A Study of Neural-Network-Based Classifiers for Material Classification

H.K. Lam, *Senior Member, IEEE*, Udeme Ekong, Hongbin Liu *Member, IEEE*, Bo Xiao, Hugo Araujo, Sai Ho Ling, *Senior Member, IEEE* and Kit Yan Chan

Abstract

In this paper, the performance of the commonly used neural-network-based classifiers is investigated on solving a classification problem which aims to identify the object nature based on surface features of the object. When the surface data is obtained, a proposed feature extraction method is used to extract the surface feature of the object. The extracted features are then used as the inputs for the classifier. This research studies eighteen household objects which are requisite to our daily life. Six commonly used neural-network-based classifiers, namely one-against-all, weighted one-against-all, binary coded, parallel-structured, weighted parallel structured and tree-structured, are investigated. The performance for the six neural-network-based classifiers is evaluated based on recognition accuracy for individual object. Also, two traditional classifiers, namely k-nearest neighbor classifier and naive Bayes classifier, are employed for the comparison purposes. To evaluate robustness property of the classifiers, the original clean data is contaminated with Gaussian white noise. Experimental results show that the parallel-structured, tree-structured and the naive Bayes classifiers outperform the others under the noise-free data. The tree-structured classifier demonstrates the best robustness property under the noisy data.

Index Terms

Classifier, Material Classification, Neural Networks.

I. Introduction

This work was partially supported by King's College London, National Natural Science Foundation of China (61304003 and 61203002), Fundação para a Ciência e Tecnologia (grant number SFHR/BD/44162/2008) and European Social Fund in the POPH framework.

H.K. Lam, Udeme Ekong, Hongbin Liu, Bo Xiao and Hugo Araujois are with the Department of Informatics, King's College London, Strand, London, WC2R 2LC, United Kingdom. e-mail: {hak-keung.lam, udeme.ekong, hongbin.liu, bo.xiao, hugo.araujo}@kcl.ac.uk.

Sai Ho Ling is with the Centre for Health Technologies, Faculty of Engineering and Information Technology, University of Technology, Sydney, NSW, Australia. e-mail: Steve.Ling@uts.edu.au.

Kit Yan Chan is with the Department of Electrical and Computer Engineering, Curtin University, Perth, Australia. e-mail: kit.chan@curtin.edu.au.

Manuscript received 2013.

OBJECT classification aims to classify an unknown object into a pre-determined group which consists of a set of pre-classified objects with similar features to that unknown object. This is a very important field of study that has a diverse number of applications such as risk management of investment [1], hand-writing recognition [2] and speech recognition [3].

In the literature, the main methods of classification can be found as logic based method (e.g., decision trees), perceptron based methods (e.g., single layer perceptrons [4] and neural networks [5]), statistical approach (e.g., Bayesian classification [6]), instance-based methods (e.g., nearest neighbor algorithm [7], [8]) and support vector machine (SVM) methods [9].

The decision tree technique is a subset of the logic based classification method, it performs classification by sorting the inputs based on the inherent feature values. The nodes in a decision tree are representative of the feature values [10]. This method has been improving classification accuracy and interpretability of loan granting decisions [1]. The decision tree can improve the classification accuracy of the process, and also it is transparent and can be easily deciphered. Hence, for example, it is attractive for investment bankers who are required by law to give reason for a loan denial. The performance of the decision tree can be further improved by incorporating with neural networks, in order to utilize the distinct nature of processing adopted by both approaches [11]. However, if the splitting rule of the decision tree makes a wrong decision, it is impossible to return to the correct path resulting in an accumulation of errors. Also, an increase in the number of learned rules leads to the training algorithm trying to memorise the training set instead of discovering the rules that governing the patterns of it resulting in poor predictions.

A single layer perceptron [4] introduced by Rosenblatt in 1962 has created revolution in the artificial intelligence field, which has led to a number of perceptron-based techniques. A single layer perceptron can be simply described as a component that computes the sum of weighted inputs which is then fed to the output of the system. The outputs are then compared with the targets where the difference is employed to adjust the weights until the desired level of accuracy is derived. This approach demonstrates a major limitation that the single layer perceptron can only learn linearly separable problems, thus it is incompatible on addressing non-linearity. Despite the limitation, the single layer perceptron has been applied effectively on finger print matching [4] and image detection [12] applications.

Bayesian decision theory [6] is fundamental to statistical classification methods which provide a model for the classification procedures. The Bayesian classifier is based on the assumption that equal prior probabilities exists for all classes [13] which help in resolving conflicts that occur when two or more classes are not well separable resulting in improving the classification accuracy. However, the posterior probabilities cannot be determined directly [5]. The Bayesian classifier was applied successfully in weeds identification [6]. Recently, a hybrid Bayesian classifier [13] has been proposed, and the results demonstrated that the classification capability can be improved.

Another classification method is based on the k-nearest neighbour technique (kNN) [7] which is good in dealing with text based problems such as visual category recognition [8]. The basic principle is that objects in a data set generally exists in the neighbourhood of other objects with similar properties. The technique finds the “k” nearest objects to the particular input and determines its class by looking for the most frequent class label. In this technique, the distance between objects is more important than their individual positions. The main disadvantages of the kNN technique are the large memory requirements and the lack of a logical way of choosing “k”, this would make it difficult in a classification application as different data sets would require different optimized value of “k” in order to improve the performance of this method [10]. Furthermore, the precision accuracy can be reduced when there are too many classes or when an uneven density of training samples is presented. A clustering-based method is proposed in [14] to solve this problem as training data is being pre-processed via a clustering algorithm and then classified with a novel kNN algorithm that adjusts the value of “k” with each iteration.

Neural networks [5] consist of 3 distinct segments that the input units which have the primary responsibility of receiving information; the hidden units which carry out the processing and the output units which store the processed results [10]. The neural network is first trained on a set of data to determine the input-output mapping. The weights of connections between neurons are then established and the training network can then be used to classify a new set of data. The backpropagation algorithm [5] is a widely used method for training the neural network and improving its accuracy. It is done by calculating the error between the actual and desired output, adjusting the weights accordingly and then repeating the process until an acceptable level of accuracy is achieved. Neural network is non-linear in nature and demonstrates a universal approximation capability [15] which makes it ideal for dealing with complex input-output relationships such as classification problems. One of the classification applications of neural networks has been used in stock market prediction where different classification architectures were applied in the classification of system input (for example, historical stock market price) into “buy”, “sell” or “hold” advices for investing in the S&P 500 [16].

Support vector machines (SVMs) [9] are supervised learning methods that can be used for various data mining applications including classification and time-series analysis. The main concept of the SVMs is to obtain a hyperplane to separate two data classes. Mature theory has been developed to determine the optimal hyperplane by maximizing the distance between the hyperplane and the support vectors for reduction of generalization error for both linearly and nonlinearly separable cases. The solutions are unique and consistent and there are less occurrences of overfitting. However, it demonstrates a high algorithmic complexity and results are not transparent [10]. A novel method was proposed in [17] to tackle the problem of high complexity when large data sets are used. A two-phase approach is adopted. In the first phase, clustering techniques are applied to obtain approximate classes for all the input data, In the second phase, fine-tuning of the classified data by using the instances that are in close proximity to the approximated hyperplane obtained from the first phase is then performed.

In this paper, we consider a classification problem in material surface recognition of an unknown object using a contact sensing fingertip, which demonstrates a wide range of potential domestic and industrial applications, such as on robot-assisted surgery [18]–[21], blind grasping application [22], [23], pose classification [24], prosthetic limbs [25], quality assurance [26], shape extraction and industrial inspection [27], [28], and brain-machine-brain interface [29]. The properties of the object surface which are important for the aid of recognition are the frictional coefficients, texture, compliance and roughness. The data is obtained through an active surface exploration [30], [31] with the aid of contact-sensing fingertip which can accurately identify the normal and frictional force of the object. During the experiments, the contact sensing fingertip slides along the object with short strokes whilst increasing/decreasing the velocity as is appropriate. The properties of the vibrations caused by this action are then used as the input data. A feature vector is extracted from the raw data to reduce the number of data points used for the classification procedure. It is of utmost importance that the contact sensing fingertip is able to differentiate between the objects and that is the basis of emphasis and importance for the research being conducted in this paper.

In view of the superior learning and generalization capability of the neural networks, we are motivated to implement classifiers using neural networks to deal with the material classification problem [32]–[34]. In this study, the characteristics of the neural networks are considered for the implementation of neural-network-based classifiers, demonstrating different levels of flexibility, scalability and complexity. Six neural-networked-based classifiers, namely one-against-all, weighted one-against-all, binary coded, parallel structured, weighted parallel structured, tree-structured, are introduced for recognition of materials touched by the robot finger. In order to make a comparison, two traditional classification methods, namely k-nearest neighbor classifier and the naive Bayes classifier, are considered. Their recognition performance is investigated thoroughly using the dataset collected from experiments. To investigate the robustness property of the classifiers, Gaussian white noise is added to the test dataset and the recognition performance is evaluated. By investigating the recognition performance of the introduced classifiers, the most suitable classifier for the material surface classification problem can be recommended.

This paper is arranged as follow: After the introduction, we present the basic principles and theory behind the neural network in Section II. Section III presents the 6 neural network based classifiers and comments on their flexibility, scalability and complexity. The robustness of all the classifiers are also included. Section IV presents and discusses the results produced from the simulations under both the original testing data case and noisy data case. A conclusion is then drawn in Section V.

II. Neural Networks

In this section, a brief discussion of a fully-connected feed-forward neural networks with one input layer, n_l hidden layers and one output layer is considered. The t^{th} input of the neural network is given by $\mathbf{x}(t) = [x_1(t) \quad x_2(t) \quad \dots \quad x_{n_{in}}(t)]$ and the t^{th} output as $\mathbf{y}(t) = [y_1(t) \quad y_2(t) \quad \dots \quad y_{n_{out}}(t)]$ where n_{in} denotes the number of input nodes in the input layer and n_{out} denotes the number of output nodes in the output layer. The output of the j-th node in the input layer is given as follows:

$$f_i^{(0)}(t) = x_i(t), i = 1, 2, \dots, n_{in} \tag{1}$$

and the output of the j-th node in the n_l-th hidden layer is given as follows:

$$f_i^{(n_l)}(t) = tf_{n_l}\Big(\sum_{j=1}^{n_{n_h}^{(n_l-1)}} w_{ij}^{(n_l)} f_j^{(n_l-1)}(t) - b_j^{(n_l)} \Big), i = 1, 2, \dots, n_h^{(n_l)}, \tag{2}$$

where $tf_{n_l}(\cdot)$ denotes the transfer function; $n_h^{(n_l)}$ denotes the number of hidden nodes, $b_i^{(n_l)}$ denotes the bias in the n_l-th hidden layer; and $w_{ij}^{(n_l)}$ denotes the weight between the j-th node in the $n_h^{(n_l-1)}$-th hidden layer and the i-th node in the $n_h^{(n_l)}$-th hidden layer.

The output of the neural network is given as follows:

$$y_i(t) = tf_{n_l+1}\Big(\sum_{j=1}^{n_{n_h}^{(n_l)}} w_{ij}^{(n_l+1)} f_j^{(n_l)}(t) - b_j^{(n_l+1)}\Big), i = 1, 2, \ldots, n_{out} \tag{3}$$

Here a simple 3-layer feed-forward fully-connected neural network is considered and is illustrated in Fig. 1.

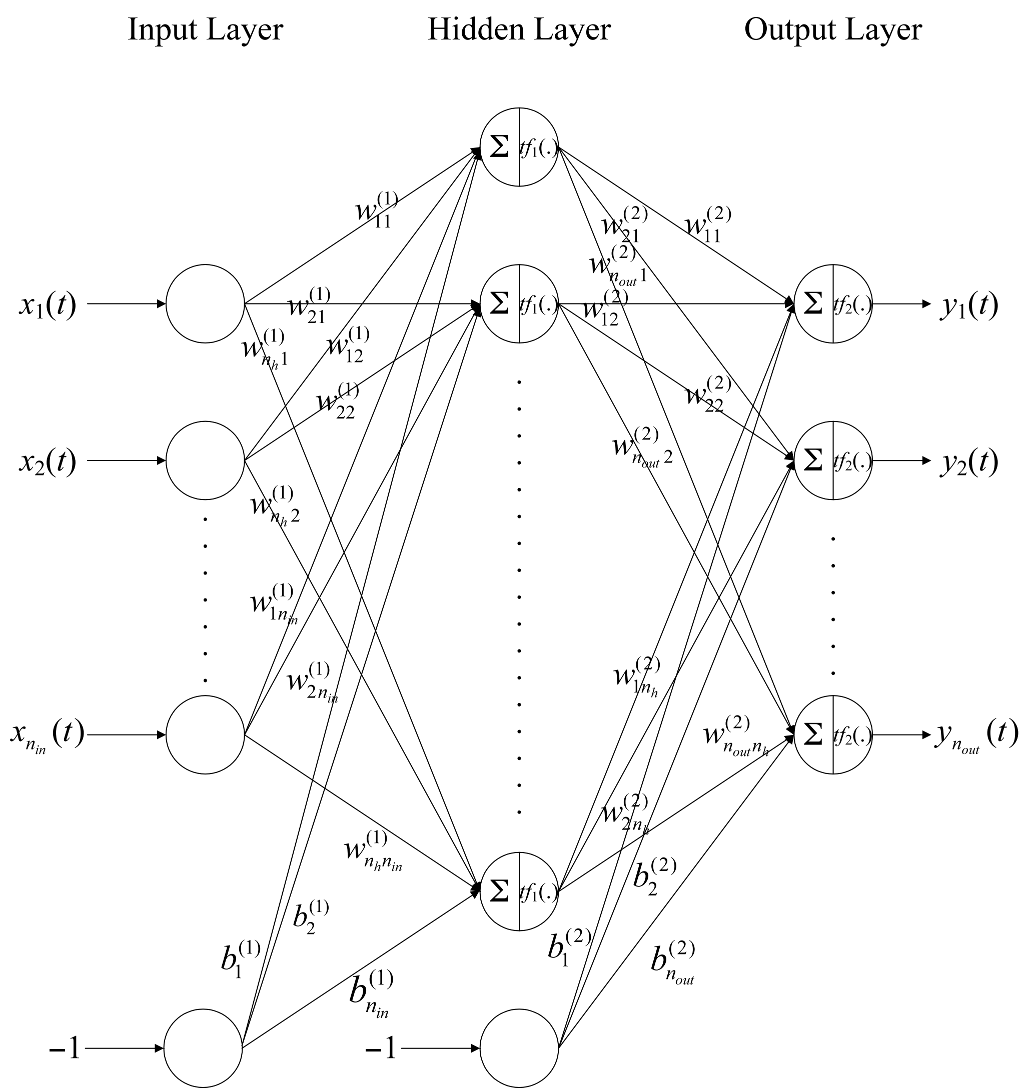

Fig. 1. 3-layer fully-connected feed-forward neural network.

III. Mechanisms of NN-Based Classifiers

In this section, six NN-based classifiers namely one-against-all, weighted one-against-all, binary coded, parallel-structured, weighted parallel-structured and tree-structured, are introduced to classify the feature patterns. In the following, the input pattern is denoted as $\mathbf{x}(t) = [x_1(t) \quad x_2(t) \quad \ldots \quad x_{n_{in}}(t)]$, which is considered as the feature vector of an object to be recognized. The purpose of these classifiers is to group the feature patterns into M classes through supervised learning.

A. One-Against-All Classifier

A one-against-all classifier is shown in Fig. 2, which can be considered as a multiple-input-single-output fully-connected feed-forward NN. It receives the feature pattern $\mathbf{x}(t)$ as input and produces a single value $y(t)$ as output. The target output $y^d(t)$ is set to be i when the input pattern $\mathbf{x}(t)$ belongs to class i. In other words, the one-against-all classifier is trained such that the output $y(t)$ is as close as possible to $y^d(t)$ according to the class of the feature pattern $\mathbf{x}(t)$.

During the operation, the feature pattern is classified as of class j which is obtained by

$$j = \arg\min_{i}\{|y(t) - i| \mid i \in \{1, 2, \dots, M\}\}, \tag{4}$$

where $|\cdot|$ is the absolute value operator. If the set j has more than one element, the first element is considered as the recognized class label.

The one-against-all classifier has a simple structure. However, it is less flexible and retraining is required when additional classes are introduced. Also, when the number of classes increases, the training time increases accordingly. For a large-scale classification problem (for example, with large dataset, large number of classes and/or high dimensional input features), the number of hidden nodes and/or layers have to be increased to achieve an acceptable recognition accuracy.

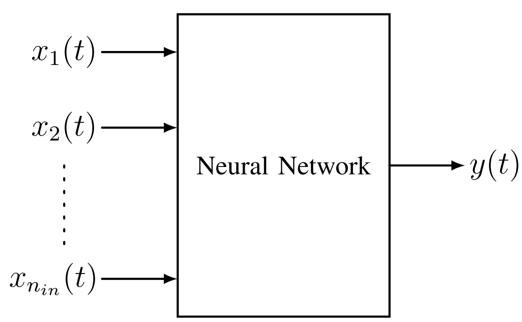

Fig. 2. NN-based one-against-all classifier.

B. Weighted One-Against-All Classifier

A weighted one-against-all classifier is shown in Fig. 3, which can be considered as a multiple-input-multiple-output fully-connected feed-forward NN. It receives a feature pattern $\mathbf{x}(t)$ as input and produces a vector $\mathbf{y}(t) = [y_1(t) \quad y_2(t) \quad \ldots \quad y_{n_{out}}(t)]$ as output where n_{out} is a non-zero positive integer pre-determined by designers. The target output vector $\mathbf{y}^d(t) = \left[y_1^d(t) \quad y_2^d(t) \quad \ldots \quad y_{n_{out}}^d(t)\right]$ is set to be $\mathbf{w}_i = [w_{i1} \quad w_{i2} \quad \ldots \quad w_{in_{out}}]$, i = 1, 2, ..., M, which is a predefined constant vector to be determined, when the input pattern $\mathbf{x}(t)$ belongs to class i. During the operation, the input pattern is classified as of class j which is obtained by

$$j = \arg\min_i \{\|\mathbf{y}(t) - \mathbf{w}_i\| \mid i \in \{1, 2, \ldots, M\}\}, \tag{5}$$

where $\|\cdot\|$ denotes the l^2 norm (i.e. Euclidean norm). If the set j has more than one element, the first element is considered as the recognized class label.

Compared with the one-against-all classifier, it offers a relatively higher flexibility to assign the target output, which could improve the recognition accuracy by examining more than one output. As weighted one-against-all classifier is based on the one-against-all classifier, it inherits the same limitations in terms of flexibility, scalability and complexity as in one-against-all Classifier discussed in Section III.B.

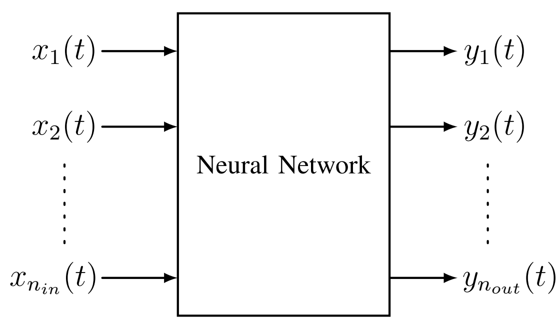

Fig. 3. NN-based weighted one-against-all or binary-coded classifier.

C. Binary-Coded Classifier

A binary-coded classifier is shown in Fig. 3, which can be considered as a multiple-input-multiple-output fully-connected feed-forward NN. It receives a feature pattern $\mathbf{x}(t)$ as input and produces a vector $\mathbf{y}(t) = [y_1(t) \quad y_2(t) \quad \ldots \quad y_{n_{out}}(t)]$ as output where $n_{out} = \left\lceil \frac{\log M}{\log 2} \right\rceil$, $\lceil \cdot \rceil$ denotes the ceiling operator rounding up the argument to the nearest integer. To reduce the number of outputs of the NN, binary string is employed to represent the class of the input patterns. Class i, i = 1, 2, ..., M, is represented by an n_{out}-bit binary string. For example, assuming that $M = 18$, a 5-bit binary string is employed to represent the class of input patterns; class 1 is represented by '00001', class 2 is represented by '00010' and so on. The target output vector $\mathbf{y}^d(t) = \begin{bmatrix} y_1^d(t) & y_2^d(t) & \ldots & y_{n_{out}}^d(t) \end{bmatrix}$ is set to be $\mathbf{w}_i = [w_{i1} \quad w_{i2} \quad \ldots \quad w_{in_{out}}]$, i = 1, 2, ..., M, which is the binary representation of i.

The binary-coded classifier is a subset of the weighted one-against-all classifier. When the weight $\mathbf{w}_i$ of the weighted one-against-all classifier is chosen to be a binary string, the classifier is configured as the binary-coded classifier. During the operation, the input pattern is recognized as of class j based on (5).

D. Parallel-Structured Classifier

A parallel-structured classifier is shown in Fig. 4, which consists of M n_{in}-input-n_{in}-output fully-connected feed-forward NNs. Fig. 4 shows that the purpose of the i^{th} NN is to recognize the input patterns of class i. To realize this purpose, the training objective is that the output of the NN corresponding to class i is the same as the input patterns of class i, i.e., the target output vector $\mathbf{y}^d(t)$ is set to be $\mathbf{x}(t)$ such that the characteristic of input patterns of class i can be learnt. Consequently, it is expected that the difference between the input and output vector of the i^{th} NN would be very small if the input patterns are of class i but relatively larger if the input pattens are not of class i. The class determiner in Fig. 4 will determine the input patten to be of class i if the i^{th} NN produces the least input-output difference. During the operation, the feature pattern is classified as class j which is obtained by

$$j = \arg\min_{i} \{ \|\mathbf{y}_i(t) - \mathbf{x}(t)\| \mid i \in \{1, 2, \ldots, M\} \}. \tag{6}$$

If the set j has more than one element, the first element is considered as the recognized class label.

The i^{th} NN is trained with the feature patterns of class i implying that the complexity of the NN is lower compared with those in one-against-all, weighted one-against-all and binary-coded classifier that feature patterns of all classes are used for training the NN. Moreover, it is more flexible to add extra classes and retraining of all existing NNs is not necessary. It is thus more suitable to handle large-scale recognition problem.

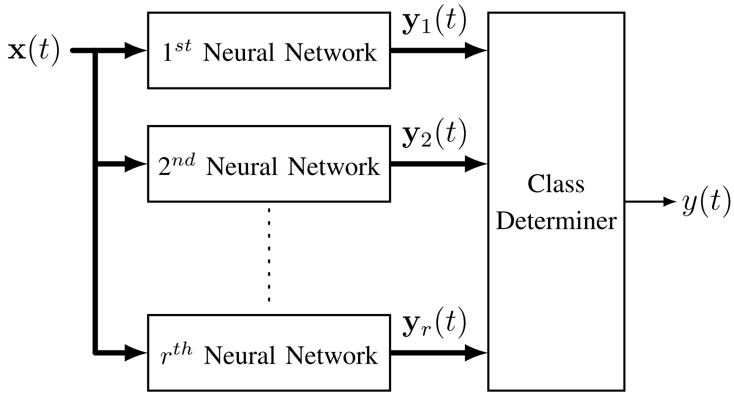

Fig. 4. NN-based parallel-structured classifier.

E. Weighted Parallel-Structured Classifier

A weighted parallel-structured classifier is a variant of parallel-structured classifier, which consists of $\lceil \frac{M}{G} \rceil$ n_{in}-input-n_{in}-output fully-connected feed-forward NNs. Each NN in the parallel-structured classifier is able to learn the characteristic of one single class of input patterns and the recognition is realized by looking into the least input-output difference. The weighted parallel-structured classifier allows each NN to learn the characteristic of more than one class of feature patterns such that each NN can classify more than one class. It reduces the number of NNs to implement the weighted parallel-structured classifier.

Let $G \leq M$ be the number of classes recognized by each NN. The i^{th} NN is trained such that the target output vector $\mathbf{y}^d(t)$ is set to be $\mathbf{W}_k\mathbf{x}(t)$ where k = $(i-1)\lceil \frac{M}{G} \rceil + 1$, $(i-1)\lceil \frac{M}{G} \rceil + 2$, ..., $(i-1)\lceil \frac{M}{G} \rceil + G$, i = 1, 2, ..., $\lceil \frac{M}{G} \rceil$, when the feature pattern $\mathbf{x}(t)$ belongs to class k; $\mathbf{W}_k =$ diag$\{w_{k1}, w_{k2}, \ldots, w_{kn_{in}}\}$ is a constant matrix determined by the designers. Consequently, when the input pattern $\mathbf{x}(t)$ is input to the i^{th} NN, the l^2 norm of the difference between the weighted input and output, i.e., $\|\mathbf{y}_i(t) - \mathbf{W}_k\mathbf{x}(t)\|$ should be very small when $\mathbf{x}(t)$ belongs to class k, otherwise, a relatively larger l^2 norm of the difference should be obtained. The class determiner will determine the class of the input pattern based on the least l^2 norm of the difference.

During the operation, the feature pattern is classified as of class j which is obtained by

$$\begin{aligned} j = \arg\min_k \{\|\mathbf{y}_i(t) - \mathbf{W}_k\mathbf{x}(t)\| \mid i \in \{1, 2, \ldots, \left\lceil \frac{M}{G} \right\rceil\}; \\ k \in \{(i-1)\left\lceil \frac{M}{G} \right\rceil + 1, (i-1)\left\lceil \frac{M}{G} \right\rceil + 2, \ldots, (i-1)\left\lceil \frac{M}{G} \right\rceil + G\}\}. \end{aligned} \tag{7}$$

If the set j has more than one element, the first element is considered as the recognized class label.

F. Tree-Structured Classifier

A tree-structured classifier is shown in Fig. 5, which consists of a single group classifier and $\lceil \frac{M}{G} \rceil$ sub-classifiers making a total of $1 + \lceil \frac{M}{G} \rceil$ NNs. We firstly divide the total number of classes into $\lceil \frac{M}{G} \rceil$ groups such that each group has G sub-classes. The group classifier is an n_{in}-input-$\left\lceil \frac{M}{G} \right\rceil$-output NN. The group classifier indicates which group the input pattern belongs to and then select the corresponding sub-classifier to perform recognition. During the training, the target output $z_k^d(t)$ for output $z_k(t)$, k = 1, 2, ..., $\left\lceil \frac{M}{G} \right\rceil$, is set to be 1 if the input pattern belongs to group k, otherwise, 0. When output $z^k(t)$ of the group classifier is closer to 1, which suggests that the input pattern belongs to group k, the k^{th} sub-classifier is selected to determine which sub-class the input pattern belongs to in this group.

During the operation, the feature pattern is classified as of group j which is obtained by

$$j = \arg\min_k \{|z_k(t) - 1| \mid k \in \{1, 2, \ldots, \left\lceil \frac{M}{G} \right\rceil\}\}. \tag{8}$$

If the set j has more than one element, the first element is considered as the recognized class label.

After the input pattern is recognized as of group j, the j^{th} sub-classifier indicates which sub-class the input pattern belongs to. The sub-classifier is an n_{in}-input-G-output NN. The l^{th} output of sub-classifier being 1 is to indicate the input pattern belongs to sub-class l in group j so that the actual class of the input pattern is $(j-1)G + l$. Based on this mechanism, the target output $y_k^d(t)$ for output $y_k(t)$, k = 1, 2, ..., G, is set to be 1 if the input pattern belongs to sub-class k, otherwise, 0.

During the operation, the input pattern is classified as of sub-class l which is obtained by

$$l = \arg\min_k \{|y_k(t) - 1| \mid k \in \{1, 2, \ldots, G\}\}. \tag{9}$$

If the set l has more than one element, the first element is considered as the recognized class label.

The tree-structured classifier provides flexibility to add extra classes without retraining the sub-classifiers, however, the group classifier has to be retrained. Furthermore, the number of levels can be increased to deal with large-scale recognition problems. As the recognition error propagates to the lower levels, the recognition performance of the upper-level classifiers, i.e., the group classifier, plays an important role to the overall recognition performance of the tree-structured classifier. Unlike

Classifier	#NNs	#outputs	Flexibility	Scalability	Complexity
1	1	1	Low	Low	High
2	1	n_{out}	Low	Low	High
3	1	$\left\lceil \frac{\log M}{\log 2} \right\rceil$	Low	Low	High
4	M	n_{out}	High	High	Low
5	$\left\lceil \frac{M}{G} \right\rceil$	n_{out}	High	High	Medium
6	$1 + \left\lceil \frac{M}{G} \right\rceil$	$\left\lceil \frac{\log M}{\log 2} \right\rceil$ or $\left\lceil \frac{M}{G} \right\rceil$	Medium	Medium	Medium

TABLE I
COMPARISON OF VARIOUS NN-BASED CLASSIFIERS. CLASSIFIER 1: ONE-AGAINST-ALL CLASSIFIER, CLASSIFIER 2: WEIGHTED ONE-AGAINSY-ALL CLASSIFIER, CLASSIFIER 3: BINARY-CODED CLASSIFIER, CLASSIFIER 4: PARALLEL-STRUCTURED CLASSIFIER, CLASSIFIER 5: PARALLEL-STRUCTURED CLASSIFIER, CLASSIFIER 6: TREE-STRUCTURED CLASSIFIER.

other classifiers introduced above, the processing time for recognition is relatively longer as the lower-level classifiers cannot start to work until result has been received from the upper levels. As the sub-classifiers only need to deal with sub-classes, the complexity of NN is relatively lower compared with the classifiers with a single NN.

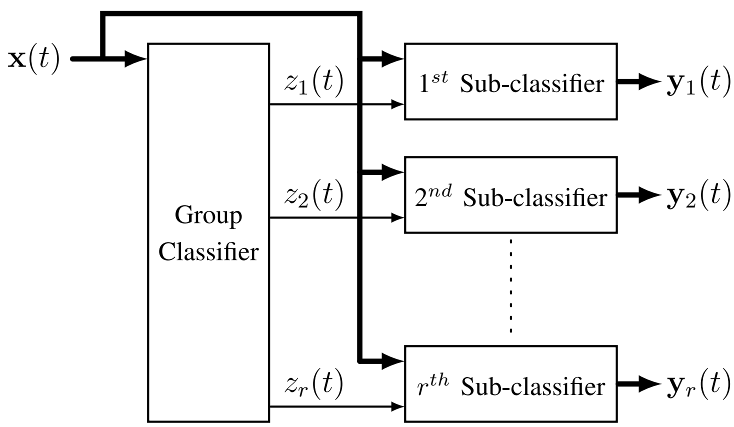

Fig. 5. NN-based tree-structured classifier.

The properties of the NN-classifiers are summarized in Table I, which compares the number of NNs used, number of outputs of NNs, flexibility adding extra classes, scalability in handling large-scale recognition problems and complexity of NNs used in the classifiers.

IV. EXPERIMENTAL RESULTS

The recognition performance of the introduced NN-based classifiers is investigated using the data collected from a robotic testing platform. The testing platform includes a robot arm Mitsubishi RV-

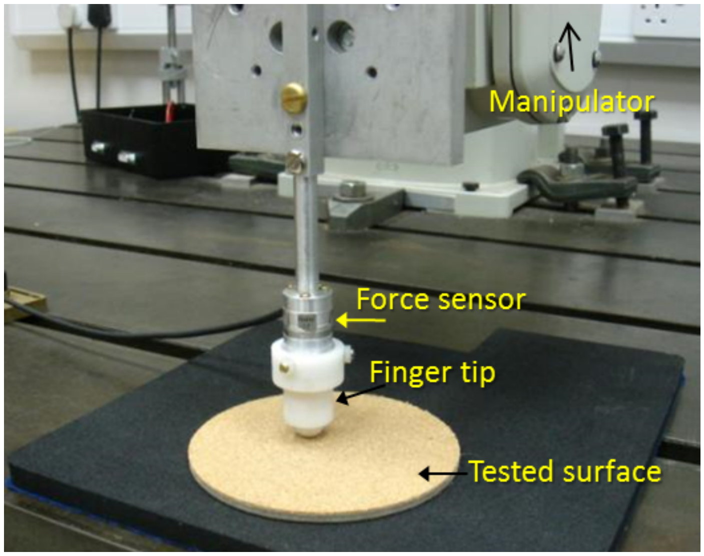

Fig. 6. The test platform.

6SL, a 6-axis force/torque sensor ATI Nano17 (resolution = 0.003 N, sampling rate = 100 Hz) and a hemispherical plastic fingertip as shown in Fig 1. During experiments, the fingertip which is rigidly attached to the robot arm was commanded to slide on a selected object surface, keeping the normal force around 2 N. The fingertip was kept perpendicular to the surface all the time. To obtain the dynamic relationship of friction and velocity, within one stroke, the sliding velocity was increased from zero to 15 mm/s with a constant acceleration rate of 3mm/s2. In total, surface of 18 materials were investigated and the raw data of fractional force are collected through the force/torque sensor. The 18 materials used in this experiment are summarized in Table II. Each time the fingertip slides along a material surface, 100 numerical values reflecting the material characteristics are collected. 60 sets of data for each material were collected and each set of data contains 100 numerical values. Detailed description of the experiment setup and technical details of raw data collection can be found in [35]. The objective of this experiment is to employ the introduced NN-based classifiers for classifying the 18 materials and comparing their recognition performance.

A. *Feature Extraction*

Before applying the introduced NN-based classifiers to recognize the materials, feature vector will be extracted from the raw data consisting of 100 numerical values to reduce the number of data points used for the classification processes. In these experiments, feature vectors of 3, 4 and 5 are extracted from the raw data, which will be used as the input of the NN-based classifiers. As a result, the raw data

Class label	Material
1	Un-laminated wood
2	Fine polished aluminium
3	Unpolished aluminium
4	Polished brass
5	Ceramic plate
6	Cloth liner
7	Glass
8	Artificial leather
9	Mouse pad (liner surface)
10	A4 paper
11	Laminated book cover
12	Plastic PC mouse
13	Plastic CD cover
14	Polymer composite (smooth surface)
15	Kitchen sponge
16	Stainless steel knife
17	Rubber tape
18	Un-laminated paper package

TABLE II
18 MATERIALS USED IN THE EXPERIMENT.

of each pattern (100 numerical values) is represented by 3, 4 or 5 features, which significantly reduces the dimensions of the input features, implying reduced computational demand and implementation complexity.

Details of feature extraction are given below. The raw data of 100 numerical values of each pattern is first divided into P portions where $P = 4$ is chosen in this experiment. Denote the raw data of 100 numerical values as $\mathbf{p} = [p_1 \quad p_2 \quad \dots \quad p_{100}]$, the first to the forth portions of raw data are defined as: $\mathbf{p}_1 = [p_1 \quad p_2 \quad \dots \quad p_{25}]$, $\mathbf{p}_2 = [p_{26} \quad p_{27} \quad \dots \quad p_{50}]$, $\mathbf{p}_3 = [p_{51} \quad p_{52} \quad \dots \quad p_{75}]$ and $\mathbf{p}_4 = [p_{76} \quad p_{77} \quad \dots \quad p_{100}]$. Define

$$f_1(\mathbf{z}) = \frac{1}{S}\sum_{i=1}^{S} z_i, \tag{10}$$

$$f_2(\mathbf{p}) = \sum_{i=1}^{4} |f_1(\mathbf{p}_{i+1}) - f_1(\mathbf{p}_i)|, \tag{11}$$

$$f_3(\mathbf{z}) = \frac{1}{S-1}\sum_{i=1}^{S}(z_i - f_1(\mathbf{z}))^2, \tag{12}$$

where $\mathbf{z} = [z_1 \quad z_2 \quad \ldots \quad z_S]$.

Feature vectors of 3 to 5 feature points are defined as follows:

Feature vector with 3 points:

$$\mathbf{x} = \left[\begin{array}{ccc} \sum_{i=1}^{4} f_1(\mathbf{p}_i) & 50 f_2(\mathbf{p}_i) & 50\sum_{i=1}^{4} f_3(\mathbf{p}_i) \end{array}\right]. \tag{13}$$

Feature vector with 4 points:

$$\mathbf{x} = \left[\begin{array}{cccc} \sum_{i=1}^{4} f_1(\mathbf{p}_i) & 50 f_2(\mathbf{p}_i) & 50\sum_{i=1}^{4} f_3(\mathbf{p}_i) & 20\sum_{i=1}^{4} \sqrt{f_3(\mathbf{p}_i)} \end{array}\right]. \tag{14}$$

Feature vector with 5 points:

$$\mathbf{x} = \left[\begin{array}{ccccc} \sum_{i=1}^{4} f_1(\mathbf{p}_i) & 50|f_1(\mathbf{p}_2) - f_1(\mathbf{p}_1)| & 50|f_1(\mathbf{p}_3) - f_1(\mathbf{p}_2)| & 50|f_1(\mathbf{p}_4) - f_1(\mathbf{p}_3)| & 50\sum_{i=1}^{4} f_3(\mathbf{p}_i) \end{array}\right]. \tag{15}$$

It can be seen from (10) to (12) that $f_1(\mathbf{z})$ is the mean of $\mathbf{z}$, $f_2(\mathbf{z})$ is the sum of the difference of the mean of the consecutive portions of raw data, $f_3(\mathbf{z})$ is the variance of $\mathbf{z}$.

B. NN-based Classification

The 6 NN-based classifiers are employed to recognize the 18 materials using the feature vectors of 3, 4, and 5 points. The introduced classifiers were implemented on Matlab.The Levenberg-Marquardt back-propagation is used to develop the classifiers by minimizing the mean square error.

In this experiment, recalling that 60 sets of raw data being collected for each material, 40 of them are be used for the training of NNs and 20 of them are used for testing. Various transfer functions and different number of hidden nodes and hidden layers have been tried in this study. In the following, only the appropriate configurations (number of hidden nodes, transfer functions, etc.) which can achieve acceptable recognition accuracy are reported. The linear transfer function is used in the output layer of all classifiers. For comparison purposes, the traditional kNN classifier and naive Bayes classifier are employed for the classification problem. To investigate how the noise influences the recognition performance of the classifiers, which is inevitable in real world, the test dataset contaminated by

Gaussian white noise with variance of 0.005 is employed. It should be noted that the simulations for all classifiers tested with noisy test dataset are conducted 10 times for fair comparison, as different solutions can be obtained by the Levenberg-Marquardt back-propagation algorithm with different initial guesses. Statistical information of the tests including the average recognition accuracy for individual class, maximum and minimum recognition accuracy and standard deviation of the 10 tests is reported.

In the following, the recognition performance of the 6 NN-based classifiers, kNN classifier and naive Bayes classifier for the classification problem subject to noise-free and noisy datasets is reported.

1) One-Against-All Classifiers: An NN with 3 layers as shown in Fig. 1 is employed to implement the one-against-all classifier. The number of hidden nodes was chosen to be 30 and the transfer function of hidden nodes was chosen to be a logarithmic sigmoid transfer function. The recognition accuracy in percentage for the one-against-all classifier with feature vector of 3 to 5 feature points corresponding to each material is summarized in Table V to Table VII.

Referring to these tables, it can be seen that the average testing recognition accuracy is about 96% for the one-against-all classifier using feature vector of 3 to 5 feature points. However, looking into the testing recognition accuracy of individual material, the one-against-all classifier using 3 feature points offers 80% recognition accuracy for material 11 while the one-against-all classifier using 4 or 5 feature points offers 85% testing recognition accuracy in the worst case. It suggests that the feature vector of 3 feature points may not work well with the one-against-all classifier.

The recognition accuracy for the test data subject to Gaussian white noise is shown in Table XXIX to Table XXXI. It can be seen from the tables that the recognition accuracy of the one-against-all classifiers subject to noisy data has declined to about 92%, 83% and 93% for 3, 4 and 5 feature vectors, respectively. The classifier with 4 feature points performs the worst when the noise exists. Also, it is found that materials 12 and 15 are very sensitive to the noise.

2) Weighted One-Against-All Classifiers: An NN with 3 layers is employed to implement the weighted one-against-all classifier. The elements of the weighting vector $\mathbf{w}_i$ were all chosen to be $\lceil i - 9.5 \rceil$, i = 1, 2, . . ., 18. The number of hidden nodes was chosen to be 30 and the transfer function of hidden nodes was chosen to be hyperbolic tangent sigmoid transfer function. The recognition accuracy in percentage for the weighted one-against-all classifier with the feature vector of 3 to 5 feature points corresponding to each material is summarized in Table VIII to Table X for noise-free dataset and Table XXXII to Table XXXIV for noisy dataset.

Referring to Table VIII to Table X, the average testing recognition accuracy is about 96% for the weighted one-against-all classifier using the feature vector of 3 or 4 feature points. However, the average testing recognition accuracy is improved to about 98% for the feature vector of 5 feature points. Looking into the worst individual testing recognition accuracy, the weighted one-against-all classifier using feature vector of 3 feature points offers 80% for material 13 while the weighted one-against-all classifier using 4 or 5 feature points offers 85% testing recognition accuracy in the worst case. Similar conclusion that the feature vector of 3 feature points may not work effectively can be drawn.

Referring to Table XXXII to Table XXXIV the performance of weighted one-against-all classifier under noisy data has declined to about 85%, 85% and 95% respectively. Similar observation is found as in the results from one-against-all classifiers as the same mechanism is used on both one-against-all and weighted one-against-all classifiers.

3) Binary-Coded Classifiers: An NN with 3 layers as shown in Fig. 1 is employed to implement the binary-coded classifier. The number of hidden nodes was chosen to be 30 and the transfer function of hidden nodes was chosen to be logarithmic sigmoid transfer function. The recognition accuracy in percentage for the binary-coded classifier with feature vector of 3 to 5 feature points corresponding to each material is summarized in Table XI to Table XIII for noise-free dataset and Table XXXV to Table XXXVII for noisy dataset.

Referring to Table XI to Table XIII, the average testing recognition accuracy for the binary-coded classifier with feature vector of 3 or 4 feature points is 98% while with feature vector of 5 feature points is about 99%. The worst individual testing recognition accuracy is 95% for all binary-coded classifier with feature vector of 3 to 5 feature points. Comparing with the one-against-all or the weighted one-against-all classifier, the recognition performance of binary-coded classifier is less sensitive to the number of feature points.

Referring to Table XXXV to Table XXXVII, the recognition performance of binary-coded classifier under noisy data can be observed. The binary-coded classifier is able to offer a relatively higher performance compared with the one-against-all and weighted one-against-all classifiers. Corresponding to the number of feature points as 3, 4 and 5, the average recognition accuracy can achieve about 97%, 94% and 99%, respectively. It is again showing that the dataset with 4 feature points produces the worst result. It is observed that materials 12 and 15 are the most difficult classes to be recognized but their recognition accuracy can be significantly improved compared with the previous discussed classifiers.

4) Parallel-Structured Classifiers: In the parallel-structured classifier, all NNs are with 3 layers where the number of hidden nodes was chosen to be 10 and the transfer function of hidden nodes was chosen to be logarithmic sigmoid transfer function. Compared with the NNs used in the above classifiers, the number of hidden nodes is significantly reduced, which supports the comment in Table I that the complexity of NN is relatively lower. The recognition accuracy in percentage for the parallel-structured classifier with feature vector of 3 to 5 feature points corresponding to each material is summarized in Table XIV to Table XVI for noise-free dataset and Table XXXVIII to Table XL for noisy dataset.

Referring to Table XIV to Table XVI, the individual training and testing recognition accuracy are all 100% irregardless of the number of feature points used. Of all classifiers, the parallel-structured classifier offers the best recognition performance. Based on the recognition accuracy, it suggests that 3 features points are sufficient for recognition purposes.

Referring to Table XXXVIII to Table XL the performance of parallel-structured classifier under noisy data can be observed. It can be seen from these 3 tables that the parallel-structured classifier is still able to offer a tolerable performance. Corresponding to the number of feature points as 3, 4 and 5, the average recognition accuracy can achieve about 94%, 93% and 96%, respectively. When the noisy dataset is considered, the recognition performance is not as good as but comparable to that of the binary-coded classifiers. Also, materials 12 and 15 are the most difficult classes to be recognized.

5) Weighted Parallel-Structured Classifiers Classifiers: In the parallel-structured classifier, all NNs are with 3 layers where the number of hidden nodes was chosen to be 15, the transfer function of hidden nodes was chosen to be hyperbolic tangent sigmoid transfer function and $G = 3$. The weighting vector $\mathbf{w}_i$ were chosen as follows.

Feature vector of 3 feature points:

$$\mathbf{w}_i = \begin{bmatrix} -1 & -1 & -1 \end{bmatrix}, i = 1, 4, 7, 10, 13, 16.$$

$$\mathbf{w}_i = \begin{bmatrix} 1 & 1 & -1 \end{bmatrix}, i = 2, 5, 8, 11, 14, 17.$$

$$\mathbf{w}_i = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}, i = 3, 6, 9, 12, 15, 18.$$

Feature vector of 4 feature points:

$$\mathbf{w}_i = [-1 \quad -1 \quad -1 \quad -1]\,, i = 1, 4, 7, 10, 13, 16.$$

$$\mathbf{w}_i = [1 \quad 1 \quad -1 \quad -1]\,, i = 2, 5, 8, 11, 14, 17.$$

$$\mathbf{w}_i = [1 \quad 1 \quad 1 \quad 1]\,, i = 3, 6, 9, 12, 15, 18.$$

Feature vector of 5 feature points:

$$\mathbf{w}_i = [-1 \quad -1 \quad -1 \quad -1 \quad -1]\,, i = 1, 4, 7, 10, 13, 16.$$

$$\mathbf{w}_i = [1 \quad 1 \quad 1 \quad -1 \quad -1]\,, i = 2, 5, 8, 11, 14, 17.$$

$$\mathbf{w}_i = [1 \quad 1 \quad 1 \quad 1 \quad 1]\,, i = 3, 6, 9, 12, 15, 18.$$

The recognition accuracy in percentage for the weighted parallel-structured classifier with feature vector of 3 to 5 feature points corresponding to each material is summarized in Table XVII to Table XIX for noise-free dataset and Table XLI to Table XLIII for noisy dataset.

Referring to Table XVII to Table XIX, it can be seen that the weighted parallel-structured classifier with feature vector of 5 feature points offers the best average testing recognition accuracy of about 99% with the worst individual recognition accuracy of 95%. Although the weighted parallel-structured classifier with feature vector of 3 or 4 feature points does not have a bad performance with an average testing recognition accuracy of about 98%, the individual testing recognition accuracy is 90% for 3 feature points and 80% for 4 feature points.

Referring to Table XLI to Table XLIII, the performance of weighted parallel-structured classifier under noisy data can be observed. The weighted parallel-structured classifier is able to offer tolerable average recognition accuracy of about 90%, 93% and 97%, corresponding to 3, 4 and 5 points of feature vectors, respectively.

6) Tree-Structured Classifiers: In the tree-structured classifier, all NNs are with 3 layers where the number of hidden nodes was chosen to be 20 for the group classifier, 5 for each sub-classifier, the transfer function of hidden nodes was chosen to be logarithmic sigmoid transfer function for both the group classifier and sub-classifiers. The number of sub-classes is chosen to be $G = 3$. Compared with the NNs used in the above classifiers, the number of hidden nodes in the sub-classifier is small as only 3 sub-classes need to be handled. The recognition accuracy in percentage for the parallel-structured classifier with feature vector of 3 to 5 feature points corresponding to each material is summarized in Table XX to Table XXII for noise-free dataset and Table XLIV to Table XLVI for noisy dataset.

Referring to Table XX to Table XXII, the tree-structured classifier with feature vector of 5 feature points offers 100% training and testing recognition accuracy while the one with 3 or 4 feature points offers about 99% testing recognition accuracy and the worst individual testing recognition accuracy of 90%.

It can be seen from Table XLIV to Table XLVI that the tree-structured classifier under noisy data is still able to offer a relatively high recognition accuracy. Corresponding to 3, 4 and 5 points of feature vectors, the average recognition accuracy of about 98%, 97% and 100%, respectively, can be achieved. Among all NN-based classifiers, the tree-structured classifiers are more robust to the noisy input.

C. Traditional Classifiers

In order to show the superiority and adaptability of the NN-based classifiers, two traditional classifiers, namely kNN classifier and the Naive Bayes classifier, are employed to accomplish the classification of the 18 materials using the features vectors of 3, 4, and 5 points.

1) K-Nearest Neighbor Classifier: In this experiment, fixing the k-nearest to 1, the recognition accuracy in percentage for the kNN classifiers with feature vectors of 3 to 5 feature points are summarized in Table XXIII to Table XXV.

From Table XXIII to Table XXV, it can be seen that the average recognition accuracy for the training dataset is 100% for 3, 4 and 5 points of feature vectors. However, when test dataset is considered, the kNN classifiers with 3 feature points can achieve average recognition accuracy of about 96%, which is higher than that of the kNN classifiers with 4 and 5 feature points, which can achieve only 94% and 90% of average recognition accuracy.

From Table XLVII to Table XLIX, it can be found that the recognition performance of the kNN classifiers with noisy dataset has declined to some extent. The best average recognition accuracy of about 94% is obtained for the kNN classifier the feature vector of 3 points while the average recognition accuracy is dropped to about 88% and 89% for the kNN classifiers with the feature vector of 4 and 5 points, respectively. It is interestingly observed that materials 12 and 15 can be recognized well.

However, material 17 becomes the most difficult class to be recognized.

2) Naive Bayes Classifier: The recognition accuracy in percentage for the naive Bayes classifier with feature vector of 3 to 5 points is summarized in Table XXVI and Table XXVIII for noise-free dataset. From these 3 tables, it can be seen that the recognition accuracy of the naive Bayes classifier for training dataset can be achieved as 100% for 3, 4 and 5 points of feature vectors, the recognition accuracy for the data set are, 99%, 100% and 100%.

When noise is considered in the test dataset, the recognition performance is given in Table L and Table LII. The best recognition accuracy for the best is about 93.9% which is achieved by the classifier with the feature vector of 5 points. The worst is about 93% which is achieved by the classifier with the feature vector of 4 points.

D. Discussion

Giving an overall picture of the recognition performance, Table III summarizes the overall recognition performance of the 6 NN-based classifiers and two traditional classifiers and Table IV summarizes the overall recognition performance under noisy test dataset. In these two tables, the average recognition accuracy is the overall recognition accuracy, which is the average recognition accuracy of all classes; the worst recognition accuracy is the worst recognition accuracy in the 18 classes.

It can be seen from Table III that in general the classifiers with 5 feature points perform better in terms of the worst individual training and testing recognition accuracy, and the average training and testing recognition accuracy. When 5 feature points are considered, the parallel-structured, tree-structured classifier and naive Bayes classifer are able to offer the training and testing recognition accuracy of 100%. The second best is the binary-coded classifier which is able to offer the training and testing recognition accuracy around 99%. The worst one is the one-against-all classifier which is only able to offer a testing recognition accuracy around 96%. When considering the kNN classifier, the overall recognition accuracy for the training dataset is 100%. However, among all classifiers, the kNN classifier offers the worst recognition accuracy for the test dataset.

Under the noisy test dataset, referring to Table IV, in general, the recognition performance declines for all classifiers. The overall average recognition accuracy drops below 90% for weighted one-against-all classifier when feature vector of 3 points is employed; for one-against-all classifier, weighted one-against-all classifier and kNN classifier when feature vector of 4 points is employed; for kNN classifier when feature vector of 5 points is employed. It is observed that majority of classifiers can obtain better recognition accuracy when feature vector of 5 points is employed. By looking into the details, it can be seen that the tree-structured classifier can obtain the best recognition accuracy. In particular, when feature vector of 5 points is employed, the tree-structured classifier is able to achieve overall average recognition accuracy of 99.7778%, outperforming the rest classifiers. It can also been seen that the tree-structured classifier demonstrate consistent recognition performance subject to noisy input with the smallest standard deviation among all classifiers. The second best is the binary-coded classifier which can obtain the overall average recognition accuracy of 98.8611% but the standard derivation is more or less 5 times higher than that of the tree-structured classifier. The worst one is the kNN classifier which can obtain the overall average recognition accuracy of 88.6389% with a significant higher standard deviation. It is interestingly found that the parallel-structured classifier is less sensitive to the number of feature points used, which is able to offer more or less the same overall average recognition accuracy regardless of the number of feature points under noise-free and noisy conditions.

From the summary tables, it can be concluded that the binary-coded classifier and tree-structured classifier are more suitable for the application of material recognition when feature vector of 5 points are used.

V. Conclusion

This paper has introduced 6 neural-network-based classifiers (namely one-against-all, weighted one-against-all, binary coded, parallel structured, weighted parallel structured, tree-structured classifier) and two traditional classifiers (namely k-nearest neighbor classifier and naive Bayes classifier) to deal with a material classification problem where the data was collected from a robot finger installed with tactile sensors. In total 18 materials have been considered in the experiment. The properties of each classifier have been discussed and its mechanism of performing classification has been detailed. To perform the classification, feature vectors of size 3, 4 and 5 are extracted for each material. Supervised learning approach has been adopted to train the neural-network-based classifier, kNN classifier and naive Bayes classifier for the recognition of materials. The performance of each classifiers has been fully investigated and compared with each other in terms of recognition accuracy. In the noisy-free case, the results has shown that the parallel-structured classifier produces the best performance among all 8 classifiers when 3, 4 and 5 feature points are used. However, under the noisy case, the tree-structured

	Recognition Accuracy (%)				
#feature points	Classifier	Worst (Training)	Average (Training)	Worst (Testing)	Average (Testing)
3	1	100	100	80	96.9444
3	2	92.5	98.8889	80	96.3889
3	3	97.5	99.7222	90	98.6111
3	4	100	100	100	100
3	5	95	99.3056	90	98.0556
3	6	97.5	99.8611	90	99.1667
3	7	100	100	80	95.8333
3	8	100	100	90	99.4444
4	1	100	100	85	96.3889
4	2	97.5	99.8611	85	96.3889
4	3	97.5	99.5833	90	98.8889
4	4	100	100	100	100
4	5	87.5	99.0278	80	98.3333
4	6	100	100	90	98.6111
4	7	100	100	70	93.6111
4	8	100	100	100	100
5	1	100	100	85	96.1111
5	2	100	100	95	98.3333
5	3	97.5	99.8611	95	99.7222
5	4	100	100	100	100
5	5	97.5	99.8611	95	99.1667
5	6	100	100	100	100
5	7	100	100	40	89.7222
5	8	100	100	100	100

TABLE III
SUMMARY OF RECOGNITION PERFORMANCE OF THE 6 NN-BASED CLASSIFIERS, KNN CLASSIFIER AND NAIVE BAYES CLASSIFIER UNDER NOISE-FREE DATASET. CLASSIFIER 1: ONE-AGAINST-ALL CLASSIFIER, CLASSIFIER 2: WEIGHTED ONE-AGAINSY-ALL CLASSIFIER, CLASSIFIER 3: BINARY-CODED CLASSIFIER, CLASSIFIER 4: PARALLEL-STRUCTURED CLASSIFIER, CLASSIFIER 5: PARALLEL-STRUCTURED CLASSIFIER, CLASSIFIER 6: TREE-STRUCTURED CLASSIFIER. CLASSIFIER 7: K-NEAREST NEIGHBOR CLASSIFIER, 8: NAIVE BAYES CLASSIFIER

classifier has achieved the best performance among all the classifiers when 3, 4 and 5 feature points are used.

	Recognition Accuracy (%)				
#feature points	Classifier	Worst	Average	Best	Std
3	1	88.3333	92.9722	96.3889	2.7212
3	2	80.0000	85.2500	89.4444	3.4430
3	3	94.4444	96.6944	98.8889	1.4593
3	4	90.0000	93.9722	96.3889	2.2056
3	5	86.9444	90.1944	92.5000	1.9889
3	6	96.3889	97.8611	99.1667	0.9679
3	7	88.6111	93.8889	98.3333	3.3120
3	8	92.5000	93.5278	94.1667	0.5826
4	1	78.6111	83.0000	86.6667	2.8315
4	2	81.6667	85.3889	88.6111	2.3061
4	3	90.0000	93.9444	96.3889	2.1650
4	4	91.1111	92.7778	93.8889	1.0273
4	5	90.2778	93.1667	95.2778	1.9215
4	6	95.5556	97.3611	99.1667	1.0499
4	7	83.6111	87.5926	92.2222	4.5695
4	8	92.2222	93.0278	94.1667	0.6887
5	1	87.7778	93.0278	97.2222	3.2587
5	2	92.2222	94.8333	97.5000	1.9245
5	3	96.6667	98.8611	99.7222	1.1066
5	4	94.7222	95.8056	96.6667	0.6879
5	5	92.5000	96.5833	98.8889	2.1120
5	6	99.1667	99.7778	100.0000	0.2869
5	7	83.3333	88.6389	92.7778	3.0588
5	8	93.3333	93.9167	94.4444	0.4086

TABLE IV

NOISE:SUMMARY OF RECOGNITION PERFORMANCE OF THE 6 NN-BASED CLASSIFIERS, KNN CLASSIFIER AND NAIVE BAYES CLASSIFIER. CLASSIFIER 1: ONE-AGAINST-ALL CLASSIFIER, CLASSIFIER 2: WEIGHTED ONE-AGAINSY-ALL CLASSIFIER, CLASSIFIER 3: BINARY-CODED CLASSIFIER, CLASSIFIER 4: PARALLEL-STRUCTURED CLASSIFIER, CLASSIFIER 5: PARALLEL-STRUCTURED CLASSIFIER, CLASSIFIER 6: TREE-STRUCTURED CLASSIFIER. CLASSIFIER 7: K-NEAREST NEIGHBOR CLASSIFIER, 8: NAIVE BAYES CLASSIFIER

APPENDIX

TRAINING AND TESTING RECOGNITION ACCURACY

	Recognition Accuracy (%)	
Material	Training	Testing
1	100.0000	95.0000
2	100.0000	90.0000
3	100.0000	100.0000
4	100.0000	100.0000
5	100.0000	100.0000
6	100.0000	100.0000
7	100.0000	100.0000
8	100.0000	95.0000
9	100.0000	100.0000
10	100.0000	100.0000
11	100.0000	80.0000
12	100.0000	100.0000
13	100.0000	100.0000
14	100.0000	95.0000
15	100.0000	100.0000
16	100.0000	95.0000
17	100.0000	95.0000
18	100.0000	100.0000
Average	100.0000	96.9444

TABLE V
NN-BASED ONE-AGAINST-ALL CLASSIFIER USING 3 FEATURE POINTS. NUMBER OF HIDDEN NODES: 30, TRANSFER FUNCTION OF HIDDEN NODES: LOGARITHMIC SIGMOID TRANSFER FUNCTION.

	Recognition Accuracy (%)	
Material	Training	Testing
1	100.0000	100.0000
2	100.0000	90.0000
3	100.0000	100.0000
4	100.0000	85.0000
5	100.0000	95.0000
6	100.0000	100.0000
7	100.0000	100.0000
8	100.0000	95.0000
9	100.0000	100.0000
10	100.0000	90.0000
11	100.0000	90.0000
12	100.0000	95.0000
13	100.0000	95.0000
14	100.0000	100.0000
15	100.0000	100.0000
16	100.0000	100.0000
17	100.0000	100.0000
18	100.0000	100.0000
Average	100.0000	96.3889

TABLE VI
NN-BASED ONE-AGAINST-ALL CLASSIFIER USING 4 FEATURE POINTS. NUMBER OF HIDDEN NODES: 30, TRANSFER FUNCTION OF HIDDEN NODES: LOGARITHMIC SIGMOID TRANSFER FUNCTION.

	Recognition Accuracy (%)	
Material	Training	Testing
1	100.0000	90.0000
2	100.0000	85.0000
3	100.0000	95.0000
4	100.0000	95.0000
5	100.0000	100.0000
6	100.0000	100.0000
7	100.0000	100.0000
8	100.0000	85.0000
9	100.0000	100.0000
10	100.0000	100.0000
11	100.0000	100.0000
12	100.0000	100.0000
13	100.0000	95.0000
14	100.0000	100.0000
15	100.0000	95.0000
16	100.0000	100.0000
17	100.0000	95.0000
18	100.0000	95.0000
Average	100.0000	96.1111

TABLE VII
NN-BASED ONE-AGAINST-ALL CLASSIFIER USING 5 FEATURE POINTS. NUMBER OF HIDDEN NODES: 30, TRANSFER FUNCTION OF HIDDEN NODES: LOGARITHMIC SIGMOID TRANSFER FUNCTION.

	Recognition Accuracy (%)	
Material	Training	Testing
1	97.5000	95.0000
2	92.5000	90.0000
3	100.0000	95.0000
4	97.5000	100.0000
5	100.0000	100.0000
6	100.0000	100.0000
7	100.0000	100.0000
8	100.0000	95.0000
9	100.0000	100.0000
10	100.0000	100.0000
11	100.0000	95.0000
12	100.0000	100.0000
13	100.0000	80.0000
14	100.0000	100.0000
15	100.0000	100.0000
16	95.0000	90.0000
17	100.0000	100.0000
18	97.5000	95.0000
Average	98.8889	96.3889

TABLE VIII

NN-based weighted one-against-all classifier using 3 feature points. Number of hidden nodes: 20, transfer function of hidden nodes: hyperbolic tangent sigmoid transfer function.

	Recognition Accuracy (%)	
Material	Training	Testing
1	100.0000	100.0000
2	97.5000	90.0000
3	100.0000	100.0000
4	100.0000	100.0000
5	100.0000	100.0000
6	100.0000	95.0000
7	100.0000	95.0000
8	100.0000	90.0000
9	100.0000	100.0000
10	100.0000	100.0000
11	100.0000	90.0000
12	100.0000	95.0000
13	100.0000	100.0000
14	100.0000	100.0000
15	100.0000	100.0000
16	100.0000	95.0000
17	100.0000	85.0000
18	100.0000	100.0000
Average	99.8611	96.3889

TABLE IX

NN-based weighted one-against-all classifier using 4 feature points. Number of hidden nodes: 20, transfer function of hidden nodes: hyperbolic tangent sigmoid transfer function.

	Recognition Accuracy (%)	
Material	Training	Testing
1	100.0000	95.0000
2	100.0000	100.0000
3	100.0000	100.0000
4	100.0000	100.0000
5	100.0000	100.0000
6	100.0000	100.0000
7	100.0000	95.0000
8	100.0000	95.0000
9	100.0000	100.0000
10	100.0000	95.0000
11	100.0000	100.0000
12	100.0000	100.0000
13	100.0000	100.0000
14	100.0000	100.0000
15	100.0000	95.0000
16	100.0000	100.0000
17	100.0000	95.0000
18	100.0000	100.0000
Average	100.0000	98.3333

TABLE X

NN-based weighted one-against-all classifier using 5 feature points. Number of hidden nodes: 20, transfer function of hidden nodes: hyperbolic tangent sigmoid transfer function.

	Recognition Accuracy (%)	
Material	Training	Testing
1	100.0000	100.0000
2	100.0000	100.0000
3	100.0000	100.0000
4	100.0000	100.0000
5	100.0000	100.0000
6	100.0000	100.0000
7	100.0000	100.0000
8	97.5000	95.0000
9	100.0000	100.0000
10	100.0000	100.0000
11	97.5000	90.0000
12	100.0000	100.0000
13	100.0000	95.0000
14	100.0000	100.0000
15	100.0000	100.0000
16	100.0000	100.0000
17	100.0000	95.0000
18	100.0000	100.0000
Average	99.7222	98.6111

TABLE XI

NN-based binary-coded-output classifier using 3 feature points. Number of hidden nodes: 30, transfer function of hidden nodes: logarithmic sigmoid transfer function.

	Recognition Accuracy (%)	
Material	Training	Testing
1	97.5000	100.0000
2	100.0000	100.0000
3	97.5000	100.0000
4	97.5000	100.0000
5	100.0000	100.0000
6	100.0000	100.0000
7	100.0000	100.0000
8	100.0000	100.0000
9	100.0000	100.0000
10	100.0000	100.0000
11	100.0000	95.0000
12	100.0000	100.0000
13	100.0000	90.0000
14	100.0000	100.0000
15	100.0000	100.0000
16	100.0000	100.0000
17	100.0000	95.0000
18	100.0000	100.0000
Average	99.5833	98.8889

TABLE XII
NN-BASED BINARY-CODED-OUTPUT CLASSIFIER USING 4 FEATURE POINTS. NUMBER OF HIDDEN NODES: 30, TRANSFER FUNCTION OF HIDDEN NODES: LOGARITHMIC SIGMOID TRANSFER FUNCTION.

	Recognition Accuracy (%)	
Material	Training	Testing
1	100.0000	100.0000
2	100.0000	100.0000
3	100.0000	100.0000
4	100.0000	100.0000
5	100.0000	100.0000
6	100.0000	100.0000
7	100.0000	100.0000
8	100.0000	100.0000
9	100.0000	100.0000
10	100.0000	100.0000
11	100.0000	100.0000
12	100.0000	100.0000
13	100.0000	100.0000
14	100.0000	100.0000
15	100.0000	100.0000
16	100.0000	100.0000
17	100.0000	95.0000
18	97.5000	100.0000
Average	99.8611	99.7222

TABLE XIII
NN-BASED BINARY-CODED-OUTPUT CLASSIFIER USING 5 FEATURE POINTS. NUMBER OF HIDDEN NODES: 30, TRANSFER FUNCTION OF HIDDEN NODES: LOGARITHMIC SIGMOID TRANSFER FUNCTION.

	Recognition Accuracy (%)	
Material	Training	Testing
1	100.0000	100.0000
2	100.0000	100.0000
3	100.0000	100.0000
4	100.0000	100.0000
5	100.0000	100.0000
6	100.0000	100.0000
7	100.0000	100.0000
8	100.0000	100.0000
9	100.0000	100.0000
10	100.0000	100.0000
11	100.0000	100.0000
12	100.0000	100.0000
13	100.0000	100.0000
14	100.0000	100.0000
15	100.0000	100.0000
16	100.0000	100.0000
17	100.0000	100.0000
18	100.0000	100.0000
Average	100.0000	100.0000

TABLE XIV
NN-BASED PARALLEL-STRUCTURED CLASSIFIER USING 3 FEATURE POINTS. NUMBER OF HIDDEN NODES: 10, TRANSFER FUNCTION OF HIDDEN NODES: LOGARITHMIC SIGMOID TRANSFER FUNCTION.

	Recognition Accuracy (%)	
Material	Training	Testing
1	100.0000	100.0000
2	100.0000	100.0000
3	100.0000	100.0000
4	100.0000	100.0000
5	100.0000	100.0000
6	100.0000	100.0000
7	100.0000	100.0000
8	100.0000	100.0000
9	100.0000	100.0000
10	100.0000	100.0000
11	100.0000	100.0000
12	100.0000	100.0000
13	100.0000	100.0000
14	100.0000	100.0000
15	100.0000	100.0000
16	100.0000	100.0000
17	100.0000	100.0000
18	100.0000	100.0000
Average	100.0000	100.0000

TABLE XV
NN-BASED PARALLEL-STRUCTURED CLASSIFIER USING 4 FEATURE POINTS. NUMBER OF HIDDEN NODES: 10, TRANSFER FUNCTION OF HIDDEN NODES: LOGARITHMIC SIGMOID TRANSFER FUNCTION.

	Recognition Accuracy (%)	
Material	Training	Testing
1	100.0000	100.0000
2	100.0000	100.0000
3	100.0000	100.0000
4	100.0000	100.0000
5	100.0000	100.0000
6	100.0000	100.0000
7	100.0000	100.0000
8	100.0000	100.0000
9	100.0000	100.0000
10	100.0000	100.0000
11	100.0000	100.0000
12	100.0000	100.0000
13	100.0000	100.0000
14	100.0000	100.0000
15	100.0000	100.0000
16	100.0000	100.0000
17	100.0000	100.0000
18	100.0000	100.0000
Average	100.0000	100.0000

TABLE XVI
NN-BASED PARALLEL-STRUCTURED CLASSIFIER USING 5 FEATURE POINTS. NUMBER OF HIDDEN NODES: 10, TRANSFER FUNCTION OF HIDDEN NODES: LOGARITHMIC SIGMOID TRANSFER FUNCTION.

	Recognition Accuracy (%)	
Material	Training	Testing
1	95.0000	95.0000
2	100.0000	100.0000
3	100.0000	95.0000
4	95.0000	95.0000
5	97.5000	95.0000
6	100.0000	100.0000
7	100.0000	100.0000
8	100.0000	100.0000
9	100.0000	100.0000
10	100.0000	100.0000
11	100.0000	90.0000
12	100.0000	100.0000
13	100.0000	100.0000
14	100.0000	100.0000
15	100.0000	100.0000
16	100.0000	100.0000
17	100.0000	95.0000
18	100.0000	100.0000
Average	99.3056	98.0556

TABLE XVII
NN-BASED WEIGHTED PARALLEL-STRUCTURED CLASSIFIER USING 3 FEATURE POINTS. NUMBER OF HIDDEN NODES: 15, TRANSFER FUNCTION OF HIDDEN NODES: HYPERBOLIC TANGENT SIGMOID TRANSFER FUNCTION.

	Recognition Accuracy (%)	
Material	Training	Testing
1	100.0000	100.0000
2	100.0000	100.0000
3	100.0000	100.0000
4	100.0000	100.0000
5	100.0000	100.0000
6	100.0000	95.0000
7	100.0000	100.0000
8	100.0000	100.0000
9	100.0000	100.0000
10	100.0000	100.0000
11	87.5000	80.0000
12	100.0000	100.0000
13	100.0000	100.0000
14	100.0000	100.0000
15	100.0000	100.0000
16	100.0000	100.0000
17	95.0000	95.0000
18	100.0000	100.0000
Average	99.0278	98.3333

TABLE XVIII
NN-BASED WEIGHTED PARALLEL-STRUCTURED CLASSIFIER USING 4 FEATURE POINTS. NUMBER OF HIDDEN NODES: 15, TRANSFER FUNCTION OF HIDDEN NODES: HYPERBOLIC TANGENT SIGMOID TRANSFER FUNCTION.

	Recognition Accuracy (%)	
Material	Training	Testing
1	100.0000	95.0000
2	100.0000	100.0000
3	100.0000	100.0000
4	97.5000	95.0000
5	100.0000	100.0000
6	100.0000	95.0000
7	100.0000	100.0000
8	100.0000	100.0000
9	100.0000	100.0000
10	100.0000	100.0000
11	100.0000	100.0000
12	100.0000	100.0000
13	100.0000	100.0000
14	100.0000	100.0000
15	100.0000	100.0000
16	100.0000	100.0000
17	100.0000	100.0000
18	100.0000	100.0000
Average	99.8611	99.1667

TABLE XIX
NN-BASED WEIGHTED PARALLEL-STRUCTURED CLASSIFIER USING 5 FEATURE POINTS. NUMBER OF HIDDEN NODES: 15, TRANSFER FUNCTION OF HIDDEN NODES: HYPERBOLIC TANGENT SIGMOID TRANSFER FUNCTION.

	Recognition Accuracy (%)	
Material	Training	Testing
1	100.0000	100.0000
2	100.0000	100.0000
3	100.0000	100.0000
4	100.0000	100.0000
5	100.0000	100.0000
6	100.0000	100.0000
7	100.0000	100.0000
8	100.0000	100.0000
9	100.0000	100.0000
10	100.0000	100.0000
11	97.5000	90.0000
12	100.0000	100.0000
13	100.0000	100.0000
14	100.0000	100.0000
15	100.0000	100.0000
16	100.0000	95.0000
17	100.0000	100.0000
18	100.0000	100.0000
Average	99.8611	99.1667

TABLE XX
NN-BASED TREE-STRUCTURED CLASSIFIER USING 3 FEATURE POINTS. NUMBER OF HIDDEN NODES: 20, TRANSFER FUNCTION OF HIDDEN NODES: HYPERBOLIC TANGENT SIGMOID TRANSFER FUNCTION.

	Recognition Accuracy (%)	
Material	Training	Testing
1	100.0000	100.0000
2	100.0000	100.0000
3	100.0000	100.0000
4	100.0000	100.0000
5	100.0000	100.0000
6	100.0000	100.0000
7	100.0000	100.0000
8	100.0000	100.0000
9	100.0000	100.0000
10	100.0000	100.0000
11	100.0000	90.0000
12	100.0000	100.0000
13	100.0000	90.0000
14	100.0000	100.0000
15	100.0000	100.0000
16	100.0000	100.0000
17	100.0000	95.0000
18	100.0000	100.0000
Average	100.0000	98.6111

TABLE XXI
NN-BASED TREE-STRUCTURED CLASSIFIER USING 4 FEATURE POINTS. NUMBER OF HIDDEN NODES: 20, TRANSFER FUNCTION OF HIDDEN NODES: HYPERBOLIC TANGENT SIGMOID TRANSFER FUNCTION.

	Recognition Accuracy (%)	
Material	Training	Testing
1	100.0000	100.0000
2	100.0000	100.0000
3	100.0000	100.0000
4	100.0000	100.0000
5	100.0000	100.0000
6	100.0000	100.0000
7	100.0000	100.0000
8	100.0000	100.0000
9	100.0000	100.0000
10	100.0000	100.0000
11	100.0000	100.0000
12	100.0000	100.0000
13	100.0000	100.0000
14	100.0000	100.0000
15	100.0000	100.0000
16	100.0000	100.0000
17	100.0000	100.0000
18	100.0000	100.0000
Average	100.0000	100.0000

TABLE XXII

NN-BASED TREE-STRUCTURED CLASSIFIER USING 5 FEATURE POINTS. NUMBER OF HIDDEN NODES: 20, TRANSFER FUNCTION OF HIDDEN NODES: HYPERBOLIC TANGENT SIGMOID TRANSFER FUNCTION.

	Recognition Accuracy (%)	
Material	Training	Testing
1	100.0000	95.0000
2	100.0000	95.0000
3	100.0000	100.0000
4	100.0000	95.0000
5	100.0000	95.0000
6	100.0000	100.0000
7	100.0000	100.0000
8	100.0000	100.0000
9	100.0000	95.0000
10	100.0000	100.0000
11	100.0000	85.0000
12	100.0000	100.0000
13	100.0000	100.0000
14	100.0000	100.0000
15	100.0000	100.0000
16	100.0000	95.0000
17	100.0000	80.0000
18	100.0000	90.0000
Average	100.0000	95.8333

TABLE XXIII

K-NEAREST NEIGHBOR CLASSIFIER USING 3 FEATURE POINTS.

	Recognition Accuracy (%)	
Material	Training	Testing
1	100.0000	100.0000
2	100.0000	90.0000
3	100.0000	100.0000
4	100.0000	70.0000
5	100.0000	95.0000
6	100.0000	100.0000
7	100.0000	100.0000
8	100.0000	100.0000
9	100.0000	100.0000
10	100.0000	95.0000
11	100.0000	90.0000
12	100.0000	100.0000
13	100.0000	95.0000
14	100.0000	100.0000
15	100.0000	100.0000
16	100.0000	80.0000
17	100.0000	70.0000
18	100.0000	100.0000
Average	100.0000	93.6111

TABLE XXIV
K-NEAREST NEIGHBOR CLASSIFIER USING 4 FEATURE POINTS.

	Recognition Accuracy (%)	
Material	Training	Testing
1	100.0000	100.0000
2	100.0000	90.0000
3	100.0000	100.0000
4	100.0000	95.0000
5	100.0000	100.0000
6	100.0000	100.0000
7	100.0000	100.0000
8	100.0000	70.0000
9	100.0000	100.0000
10	100.0000	100.0000
11	100.0000	100.0000
12	100.0000	100.0000
13	100.0000	80.0000
14	100.0000	100.0000
15	100.0000	100.0000
16	100.0000	70.0000
17	100.0000	40.0000
18	100.0000	70.0000
Average	100.0000	89.7222

TABLE XXV
K-NEAREST NEIGHBOR CLASSIFIER USING 5 FEATURE POINTS.

	Recognition Accuracy (%)	
Material	Training	Testing
1	100.0000	100.0000
2	100.0000	100.0000
3	100.0000	100.0000
4	100.0000	100.0000
5	100.0000	100.0000
6	100.0000	100.0000
7	100.0000	100.0000
8	100.0000	100.0000
9	100.0000	100.0000
10	100.0000	100.0000
11	100.0000	90.0000
12	100.0000	100.0000
13	100.0000	100.0000
14	100.0000	100.0000
15	100.0000	100.0000
16	100.0000	100.0000
17	100.0000	100.0000
18	100.0000	100.0000
Average	100.0000	99.4444

TABLE XXVI
NAIVE BAYES CLASSIFIER USING 3 FEATURE POINTS.

	Recognition Accuracy (%)	
Material	Training	Testing
1	100.0000	100.0000
2	100.0000	100.0000
3	100.0000	100.0000
4	100.0000	100.0000
5	100.0000	100.0000
6	100.0000	100.0000
7	100.0000	100.0000
8	100.0000	100.0000
9	100.0000	100.0000
10	100.0000	100.0000
11	100.0000	100.0000
12	100.0000	100.0000
13	100.0000	100.0000
14	100.0000	100.0000
15	100.0000	100.0000
16	100.0000	100.0000
17	100.0000	100.0000
18	100.0000	100.0000
Average	100.0000	100.0000

TABLE XXVII
NAIVE BAYES CLASSIFIER USING 4 FEATURE POINTS.

	Recognition Accuracy (%)	
Material	Training	Testing
1	100.0000	100.0000
2	100.0000	100.0000
3	100.0000	100.0000
4	100.0000	100.0000
5	100.0000	100.0000
6	100.0000	100.0000
7	100.0000	100.0000
8	100.0000	100.0000
9	100.0000	100.0000
10	100.0000	100.0000
11	100.0000	100.0000
12	100.0000	100.0000
13	100.0000	100.0000
14	100.0000	100.0000
15	100.0000	100.0000
16	100.0000	100.0000
17	100.0000	100.0000
18	100.0000	100.0000
Average	100.0000	100.0000

TABLE XXVIII
NAIVE BAYES CLASSIFIER USING 5 FEATURE POINTS.

	Recognition Accuracy (%)			
Material	Average	Min	Max	Std
1	99.5000	95.0000	100.0000	1.5811
2	85.5000	80.0000	90.0000	2.8382
3	90.5000	75.0000	100.0000	7.2457
4	97.0000	90.0000	100.0000	3.4960
5	100.0000	100.0000	100.0000	0.0000
6	100.0000	100.0000	100.0000	0.0000
7	100.0000	100.0000	100.0000	0.0000
8	92.5000	90.0000	95.0000	2.6352
9	99.5000	95.0000	100.0000	1.5811
10	95.5000	90.0000	100.0000	3.6893
11	82.0000	75.0000	90.0000	4.2164
12	59.5000	45.0000	75.0000	9.5598
13	100.0000	100.0000	100.0000	0.0000
14	99.5000	95.0000	100.0000	1.5811
15	80.5000	70.0000	90.0000	7.9757
16	95.0000	95.0000	95.0000	0.0000
17	97.0000	95.0000	100.0000	2.5820
18	100.0000	100.0000	100.0000	0.0000
Average	92.9722	88.3333	96.3889	2.7212

TABLE XXIX
NN-BASED ONE-AGAINST-ALL CLASSIFIER USING 3 FEATURE POINTS OF NOISY TEST DATASET. NUMBER OF HIDDEN NODES: 30, TRANSFER FUNCTION OF HIDDEN NODES: LOGARITHMIC SIGMOID TRANSFER FUNCTION.

	Recognition Accuracy (%)			
Material	Average	Min	Max	Std
1	96.0000	90.0000	100.0000	3.9441
2	87.0000	80.0000	95.0000	6.7495
3	75.5000	65.0000	80.0000	4.9721
4	64.5000	50.0000	75.0000	7.9757
5	96.0000	95.0000	100.0000	2.1082
6	99.0000	95.0000	100.0000	2.1082
7	100.0000	100.0000	100.0000	0.0000
8	95.0000	95.0000	95.0000	0.0000
9	100.0000	100.0000	100.0000	0.0000
10	82.0000	75.0000	95.0000	5.8689
11	86.5000	75.0000	95.0000	5.2967
12	19.0000	10.0000	25.0000	6.1464
13	95.0000	95.0000	95.0000	0.0000
14	99.0000	95.0000	100.0000	2.1082
15	0.5000	0.0000	5.0000	1.5811
16	100.0000	100.0000	100.0000	0.0000
17	99.0000	95.0000	100.0000	2.1082
18	100.0000	100.0000	100.0000	0.0000
Average	83.0000	78.6111	86.6667	2.8315

TABLE XXX

NN-based one-against-all classifier using 4 feature points of noisy test dataset. Number of hidden nodes: 30, transfer function of hidden nodes: logarithmic sigmoid transfer function.

	Recognition Accuracy (%)			
Material	Average	Min	Max	Std
1	84.5000	75.0000	90.0000	4.9721
2	91.5000	85.0000	100.0000	4.7434
3	86.5000	75.0000	90.0000	5.2967
4	96.5000	90.0000	100.0000	3.3747
5	99.5000	95.0000	100.0000	1.5811
6	100.0000	100.0000	100.0000	0.0000
7	98.5000	95.0000	100.0000	2.4152
8	86.0000	80.0000	90.0000	3.9441
9	95.0000	90.0000	100.0000	4.0825
10	100.0000	100.0000	100.0000	0.0000
11	95.0000	85.0000	100.0000	5.2705
12	99.0000	95.0000	100.0000	2.1082
13	97.5000	95.0000	100.0000	2.6352
14	100.0000	100.0000	100.0000	0.0000
15	56.5000	40.0000	85.0000	12.4833
16	98.5000	95.0000	100.0000	2.4152
17	95.0000	95.0000	95.0000	0.0000
18	95.0000	90.0000	100.0000	3.3333
Average	93.0278	87.7778	97.2222	3.2587

TABLE XXXI

NN-based one-against-all classifier using 5 feature points of noisy test dataset. Number of hidden nodes: 30, transfer function of hidden nodes: logarithmic sigmoid transfer function.

	Recognition Accuracy (%)			
Material	Average	Min	Max	Std
1	76.5000	70.0000	85.0000	5.2967
2	80.5000	70.0000	90.0000	6.4334
3	82.0000	75.0000	85.0000	3.4960
4	97.0000	95.0000	100.0000	2.5820
5	98.5000	95.0000	100.0000	2.4152
6	94.0000	90.0000	100.0000	3.9441
7	100.0000	100.0000	100.0000	0.0000
8	96.0000	90.0000	100.0000	3.9441
9	67.5000	45.0000	80.0000	11.1181
10	99.5000	95.0000	100.0000	1.5811
11	95.5000	90.0000	100.0000	3.6893
12	89.0000	80.0000	95.0000	5.1640
13	86.0000	85.0000	90.0000	2.1082
14	100.0000	100.0000	100.0000	0.0000
15	0.0000	0.0000	0.0000	0.0000
16	89.5000	85.0000	90.0000	1.5811
17	97.5000	95.0000	100.0000	2.6352
18	85.5000	80.0000	95.0000	5.9861
Average	85.2500	80.0000	89.4444	3.4430

TABLE XXXII

NN-based weighted one-against-all classifier using 3 feature points of noisy test dataset. Number of hidden nodes: 20, transfer function of hidden nodes: hyperbolic tangent sigmoid transfer function.

	Recognition Accuracy (%)			
Material	Average	Min	Max	Std
1	78.0000	70.0000	85.0000	4.2164
2	87.5000	85.0000	90.0000	2.6352
3	71.5000	60.0000	80.0000	5.2967
4	100.0000	100.0000	100.0000	0.0000
5	96.5000	95.0000	100.0000	2.4152
6	99.0000	95.0000	100.0000	2.1082
7	99.0000	95.0000	100.0000	2.1082
8	92.5000	90.0000	95.0000	2.6352
9	100.0000	100.0000	100.0000	0.0000
10	100.0000	100.0000	100.0000	0.0000
11	95.5000	90.0000	100.0000	2.8382
12	38.5000	20.0000	55.0000	11.5590
13	100.0000	100.0000	100.0000	0.0000
14	100.0000	100.0000	100.0000	0.0000
15	0.0000	0.0000	0.0000	0.0000
16	95.5000	95.0000	100.0000	1.5811
17	83.5000	75.0000	90.0000	4.1164
18	100.0000	100.0000	100.0000	0.0000
Average	85.3889	81.6667	88.6111	2.3061

TABLE XXXIII

NN-based weighted one-against-all classifier using 4 feature points of noisy test dataset. Number of hidden nodes: 20, transfer function of hidden nodes: hyperbolic tangent sigmoid transfer function.

	Recognition Accuracy (%)			
Material	Average	Min	Max	Std
1	91.5000	80.0000	100.0000	5.7975
2	99.0000	95.0000	100.0000	2.1082
3	96.0000	85.0000	100.0000	5.1640
4	100.0000	100.0000	100.0000	0.0000
5	97.5000	95.0000	100.0000	2.6352
6	100.0000	100.0000	100.0000	0.0000
7	96.5000	95.0000	100.0000	2.4152
8	95.5000	95.0000	100.0000	1.5811
9	100.0000	100.0000	100.0000	0.0000
10	100.0000	100.0000	100.0000	0.0000
11	98.0000	95.0000	100.0000	2.5820
12	100.0000	100.0000	100.0000	0.0000
13	100.0000	100.0000	100.0000	0.0000
14	96.5000	95.0000	100.0000	2.4152
15	43.0000	35.0000	60.0000	7.5277
16	100.0000	100.0000	100.0000	0.0000
17	93.5000	90.0000	95.0000	2.4152
18	100.0000	100.0000	100.0000	0.0000
Average	94.8333	92.2222	97.5000	1.9245

TABLE XXXIV

NN-BASED WEIGHTED ONE-AGAINST-ALL CLASSIFIER USING 5 FEATURE POINTS OF NOISY TEST DATASET. NUMBER OF HIDDEN NODES: 20, TRANSFER FUNCTION OF HIDDEN NODES: HYPERBOLIC TANGENT SIGMOID TRANSFER FUNCTION.

	Recognition Accuracy (%)			
Material	Average	Min	Max	Std
1	100.0000	100.0000	100.0000	0.0000
2	94.0000	90.0000	100.0000	3.1623
3	90.5000	80.0000	100.0000	5.9861
4	100.0000	100.0000	100.0000	0.0000
5	100.0000	100.0000	100.0000	0.0000
6	100.0000	100.0000	100.0000	0.0000
7	99.5000	95.0000	100.0000	1.5811
8	94.5000	90.0000	95.0000	1.5811
9	100.0000	100.0000	100.0000	0.0000
10	100.0000	100.0000	100.0000	0.0000
11	96.5000	95.0000	100.0000	2.4152
12	100.0000	100.0000	100.0000	0.0000
13	95.0000	95.0000	95.0000	0.0000
14	100.0000	100.0000	100.0000	0.0000
15	74.5000	65.0000	90.0000	6.8516
16	100.0000	100.0000	100.0000	0.0000
17	97.0000	95.0000	100.0000	2.5820
18	99.0000	95.0000	100.0000	2.1082
Average	96.6944	94.4444	98.8889	1.4593

TABLE XXXV

NN-BASED BINARY-CODED-OUTPUT CLASSIFIER USING 3 FEATURE POINTS OF NOISY TEST DATASET. NUMBER OF HIDDEN NODES: 30, TRANSFER FUNCTION OF HIDDEN NODES: LOGARITHMIC SIGMOID TRANSFER FUNCTION.

	Recognition Accuracy (%)			
Material	Average	Min	Max	Std
1	92.0000	80.0000	100.0000	5.3748
2	99.5000	95.0000	100.0000	1.5811
3	64.5000	50.0000	75.0000	7.9757
4	100.0000	100.0000	100.0000	0.0000
5	100.0000	100.0000	100.0000	0.0000
6	100.0000	100.0000	100.0000	0.0000
7	100.0000	100.0000	100.0000	0.0000
8	100.0000	100.0000	100.0000	0.0000
9	100.0000	100.0000	100.0000	0.0000
10	100.0000	100.0000	100.0000	0.0000
11	98.0000	90.0000	100.0000	3.4960
12	75.0000	60.0000	90.0000	11.0554
13	95.0000	95.0000	95.0000	0.0000
14	100.0000	100.0000	100.0000	0.0000
15	72.5000	60.0000	80.0000	7.9057
16	100.0000	100.0000	100.0000	0.0000
17	94.5000	90.0000	95.0000	1.5811
18	100.0000	100.0000	100.0000	0.0000
Average	93.9444	90.0000	96.3889	2.1650

TABLE XXXVI
NN-BASED BINARY-CODED-OUTPUT CLASSIFIER USING 4 FEATURE POINTS OF NOISY TEST DATASET. NUMBER OF HIDDEN NODES: 30, TRANSFER FUNCTION OF HIDDEN NODES: LOGARITHMIC SIGMOID TRANSFER FUNCTION.

	Recognition Accuracy (%)			
Material	Average	Min	Max	Std
1	97.5000	95.0000	100.0000	2.6352
2	100.0000	100.0000	100.0000	0.0000
3	99.5000	95.0000	100.0000	1.5811
4	100.0000	100.0000	100.0000	0.0000
5	100.0000	100.0000	100.0000	0.0000
6	100.0000	100.0000	100.0000	0.0000
7	100.0000	100.0000	100.0000	0.0000
8	99.5000	95.0000	100.0000	1.5811
9	100.0000	100.0000	100.0000	0.0000
10	100.0000	100.0000	100.0000	0.0000
11	96.5000	90.0000	100.0000	3.3747
12	100.0000	100.0000	100.0000	0.0000
13	100.0000	100.0000	100.0000	0.0000
14	100.0000	100.0000	100.0000	0.0000
15	94.0000	80.0000	100.0000	6.5828
16	98.0000	95.0000	100.0000	2.5820
17	95.0000	95.0000	95.0000	0.0000
18	99.5000	95.0000	100.0000	1.5811
Average	98.8611	96.6667	99.7222	1.1066

TABLE XXXVII
NN-BASED BINARY-CODED-OUTPUT CLASSIFIER USING 5 FEATURE POINTS OF NOISY TEST DATASET. NUMBER OF HIDDEN NODES: 30, TRANSFER FUNCTION OF HIDDEN NODES: LOGARITHMIC SIGMOID TRANSFER FUNCTION.

	Recognition Accuracy (%)			
Material	Average	Min	Max	Std
1	99.5000	95.0000	100.0000	1.5811
2	97.5000	95.0000	100.0000	2.6352
3	72.0000	60.0000	85.0000	6.7495
4	100.0000	100.0000	100.0000	0.0000
5	100.0000	100.0000	100.0000	0.0000
6	100.0000	100.0000	100.0000	0.0000
7	100.0000	100.0000	100.0000	0.0000
8	100.0000	100.0000	100.0000	0.0000
9	97.0000	95.0000	100.0000	2.5820
10	100.0000	100.0000	100.0000	0.0000
11	99.0000	95.0000	100.0000	2.1082
12	75.0000	55.0000	90.0000	11.7851
13	100.0000	100.0000	100.0000	0.0000
14	100.0000	100.0000	100.0000	0.0000
15	51.5000	25.0000	60.0000	12.2588
16	100.0000	100.0000	100.0000	0.0000
17	100.0000	100.0000	100.0000	0.0000
18	100.0000	100.0000	100.0000	0.0000
Average	93.9722	90.0000	96.3889	2.2056

TABLE XXXVIII

NN-BASED PARALLEL-STRUCTURED CLASSIFIER USING 3 FEATURE POINTS OF NOISY TEST DATASET. NUMBER OF HIDDEN NODES: 10, TRANSFER FUNCTION OF HIDDEN NODES: LOGARITHMIC SIGMOID TRANSFER FUNCTION.

	Recognition Accuracy (%)			
Material	Average	Min	Max	Std
1	95.5000	85.0000	100.0000	4.3780
2	97.5000	95.0000	100.0000	2.6352
3	80.5000	75.0000	85.0000	4.3780
4	100.0000	100.0000	100.0000	0.0000
5	100.0000	100.0000	100.0000	0.0000
6	100.0000	100.0000	100.0000	0.0000
7	100.0000	100.0000	100.0000	0.0000
8	100.0000	100.0000	100.0000	0.0000
9	100.0000	100.0000	100.0000	0.0000
10	100.0000	100.0000	100.0000	0.0000
11	100.0000	100.0000	100.0000	0.0000
12	99.0000	95.0000	100.0000	2.1082
13	100.0000	100.0000	100.0000	0.0000
14	100.0000	100.0000	100.0000	0.0000
15	2.5000	0.0000	5.0000	2.6352
16	100.0000	100.0000	100.0000	0.0000
17	100.0000	100.0000	100.0000	0.0000
18	95.0000	90.0000	100.0000	2.3570
Average	92.7778	91.1111	93.8889	1.0273

TABLE XXXIX

NN-BASED PARALLEL-STRUCTURED CLASSIFIER USING 4 FEATURE POINTS OF NOISY TEST DATASET. NUMBER OF HIDDEN NODES: 10, TRANSFER FUNCTION OF HIDDEN NODES: LOGARITHMIC SIGMOID TRANSFER FUNCTION.

	Recognition Accuracy (%)			
Material	Average	Min	Max	Std
1	100.0000	100.0000	100.0000	0.0000
2	100.0000	100.0000	100.0000	0.0000
3	100.0000	100.0000	100.0000	0.0000
4	100.0000	100.0000	100.0000	0.0000
5	100.0000	100.0000	100.0000	0.0000
6	100.0000	100.0000	100.0000	0.0000
7	100.0000	100.0000	100.0000	0.0000
8	100.0000	100.0000	100.0000	0.0000
9	100.0000	100.0000	100.0000	0.0000
10	100.0000	100.0000	100.0000	0.0000
11	99.5000	95.0000	100.0000	1.5811
12	100.0000	100.0000	100.0000	0.0000
13	100.0000	100.0000	100.0000	0.0000
14	100.0000	100.0000	100.0000	0.0000
15	25.0000	10.0000	40.0000	10.8012
16	100.0000	100.0000	100.0000	0.0000
17	100.0000	100.0000	100.0000	0.0000
18	100.0000	100.0000	100.0000	0.0000
Average	95.8056	94.7222	96.6667	0.6879

TABLE XL
NN-BASED PARALLEL-STRUCTURED CLASSIFIER USING 5 FEATURE POINTS OF NOISY TEST DATASET. NUMBER OF HIDDEN NODES: 10, TRANSFER FUNCTION OF HIDDEN NODES: LOGARITHMIC SIGMOID TRANSFER FUNCTION.

	Recognition Accuracy (%)			
Material	Average	Min	Max	Std
1	67.5000	55.0000	75.0000	6.7700
2	99.0000	95.0000	100.0000	2.1082
3	75.0000	70.0000	80.0000	3.3333
4	91.0000	80.0000	100.0000	5.6765
5	95.0000	90.0000	100.0000	2.3570
6	99.0000	95.0000	100.0000	2.1082
7	100.0000	100.0000	100.0000	0.0000
8	100.0000	100.0000	100.0000	0.0000
9	100.0000	100.0000	100.0000	0.0000
10	100.0000	100.0000	100.0000	0.0000
11	94.5000	90.0000	100.0000	4.3780
12	100.0000	100.0000	100.0000	0.0000
13	100.0000	100.0000	100.0000	0.0000
14	100.0000	100.0000	100.0000	0.0000
15	5.5000	0.0000	10.0000	4.3780
16	99.0000	95.0000	100.0000	2.1082
17	98.0000	95.0000	100.0000	2.5820
18	100.0000	100.0000	100.0000	0.0000
Average	90.1944	86.9444	92.5000	1.9889

TABLE XLI
NN-BASED WEIGHTED PARALLEL-STRUCTURED CLASSIFIER USING 3 FEATURE POINTS OF NOISY TEST DATASET. NUMBER OF HIDDEN NODES: 15, TRANSFER FUNCTION OF HIDDEN NODES: HYPERBOLIC TANGENT SIGMOID TRANSFER FUNCTION.

	Recognition Accuracy (%)			
Material	Average	Min	Max	Std
1	99.5000	95.0000	100.0000	1.5811
2	97.5000	95.0000	100.0000	2.6352
3	96.5000	95.0000	100.0000	2.4152
4	99.0000	95.0000	100.0000	2.1082
5	96.5000	90.0000	100.0000	4.1164
6	99.0000	95.0000	100.0000	2.1082
7	100.0000	100.0000	100.0000	0.0000
8	100.0000	100.0000	100.0000	0.0000
9	100.0000	100.0000	100.0000	0.0000
10	99.5000	95.0000	100.0000	1.5811
11	86.5000	80.0000	95.0000	6.2583
12	10.5000	0.0000	25.0000	7.6194
13	100.0000	100.0000	100.0000	0.0000
14	100.0000	100.0000	100.0000	0.0000
15	100.0000	100.0000	100.0000	0.0000
16	99.5000	95.0000	100.0000	1.5811
17	93.0000	90.0000	95.0000	2.5820
18	100.0000	100.0000	100.0000	0.0000
Average	93.1667	90.2778	95.2778	1.9215

TABLE XLII
NN-BASED WEIGHTED PARALLEL-STRUCTURED CLASSIFIER USING 4 FEATURE POINTS OF NOISY TEST DATASET. NUMBER OF HIDDEN NODES: 15, TRANSFER FUNCTION OF HIDDEN NODES: HYPERBOLIC TANGENT SIGMOID TRANSFER FUNCTION.

	Recognition Accuracy (%)			
Material	Average	Min	Max	Std
1	90.0000	85.0000	100.0000	4.7140
2	99.0000	95.0000	100.0000	2.1082
3	98.5000	95.0000	100.0000	2.4152
4	89.0000	75.0000	95.0000	6.5828
5	99.5000	95.0000	100.0000	1.5811
6	97.5000	95.0000	100.0000	2.6352
7	100.0000	100.0000	100.0000	0.0000
8	100.0000	100.0000	100.0000	0.0000
9	100.0000	100.0000	100.0000	0.0000
10	100.0000	100.0000	100.0000	0.0000
11	100.0000	100.0000	100.0000	0.0000
12	91.5000	70.0000	100.0000	8.8349
13	100.0000	100.0000	100.0000	0.0000
14	100.0000	100.0000	100.0000	0.0000
15	73.5000	55.0000	85.0000	9.1439
16	100.0000	100.0000	100.0000	0.0000
17	100.0000	100.0000	100.0000	0.0000
18	100.0000	100.0000	100.0000	0.0000
Average	96.5833	92.5000	98.8889	2.1120

TABLE XLIII
NN-BASED WEIGHTED PARALLEL-STRUCTURED CLASSIFIER USING 5 FEATURE POINTS OF NOISY TEST DATASET. NUMBER OF HIDDEN NODES: 15, TRANSFER FUNCTION OF HIDDEN NODES: HYPERBOLIC TANGENT SIGMOID TRANSFER FUNCTION.

	Recognition Accuracy (%)			
Material	Average	Min	Max	Std
1	100.0000	100.0000	100.0000	0.0000
2	100.0000	100.0000	100.0000	0.0000
3	89.5000	85.0000	95.0000	3.6893
4	100.0000	100.0000	100.0000	0.0000
5	100.0000	100.0000	100.0000	0.0000
6	100.0000	100.0000	100.0000	0.0000
7	100.0000	100.0000	100.0000	0.0000
8	100.0000	100.0000	100.0000	0.0000
9	100.0000	100.0000	100.0000	0.0000
10	100.0000	100.0000	100.0000	0.0000
11	95.0000	90.0000	100.0000	2.3570
12	100.0000	100.0000	100.0000	0.0000
13	100.0000	100.0000	100.0000	0.0000
14	100.0000	100.0000	100.0000	0.0000
15	80.5000	65.0000	90.0000	8.9598
16	96.5000	95.0000	100.0000	2.4152
17	100.0000	100.0000	100.0000	0.0000
18	100.0000	100.0000	100.0000	0.0000
Average	97.8611	96.3889	99.1667	0.9679

TABLE XLIV
NN-BASED TREE-STRUCTURED CLASSIFIER USING 3 FEATURE POINTS OF NOISY TEST DATASET. NUMBER OF HIDDEN NODES: 20, TRANSFER FUNCTION OF HIDDEN NODES: HYPERBOLIC TANGENT SIGMOID TRANSFER FUNCTION.

	Recognition Accuracy (%)			
Material	Average	Min	Max	Std
1	95.0000	90.0000	100.0000	2.3570
2	95.5000	90.0000	100.0000	2.8382
3	75.5000	65.0000	90.0000	6.8516
4	100.0000	100.0000	100.0000	0.0000
5	100.0000	100.0000	100.0000	0.0000
6	100.0000	100.0000	100.0000	0.0000
7	100.0000	100.0000	100.0000	0.0000
8	100.0000	100.0000	100.0000	0.0000
9	100.0000	100.0000	100.0000	0.0000
10	100.0000	100.0000	100.0000	0.0000
11	96.0000	90.0000	100.0000	3.1623
12	99.5000	95.0000	100.0000	1.5811
13	91.0000	90.0000	95.0000	2.1082
14	100.0000	100.0000	100.0000	0.0000
15	100.0000	100.0000	100.0000	0.0000
16	100.0000	100.0000	100.0000	0.0000
17	100.0000	100.0000	100.0000	0.0000
18	100.0000	100.0000	100.0000	0.0000
Average	97.3611	95.5556	99.1667	1.0499

TABLE XLV
NN-BASED TREE-STRUCTURED CLASSIFIER USING 4 FEATURE POINTS OF NOISY TEST DATASET. NUMBER OF HIDDEN NODES: 20, TRANSFER FUNCTION OF HIDDEN NODES: HYPERBOLIC TANGENT SIGMOID TRANSFER FUNCTION.

	Recognition Accuracy (%)			
Material	Average	Min	Max	Std
1	100.0000	100.0000	100.0000	0.0000
2	96.0000	85.0000	100.0000	5.1640
3	100.0000	100.0000	100.0000	0.0000
4	100.0000	100.0000	100.0000	0.0000
5	100.0000	100.0000	100.0000	0.0000
6	100.0000	100.0000	100.0000	0.0000
7	100.0000	100.0000	100.0000	0.0000
8	100.0000	100.0000	100.0000	0.0000
9	100.0000	100.0000	100.0000	0.0000
10	100.0000	100.0000	100.0000	0.0000
11	100.0000	100.0000	100.0000	0.0000
12	100.0000	100.0000	100.0000	0.0000
13	100.0000	100.0000	100.0000	0.0000
14	100.0000	100.0000	100.0000	0.0000
15	100.0000	100.0000	100.0000	0.0000
16	100.0000	100.0000	100.0000	0.0000
17	100.0000	100.0000	100.0000	0.0000
18	100.0000	100.0000	100.0000	0.0000
Average	99.7778	99.1667	100.0000	0.2869

TABLE XLVI
NN-BASED TREE-STRUCTURED CLASSIFIER USING 5 FEATURE POINTS OF NOISY TEST DATASET. NUMBER OF HIDDEN NODES: 20, TRANSFER FUNCTION OF HIDDEN NODES: HYPERBOLIC TANGENT SIGMOID TRANSFER FUNCTION.

	Recognition Accuracy (%)			
Material	Average	Min	Max	Std
1	93.0000	90.0000	100.0000	4.2164
2	88.5000	75.0000	100.0000	8.8349
3	93.0000	85.0000	100.0000	4.8305
4	92.0000	85.0000	100.0000	4.2164
5	95.0000	90.0000	100.0000	3.3333
6	100.0000	100.0000	100.0000	0.0000
7	100.0000	100.0000	100.0000	0.0000
8	100.0000	100.0000	100.0000	0.0000
9	94.5000	90.0000	100.0000	2.8382
10	98.5000	95.0000	100.0000	2.4152
11	84.0000	75.0000	90.0000	5.1640
12	91.0000	80.0000	100.0000	6.1464
13	100.0000	100.0000	100.0000	0.0000
14	100.0000	100.0000	100.0000	0.0000
15	91.0000	75.0000	100.0000	7.3786
16	92.5000	90.0000	95.0000	2.6352
17	84.5000	75.0000	90.0000	4.9721
18	92.5000	90.0000	95.0000	2.6352
Average	93.8889	88.6111	98.3333	3.3120

TABLE XLVII
K-NEAREST NEIGHBOR CLASSIFIER USING 3 FEATURE POINTS OF NOISY TEST DATASET.

	Recognition Accuracy (%)			
Material	Average	Min	Max	Std
1	86.6667	80.0000	90.0000	5.7735
2	71.6667	65.0000	80.0000	7.6376
3	66.6667	55.0000	80.0000	12.5831
4	90.0000	90.0000	90.0000	0.0000
5	91.6667	85.0000	100.0000	7.6376
6	96.6667	95.0000	100.0000	2.8868
7	100.0000	100.0000	100.0000	0.0000
8	100.0000	100.0000	100.0000	0.0000
9	96.6667	95.0000	100.0000	2.8868
10	90.0000	85.0000	100.0000	8.6603
11	88.3333	85.0000	90.0000	2.8868
12	81.6667	75.0000	90.0000	7.6376
13	95.0000	95.0000	95.0000	0.0000
14	96.6667	95.0000	100.0000	2.8868
15	75.0000	65.0000	85.0000	10.0000
16	76.6667	75.0000	80.0000	2.8868
17	75.0000	70.0000	80.0000	5.0000
18	98.3333	95.0000	100.0000	2.8868
Average	87.5926	83.6111	92.2222	4.5695

TABLE XLVIII
K-NEAREST NEIGHBOR CLASSIFIER USING 4 FEATURE POINTS OF NOISY TEST DATASET.

	Recognition Accuracy (%)			
Material	Average	Min	Max	Std
1	96.5000	90.0000	100.0000	3.3747
2	92.5000	90.0000	95.0000	2.6352
3	99.5000	95.0000	100.0000	1.5811
4	93.0000	90.0000	100.0000	3.4960
5	100.0000	100.0000	100.0000	0.0000
6	100.0000	100.0000	100.0000	0.0000
7	100.0000	100.0000	100.0000	0.0000
8	71.5000	65.0000	75.0000	3.3747
9	100.0000	100.0000	100.0000	0.0000
10	100.0000	100.0000	100.0000	0.0000
11	97.5000	90.0000	100.0000	3.5355
12	94.5000	85.0000	100.0000	4.3780
13	81.5000	75.0000	90.0000	4.7434
14	100.0000	100.0000	100.0000	0.0000
15	90.5000	75.0000	100.0000	8.3166
16	66.0000	50.0000	75.0000	8.0966
17	39.5000	35.0000	50.0000	4.3780
18	73.0000	60.0000	85.0000	7.1492
Average	88.6389	83.3333	92.7778	3.0588

TABLE XLIX
K-NEAREST NEIGHBOR CLASSIFIER USING 5 FEATURE POINTS OF NOISY TEST DATASET.

	Recognition Accuracy (%)			
Material	Average	Min	Max	Std
1	100.0000	100.0000	100.0000	0.0000
2	99.5000	95.0000	100.0000	1.5811
3	86.0000	80.0000	90.0000	3.1623
4	100.0000	100.0000	100.0000	0.0000
5	100.0000	100.0000	100.0000	0.0000
6	100.0000	100.0000	100.0000	0.0000
7	100.0000	100.0000	100.0000	0.0000
8	100.0000	100.0000	100.0000	0.0000
9	98.0000	95.0000	100.0000	2.5820
10	100.0000	100.0000	100.0000	0.0000
11	99.5000	95.0000	100.0000	1.5811
12	100.0000	100.0000	100.0000	0.0000
13	100.0000	100.0000	100.0000	0.0000
14	100.0000	100.0000	100.0000	0.0000
15	0.5000	0.0000	5.0000	1.5811
16	100.0000	100.0000	100.0000	0.0000
17	100.0000	100.0000	100.0000	0.0000
18	100.0000	100.0000	100.0000	0.0000
Average	93.5278	92.5000	94.1667	0.5826

TABLE L
NAIVE BAYES CLASSIFIER USING 3 FEATURE POINTS OF NOISY TEST DATASET.

	Recognition Accuracy (%)			
Material	Average	Min	Max	Std
1	100.0000	100.0000	100.0000	0.0000
2	97.0000	95.0000	100.0000	2.5820
3	77.0000	65.0000	90.0000	8.2327
4	100.0000	100.0000	100.0000	0.0000
5	100.0000	100.0000	100.0000	0.0000
6	100.0000	100.0000	100.0000	0.0000
7	100.0000	100.0000	100.0000	0.0000
8	100.0000	100.0000	100.0000	0.0000
9	100.0000	100.0000	100.0000	0.0000
10	100.0000	100.0000	100.0000	0.0000
11	100.0000	100.0000	100.0000	0.0000
12	100.0000	100.0000	100.0000	0.0000
13	100.0000	100.0000	100.0000	0.0000
14	100.0000	100.0000	100.0000	0.0000
15	0.5000	0.0000	5.0000	1.5811
16	100.0000	100.0000	100.0000	0.0000
17	100.0000	100.0000	100.0000	0.0000
18	100.0000	100.0000	100.0000	0.0000
Average	93.0278	92.2222	94.1667	0.6887

TABLE LI
NAIVE BAYES CLASSIFIER USING 4 FEATURE POINTS OF NOISY TEST DATASET.

	Recognition Accuracy (%)			
Material	Average	Min	Max	Std
1	100.0000	100.0000	100.0000	0.0000
2	100.0000	100.0000	100.0000	0.0000
3	90.0000	80.0000	95.0000	5.7735
4	100.0000	100.0000	100.0000	0.0000
5	100.0000	100.0000	100.0000	0.0000
6	100.0000	100.0000	100.0000	0.0000
7	100.0000	100.0000	100.0000	0.0000
8	100.0000	100.0000	100.0000	0.0000
9	100.0000	100.0000	100.0000	0.0000
10	100.0000	100.0000	100.0000	0.0000
11	100.0000	100.0000	100.0000	0.0000
12	100.0000	100.0000	100.0000	0.0000
13	100.0000	100.0000	100.0000	0.0000
14	100.0000	100.0000	100.0000	0.0000
15	0.5000	0.0000	5.0000	1.5811
16	100.0000	100.0000	100.0000	0.0000
17	100.0000	100.0000	100.0000	0.0000
18	100.0000	100.0000	100.0000	0.0000
Average	93.9167	93.3333	94.4444	0.4086

TABLE LII
NAIVE BAYES CLASSIFIER USING 5 FEATURE POINTS OF NOISY TEST DATASET.

REFERENCES

[1] J. Zurada, "Could decision trees improve the classification accuracy and interpretability of loan granting decisions?" in *Proceedings of the 43rd Hawaii International Conference on System Sciences (HICSS)*, Jan. 2010, pp. 1–9.

[2] J. Pradeep, E. Srinivasan, and S. Himavathi, "Neural network based handwritten character recognition system without feature extraction," in *Proceedings of the International Conference on Computer, Communication and Electrical Technology (ICCCET)*, Jan. 2011, pp. 40–44.

[3] J.-T. Chien, "Linear regression based bayesian predictive classification for speech recognition," *IEEE Transactions on Speech and Audio Processing*, vol. 11, no. 1, pp. 70–79, Jan. 2003.

[4] G. L. Marcialis and F. Roli, "Fusion of multiple fingerprint matchers by single-layer perceptron with class-separation loss function," *Pattern Recognition Letters*, vol. 26, no. 12, pp. 1830–1839, Sep. 2005.

[5] G. P. Zhang, "Neural networks for classification: a survey," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 30, no. 4, pp. 451–462, Nov. 2000.

[6] A. Tellaeche, X. P. Burgos-Artizzu, G. Pajares, and A. Ribeiro, "A vision-based method for weeds identification through the Bayesian decision theory," *Pattern Recognition*, vol. 41, no. 2, pp. 521–530, Feb 2008.

[7] M.-L. Zhang and Z.-H. Zhou, "A k-nearest neighbor based algorithm for multi-label classification," in *Proceedings of the IEEE International Conference on Granular Computing*, vol. 2, 2005, pp. 718–721.

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

[9] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition," *Data mining and knowledge discovery*, vol. 2, no. 2, pp. 121–167, 1998.

[10] S. Kotsiantis, "Supervised machine learning: A review of classification techniques," *Informatica*, vol. 31, pp. 249–268, Jul. 2007.

[11] Y.-S. Chen and T.-H. Chu, "A neural network classification tree," in *Proceedings of the IEEE International Conference on Neural Networks*, vol. 1, Dec. 1995, pp. 409–413.

[12] A. Santillana Fernandez, C. Delgado-Mata, and R. Velazquez, "Training a single-layer perceptron for an approximate edge detection on a digital image," in *Proceedings of the International Conference on Technologies and Applications of Artificial Intelligence (TAAI)*, Nov. 2011, pp. 189–193.

[13] U. Kumar, S. K. Raja, C. Mukhopadhyay, and T. Ramachandra, "Hybrid bayesian classifier for improved classification accuracy," *IEEE Geoscience and Remote Sensing Letters*, vol. 8, no. 3, pp. 474–477, Nov. 2011.

[14] L. Zhou, L. Wang, X. Ge, and Q. Shi, "A clustering-based knn improved algorithm CLKNN for text classification," in *Proceedings of the 2^{nd} International Asia Conference on Informatics in Control, Automation and Robotics (CAR)*, vol. 3, 2010, pp. 212–215.

[15] K. Hornik, M. Stinchcombe, and H. White, "Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks," *Neural networks*, vol. 3, no. 5, pp. 551–560, 1990.

[16] K. Schierholt and C. H. Dagli, "Stock market prediction using different neural network classification architectures," in *Proceedings of the IEEEIAFE 1996 Conference on Computational Intelligence for Financial Engineering*, 1996, pp. 72–78.

[17] X. Li, J. Cervantes, and W. Yu, "A novel SVM classification method for large data sets," in *IEEE International Conference on Granular Computing (GrC)*, 2010, pp. 297–302.

[18] A. M. Okamura, "Methods for haptic feedback in teleoperated robot-assisted surgery," *Industrial Robot: An International Journal*, vol. 31, no. 6, pp. 499–508, 2004.

[19] O. A. J. Van der Meijden and M. P. Schijven, "The value of haptic feedback in conventional and robot-assisted minimal invasive surgery and virtual reality training: a current review," *Surgical endoscopy*, vol. 23, no. 6, pp. 1180–1190, 2009.

[20] C.-H. King, M. O. Culjat, M. L. Franco, C. E. Lewis, E. P. Dutson, W. S. Grundfest, and J. W. Bisley, "Tactile feedback induces reduced grasping force in robot-assisted surgery," *Haptics, IEEE Transactions on*, vol. 2, no. 2, pp. 103–110, 2009.

[21] G. I. Barbash and S. A. Glied, "New technology and health care costs-the case of robot-assisted surgery," *New England Journal of Medicine*, vol. 363, no. 8, pp. 701–704, 2010.

[22] R. Ozawa, J.-H. Bae, and S. Arimoto, "Multi-fingered dynamic blind grasping with tactile feedback in a horizontal plane," in *Proceedings of the 2006 IEEE International Conference on Robotics and Automation (ICRA)*, 2006, pp. 1006–1011.

[23] H. Dang, J. Weisz, and P. K. Allen, "Blind grasping: Stable robotic grasping using tactile feedback and hand kinematics," in *Proceedings of the 2011 IEEE International Conference on Robotics and Automation (ICRA)*, 2011, pp. 5917–5922.

[24] H. Liu, X. Song, T. Nanayakkara, L. D. Seneviratne, and K. Althoefer, "A computationally fast algorithm for local contact shape and pose classification using a tactile array sensor," in *Proceedings of 2012 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2012, pp. 1410–1415.

[25] R. E. Fan, M. O. Culjat, C.-H. King, M. L. Franco, R. Boryk, J. W. Bisley, E. Dutson, and W. S. Grundfest, "A haptic feedback system for lower-limb prostheses," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 16, no. 3, pp. 270–277, 2008.

[26] N. Jamali, "Material classification by tactile sensing using surface textures," in *Proceedings of the 2007 IEEE International Conference on Image Processing*, May 2010, pp. 2336–2341.

[27] V. Thilak, "Material classification using passive polametric imagery," in *Proceedings of the 2007 IEEE International Conference on Image Processing*, Sept. 2007, pp. 121–124.

[28] N. Charniya, "Neural network based sensor for classification of material type and its surface properties," in *Proceedings of the 2007 IEEE International Joint Conference on Neural Networks*, Aug. 2007, pp. 424–429.

[29] J. E. ODoherty, M. A. Lebedev, P. J. Ifft, K. Z. Zhuang, S. Shokur, H. Bleuler, and M. A. L. Nicolelis, "Active tactile exploration using a brain-machine-brain interface," *Nature*, vol. 479, no. 7372, pp. 228–231, 2011.

[30] X. Song, H. Liu, J. Bimbo, K. Althoefer, and L. D. Seneviratne, "A novel dynamic slip prediction and compensation approach based on haptic surface exploration," in *Proceedings of 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS),*. IEEE, 2012, pp. 4511–4516.

[31] X. Song, H. Liu, K. Althoefer, T. Nanayakkara, and L. D. Seneviratne, "Efficient break-away friction ratio and slip prediction based on haptic surface exploration," *IEEE Transactions on Robotics*, 2013, DOI: 10.1109/TRO.2013.2279630.

[32] K. F. Leung, F. H. F. Leung, H. K. Lam, and S. H. Ling, "On interpretation of graffiti digits and characters for eBooks: neural-fuzzy network and genetic algorithm approach," *IEEE Trans. on Industrial Electronics*, vol. 51, no. 2, pp. 464–471, Apr. 2004.

[33] H. K. Lam and F. H. F. Leung, "Digit and command interpretation for electronic book using neural network and genetic algorithm," *IEEE Trans. on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 34, no. 6, pp. 2273–2283, Dec. 2004.

[34] H. K. Lam and J. Prada, "Interpretation of handwritten single-stroke graffiti using support vector machines," *International Journal of Computational Intelligence and Applications*, vol. 8, no. 04, pp. 369–393, Dec. 2009.

[35] H. Liu, X. Song, J. Bimbo, L. Seneviratne, and K. Althoefer, "Surface material recognition through haptic exploration using an

intelligent contact sensing finger," in *Proceedings of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2012, pp. 52–57.