

Research on Handwritten Note Recognition in Digital Music Classroom Based on Deep Learning

Yanfang Wang

Department of Humanities Management, Ordos Vocational College, China
wyf201096@163.com

Abstract

Music is an indispensable subject in quality education, which plays an important role in improving students' overall quality. Traditional music teaching is mainly a one-way teaching led by teachers. The teaching style is monotonous and the teaching resources are lacking. It is in sharp contrast with diversified music learning, which affects students' mastery of basic music skills. The research of this paper is mainly based on the metaphor of "paper and pen", and the user-centered natural interaction method is used as the design idea to identify the handwritten notes in music teaching in order to provide a natural and efficient teaching environment for music education. Handwritten note recognition draws on the idea of metric learning. Based on the deep Gaussian process model, a non-parametric model, a deep Gaussian matching network for small batch handwritten note recognition is proposed. The framework can adaptively learn a deep structure that can effectively map the labeled support set and unlabeled samples to its label, while avoiding overfitting due to insufficient training data. In the training stage of the deep Gaussian process model, the standardized flow method is used to construct a flexible variational distribution, which improves the quality of inference. Gaussian Processes (GP) are type of supervised learning system that can be used to solve problems like regression and probabilistic classification. Gaussian processes have the following advantages: The forecast generalizes the data from the observations. And when sparse the Gaussian model to reduce the amount of calculation, the optimal k-means method is used to find false points. Experiments were carried out on the handwritten note data set collected in the digital music classroom. The experimental results show that compared with the traditional deep neural network model, the accuracy of the algorithm proposed in this paper has increased from 88% to 94.7% in a single learning sample, and the model proposed in this paper does not rely on fine-tuning and controls the actual calculation amount. The handwritten note recognition effect is better, and it has good application prospects in digital music classrooms.

Keywords: Digital music classroom, Handwritten note recognition, Deep learning, Gaussian process, Non-parametric estimation

1 Introduction

Music is an indispensable subject in quality education. It plays an important role in improving the overall quality of students. It allows learners to not be limited to the accumulation of rational and objective knowledge content, but can expand their perspective to spiritual Feeling and nurturing, so that the individual becomes a person with comprehensive development of knowledge and ideological quality [1]. Traditional music teaching is mainly one-way teaching led by teachers. The teaching style is monotonous and the teaching resources are lacking. It is in sharp contrast with the diversified music learning, which affects students' mastery of basic music skills. However, teaching in a digital environment has: diversity; interactivity; fast and efficient; non-linear structure of information organization. In this kind of open teaching environment with rich teaching resources and forms, teachers can teach easily and students can learn happily, thus ensuring the teaching progress and improving the teaching effect [2]. It can be seen that the introduction of digitization will not only bring material and technical impacts to traditional music teaching, but will also positively promote the innovation of teaching concepts and teaching methods. Breaking the traditional music teaching model, concreteizing and visualizing the "talking on paper" abstract music theory teaching, enriching teaching resources and forms, and conducive to the cultivation of students' music basic technical ability and the improvement of national music basic education [3].

At the same time, the combination of hearing and vision in digital technology can greatly improve learning efficiency. Studies have shown that hearing alone can maintain the acquired knowledge for about three hours. After three days, it will drop to about three hours by vision alone, and keep it for the next three days. If it can be maintained for three hours, it will be

maintained for a total of five days. Increase the depth and breadth of the teaching classroom, reduce the teacher's lesson preparation time, and improve the teacher's work efficiency [4]. At the same time, the intuitive music theory teaching method can easily stimulate students' interest in learning and create an open and creative space for music teaching. The music theory is the study of possibilities and practices of music. The term music defines three interrelated uses of "music theory". The theory of musicology differs from "because output as its starting point, but the basic from which it is produced." Like music theory helps one to consider the significance of a musical composition. Second, the philosophy of music allows one in a shared language to communicate to other musicians. It is an abbreviation for listing significant musical points. The improvement of teaching staff and the limitation of teaching cost provide a way out. The use of various intelligent teaching tools and virtual music teaching tools can reduce the amount of teaching tasks for teachers, help the training of all-round teachers, and make one-to-many. The virtual teaching is nothing but the teachers use digital technology to enhance the teaching process. The virtual teaching tools are MOOCs, content platforms an online courses, and these kinds of tools are used to save organization time and money. The one-to-one effect of teaching can also alleviate the difficulty of being unable to pay due to the expensive musical instruments, and make up for the lack of instrumental music teaching [5]. Through analysis, the use of advanced technology to improve, and enhance the overall effect of music education digitalization will be an inevitable way to popularize music education and improve the music literacy of the whole people.

Humans can often acquire the ability to recognize certain things by learning a small number of labeled samples. Even four or five-year-old children can accurately recognized tigers after seeing a few pictures of tigers. Generally, deep learning systems often require a large amount of labeled training data to have limited recognition capabilities. This significant gap between humans and deep learning systems arouses people's interest in small-sample learning [6]. Small sample learning focuses on how to learn useful information from a few training samples, and its purpose is to learn a classifier that has good generalization ability when there are few training samples. The current mainstream small sample learning methods are roughly divided into three categories: model-based methods [7], metric-based methods [8] and optimization-based methods [9]. The model-based method (MBD) is used to address the problems in design of communication systems, signal processing and dynamic control. It has many uses in motion control, industrial machinery, aerospace, and automotive. The metric-based approach is used to define and predict the total no. of defects through

testing. In

this, the user's needs are fulfilled, and the software specifications are attained by the system. The main aim of the optimization method is used to determine computational efforts optimal solution. It is also used as decision tool to predict best solution for the problem based on the constraints.

The difficulty of small-sample learning is that because there are few training samples, it is difficult to extract enough features from it to meet the needs of the model. Therefore, it is necessary to make full use of the limited labeled samples and use the deep learning model to extract the "deep features". In recent years, many great breakthroughs in deep neural networks (DNN) [10] are quite dependent on the large-scale labeled training set. The small sample area lacks enough samples to update a large number of parameters in the traditional DNN model, so it cannot be trained to the ideal model. At the same time, DNN is extremely dependent on the training process of the network. When the training set is small and the network is deep, it is very easy to overfitting. Moreover, a small number of labeled samples cannot represent the true distribution of the data, resulting in a large variance of the obtained classifier, which leads to a weak generalization ability of the model. The commonly used fine-tuning technique will also produce over-fitting in small sample learning. The common fine-tuning procedure is to truncate the pre-trained network's last layer (SoftMax layer) and substitute it with our own new SoftMax layer that is important to our own problem. To train the network, use a slower learning pace. Also, it is used to train the model for a task and then fine tune the same model to perform another task. The main advantage of fine-tuning technique is to improve the forecasting accuracy.

In order to make full use of a small number of labeled samples, this paper introduces a Gaussian process model with data efficiency, and at the same time gives it a deep structure for extracting deep abstract features of the sample to improve the quality of model inference and learning, from both theoretical and experimental aspects Choose the appropriate model training method. The Gaussian process is a non-linear process used to collect finite random variables with multivariate normal distribution. It is the simplified version of Gaussian probability distribution. For classification and regression, Gaussian Processes may be used as the base for complex non-parametric machine learning algorithms. This paper proposes a small sample learning model based on the deep Gaussian process, using the multi-layer Gaussian process model to fully extract the depth features of the sample, and obtain an improved small sample learning method based on metric, which performs well in the field of small batch handwritten note recognition.

2 Overview of Handwritten Note Recognition in Digital Music Classroom

2.1 Handwritten Notes in Digital Music Classroom

On-line handwritten note recognition has the same principle as on-line handwritten character recognition. However, they also have certain differences in some aspects: (In terms of shape, notes are different from characters. For example, a note can have multiple heads, dots and tails, as shown in the figure. As shown, the number of them is not fixed, and the operator can add or delete them at any time as needed [11]. (In terms of structure, there is generally no obvious connection between handwritten characters, and they exist as an independent individual. The handwritten notes are different. Some notes are interdependent, that is, the formation of a certain note is based on another note. For example, adding a tail to a quarter note can form an eighth A quaver note, adding a tail to an eighth note can form a sixteenth note, etc. The characteristics of the above notes make us need to pay attention to handwritten notes that are more random and varied than other handwriting recognition when conducting recognition research. On-line handwriting recognition is generally divided into two methods: one is based on the entire handwritten character recognition method, and the other is based on the stroke recognition method. Online-recognition is the technique, which automatically converts the text using digitizer. The functions of online handwriting recognition are to convert the text into user write special digitizer. Also, the user inputs are determined and the letter are changed into codes in computer system. Meanwhile, on-line character recognition is used to recognize the number or alphabets. In this article, the second method is used for real-time recognition of strokes [12]. Therefore, stroke recognition is the premise and key of handwritten note recognition in this article. Compared with handwritten character recognition systems, according to the randomness and relevance of handwritten notes, we use single stroke recognition as the basis, and then multiple single strokes Combine to form a complete note.

2.2 The Design Process of Handwritten Notes

The design of handwritten notes is based on the above two forms of handwriting of notes, as well as a survey of professional musicians' handwriting habits, comprehensive analysis, consideration and research have decided to combine the advantages of the two handwritten forms, and the design of handwritten notes is based on the following principles [13]: User-centered, the design of the notes should avoid users' memory of a large amount of it, and reduce the user's cognitive burden; cover a wide range of users, try to

















meet the handwriting habits of most users; the designed notes should be simple to write Convenient, less calculation, and easy to be identified. According to the design principles of the above handwritten notes, most symbols are derived from the handwritten note stroke set. In this handwritten set, the musical symbols are mainly composed of five primitives: point, line, circle, arc, and three-point polyline. Through the combination of these primitives, basic notes (such as basic notes, rests, temporary notes, etc.) can be generated, and further combinations can be used to obtain higher-level symbols (such as harmony, connection notes, etc.). These primitives are all basic geometric shapes and are easy to be classified, and they represent different meanings and varying degrees of change depending on the use situation. For example, a straight line is shorter when used as a talisman and becomes longer when used as a connector. The musical note primitive set and its notes are shown in Table 1.

The primitive set and its notes are represented in Table 1, in that the index value are taken from 0 to 19. Then, for primitive sets the note primitives and its notes are represented.

2.3 Feature Extraction Process of Handwritten Notes

The feature extraction of handwritten notes combines its geometric structure features and statistical features. A geometric structure is the geometric detail that remains same after extracting position, size, direction, and reflection from a geometric object's definition. Polygons are such shapes, and they contain circles, squares, and pentagons. Curves, such as circle or ellipse, may be used to describe other shapes, but the statistical features are the frequently used concept in data science. In image processing, the statistical features are determined from the statistical distribution of pixels. Based on the design of handwritten note strokes, it can be seen that these strokes mainly include three geometric types: straight line segments, polyline segments, and quadratic curves, and they are consistent with the recognition idea of single-stroke gestures. A line segment is a component of a line constrained by two separate end points and includes any point on an endpoint line. It is divided into two different types such as open line segment and closed line segment. In that, closed line segment comprises both end points and the open line segment ignores both end points. A polyline is a line which involves in line segment joining in end with one or more lines. The Polylines are not polygons. The quadratic curves are a U-shaped curve, which is also known as parabola. In general quadratic curve is represented in graphs, with up or down open. Each stroke is completed by one stroke. In this way, the selection of handwritten stroke features has a great influence on the recognition result. Computable stroke features have always been the core of single-stroke gesture recognition. After a large number of experimental

Table 1. Comparison table of note strokes

Index	Note primitives	Notes	Index	Note primitives	Notes
0	•	Rest	10		Note head
1	—	L mark	11		Rise note tail
2		Note stem	12		Note tail
3	/	Rise mark	13		Down note
4		High spectrum	14		Up note
5		Medium spectrum	15		Whole rest
6		Left back	16		Half rest
7		Right back	17		Quarter rest
8		Drop mark	18		Eighth rest
9		Whole note	19		Sixteen rest

verifications, domestic and foreign researchers have obtained many features for different recognition problems, including curvature, density, direction, size, pressure, time and speed Wait for several categories. People can choose to add or remove some of the features according to their specific problem situations to improve the recognition rate [14]. At present, the commonly used single-stroke-based recognition algorithms include a numerical feature with obvious geometric characteristics and a numerical feature related to time. Single-stroke-base recognition algorithm is used to implement appropriate results for user interface prototypes. The Graffiti recognition is one of the basic single-stroke shorthand handwriting recognition system. However, after repeated experimental research, this feature is not completely suitable for the handwritten note strokes designed in this paper, and the recognition rate is relatively low. In order to get a better recognition effect, we selected a feature, which is mainly composed of four categories: curvature, density, size, and direction, and extracted the features of the collected handwritten note handwriting to obtain a feature database set. After analyzing these data Analyze, find a feature with large data value difference, as shown in the table, and set an appropriate threshold for it. After analyzing the experimental feature data, these features are combined with the features of the eight-direction chain code of the strokes to complete the rough classification of linear strokes and nonlinear strokes and the fine classification between strokes. An eight-direction chain used to capture any object shape grid intersection points together. It is used to define eight directions

based on moving perspective in relation.

2.4 The Status of Research on Recognition of Handwritten Content in Deep Learning

The Bayesian non-parametric model represented by Gaussian process [15] has the advantage of data efficiency, and often requires less data to infer the distribution of the fitting function, and this inference generally occurs in testing Phase, significantly reduces the workload of the training phase, and the trained model can still be flexibly adjusted during testing. At the same time, the Gaussian process model has a good ability to measure uncertainty, and Khémiri [16] proved as early as 1998 that a single layer neural network with infinite nodes is equivalent to a Gaussian process with a specific covariance function. In order to make the model have the characteristics of both parametric and non-parametric models [17], Xie [18] proposed the Deep Gaussian Process (DGP) in 2013 by combining deep neural networks and Gaussian processes. The deep Gaussian process is a deep model with a structure similar to DNN, which is good at processing abstract features.

Compared with the standard DNN, DGP has only a small number of kernel parameters and variational parameters, so it can learn complex functions through a small amount of training data. Variational methods are mainly useful when applied to highly coupled systems. These methods achieve system decoupling by introducing additional parameters known as variational parameters, which essentially serve as low-dimensional surrogates for the system's high-dimensional couplings.

But Kernel boot parameters are strings of text that the kernel interprets to modify particular behaviors and allow or disable functions. At the same time, DGP as a non-parametric model, the data generation process does not depend on the huge parameter set of DNN, and different from DNN by introducing nonlinear functions to make DNN have the ability to deal with nonlinear problems, DGP is a combination of random Gaussian functions Automatically obtain the ability to deal with nonlinearities. This kind of processing ability is particularly advantageous when dealing with drastically changing data (the smaller the data set, the more obvious the data fluctuation). At the same time, the architecture of DNN often relies on subjective experience, and it is difficult to arrive at an optimal selection strategy. Data generating process (DGP) is the mixture of random Gaussian functions, which automatically have the ability to deal with non-linear data but in DNN model the non-functions needs to be generated manually.

Because DGP retains many advantages of the single-layer Gaussian model, such as optimizing the edge likelihood function with regard to kernel parameters, it effectively avoids the occurrence of over-fitting, and at the same time, it can adaptively select the network structure. This article starts from another angle and reduces the influence of the regular term by looking for more complex and flexible variational distributions. Introducing the normalizing flow method (Normalizing flow [19]), by repeatedly applying a series of reversible transformations to a simple distribution, a flexible and controllable posterior distribution family is obtained, combined with the variational inference method, and a tractable The variational lower bound of, the kernel parameters and variational parameters are updated by optimizing this lower bound.

3 Handwritten Note Recognition Algorithm in Digital Music Classroom Based on Deep Learning

3.1 Gaussian Process Regression Model

Stochastic processes have been widely used in the field of machine learning. With some observational data, using Bayesian rules to infer the forecast distribution under the framework of a random process can make the model have the ability to use data efficiently, which happens to be lacking in neural networks. The Gaussian process (GP) [20] can be completely determined by a mean function and a covariance function (existence theorem of Gaussian process), where the covariance based function contains a priori assumptions about the function we want to model, such as the smoothness. Suppose $X = \{x_i | i = 1, 2, \dots, N\}$, the target value corresponding to

$x_i \in R^d$ is t_i , and $T = \{t_i | i = 1, 2, \dots, N\}$. Given the set $D = \{(x_i, t_i) | i = 1, 2, \dots, N\}$, for the new data point x' , we hope to find the predicted distribution of its corresponding target value t' . Suppose the objective function is f , and write $F_N = [f_1, \dots, f_N] = [f(x_1), \dots, f(x_N)]$, and $F_{N+1} = [f_1, \dots, f_{N+1}] = [f(x_1), \dots, f(x_{N+1})]$, where $x' = x_{N+1}$. The Gaussian process prior of the objective function f can be implicitly expressed as:

$$p(F_N | X, \Theta) = \frac{1}{(2\pi)^{N/2} |\Sigma|^{1/2}} e^{-\frac{1}{2}(F_N - \mu)^T \Sigma (F_N - \mu)} \quad (1)$$

From the properties of the multivariate joint Gaussian distribution, the predicted distribution can be obtained:

$$\begin{aligned} p(f_{N+1} | D, x_{N+1}, \Theta) &= \frac{p(F_{N+1} | \Theta, x_{N+1}, X)}{p(F_N | \Theta, X)} \\ &= \frac{H_N}{H_{N+1}} e^{-\frac{1}{2}(F_{N+1}^T \Sigma_{N+1}^{-1} F_{N+1} - F_N^T \Sigma_N^{-1} F_N)} \end{aligned} \quad (2)$$

where H_N and H_{N+1} are two normalization constants. The relationship between Σ_N and Σ_{N+1} is:

$$\Sigma_{N+1} = \begin{pmatrix} \Sigma_N & K \\ K^T & K_{x'x'} \end{pmatrix} \quad (3)$$

where $K = [k(x', x_1; \Theta), \dots, k(x', x_N; \Theta)]^T$, $K_{x'x'} = k(x', x'; \Theta)$, K is the kernel function. Finally, the predicted distribution is:

$$p(f_{N+1} | D, x_{N+1}, \Theta) = N(K^T \Sigma^{-1} t, K_{x'x'} - K^T \Sigma^{-1} K) \quad (4)$$

3.2 Deep Gaussian Matching Network Recognition Algorithm

Although DNN has achieved great success in the field of handwriting classification, when the training data is insufficient, due to its massive parameters, DNN is prone to overfitting [21]. Compared with DNN, DGP has a much smaller number of parameters and has Bayesian properties. It is an ideal model for small sample learning. In order to improve the expressive ability of neural network models and introduce deep structure, Hinton et al. proposed a deep neural network. Also, in order to improve the Gaussian model, by combining random processes instead of functions and introducing a deep structure, Diamianou obtains a deep Gaussian process model by stacking several Gaussian process models. The deep Gaussian model is a deep directed graph model that contains multiple hidden layers and uses Gaussian processes to control the mapping relationship between layers. If all the variables are continuous and all of the variables and their parents obey a linear Gaussian model, then the irected graphical model is known as Gaussian directed

graphical model (or Bayesian network). Figure 1 shows the handwritten note recognition deep convolutional neural network architecture used in this digital music classroom.

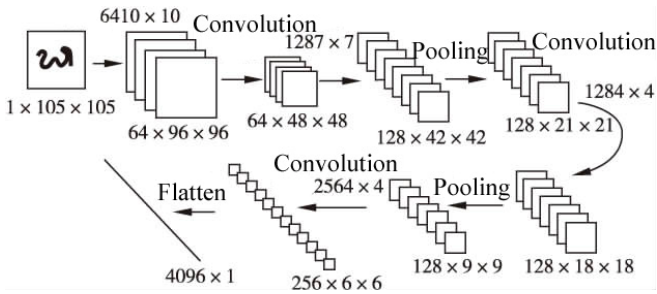


Figure 1. Handwritten note recognition deep convolutional neural network architecture used in the digital music classroom in this article

Convolutional Neural Networks is the different kind of multi-layer neural networks, which is used to recognise visual patterns take image pixels directly with minimal pre-processing. In this proposed architecture, the input image is taken as 1*105*105 then the image feature is extracted and converted into multiple layer of 64*96*96 pixels, without damaging the image in convolution layer, then the extracted image is reduced into appropriate size without pixel loss of about 64*48*48. Further, the extracted image is processed in the pooling layer. The pooling layer is considered as the building block of CNN, which is used to minimize the spatial size and parameter in the network. But in image, it is used to reduce the dimension. Again, the input set are processed in pooling and convolution layer to flatten the image. The parametric equations are aligned properly. In statics and probability, the LOTUS (law of the unconscious satisfaction) satisfaction theorem determines the random variable X for a function $g(X)$ as expected value, but when the probability distribution of X is known but the distribution of $g(X)$ is unknown. Compared with the pre-given non-linear function in the standard parameterized deep neural network, the mapping relationship between the deep Gaussian process layer and the layer has stronger expressive ability and data dependent. And compared with DNN, DGP model has fewer parameters (only a few nuclear hyperparameters and variational parameters). And as a Bayesian model, DGP can effectively avoid the overfitting phenomenon in DNN when the data is insufficient. The random nature inherited from the Gaussian model enables DGP to deal with the problem of data uncertainty well. Denote the input of a DGP model with L hidden layers as x and the output as y . The model definition is as follows. The input variables are often regarded as hidden variables, denoted as input $x = h^0$, output $y = h^L$, and the noise of each layer is $\varepsilon_i \sim N(0, \sigma_i^2 I)$:

$$f^l \sim GP(\mu^l, k(\cdot)^l), h^l = f^l(h^0) + \varepsilon_1, u^l = f^l(z^0) \quad (5)$$

$$f^l \sim GP(\mu^l, k(\cdot)^l), h^l = f^l(h^{l-1}) + \varepsilon_l, u^l = f^l(z^{l-1}) \quad (6)$$

$$f^l \sim GP(\mu^L, k(\cdot)^L), h^L = f^L(h^{L-1}) + \varepsilon_L, u^L = f^L(z^{L-1}) \quad (7)$$

Sometimes in order to reduce the number of variational parameters, the Gaussian noise term is often placed in the kernel function. It should be noted that each Gaussian function has its own corresponding kernel function and kernel parameters, so there may be multiple kernels in a layer. At this time, the joint distribution of the model is:

$$P(y, \{h^l, u^l\}_{l=1}^{L-1}, x) = P(y | h^{L-1}) \left\{ \prod_{l=1}^{L-1} P(h^l | u^l; h^{l-1}, z^{l-1}) P(u^l, z^{l-1}) \right\} P(h^0) \quad (8)$$

By integrating all the hidden variables, the marginal likelihood function as the objective function of the model can be obtained. A marginal likelihood function, also known as integrated likelihood in mathematics, which is a likelihood function with certain parameter variables marginalized. For simplicity, the kernel parameter σ is omitted from all probability distributions. But what is obtained is still a complex integral, which makes inference difficult, so approximate methods are needed, such as variational inference [22] and expectation propagation [23].

Cosine distance is often also called cosine similarity, which uses the cosine of the angle between two vectors as a measure of the difference between two vectors. When $X = (x_1, x_2, \dots, x_n)$, $Y = (y_1, y_2, \dots, y_n)$, the cosine similarity of X and Y is:

$$\text{Similarity}(X, Y) = \frac{x_1 y_1 + x_2 y_2 + \dots + x_n y_n}{\sqrt{x_1^2 + x_2^2 + \dots + x_n^2} \sqrt{y_1^2 + y_2^2 + \dots + y_n^2}} \quad (9)$$

Cosine similarity is often not sensitive to absolute values, but pays attention to the difference in two vector directions, that is, cosine similarity is more of a relative difference between the two. The cosine similarity is the metric which used to evaluate the dissimilarity between same documents with different size. It also measures the cosine similarity angle between n-dimensional space and n-dimensional vectors. The cosine similarity is a vector similarity metric. The idea is that if two vectors are exactly identical, then the similarity becomes 1 (angle=0), and distance becomes 0(1-1=0). Similarity, the consine distance for the resultsing similarity value spectrum is defined. In order to make full use of the data, a support set $S = \{x_i, y_i\}_{i=1}^K$ is introduced, where x_i is the sample, y_i is the label, and S contains a small number of

images that are not in the same category, at least one of which is the same as the image to be recognized Be in the same category. In the training process, for the sample x^* to be tested, if its category is $l_s(x^*)$, define a mapping: $S \rightarrow l_s(x^*)$, expressed in the form of conditional probability: $P(y^*|x^*, S)$, this mapping needs to be learned. Therefore, in the process of testing, for a given new support set, the trained model can be used to predict the label corresponding to the test sample. Figure 2 shows the deep Gaussian matching network architecture for handwritten note recognition. Deep Gaussian matching network architecture for handwritten note recognition shown in Figure 2, in that first the convolutional neural network is used to extract the features form the structure for support set and test set for images, with image input set of 105×105 . Further, two DGP functions are taken as coding function (φ and ψ) parameters. Then, to complete the image classification, measure the cosine distance between the test image feature and the support set image feature.

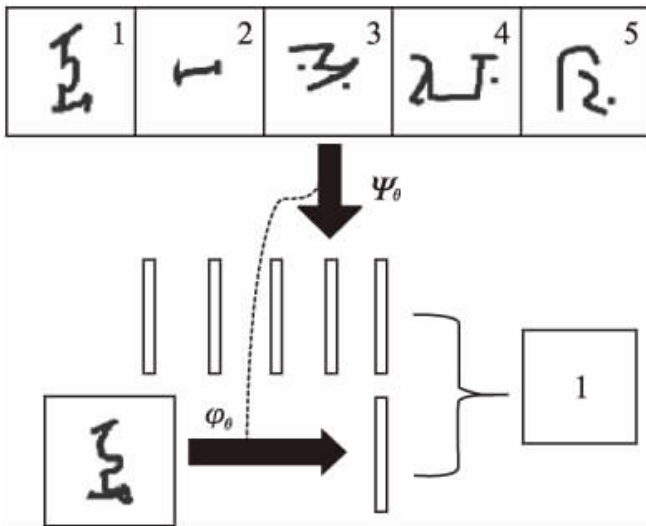


Figure 2. Deep Gaussian matching network architecture for handwritten note recognition

Step 1: Use the convolutional neural network with the following structure to extract the abstract features of the test set and support set images, and flatten the 105×105 image into a vector of length 4096.

Step 2: Two DGP structures are used as coding functions, and the resulting feature vectors are embedded in the feature space. Then calculate the cosine distance between the test image feature and the support set image feature to complete the image classification, as follows.

In order to obtain an end-to-end handwritten note recognition model similar to KNN, similar to the literature [5], the attention core is introduced, based on the idea of k-nearest neighbor method, there are:

$$y^* = \sum_{i=1}^K a(x^*, x_i) y_i \tag{10}$$

where y_i is a vector with only one dimension of 1, and the rest are all zeros. The attention core is:

$$a(x^*, x_i) = \frac{e^{c(\varphi(x^*), \psi(x_i))}}{\sum_{j=1}^K e^{c(\varphi(x^*), \psi(x_j))}} \tag{11}$$

where φ and ψ are coding functions parameterized by the deep Gaussian process model, and their parameters can be adjusted by the accuracy of the training set classification. The function of φ and ψ is to extract the abstract features of the image and embed and into the feature space.

In order to ensure that the training process and the test process occur under the same conditions, first sample a small data set W in the original data set D . For each category in W , randomly select K samples to construct the support set S , and randomly select some samples are used as the training set T . Due to the complex correlation between layers, this paper introduces dummy data and uses a sparse method to simplify the correlation between the levels of the DGP model. Next, the stochastic gradient descent method is used to update the parameters of the parameter embedding function and the position of the dummy data by maximizing the recognition rate, using the log function as an increasing function, and the resulting objective function is:

$$E_{W \sim D} [E_{S, T \sim W} [\sum_{(x, y) \in T} \log P(y | x, S, \Theta)]] \tag{12}$$

The parameters that the model needs to learn are pseudo points and the core parameters of each node in each layer:

$$\Theta = \{ \{ u_d^l, z_d^l, \theta_d^l \}_{d=1}^{D_l} \}_{l=1}^L \tag{13}$$

where L is the number of layers in the model, and D_l is the number of nodes in the l layer. If both pseudo-output and pseudo-input are learned, since more parameters will increase the since more parameters will increase the probability of overfitting, the model will lose part of the standardization advantage of the Bayesian model. For the sake of simplicity, first consider the case of a single layer. Errors can be propagated from layer to layer through variational inference. The inference method between the layers is as follows. The input x is often regarded as a hidden variable, denoting $f = f(x)$. For the sake of brevity, the pseudo data is omitted. At this time, the edge likelihood function of the model is:

$$p(y) = \int p(y, x, f) dx df \tag{14}$$

Variational inference is to find an approximate posterior distribution close to the true posterior distribution based on the KL divergence between two distributions. The approximate distribution is often

limited to some easy-to-handle distribution families, such as the Gaussian distribution. In this way, the complex inference problem is turned into an optimization problem of minimizing the KL divergence between distributions. The same objective function can be obtained by using Jensen’s inequality in the marginal likelihood function. At the same time, it is assumed that each approximate distribution satisfies the mean field hypothesis, that is,. At this time, the true edge likelihood function of the model is:

$$p(y) = E_{x,f}[\log[p(y|x, f) \frac{p(f)p(x)}{q(f)q(x)}]] \quad (15)$$

The expectation in equation (15) is the expectation about the variational distribution, and both sides take the logarithm at the same time. According to Jensen’s inequality, there are:

$$\log p(y) \geq E_{x,f}[\log[p(y|x, f) \frac{p(f)p(x)}{q(f)q(x)}]] \quad (16)$$

In general, in order to simplify the calculation, some simple distributions are used to approximate the true posterior distribution, and it is often assumed that the approximate distribution satisfies the simple structural characteristics similar to the mean field. This greatly affects the inference effect of the variational method, and often leads to underestimation of the prediction distribution variance, making the decision unreliable. At the same time, the limited capacity of the approximate posterior distribution family will lead to the degradation of the MAP estimation of the model parameters. Into a biased estimate. In order to solve this problem, this paper selects the approximate distribution from the flexible and arbitrarily complex distribution family constructed by the standardized flow method, so that the approximate distribution is more likely to be close to the true posterior distribution.

3.3 DGPMN Model Training Method Based on Standardized Flow

It can be seen from the lower bound that we hope that the variational distribution can be as close as possible to the posterior distribution of real z, but if it is only based on the prior assumption that the variational distribution is a mutually independent Gaussian distribution or other mean field assumptions, this is very Difficult to achieve [24]. In fact, this is the biggest limitation of the variational method. The approximate distribution family is not flexible enough. Even in an approximate area, a good approximation of the true posterior cannot be obtained. A truly ideal family of variational distributions should be very flexible and contain true posterior. The standardized flow is to transform a simple density function through a series of reversible transformations, and finally get a more complex probability distribution with stronger expressive power. If we assume that the distribution

function of the random variable f is $p(f)$, $f' = g(f)$, using the inverse function theorem and the chain rule, we get the density distribution function of $p(f')$:

$$p(f') = p(f) \left| \det \frac{\partial g^{-1}}{\partial f'} \right| = p(f) \det \left| \frac{\partial g}{\partial f} \right|^{-1} \quad (17)$$

Through the stacking of these simple rever-sible functions, we can construct arbitrarily complex density functions:

$$f_M = g_M \dots g_2 g_1(f_0) \quad (18)$$

$$\ln p_M(f_M) = \ln p_0(f_0) - \sum_{m=1}^M \ln \left| \det \frac{\partial g_k}{\partial f_{k-1}} \right| \quad (19)$$

The reason for this is that a more complex distribution can be obtained by nonlinearly transforming a simple distribution. As the statistician LOTUS once proposed, when looking for expectations about the transformed distribution, it is not necessary to know the specific form of the transformed distribution. This expectation can be obtained only by the original simple distribution and transformation function, namely:

$$E_{p_M(f_M)}[h(f_M)] = E_{p_0(f_0)}[h(g_M g_{M-1} \dots g_1(f_0))] \quad (20)$$

And if h and p_M have nothing to do, there is no need to calculate the Jacobian. Consider the plane flow $q(f) = f + v \bullet \rho(w^T f + b)$, where the parameter $\Omega = \{w, v, b\}$. If $\zeta(f) = \rho'(w^T f + b)$, then the Jacobian determinant:

$$\left| \det \frac{\partial g}{\partial f} \right| = \left| \det(I + v\zeta(f)^T) \right| = \left| 1 + v\zeta(f)^T \right| \quad (21)$$

Then the variational distribution obtained by the initial density function $q_0(f_0)$ through a series of reversible transformations can be expressed as:

$$\ln p_M(f_M) = \ln p_0(f_0) - \sum_{m=1}^M \ln \left| 1 + u_m^T \zeta_m(f_{m-1}) \right| \quad (22)$$

At this time, we use a stream of length M to parameterize the approximate posterior distribution, that is, set $q(x) \triangleq q_M(x_M)$, $q(f) \triangleq q_M(f_M)$, and the variational lower bound of equation (15) can be Written as:

$$\Gamma = E_{q_0(x_0)q_0(f_0)}[\log p(y, x_M, f_M) - \log q_0(x_0) - \log q_0(f_0)] + E_{q_0(x_0)q_0(f_0)} \left[\sum_{m=1}^M \ln \left| 1 + u_m^T \zeta_m(x_{m-1}) \right| + \sum_{m=1}^M \ln \left| 1 + u_m^T \zeta_m(f_{m-1}) \right| \right] \quad (23)$$

Each layer is inferred in a similar way. The process of the recognition algorithm proposed in this paper is

as follows, and the effectiveness of the algorithm is verified on the real data set in the next section. Regarding the noise term in the model, we found that when optimizing the objective function given by Khamparia [25], due to strong model assumptions, the model is prone to noise degradation. That is, the signal-to-noise ratio tends to zero. In order to alleviate this problem, we use aggregate noise to replace simple additive noise $[h, \varepsilon]$, which is also regarded as a hidden variable.

4 Simulation Experiment and Result Analysis

4.1 Analysis of Handwritten Note Recognition Results in Digital Music Classroom

In order to verify the effect of the single-stroke recognition algorithm for notes studied in this article, first, the data and environment for the experiment are

prepared. The stroke recognition method is used to determine the strokes in handwritten note. The detected strokes are grouped together with the help of spectral clustering algorithm in order to find the similarity between features and to reduce the dimension of feature for grouping. The stroke determines the arrows, symbols and lines from the text. We collected a total of 10,000 handwritten strokes from 25 people, using python 3.6 as the experimental platform and programming language to implement the algorithm. Secondly, 34 standard eight-direction chain code templates for dynamic matching have been prepared. Finally, the recognition effect of this algorithm is compared with the RUBINE algorithm [26], which is commonly used for single-stroke recognition algorithm, to reflect the usability of the recognition algorithm in this article. The recognition results and total recognition results of each type of stroke are shown in Table 2.

Table 2. Recognition results and total recognition results of each type of stroke of handwritten notes

Note strokes	Mistaken recognized note strokes (%)	Note strokes	Mistaken recognized note strokes (%)
3 /	1 — (2.35)	13.1	10
5	11.1 (0.32), 16 (0.43)	13.2	5 (0.15)
6	14 (0.88), 15 (0.42)	14	6 (6.35), 15 (0.36)
7	5 (4.58)	15	14 (0.36), 8 (0.17)
8	14 (1.36)	16	5 (0.43), 13.2 (1.87)
9	10 (0.29)	17	11.1 (1.53), 13.1 (1.05)
10	9 (0.29)	18	5 (0.16)
12	3 / (13.56), 5 (2.87)	19	1 — (0.29)

According to the results of the above data, it can be seen that the algorithm in this article has obtained a relatively satisfactory recognition effect. The recognition rate of most strokes is above. Only a few strokes have a low recognition rate, mainly composite stroke recognition effects. It is not particularly ideal, but this situation will not have too much impact on writing. In general, the recognition effect is better than the algorithm specifically used for single stroke recognition. Especially for solid symbol heads that are not easy to recognize, the recognition rate has increased from 94.12% in the algorithm to 99.71%,

which is a significant improvement. Furthermore, from the recognition data, we can see that the recognition algorithm in this article is effective. The recognition effect of linear strokes, such as points, straight lines, and diagonal lines, almost reaches a recognition rate of 100%, which shows that the selection of geometric features and the setting of thresholds are reasonable.

However, due to the existence of the similarity problem between strokes, some strokes may be misidentified as other types of strokes. The misrecognized strokes and their percentages are given in the table. Here, we use the following formula to

determine each type of strokes. The percentage of misrecognition is calculated: among them, it means that the judgment is correct, and the category belonging to the object classification is marked as the correct classification of the object; it is negative misrecognition, and the category belonging to the object classification is marked as the misclassification that does not belong to the object; In the table above, we can see that 13.56% of ㄥ were misidentified as ㄥ and 5.35% of ㄥ were misidentified as ㄥ. Obviously, they are indistinguishable in appearance. The larger the percentage is, the greater the similarity is between them. However, for the misrecognition problem caused by the similarity between these strokes, we can adjust the stroke combination method and complete the recognition correctly.

4.2 Performance Analysis of Handwritten Character Recognition in Digital Music Classroom

In order to illustrate the performance of this model in the classification of small samples of handwritten notes, we collected 623 handwritten notes, each of which was handwritten by 20 different people. The experimental platform of this paper is python3.6. The Gaussian process model is implemented on the GPflow

platform, and the parameters are optimized using the automatic derivative function of tensorflow. The GPflow is the open source software using tensor flow in python programming language. The GPflow is the package used to construct Gaussian process models. It uses composable kernels and likelihoods to introduce modern Gaussian method inference. The experimental environment is a Core i5 processor, the frequency is 2.6GHz, and the RAM is 8G. First, the performance of the training algorithm in this paper and the traditional training algorithm is compared through the error curve. Figure 3 shows the error convergence graph of the training set and test set (left: DGPMN, right: DGP). Based on the error curve of perplexity, we found that compared with the traditional DGP model, the DGPMN model proposed in this paper has a faster convergence speed and lower error. MNIST is a commonly used handwritten digit data set. Figure 4 shows the image of the optimized binary hidden space projected from the MNIST data set. Different types of data points are represented by different colors, and the different types of data points are more differentiated. It means that the optimization effect of the hidden space is better. The results show that the optimization effect of the training algorithm in this paper is better than that of the traditional DGP.

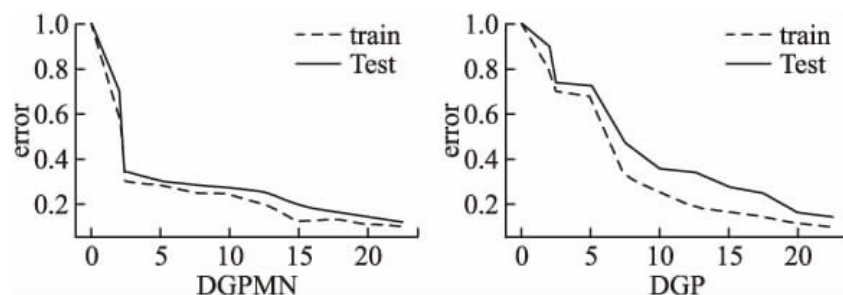


Figure 3. Error convergence graph of training set and test set (left: DGPMN, right: DGP)

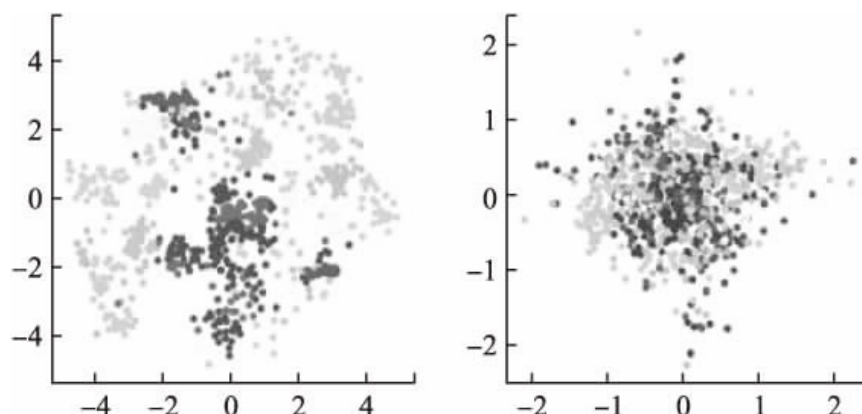


Figure 4. Projection of binary hidden space

In order to illustrate the effect of the model in this paper, the method based on pixel matching (pixels) and the classical convolutional twin network (CNN-S) network and the model in this paper are used for multiple comparison experiments. The performance

test uses the handwritten note data set collected in this paper. We select 30 regions of the alphabet as the training set, among which there are 964 classes. The alphabets of the remaining 20 regions are used as the test set, of which there are 659 categories. This means

that the model encountered during the test is never seen before. In the training process, 20 classes are randomly selected from 964 classes in each iteration, and 5 of them are selected as the support set. During the test, M classes are randomly selected from 659 classes, and each class provides K samples to generalize the model, which is the so-called M-Way, K-shot learning task [27]. The probability of randomly guessing the correct

result is 1/M. Table 3 shows the performance of this model on the collected handwritten note data set. From the experimental results in Table 3, it can be seen that the model in this paper improves the prediction accuracy from 88% to 94.7% compared to the traditional CNN-S network in the 20-channel single-sample learning task we are more concerned about.

Table 3. The performance of this model on the collected handwritten note data set

Models	5-way		15-way		20-way	
	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot
PIXELS	41.65%	62.85%	37.65%	51.46%	25.31%	42.68%
CNN-S	65.87%	96.59%	89.96%	94.87%	87.68%	93.65%
DGPMN	97.63%	98.67%	93.56%	95.87%	93.68%	94.87%

In small sample learning, due to the small training set, at this time, the result of fine-tuning the network will be much better than retraining the network. Therefore, we conducted a fine-tuning experiment. Figure 5 shows the effect of fine-tuning on the model’s prediction of Omniglot’s effect. From the results in Figure 5, we find that the model in this paper does not depend on the fine-tuning operation of the network, which can effectively avoid the over-fitting phenomenon caused by the fine-tuning. The experimental results are shown in Table 3, where N means no fine-tuning is used. Y means fine tuning is used.

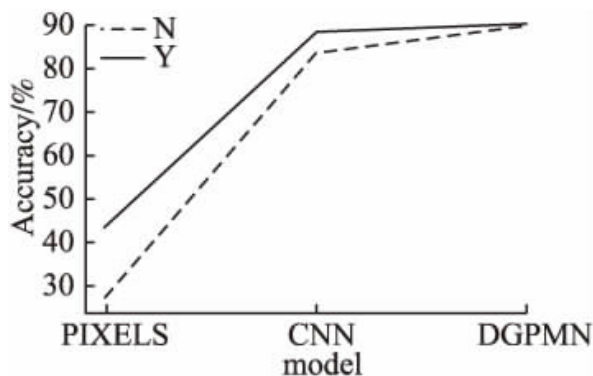


Figure 5. The effect of fine-tuning on the model’s prediction of Omniglot’s effect

The handwritten note image is relatively simple. In order to further illustrate the effect of this model, we conduct experiments on other public MiniImage datasets [27]. The MiniImage data set is extracted from the ImageNet data set for small sample learning problems, which is more complicated than the commonly used CIFAR10 data set. The MiniImage data set contains 60,000 84×84 color images in 100 categories, each with 600 samples. We use 80 categories as the training set, and the other 20 categories as the test set, and conduct comparative experiments. The results are shown in Table 4.

Table 4. The performance of this model on the MiniImage dataset

Comparative models		PIXELS	CNN-S	DGPMN
5-way	1-shot	22.46%	44.62%	45.89%
	5-shot	26.53%	48.19%	57.68%

The handwritten image notes are analysed by using public miniImage dataset. The miniImageNet dataset comprises 100 groups, each with 600 images of size 8484 pixels, selected at random from the ImageNet ILSVRC-2012 challenge. There are 64 base classes, 16 validation classes, and 20 novel classes in all. The miniImage dataset is extracted from the imageNet data set. The proposed algorithm achieves better results than the existing algorithm.

5 Conclusion

This paper constructs a flexible variational distribution based on the standardized flow, and uses the optimal K-means clustering method to select pseudo points, which improves the training effect of the deep Gaussian process model. Compared with the shallower model, the deeper model must have stronger predictive ability once trained, so it is very meaningful to explore more feasible training methods of DGP model. Although the recognition algorithm in this article has obtained a relatively ideal recognition effect, it also has many shortcomings, and some issues need to be comprehensively considered to be further improved. The following is a further outlook for future research work: In this article, only the experimental analysis of the recognition rate of the recognition algorithm is carried out, but the quality of an algorithm cannot only consider its output results, and the time complexity of the algorithm is also an important indicator that needs to be considered. one. Next, I need to verify the reaction time of the recognition process, including the recognition time of a single stroke, the combination time of strokes, and the output time of regular notes.

Since there are many positional relations between the note point and the note head, different positional relations represent different meanings. In order to better distinguish different situations, we need to further determine the position between the note point and the note head. Relations make corresponding rules. The relationship between one talisman stem and multiple talisman heads in harmony needs to be further designed and perfected, and try to avoid the problem of combination errors due to the position of strokes in the process of combining notes. In short, whether it is the algorithm or the implementation of the final score editor, there are many things to consider. In the future research work, we will try our best to meet the needs of all users and make the recognition research of handwritten notes more standard, more accurate, more natural and harmonious.

References

- [1] A. El-Sawy, M. Loey, H. El-Bakry, Arabic handwritten characters recognition using convolutional neural network, *WSEAS Transactions on Computer Research*, Vol. 5, pp. 11-19, 2017.
- [2] C. Boufenar, A. Kerboua, M. Batouche, Investigation on deep learning for off-line handwritten Arabic character recognition, *Cognitive Systems Research*, Vol. 50, pp. 180-195, August, 2018.
- [3] K. Peymani, M. Soryani, From machine generated to handwritten character recognition; a deep learning approach, *2017 3rd International Conference on Pattern Recognition and Image Analysis (IPRIA)*, Shahrekord, Iran, 2017, pp. 243-247.
- [4] B. Balci, D. Saadati, D. Shiferaw, *Handwritten text recognition using deep learning*, CS231n: Convolutional Neural Networks for Visual Recognition, Stanford University, Course Project Report, Spring, 2017.
- [5] W. Wang, J. Zhang, J. Du, Z.-R. Wang, Y. Zhu, DenseRAN for offline handwritten Chinese character recognition, *2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, Niagara Falls, NY, USA, 2018, pp. 104-109.
- [6] X. Y. Zhang, Y. Bengio, C. L. Liu, Online and offline handwritten chinese character recognition: A comprehensive study and new benchmark, *Pattern Recognition*, Vol. 61, pp. 348-360, January, 2017.
- [7] S. Kundu, S. Paul, P. K. Singh, R. Sarkar, M. Nasipuri, Understanding NFC-Net: a deep learning approach to word-level handwritten Indic script recognition, *Neural Computing and Applications*, Vol. 32, No. 12, pp. 7879-7895, June, 2020.
- [8] C. Neche, A. Belaid, A. Kacem-Echi, Arabic handwritten documents segmentation into text-lines and words using deep learning, *2019 International Conference on Document Analysis and Recognition Workshops (ICDARW)*, Sydney, NSW, Australia, 2019, pp. 6: 19-24.
- [9] A. Baró, P. Riba, J. Calvo-Zaragoza, A. Fornes, From optical music recognition to handwritten music recognition: A baseline, *Pattern Recognition Letters*, Vol. 123, pp. 1-8, May, 2019.
- [10] S. R. Kulkarni, B. Rajendran, Spiking neural networks for handwritten digit recognition—Supervised learning and network optimization, *Neural Networks*, Vol. 103, pp. 118-127, July, 2018.
- [11] G. M. De Buy Wenniger, L. Schomaker, A. Way, No padding please: Efficient neural handwriting recognition, *2019 International Conference on Document Analysis and Recognition (ICDAR)*, Sydney, NSW, Australia, 2019, pp. 355-362.
- [12] H. Ding, K. Chen, Y. Yuan, M. Cai, L. Sun, S. Liang, Q. Huo, A compact CNN-DBLSTM based character model for offline handwriting recognition with Tucker decomposition, *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, Kyoto, Japan, 2017, pp. 1: 507-512.
- [13] Y. Liu, L. Jin, S. Lai, Automatic labeling of large amounts of handwritten characters with gate-guided dynamic deep learning, *Pattern Recognition Letters*, Vol. 119, pp. 94-102, March, 2019.
- [14] A. Pacha, K. Y. Choi, B. Coüasnon, Y. Ricquebourg, R. Zanibbi, H. Eidenberger, Handwritten music object detection: Open issues and baseline results, *2018 13th IAPR International Workshop on Document Analysis Systems (DAS)*, Vienna, Austria, 2018, pp. 163-168.
- [15] Y. Yang, K. Liang, X. Xiao, Z. Xie, L. Jin, J. Sun, W. Zhou, Accelerating and compressing LSTM based model for online handwritten Chinese character recognition, *2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, Niagara Falls, NY, USA, 2018, pp. 110-115.
- [16] A. Khémiri, A. K. Echi, M. Elloumi, Bayesian versus convolutional networks for Arabic handwriting recognition, *Arabian Journal for Science and Engineering*, Vol. 44, No. 11, pp. 9301-9319, November, 2019.
- [17] H. Kusetogullari, A. Yavariabdi, A. Cheddad, H. Grahm, J. Hall, ARDIS: a Swedish historical handwritten digit dataset, *Neural Computing and Applications*, Vol. 32, No. 21, pp. 16505-16518, November, 2020.
- [18] Z. Xie, Z. Sun, L. Jin, H. Ni, T. Lyons, Learning spatial-semantic context with fully convolutional recurrent network for online handwritten Chinese text recognition, *IEEE transactions on pattern analysis and machine intelligence*, Vol. 40, No. 8, pp. 1903-1917, August, 2018.
- [19] R. Tolosana, R. Vera-Rodriguez, J. Fierrez, A. Morales, J. Ortega-Garcia, Do you need more data? the DeepSignDB online handwritten signature biometric database, *2019 International Conference on Document Analysis and Recognition (ICDAR)*, Sydney, NSW, Australia, 2019, pp. 1143-1148.
- [20] H. Ali, A. Ullah, T. Iqbal, S. Khattak, Pioneer dataset and automatic recognition of Urdu handwritten characters using a deep autoencoder and convolutional neural network, *SN Applied Sciences*, Vol. 2, No. 2, Article No. 152, February,

- 2020.
- [21] W. Luo, S. Kamata, Radical region based CNN for offline handwritten Chinese character recognition, *2017 4th IAPR Asian Conference on Pattern Recognition (ACPR)*, Nanjing, China, 2017, pp. 542-547.
 - [22] P. Melnyk, Z. You, K. Li, A high-performance CNN method for offline handwritten Chinese character recognition and visualization, *Soft Computing*, Vol. 24, No. 11, pp. 7977-7987, June, 2020.
 - [23] S. He, L. Schomaker, Deep adaptive learning for writer identification based on single handwritten word images, *Pattern Recognition*, Vol. 88, pp. 64-74, April, 2019.
 - [24] H. Ding, K. Chen, W. Hu, M. Cai, Q. Huo, Building compact cnn-dblstm based character models for handwriting recognition and ocr by teacher-student learning, *2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, Niagara Falls, NY, USA, 2018, pp. 139-144.
 - [25] A. Khamparia, K. M. Singh, A systematic review on deep learning architectures and applications, *Expert Systems*, Vol. 36, No. 3, Article No. e12400, June, 2019.
 - [26] A. Kölsch, A. Mishra, S. Varshneya, M. Z. Afzal, M. Liwicki, Recognizing challenging handwritten annotations with fully convolutional networks, *2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, 2018, Niagara Falls, NY, USA, pp. 25-31.
 - [27] A. Hast, M. Lind, E. Vats, Embedded prototype subspace classification: A subspace learning framework, *International Conference on Computer Analysis of Images and Patterns*, Salerno, Italy, 2019, pp. 581-592.

Biography



Yanfang Wang, graduated from Inner Mongolia Normal University in June 1999, majoring in music education. A monograph was published; One textbook has been used by more than 50 colleges and universities in China; 2 invention and utility model patents; He has published more than 20 papers in music creation, film review, China Radio and television journal and other journals.

