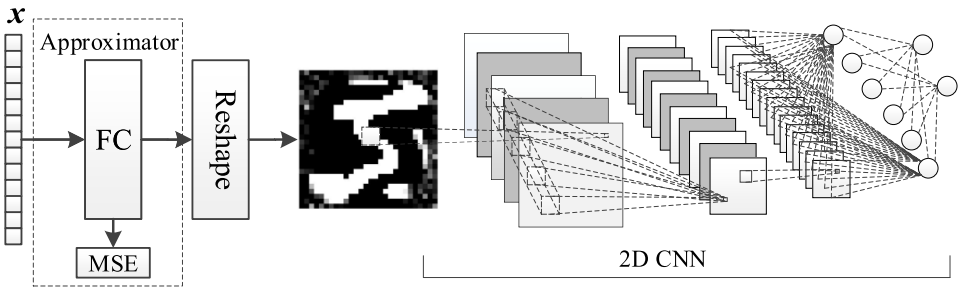


FIGURE 5. The architecture of the 2D FCLF-CNN.

there is a reshaping layer in the 2D FCLF-CNN, which is used to transform a 1D representation into a map (can also be seen as a 2D representation). In fact, FCLF-CNN is a stacking method, which stacks a approximator or an encoder with a CNN. We expect that this stacking method can provide a better performance than pure MLP and pure CNN, particularly for structured data.

III. PERFORMANCE EVALUATION

A. SENSITIVITY, SPECIFICITY AND ACCURACY

Data in both of the datasets WDBC and WBCD are divided into two classes benign (the negative class) and malignant (the positive class). Sensitivity, specificity, and accuracy are calculated using the True positive (TP), true negative (TN), false negative (FN), and false positive (FP) according to (7), (8), and (9). TP is the number of positive cases that are classified as positive. FP is the number of negative cases that are classified as positive. TN is the number of negative cases classified as negative and FN is the number of positive cases classified as negative.

$$Sensitivity = \frac{TP}{TP + FN} \tag{7}$$

$$Specificity = \frac{TN}{TN + FP} \tag{8}$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{9}$$

B. CROSS VALIDATION

We used a 5-fold cross validation to confirm sensitivity, specificity and accuracy. Specifically, the dataset was randomly divided into 5 subsets. One of the subsets was used as the validation set and the other 4 subsets were used as the training set. We sought to identify the value of TP, TN, FP, and FN and calculate sensitivity, specificity, and accuracy, according to (7), (8), and (9). All of the 5 subsets took turns as the validation set, which lead to 5 results for each model. We averaged them out to identify the result through a cross validation. We ran the cross validation 10 times and averaged these 10 results to confirm the final result for each model.

IV. EXPERIMENTS

A. DATA

The WBCD [15] and WDBC [16] datasets are from the University of California Irvine (UCI), Machine Learning Repository. The WDBC dataset consists of 569 Fine Needle Aspirate biopsy samples of human breast tissues, including 357 (62.7%) benign samples and 212 (37.3%) malignant samples. There are 30 features computed for each cell sample, which are the mean value, the extreme value, and standard error of 10 important attributes. The 10 attributes are tabulated in Table 1.

The WBCD dataset consists of 699 instances taken from human breast tissue. Each record in the database has

TABLE 1. Description of features in WDBC dataset.

Number	Description of feaures
1	radius
2	texture
3	perimeter
4	area
5	smoothness
6	compactness
7	concavity
8	concave points
9	symmetry
10	fractal dimension

TABLE 2. Description of features in WBCD dataset.

Number	Description of feaures
1	Clump thickness
2	Uniformity of cell size
3	Uniformity of cell shape
4	Marginal adhesion
5	Single epithelial cell size
6	Bare nuclei
7	Bland chromatin
8	Normal nucleoli
9	Mitoses

9 attributes, with these shown in Table 2. The measurements are assigned as an integer value between 1 and 10, with one being the closest to benign, and 10 being the most anaplastic. The label on each sample is either benign or malignant. There are 16 missing values in this dataset, which all belong to the 6th attribute. We filled these values with the linear regression method. The class has a distribution of 458 (65.5%) benign samples and 241 (34.5%) malignant samples.

B. DATA PREPROCESSING

Firstly, all of the data was preprocessed according to the following equation:

$$\hat{x}_{ij} = \frac{x_{ij} - mean(x_{i\cdot})}{std(x_{i\cdot})} \tag{10}$$

whereby, x_{ij} is the value of feature i of sample j. $mean(x_{i\cdot})$ and $std(x_{i\cdot})$ denote the mean and the standard deviation of feature i, respectively. We augmented the dataset during a cross validation. Specifically, a value randomly generated from the normal distribution $N(0, 0.01)$ is added to each feature of the training sample. We augmented the training set 7 times, with the validation set remaining unchanged.

C. SETUP

We conducted experiments on 7 models, including MLP1, MLP2, CNN, 1D FCLF-CNN training simultaneously (1D_FCLF-CNN_ST), 1D FCLF-CNN training in step-wise (1D_FCLF-CNN_SW), 2D FCLF-CNN training simultaneously (2D_FCLF-CNN_ST), and 2D FCLF-CNN training in step-wise (2D_FCLF-CNN_SW). The architectures of them are shown in Table 3. Note that, only 2 kinds of architectures are used for FCLF-CNN, including 1D FCLF-CNN and 2D FCLF-CNN. In the 1D FCLF-CNN, the fully-connected layers before the convolutional layers are seen as an encoder, while they are seen as an approximator in the 2D FCLF-CNN. Moreover, we used two different training methods for each of them. This is because there are two loss functions for each architecture (see Fig. 5 and Fig. 6), one is set for the layer before the CNN layer and one is set for the last layer. Simultaneous training refers to the entire FCLF-CNN as a multi-objective model, with all of the parameters trained at the same time. In contrast, step-wise training goes through three stages: the first stage is training the fully-connected layers before CNN; the second stage is training the subsequent CNN; and the third stage is training the FCLF-CNN as a whole, with this stage seen as fine-tuning.

TABLE 3. The Architecture for each network.

Network	Architecture
MLP1	$ReLU(fc(\boldsymbol{x}, \boldsymbol{w})) + softmax$
MLP2	$2 \cdot ReLU(fc(\boldsymbol{x}, \boldsymbol{w})) + softmax$
CNN	$Pooling(ReLU(Inception1d(\boldsymbol{x}, \boldsymbol{w}))) + fc + softmax$
1D FCLF-CNN	$Linear(fc(ReLU(fc(\boldsymbol{x}, \boldsymbol{w})))) + 1DCNN$
2D FCLF-CNN	$Linear(fc(ReLU(fc(\boldsymbol{x}, \boldsymbol{w})))) + Reshape + 2DCNN$

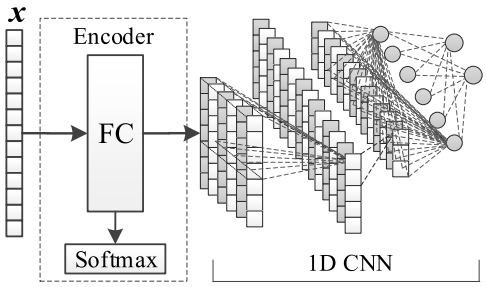


FIGURE 6. The architecture of the 1D FCLF-CNN.

Using different architectures and training methods can reduce the correlation between these 4 FCLF-CNNs, so that we can expect their ensemble models to achieve better results than any single network. A MLPn represents a MLP with n hidden layers ($n = 1$, 2 or 3). Each FCLF-CNN in this study has 2 fully-connected layers before the CNN. All of the activation functions used ReLU except for the fully-connected layer before the convolutional layers. And in this layer, linear activation functions was used to make the output approach $\boldsymbol{c}$ as well as possible.

For the convolutional layers in CNN and FCLF-CNN, we used the variants of Inception [17], that is, a bank of filters with different filter sizes. We used this because the Inception module has proved to be more effective than conventional CNN layers. The Inception module used in this study is shown in Fig. 7. For the WBCD dataset, the number of filters in each branch in the Inception module is 6, so the number of channels after the concatenation is 18. For the WDBC dataset,

TABLE 4. Results for WBCD dataset.

Network	Maximum accuracy(%)	Average accuracy(%)	Standard deviation	Sensitivity (%)	Specificity (%)
MLP1	97.56	96.94	0.30	96.62	98.12
MLP2	97.48	96.79	0.33	96.34	98.16
CNN	97.71	97.21	0.19	96.73	98.33
1D_FCLF_CNN_ST	98.41	97.63	0.18	97.51	98.89
1D_FCLF_CNN_SW	97.88	97.29	0.21	96.81	98.35
2D_FCLF_CNN_ST	98.56	97.81	0.21	97.48	99.13
2D_FCLF_CNN_SW	98.41	97.81	0.19	97.32	99.13
Ensemble FCLF-CNN	**98.71**	**98.41**	0.16	**97.60**	**99.43**

TABLE 5. Results for WDBC dataset.

Network	Maximum accuracy(%)	Average accuracy(%)	Standard deviation	Sensitivity (%)	Specificity (%)
MLP1	98.42	97.83	0.35	97.69	98.94
MLP2	98.59	97.56	0.36	97.69	99.09
CNN	98.24	97.80	0.22	97.43	98.78
1D_FCLF_CNN_ST	99.10	98.52	0.23	98.48	99.27
1D_FCLF_CNN_SW	98.59	98.01	0.24	97.83	99.16
2D_FCLF_CNN_ST	98.77	98.18	0.22	97.83	99.30
2D_FCLF_CNN_SW	98.42	97.83	0.21	97.69	99.16
Ensemble FCLF-CNN	**99.28**	**98.71**	0.18	**98.65**	**99.57**

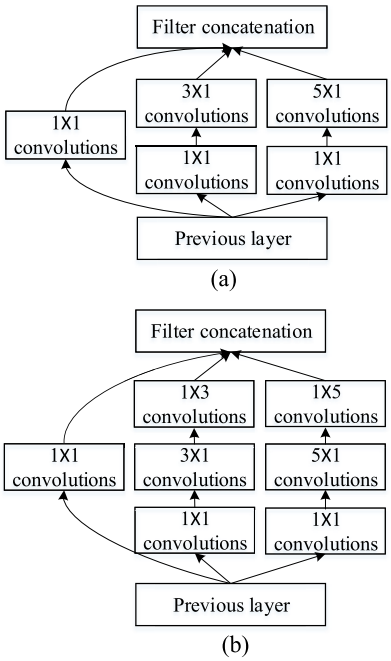


FIGURE 7. The Inception module in the 1D FCLF-CNN (a) and 2D FCLF-CNN (b).

we halved the number of channels because the WDBC data set had fewer features. We used keras [18] to build all of the FCLF-CNNs in this paper. We initialized the learning process with a learning rate of 1×10^{-3} and completed the learning process in 200 epochs (for the model training simultaneously) using a batch size of 16. The momentum was fixed to 0.9 with weight decay parameters set to 5×10^{-4} throughout the learning process. All of the pooling sizes and pooling strides were set to 2, while all the convolutional strides were set to 1. We completed the 3 stages in model training in step-wise in 200, 100, and 100 epochs, respectively.

In order to compare the classification performance between these models, we trained 100 architectures for each model. Specifically, for MLP1 and MLP2, we changed the number of hidden nodes from one to 100. The two hidden layers in MLP2 were set to be the same number. For CNN, we changed the number of nodes in the last fully-connected layer from 1 to 100. For 1D FCLF-CNN and 2D FCLF-CNN, the number of nodes in the first fully-connected layer was changed from 1 to 100. For each model, we calculated the maximum, mean, and standard deviation of the classification accuracy. These can be viewed in Table 4 and Table 5. The sensitivity and the specificity corresponding to the maximum accuracy are also shown in these tables. We observed that FCLF-CNN achieves a higher accuracy than pure MLP and pure CNN. Additionally, we ensembled the 4 FCLF-CNN models. The ensemble method used in this paper was used to add and then average the output of the last softmax layer of the 4 FCLF-CNN models. The results demonstrate that this ensemble method achieved better results. This is due to 1) the better performance of any single FCLF-CNN, and 2) the low correlation between the 4 FCLF-CNNs.

The results in Table 4 and Table 5 reveal that for structured data, pure CNN does not always work better than MLP. However, the performance of FCLF-CNN as proposed in this paper is better than MLP and CNN for both datasets. The accuracy of MLP1, CNN, and ensemble FCLF-CNN are shown in Fig. 8 and Fig.9, respectively. MLP1 is preferred, because the results shown in Table 4 and Table 5 reveal

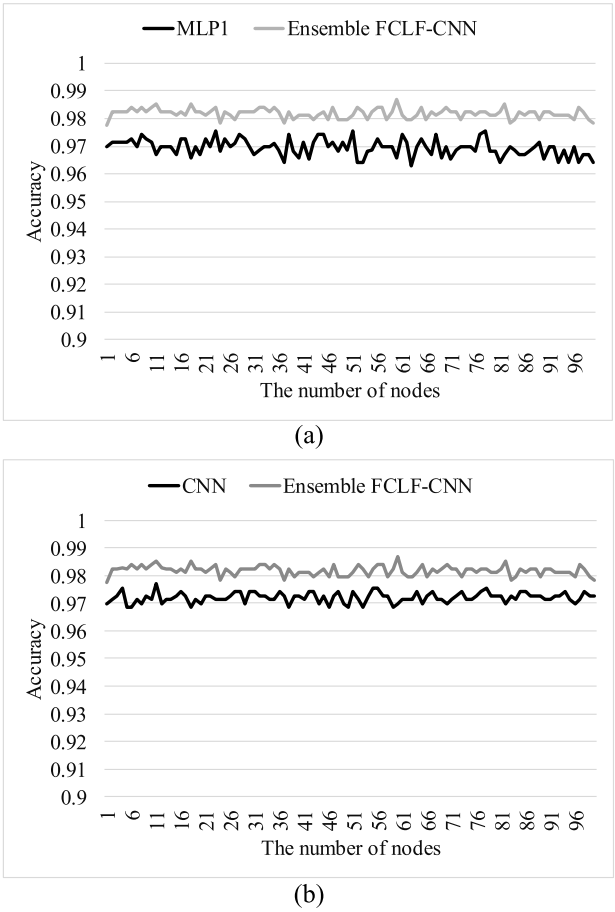


FIGURE 8. The comparison of the accuracy on WBCD dataset: Ensemble FCLF-CNN *vs.* MLP_1 (a) and Ensemble FCLF-CNN *vs.* CNN (b).

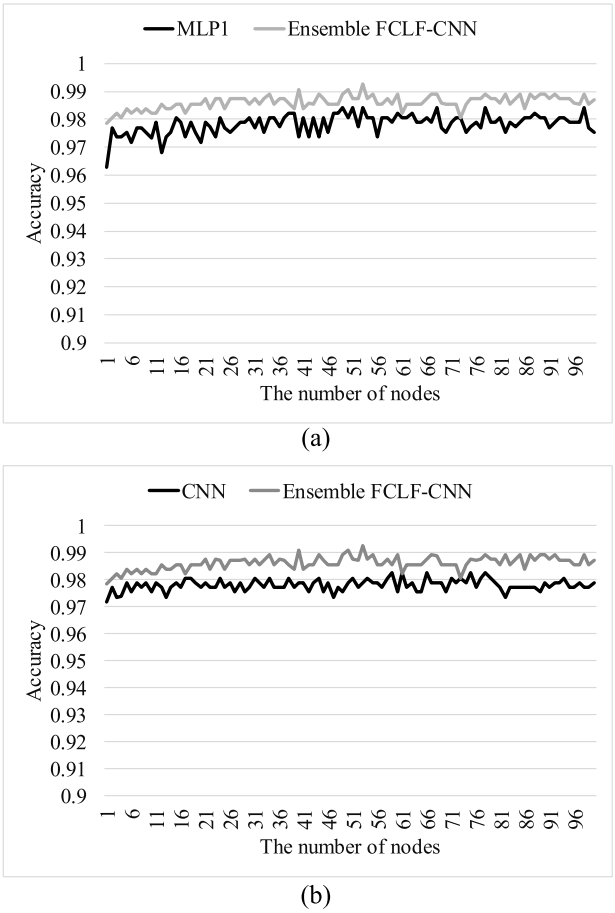


FIGURE 9. The comparison of the accuracy on WDBC dataset: Ensemble FCLF-CNN *vs.* MLP_1 (a) and Ensemble FCLF-CNN *vs.* CNN (b).

that its performance is better than MLP2, for both datasets. We observed that the accuracy curve of the ensemble FCLF-CNN is above that of the other 2 models. These results reveal that it is not accidental that the performance of FCLF-CNN is better than pure MLP and pure CNN.

Many methods have been applied to classify breast cancer data in WDBC and WBCD datasets. But a fair comparison is difficult, due to the lack of standard training sets and testing sets. Some researchers like to randomly pick the training data and the testing data with a certain proportion. Different researchers used different samples and data dividing methods. We chose to use the 5-fold cross validation method to obtain key measurements such as accuracy, sensitivity, and specificity. This is because the cross validation method provides every sample with the opportunity to be a testing sample. However, we still compared the best accuracy obtained in our experiments with the previous algorithms that have been applied to the same dataset, as shown in Table 6 and Table 7. We also show how the data is divided to get the accuracy in both of these tables. Our results were found to be better than previously used methods. It should be noted that there is a large contingency for methods that randomly splits data into training sets and testing sets.

V. DISCUSSION

A. THE PERFORMANCE ON OTHER DATASETS

In order to test FCLF-CNN on other structured datasets, we chose 5 datasets from the UCI Machine Learning Repository, including Cardiotocography (CD), Diabetic Retinopathy Debrecen (DRD), Sensorless Drive Diagnosis (SDD), Thoracic Surgery (TS), and Yeast Data (YD). For the FCLF-CNN, we used the same experimental procedure and parameter settings as used for WDBC. Note that, we did not use the MNIST data as the approximation object in this section, because the Sensorless Drive Diagnosis dataset has more than 10 categories. Therefore, we painted these images by hand (See Fig. 13 (a)). The results are shown in Table 8.

For a comparison, we also ran MLPs with 1, 2, and 3 hidden layers. This is because these 5 datasets have a larger data volume than WBCD and WDBC. We observed that an MLP with 3 hidden layers is no longer performing better than an MLP with two hidden layers. FCLF-CNN is superior to pure MLP and pure CNN, both in terms of average and maximum accuracy.

We also used other non-neural network models to classify these data sets, including Radial Basis Function-Support Vector Classifier (RBF-SVC) [30], linear-SVC [31], Random

TABLE 6. The classification accuracy comparison on WBCD dataset.

Method	Accuracy (%)
LDA [19]	96.80 (10-fold cross validation)
C4.5 [20]	94.74 (10-fold cross validation)
SVM [21]	97.20 (5-fold cross validation)
SFC [22]	97.38 (10-fold cross validation)
Self-training [23]	85.86 (10-fold cross validation)
Rough co-training [23]	92.39 (10-fold cross validation)
LDA [24]	95.61 (train:50%, test:50%)
C4.5 [24]	95.59 (train:50%, test:50%)
DIMLP [24]	96.68 (train:50%, test:50%)
SIM [24]	97.61 (train:50%, test:50%)
KDE [20]	98.53 (train:50%, test:50%)
W-NB [1]	98.54 (5-fold cross validation)
Ensemble FCLF-CNN	**98.71** (5-fold cross validation)
	98.57 (train:50%, test:50%)
	98.86 (train:75%, test:25%)

TABLE 7. The classification accuracy comparison on WDBC dataset.

Method	Accuracy (%)
CfS+LR [25]	95.95 (train:75%, test:25%)
Filtered+LR [25]	96.62 (train:75%, test:25%)
KP-SVM [26]	97.55 (train:60%, test:40%)
REF-SVM [26]	95.25 (train:60%, test:40%)
FSV [26]	95.23 (train:60%, test:40%)
Self-training [23]	85.12 (10-fold cross validation)
Rough co-training [23]	88.63 (10-fold cross validation)
BPSO-2Stage [27]	92.98 (train:70%, test:30%)
PSO [28]	93.98 (train:70%, test:30%)
LDA [24]	97.19 (train:50%, test:50%)
C4.5 [24]	94.06 (train:50%, test:50%)
DIMLP [24]	96.92 (train:50%, test:50%)
SIM [24]	98.26 (train:50%, test:50%)
PSO-KDE,GA-KDE [29]	98.45 (train:50%, test:50%)
Ensemble FCLF-CNN	**99.28** (5-fold cross validation)
	98.95 (train:50%, test:50%)
	99.30 (train:75%, test:25%)

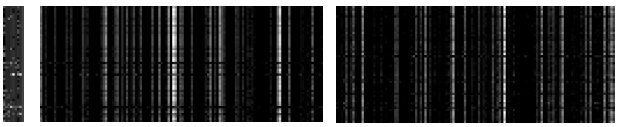
TABLE 8. Accuracy for some other 5 datasets in UCI Machine Learning Repository.

Model	CD(%)	DRD(%)	SDD	TS(%)	YD(%)
MLP1	85.37	73.28	98.83	85.11	60.72
MLP2	86.31	73.85	99.31	85.53	60.72
MLP3	85.61	73.5	98.97	85.32	60.1
CNN	85.84	75.50	98.83	85.11	60.51
mean_MLP_CNN	85.78	74.03	98.98	85.27	60.51
1D_FCLF_CNN_ST	85.79	75.59	98.73	85.53	61.99
1D_FCLF_CNN_SW	87.11	75.67	99.44	85.32	61.99
2D_FCLF_CNN_ST	86.64	74.63	99.39	85.32	62.47
2D_FCLF_CNN_SW	87.3	76.02	99.76	85.32	62.33
mean FCLF-CNN	86.71	75.48	99.34	85.37	62.20

Forest Classifier (RFC) [32], Gradient Boosting Classifier (GBC) [33], Adaboost [34], and Xgboost [35]. We compared them with the ensemble FCLF-CNN. The results are shown in Table 9. The ensemble FCLF-CNN seems not only more competitive than any single FCLF-CNN, but also better than other non-neural network models. For the completion of the non-neural network, we used the scikit-learn library [36].

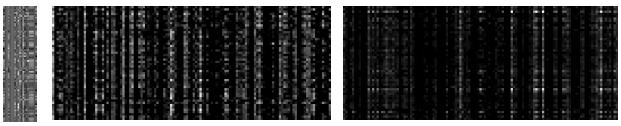
TABLE 9. Results for some other 5 datasets with other models and ensemble of FCLF-CNN.

Model	CD(%)	DRD(%)	SDD	TS(%)	YD(%)
rbf-SVC	84.95	72.54	98.59	85.11	59.1
linear-SVC	82.31	73.94	93.25	85.11	59.9
RFC	85.28	69.94	99.07	85.74	62.33
GBC	88.38	73.41	99.44	85.11	61.93
adaboost	87.63	69.24	99	84.68	60.78
XGBoost	89.04	74.76	99.83	85.32	61.12
Ensemble FCLF-CNN	**89.46**	**76.13**	**99.85**	**85.95**	**63.27**



(a) (b) (c)

FIGURE 10. The representations of 1D FCLF-CNN for WBCD, including the raw data (a), the representation output by the fully-connected layer before the convolutional layer in 1D_FCLF-CNN_ST (b), and the representation output by the fully-connected layer before the convolutional layer in 1D_FCLF-CNN_SW (c).



(a) (b) (c)

FIGURE 11. The representations of 1D FCLF-CNN for Cardiotocography, including the raw data (a), the representation output by the fully-connected layer before the convolutional layer in 1D_FCLF-CNN_ST (b), and the representation output by the fully-connected layer before the convolutional layer in 1D_FCLF-CNN_SW (c).

B. THE LOCAL STRUCTURE IN THE REPRESENTATION

In this section, we wanted to observe what the representation obtained by the fully-connected layer before the convolutional layer in FCLF-CNN, that is, whether it can give representations with more locality than raw data.

Firstly, for 1D FCLF-CNN, Fig. 10 and Fig. 11 show their observations on WBCD and Cardiotocography, respectively. Note that one row corresponds to one sample. The corresponding raw samples are also shown. We observed that there is a certain pattern in the raw data, but it is not obvious. After encoding the raw data through the fully-connected layer, a representation of a stronger pattern is obtained. Note that, this representation also obtained a better locality. However, compared to the representation of Cardiotocography, the representation of WBCD has a stronger pattern and locality, indicating that the locality of the representation is also directly related to the raw data.

For 2D FCLF-CNN, Fig. 12 and Fig. 13 show the observations on WBCD and Cardiotocography, respectively. We observed that:

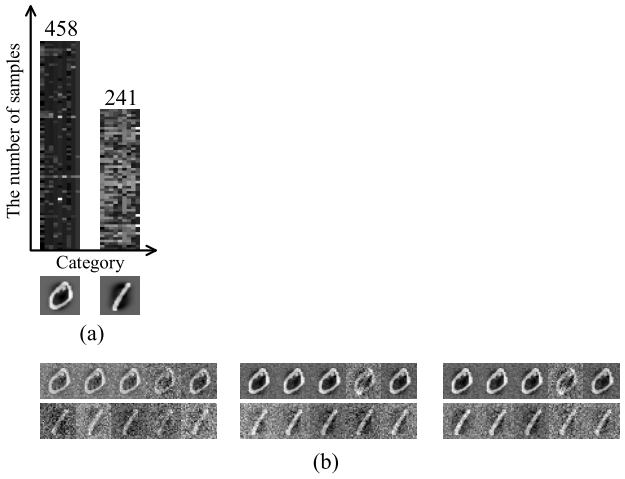


FIGURE 12. The 2D FCLF-CNN for WBCD. The raw data and the number corresponding to each category are shown in (a). Note that, one row corresponds to one sample. The output of the reshaped layer in each training method is shown in (b), including 2D_FCLF-CNN_ST (left), 2D_FCLF-CNN_SW before fine tuning (middle) and 2D FCLF-CNN after fine tuning (right).

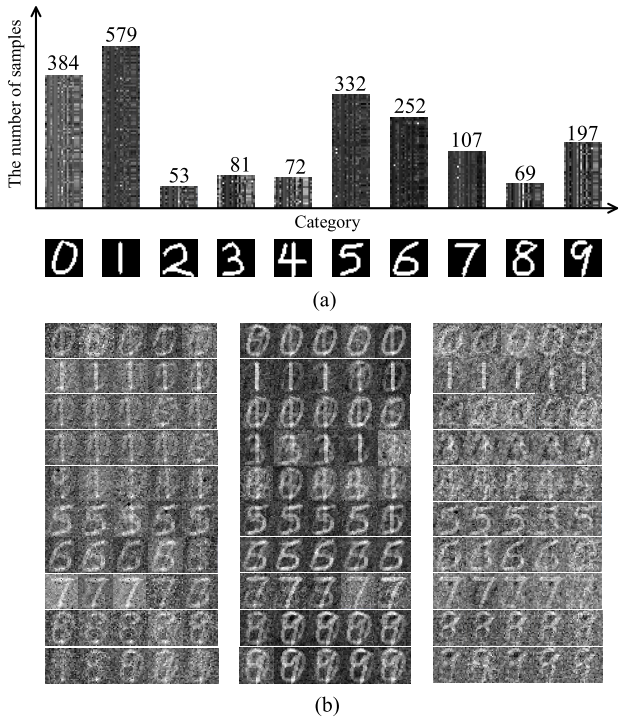


FIGURE 13. The 2D FCLF-CNN for Cardiotocography. The raw data and the number corresponding to each category are shown in (a). Note that, one row corresponds to one sample. The output of the reshape layer in each training method is shown in (b), including 2D_FCLF-CNN_ST (left), 2D_FCLF-CNN_SW before fine-tuning (middle) and 2D_FCLF-CNN after fine tuning (right).

1) The fully-connected layer really make the raw data approximate the image data through the MSE loss function, and takes the image data as the approximation target.

2) However, the effect of the approximation has a great relationship with the raw data. The WBCD's raw data provided a better approximation than Cardiotocography. Note that, the accuracy of WBCD is higher than Cardiotocography for the same model, meaning that the separability of WBCD is higher than Cardiotocography.

3) The effect of the approximation of one category depends largely on the number of samples belonging to it. For the category with a relatively large number of samples, the parameters can be adequately trained to approximate the corresponding image. On the contrary, for the category with a relatively small number of samples, the parameters cannot be adequately trained to approximate the corresponding image.

4) For some samples in the Cardiotocography dataset, the output of the reshape layer are mixed with multiple numbers. This makes the samples between some categories output more similar in approximation, and the subsequent CNN cannot make the correct classification. Even so, the model achieved a better performance than pure MLP and pure CNN.

5) Fine tuning seems to undermine the approximation of the image data, but results in the data of each category having its own (different from other categories) features.

VI. CONCLUSION

In this study, FCLF-CNN is proposed to improve the classification performance of the datasets WBCD and WDBC. The model embeds fully-connected layers before the convolutional layer. By disrupting the MNIST data, we found that the local structure results in the CNN getting a better performance for image data. We utilized two ways to transfer the structured data into representations with a better local structure. One way is to use the fully-connected layer as an encoder by using the softmax loss to the fully-connected layers before the convolutional layer. The architecture of this way corresponds to 1D FCLF-CNN (Fig.6). Another way is to use the fully-connected layer as an approximator by using the MSE loss, to make the raw data approximate the image data. The architecture of this way corresponds to 2D FCLF-CNN (Fig.5). Both of these methods are stacking methods. For each architecture, we took two kinds of training methods, including training simultaneously and training step-wise. The results demonstrate that FCLF-CNN can achieve a better performance than pure MLP and pure CNN. We ensembled 4 FCLF-CNN models, with its resulting performance better than any single FCLF-CNN network. Our results are obtained by a 5 fold cross validation. Although it is difficult to have a fair comparison with other researchers about these two datasets, the results of our experiments are still competitive compared to the results of other studies.

In terms of the complexity of the model, it is true that FCLF-CNN is more complex than the traditional CNN and MLP. But we do not think it is very bad. On the one hand, because of the rapid development of hardware, the computational complexity of this level can not be the main contradiction. Second, in the medical field, in addition to structured data, the proportion of image data is constantly rising. Combining structured data with unstructured data for joint diagnosis is an effective way to increase diagnostic

performance. In fact, the results in this paper reveal that the CNN does not only work for unstructured data, but also for structured data, as long as slight changes are made to the CNN. Therefore, it is possible to combine the structured data and the unstructured data in the same CNN network. In the medical field, a patient may be required to undertake a number of examinations and tests, including CT scans and blood tests. So, the data used to describe a patient may contain structured data and unstructured data. Integrating these data together in a unified network may lead to more comprehensive and accurate results.

REFERENCES

[1] M. Karabatak, "A new classifier for breast cancer detection based on Naïve Bayesian," *Measurement*, vol. 72, pp. 32–36, Aug. 2015.

[2] W. H. Wolberg and O. L. Mangasarian, "Multisurface method of pattern separation for medical diagnosis applied to breast cytology," *Proc. Nat. Acad. Sci. USA*, vol. 87, no. 23, pp. 9193–9196, 1990.

[3] M. Iranpour, S. Almassi, and M. Analoui, "Breast cancer detection from FNA using SVM and RBF classifier," in *Proc. 1st Joint Congr. Fuzzy Intell. Syst.*, 2007, pp. 1–7.

[4] T. C. André and R. M. Rangayyan, "Classification of breast masses in mammograms using neural networks with shape, edge sharpness, and texture features," *J. Electron. Imag.*, vol. 15, no. 1, p. 013019, 2006.

[5] R. do Espírito Santo, R. de Deus Lopes, and R. M. Rangayyan, "Classification of mammographic masses using radial basis functions and simulated annealing with shape, acutance, and texture features," *Database*, vol. 164458, p. 175, Feb. 2005.

[6] H. D. Cheng and M. Cui, "Mass lesion detection with a fuzzy neural network," *Pattern Recognit.*, vol. 37, no. 6, pp. 1189–1200, 2004.

[7] M. K. Markey, J. Y. Lo, G. D. Tourassi, and C. E. Floyd, Jr., "Self-organizing map for cluster analysis of a breast cancer database," *Artif. Intell. Med.*, vol. 27, no. 2, pp. 113–127, 2003.

[8] L. Wei, Y. Yang, R. M. Nishikawa, and Y. Jiang, "A study on several machine-learning methods for classification of malignant and benign clustered microcalcifications," *IEEE Trans. Med. Imag.*, vol. 24, no. 3, pp. 371–380, Mar. 2005.

[9] D. Ciregan, U. Meier, and J. Schmidhuber, "Multi-column deep neural networks for image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2012, pp. 3642–3649.

[10] O. Russakovsky *et al.*, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015.

[11] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 1440–1448.

[12] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 91–99.

[13] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 3431–3440.

[14] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent*, 2015, pp. 234–241.

[15] O. L. Mangasarian and W. H. Wolberg, "Cancer diagnosis via linear programming," Dept. Comput. Sci., Univ. Wisconsin–Madison, Madison, WI, USA, Tech. Rep. #958, 1990.

[16] W. N. Street, W. H. Wolberg, and O. L. Mangasarian, "Nuclear feature extraction for breast tumor diagnosis," *Proc. SPIE*, vol. 1905, pp. 861–871, Jul. 1993.

[17] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 2818–2826.

[18] F. Chollet *et al.* (2015). *Keras*. [Online]. Available: https://github.com/fchollet/keras

[19] B. Šter and A. Dobnikar, "Neural networks in medical diagnosis: Comparison with other methods," in *Proc. Int. Conf. Eng. Appl. Neural Netw.*, 1996, pp. 427–430.

[20] J. R. Quinlan, "Improved use of continuous attributes in C4.5," *J. Artif. Intell. Res.*, vol. 4, pp. 77–90, 1996.

[21] K. P. Bennett and J. A. Blue, "A support vector machine approach to decision trees," in *Proc. IEEE Int. Joint Conf. Neural Netw. IEEE World Congr. Comput. Intell.*, vol. 3. May 1998, pp. 2396–2401.

[22] J. Abonyi and F. Szeifert, "Supervised fuzzy clustering for the identification of fuzzy classifiers," *Pattern Recognit. Lett.*, vol. 24, no. 14, pp. 2195–2207, 2003.

[23] D. Miao, C. Gao, N. Zhang, and Z. Zhang, "Diverse reduct subspaces based co-training for partially labeled data," *Int. J. Approx. Reason.*, vol. 52, no. 8, pp. 1103–1117, 2011.

[24] P. Luukka and T. Leppälampi, "Similarity classifier with generalized mean applied to medical data," *Comput. Biol. Med.*, vol. 36, no. 9, pp. 1026–1040, 2006.

[25] S. M. H. Bamakan and P. Gholami, "A novel feature selection method based on an integrated data envelopment analysis and entropy model," *Proc. Comput. Sci.*, vol. 31, pp. 632–638, Jan. 2014.

[26] S. Maldonado, R. Weber, and J. Basak, "Simultaneous feature selection and classification using kernel-penalized support vector machines," *Inf. Sci.*, vol. 181, no. 1, pp. 115–128, 2011.

[27] B. Xue, M. Zhang, and W. N. Browne, "New fitness functions in binary particle swarm optimisation for feature selection," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jun. 2012, pp. 1–8.

[28] B. Xue, M. Zhang, and W. N. Browne, "Particle swarm optimisation for feature selection in classification: Novel initialisation and updating mechanisms," *Appl. Soft Comput.*, vol. 18, pp. 261–276, May 2014.

[29] R. Sheikhpour, M. A. Sarram, and R. Sheikhpour, "Particle swarm optimization for bandwidth determination and feature selection of kernel density estimation based classifiers in diagnosis of breast cancer," *Appl. Soft Comput.*, vol. 40, pp. 113–131, Mar. 2016.

[30] J. A. K. Suykens and J. Vandewalle, "Least squares support vector machine classifiers," *Neural Process. Lett.*, vol. 9, no. 3, pp. 293–300, Jun. 1999.

[31] M. A. Hearst, S. T. Dumais, E. Osman, J. Platt, and B. Scholkopf, "Support vector machines," *IEEE Intell. Syst. Appl.*, vol. 13, no. 4, pp. 18–28, Jul./Aug. 2008.

[32] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

[33] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *Ann. Statist.*, vol. 29, no. 5, pp. 1189–1232, 2001.

[34] Y. Freund and R. E. Schapire, "A desicion-theoretic generalization of on-line learning and an application to boosting," in *Proc. Eur. Conf. Comput. Learn. Theory*, 1995, pp. 23–37.

[35] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2016, pp. 785–794.

[36] F. Pedregosa *et al.*, "scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, Oct. 2011.

KUI LIU was born in Shandong, China, in 1986. He received the M.S. degree from the Shandong University of Science and Technology, Shandong, China, in 2010. He is currently pursuing the Ph.D. degree with the Beijing University of Posts and Telecommunications.

He was an Intern with the State Key Laboratory of Hisense digital multimedia in 2012. His research interests include image analysis and artificial intelligence in medical science and healthcare.

GUIXIA KANG received the M.S. degree from Tianjin University, Tianjin, China, and the Ph.D. degree in electrical engineering from the Beijing University of Posts and Telecommunications (BUPT), Beijing. She is currently a Professor of BUPT and the Director of Beijing International S&T Cooperation Base of smart medicine. She has expertise in the physical layer of 5G wireless systems and in the wireless e-Health systems.

From 2002 to 2004, she was a Research Scientist with the Future Radio Concept Department of Siemens, Munich, Germany. She is/was the Project Manager of several national projects such as Important National Science and Technology Specific Project, National 863 project, National Natural Science Foundation of China, and several international cooperation projects. She has authored one English book (Shaker Verlag, Germany), three Chinese books, and authored or co-authored over 100 journal and conference papers.

Dr. Kang was a recipient of ąřNew Century Talent of Ministry of Educationąś and ąřBeijing New Star of Science and Technologyąś. She was also a recipient of the First Prize of Science and Technology Award of China Communications Association and the Second Prize of Beijing Science and Technology Progress Award.

NINGBO ZHANG received the Ph.D. degree from the Beijing University of Posts and Telecommunications (BUPT) in 2010. He currently is an Associate Professor with BUPT. He has expertise in the physical layer of 5G wireless systems and the machine to machine communications in IoT.

From 2010 to 2014, he was a Senior Engineer with the Research and Development Wireless Department, Huawei Technologies. Since 2014, he has been the Project Manager of several national projects such as National Natural Science Foundation of China, National Science and Technology Major Project of China, and National 863 project. He has authored or co-authored over 40 journal and conference papers.

BEIBEI HOU was born in Henan, China, in 1992. She received the B.S. degree from Henan Normal University, Henan, in 2015. She is currently pursuing the M.S. and Ph.D. degrees with the Beijing University of Posts and Telecommunications. Her research interests include image analysis and pattern recognition in medical science and healthcare.