AN OPEN-SOURCE LIBRARY OF 2D-GMM-HMM BASED ON KALDI TOOLKIT AND ITS APPLICATION TO HANDWRITTEN CHINESE CHARACTER RECOGNITION

Jie-Feng Ma[1], *Zi-Rui Wang*[2], *Jun Du*[1]

[1]University of Science and Technology of China, Hefei, Anhui, P.R.China
[2]Chongqing University of Posts and Telecommunications, Chongqing, P.R.China

ABSTRACT

As an open source toolkit based on 1D-HMM framework, Kaldi toolkit is widely used in many signal processing tasks. However, when dealing with complex spatial structures, e.g. in image related tasks, 2D-HMM is more suitable since it allows free transition between hidden states in both horizontal and vertical directions. Although 2D-HMM framework has been proposed for years, there is still a lack of efficient open source toolkit for further research due to its complexity. In this paper we present a highly efficient code library of 2D-GMM-HMM based on Kaldi toolkit with implementation details. As a demonstration of its effectiveness, we apply 2D-GMM-HMM to handwritten Chinese character recognition (HCCR) task. The experiments on a 50-class HCCR task have proved that the 2D-GMM-HMM system has obvious advantages over the 1D-GMM-HMM system in terms of recognition accuracy and modeling precision. Moreover, the visual analysis shows that 2D-GMM-HMM can well segment the Chinese characters into basic components such as radicals via the hidden states in both horizontal and vertical directions while 1D-GMM-HMM can only conduct the segmentation in the horizontal direction. The project code of 2D-GMM-HMM library and its recipe on HCCR is publicly available in https://github.com/jfma-USTC/2DHMM.

***Index Terms*—** Hidden Markov model, open source library, Kaldi toolkit, handwritten Chinese character recognition

1. INTRODUCTION

As a generalization of Markov process, hidden Markov model (HMM) was first proposed in 1960s [1]. Because of its intuitive description of sequential modeling in time domain, HMM has achieved great success in many fields, e.g., automatic speech recognition (ASR) task [2], computational biology [3], protein structure prediction [4].

However, the conventional one-dimensional HMM (1D-HMM) for sequential modeling can not well handle the problem of modeling complex spatial structures, e.g., optical character recognition and semantic segmentation in the field of computer vision. In order to solve the problem of dimension mismatch, many scholars have tried to extend one-dimensional HMM to two-dimensional HMM (2D-HMM) since the 1990s. Some researchers proposed pseudo two-dimensional hidden Markov model (P2DHMM) [5] to simulate two-dimensional situation by adding super states, which is essentially the same as one-dimensional HMM with the constraints on the hidden state transition matrix. Nefian et al. took this state structure one step further in [6] by introducing the Embedded HMM (EHMM). Although the image in EHMM is scanned in a 2D manner where each observation block retains two indices, the vertical transitions among states in different superstates are still missing. The first 2D-HMM framework based on Markov random field [7] was proposed in 1998. Under the third-order Markov hypothesis, it introduced a complete formula derivation for training and testing. However, due to the complexity of the model, only the experiment on a small-scale handwritten digit database was conducted. Then, a second-order 2D-HMM framework [8] was presented in 2003, which achieved a good balance between model complexity and system performance. After that, the same framework was applied to several tasks including image classification [9] and image segmentation [10].

Unfortunately, previous works on 2D-HMM did not provide runnable source code, and no convincing visualization results of segmentation via the hidden states were given for analysing why 2D-HMM could outperform 1D-HMM by a large margin in image-related tasks. In this study, we propose an open source library of 2D-HMM with Gaussian mixture model as the output distribution (2D-GMM-HMM) based on Kaldi [11] which is a toolkit of 1D-HMM widely used for sequential modeling tasks such as speech recognition [12] and speech synthesis [13]. As a demonstration of its effectiveness, we apply 2D-GMM-HMM to handwritten Chinese character recognition (HCCR) task. The experiments on 50-class HCCR task show that the 2D-GMM-HMM system has obvious advantages over the 1D-GMM-HMM system in terms of recognition accuracy and modeling precision. Moreover, the visual analysis shows that 2D-GMM-HMM can well segment the Chinese characters into basic components such as radicals via the hidden states in both horizontal and vertical directions while 1D-GMM-HMM can only conduct the segmentation in the horizontal direction.

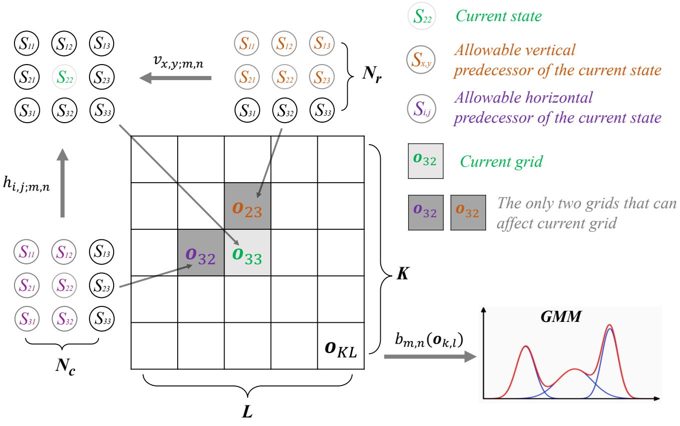
S_{22} Current state
$S_{x,y}$ Allowable vertical predecessor of the current state
$S_{i,j}$ Allowable horizontal predecessor of the current state
o_{32} Current grid
o_{32} o_{32} The only two grids that can affect current grid
$v_{x,y;m,n}$
$h_{i,j;m,n}$
N_r
N_c
K
L
o_{23}
o_{32}
o_{33}
o_{KL}
$b_{m,n}(o_{k,l})$
GMM

Fig. 1. 2D-GMM-HMM system.

2. IMPLEMENT OF 2D-HMM BASED ON KALDI

2.1. Mathematical formulation

As shown in Fig.1, an image is cut into grids of K rows and L columns, and the hidden state set of N_r rows and N_c columns is used to describe the hidden state distribution of 2D-HMM. The hidden state set can be formulated as $\mathbf{S} = \{s_{m,n}\}, 1 \leq m \leq N_r, 1 \leq n \leq N_c$, which contains all possible hidden states that may appear in a 2D-HMM system. And the observation set can be formulated as $\mathbf{O} = \{\boldsymbol{o}_{k,l}\}, 1 \leq k \leq K, 1 \leq l \leq L$ with each grid corresponding to a hidden variable $\mathbf{Q} = \{q_{k,l}\}, 1 \leq k \leq K, 1 \leq l \leq L$.

In addition, a second-order Markov hypothesis is applied to reduce the computational complexity, which means the hidden state corresponding to the current grid is only affected by the hidden state of the upper grid and the left grid. Under this assumption, the state transition can be separated into vertical state transitions $\mathbf{V}$ and horizontal state transitions $\mathbf{H}$ as in (1).

$$\begin{aligned}\mathbf{V} =& \{v_{x,y;m,n} = P\left(q_{k,l} = s_{m,n} | q_{k-1,l} = s_{x,y}\right) : \\ & 1 \leq x \leq N_r, 1 \leq y \leq N_c\} \\ \mathbf{H} =& \{h_{i,j;m,n} = P\left(q_{k,l} = s_{m,n} | q_{k,l-1} = s_{i,j}\right) : \\ & 1 \leq i \leq N_r, 1 \leq j \leq N_c\}\end{aligned} \tag{1}$$

The initial probability can also be separated into horizontal and vertical parts, respectively. Initial horizontal probability distribution can be formulated as $\mathbf{\Pi}_h = \{\pi_{h;m,n}\}$, where $\pi_{h;m,n}$ means the probability of each hidden state appearing in the first column. And initial vertical probability distribution can be formulated as $\mathbf{\Pi}_v = \{\pi_{v;m,n}\}$, where $\{\pi_{v;m,n}\}$ means the probability of each hidden state appearing in the first row.

When using Gaussian mixture model (GMM) to describe the emitting probability $\mathbf{B} = \{b_{m,n}(\boldsymbol{o}_{k,l})\}$, the likelihood of a given observation block $\boldsymbol{o}_{k,l}$ being generated by a given state $s_{m,n}$ can be formulated as (2)

$$b_{m,n}\left(\boldsymbol{o}_{k,l}\right) = \sum_{g=1}^{G_s} \frac{c_{m,n}^{(g)}}{[2\pi]^{\frac{V}{2}} \Sigma^{\frac{1}{2}}} \cdot \mathrm{e}^{-\frac{\left(\boldsymbol{o}_{k,l}-\mu_{m,n}^{(g)}\right) {\Sigma_{m,n}^{(g)}}^{-1} \left(\boldsymbol{o}_{k,l}-\mu_{m,n}^{(g)}\right)^T}{2}} \tag{2}$$

where V is the dimension of feature vector $\boldsymbol{o}_{k,l}$, and $c_{m,n}^{(g)}$, $\mu_{m,n}^{(g)}$, and $\Sigma_{m,n}^{(g)}$ are the weight, the mean, and the covariance of the g-th Gaussian component in the PDF of $s_{m,n}$, respectively.

After the explicit mathematical definition has been given, there are three key issues that need to be addressed in 2D-HMM system. The first is how to extract features from a given image, the second is how to find the optimal model parameters to maximize the probability of observation variables, and the last is how to use the well-trained HMM parameters to get the most probable label of an unlabelled image. We will describe implementation details in next sections. Detailed formulation of algorithm for above issues can be found in *APPENDIX B* of [8] and the *LEARNING* section of [14].

2.2. Feature extraction

As shown in Fig.2, an image is first divided into L frames using a $F_h \times F_w$ sliding window from left to right, with a frame shift of F_r pixels. In each frame, a smaller sliding window of size $F_p \times F_w$ is used to scan from top to bottom with a window shift of F_d pixels. After using the task specific feature extraction method, each grid can be represented by a feature vector $o_{k,l}$. If F_p is set to the same value as F_h, the number of rows K in the grids will be reduced to 1 and that is the feature extraction method used in 1D-HMM.

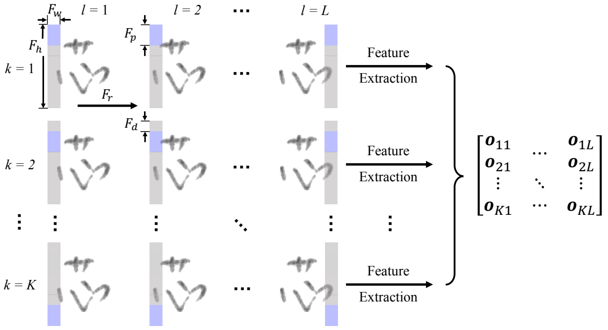
F_w
F_h
F_p
F_r
F_d
$l = 1$
$l = 2$
$l = L$
$k = 1$
$k = 2$
$k = K$
Feature Extraction
Feature Extraction
Feature Extraction
$\begin{bmatrix} o_{11} & \cdots & o_{1L} \\ o_{21} & \cdots & o_{2L} \\ \vdots & \ddots & \vdots \\ o_{K1} & \cdots & o_{KL} \end{bmatrix}$

Fig. 2. Feature extraction method in 2D-HMM.

2.3. Training process

Assume that there are C classes in total, we use a separate 2D-HMM for each class, and the parameter set can be formulated as $\mathbf{\Theta} = \{\theta^c : c = 1, 2, \ldots, C\}$. When we get labelled images of class c from the training set $\mathbf{X}_{tr}^c$, we can use the Alg.1 to find the optimal 2D-HMM parameters for every θ^c.

```
Algorithm 1: 2D HMM Training Algorithm
  Input:
      Labelled images of class $c$, $\mathbf{X}_{tr}^c$,
      Iteration number, $\mathbf{N}_0$;
  Output:
      Optimal parameters for the training set,
      $\hat{\theta}^c = \{\hat{\mathbf{\Pi}}_h^c, \hat{\mathbf{\Pi}}_v^c, \hat{\mathbf{H}}^c, \hat{\mathbf{V}}^c, \hat{\mathbf{B}}^c\}$;
  1: $\mathbf{\Pi}_{h1}^c, \mathbf{\Pi}_{v1}^c, \mathbf{H}_1^c, \mathbf{V}_1^c \leftarrow$ random initialization
  2: $\mathbf{B}_1^c \leftarrow$ first ten images of $\mathbf{X}_{tr}^c$
  3: $i \leftarrow 1$;
  4: while $i < \mathbf{N}_0$ do
  5:     $\hat{\mathbf{Q}}_i \leftarrow \arg\max_{\mathbf{Q}} P(\mathbf{Q}, \mathbf{X}_{tr}^c | \theta_i^c)$
  6:     $\theta_{i+1}^c \leftarrow \arg\max_{\theta^c} P(\theta^c | \hat{\mathbf{Q}}_i)$
  7: end while
  8: return $\hat{\theta}^c = \theta_{N_0}^c$;
```

Note that step 5 use the decoding method in *APPENDIX B* of [8] and step 6 use the *Decision-Directed* learning in [7].

2.4. Testing process

For every unlabelled image T_i from test set $\mathbf{X}_{ts}$, we need to find the most probable label $\hat{C}_i$ of it, which can be formulated as:

$$\hat{C}_i = \arg\max_c \left\{ \max_{m,n} \ln \left[P\left(q_{K,L} = s_{m,n}, O_i | \theta^c \right) \right] \right\} \quad (3)$$

The general idea of the algorithm is listed in Alg.2 .

```
Algorithm 2: 2D HMM Test Algorithm
  Input:
      Unlabelled image, $T_i \in \mathbf{X}_{ts}$,
      2D-HMM parameters, $\mathbf{\Theta} = \{\theta^c : c = 1, 2, \ldots, C\}$;
  Output:
      The most probable label, $\hat{C}_i$;
  $C_{best} \leftarrow 1$
  $P_{max} \leftarrow -\infty$
  $c \leftarrow 1$
  while $c \leq C$ do
      $p_i^c = \max_{m,n} \ln \left[ P\left( q_{K,L} = s_{m,n}, O_i | \theta^c \right) \right]$
      if $p_i^c > P_{max}$ then
          $P_{max} = p_i^c$
          $C_{best} = c$
      end if
  end while
  return $\hat{C}_i = C_{best}$;
```

2.5. Implementation details with Kaldi

We implement a complete codebase for 2D-HMM framework as shown in Fig.3, which is publicly available in https://github.com/jfma-USTC/2DHMM. The main differences between our work and the original 1D-HMM system in Kaldi are annotated with red boxes. For the feature extraction part, we use Perl scripts to integrate executable programs written in C++. For the training and test parts, we design two new classes named *TransitionModel_2D* and *HmmTopology_2D* in Kaldi. *TransitionModel_2D* is used to describe transition matrix in both vertical and horizontal directions. And *HmmTopology_2D* is used to describe 2D-HMM topology including the shape and initial transition probabilities of hidden states map. All functions of 1D-HMM used in Kaldi is rewritten as member functions of two classes. We use the decoding algorithm described in [14] instead of WFST [15] used in Kaldi, since the transition between hidden states in 2D-HMM is nonlinear and hard to be integrated with WFST framework.

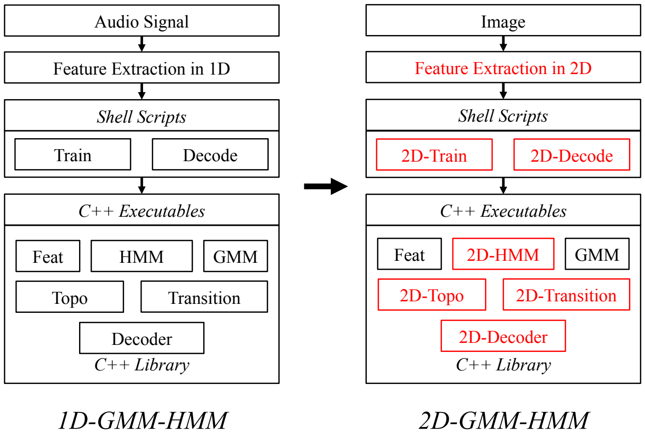

Fig. 3. Code framework comparison between 1D-GMM-HMM and 2D-GMM-HMM system with Kaldi.

3. EXPERIMENTS

3.1. Task description

Due to the variety of Chinese characters and the great differences in writing styles, handwritten recognition for Chinese character has always been a challenging problem [16]. According to the type of data acquisition, handwriting recognition can be divided into online and offline. For offline handwritten Chinese character recognition task, we need to analyze and classify a group of gray images containing Chinese characters.

Different from sound-based writing systems such as Greek and Hebrew, Chinese characters are mainly logographic and consist of basic radicals. It is natural to decompose Chinese characters to radicals and spatial structures then use these knowledge for character recognition. In previous works, CNN-based models [17], [18] treat each Chinese character as a whole without considering inner sub-structures.

Table 1. CER comparison of 1D-GMM-HMM and 2D-GMM-HMM systems with different number of hidden states

Methods	N_r	N_c	CER
1D-GMM-HMM	1	3	16.19%
	1	5	14.39%
2D-GMM-HMM	1	3	13.87%
	1	5	11.02%
	3	3	10.06%
	7	7	7.15%

N_r means the row number of hidden states
N_c means the column number of hidden states

[19] adopts 1D-HMM framework for handwritten Chinese text recognition, but only one-dimensional alignment results in horizontal direction can be obtained. Contrarily, the 2D-HMM method can provide segmentation results in both horizontal and vertical directions, with each radical modeled by one or several hidden states.

3.2. Experimental setup

We conduct our experiments on 50-class Chinese characters randomly selected from the HWDB-1.0 and HWDB-1.1 databases [20] released by Institute of Automation, Chinese Academy of Sciences. For each character class, there are 650 samples for training and 65 samples for testing in average.

The feature extraction method in 1D-GMM-HMM system is basically the same as [19]. Firstly, the image is processed by Otsu binarization, and then the height of the image is normalized to 60 pixels while the aspect ratio remains unchanged. Then, an 80 × 40 rectangular sliding window is used to scan the whole image from left to right along the center line of Chinese characters with a frame shift of 3 pixels. And a 256-dimensional feature vector is extracted for each frame [21, 22]. Finally, PCA transformation is adopted to obtain a compressed 50-dimensional feature vector fed to the training and testing process. In 2D-GMM-HMM system, a 40 × 40 sliding window is used to scan from top to bottom in each frame with a window shift of 5 pixels. The dimension of feature vector is correspondingly reduced from 256 to 128, while the feature dimension after PCA remains 50.

After feature extraction, the 1D-GMM-HMM and 2D-GMM-HMM systems are trained with iteration times set to 40 and the number of Gaussian kernels per hidden state set to 50 in average. And the system performance are compared under the character error rate (*CER*) criterion.

3.3. Result analysis

The results of 50 classes Chinese handwritten character recognition with different hidden states in both system are shown in Table 1. We can observe that under the same number of hidden states (1 × 3 and 1 × 5), the 2D-GMM-HMM system has a large *CER* reduction of 2.85% in average over 1D-GMM-HMM system. When we increase the number of hidden states to 3 × 3 and 7 × 7, the *CER* can be further reduced by 4.33% and 7.24%.

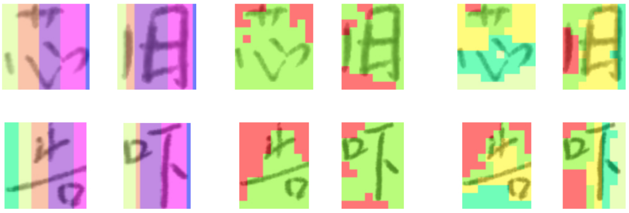
(a) 1D-HMM (b) 2D-HMM (c) 2D-HMM

Fig. 4. Hidden state alignment visualization results. (a) 1D-HMM with 5 hidden states, (b) 2D-HMM with 2 hidden states, (c) 2D-HMM with 4 hidden states

Furthermore, we compare the hidden state alignment results of the two systems. From the visualization results in Fig.4 with different colors representing different hidden states, we can observe that 1D-HMM system can only generate one-dimensional alignments from left to right, while 2D-HMM system is more suitable to capture the complex structure of Chinese radicals due to the flexibility of hidden state transitions in both vertical and horizontal directions. As the increase of hidden states number, the 2D-HMM framework can get better alignment results from simple separation of background and foreground to explicit segmentation of Chinese characters on radical level.

4. CONCLUSION

In this paper, we present a highly efficient code library of 2D-GMM-HMM based on Kaldi toolkit with detailed implementation. The experiments on 50-class HCCR task show that 2D-GMM-HMM system has obvious advantages over 1D-GMM-HMM system in both recognition accuracy and modeling precision. We will introduce the new framework of 2D-DNN-HMM using deep neural network to model state posterior probability instead of generative model like GMM in the future. The code is released in Github at https://github.com/jfma-USTC/2DHMM, hope this code library can provide convenience for researchers handling many other image related tasks.

5. REFERENCES

[1] L. E. Baum and T. Petrie, "Statistical inference for probabilistic functions of finite state markov chains," *An-*

nals of Mathematical Statistics, vol. 37, no. 6, pp. 1554–1563, 1966.

[2] A P Varga and R K Moore, "Hidden markov model decomposition of speech and noise," in *International Conference on Acoustics, Speech, and Signal Processing*, 2002.

[3] S R Eddy, "Hidden markov models," *Current Opinion in Structural Biology*, vol. 6, no. 3, pp. 361–5, 1996.

[4] A Krogh, B Larsson, Heijne G. Von, and E L Sonnhammer, "Predicting transmembrane protein topology with a hidden markov model: application to complete genomes," *Journal of Molecular Biology*, vol. 305, no. 3, 2001.

[5] R Bippus and V Margner, "Script recognition using inhomogeneous p2dhmm and hierarchical search space reduction," 1999, pp. 773–776.

[6] A. Nefian and A. Nefian, "An embedded hmm-based approach for face detection and recognition," in *1999 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings. ICASSP99*, Los Alamitos, CA, USA, mar 1999, vol. 6, pp. 3553–3556, IEEE Computer Society.

[7] Park Hee-Seon and Lee Seong-Whan, "A truly 2-d hidden markov model for off-line handwritten character recognition," *Pattern Recognition*, vol. 31, pp. 1849–1864, 12 1998.

[8] H Othman and T Aboulnasr, "A separable low complexity 2d hmm with application to face recognition," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 25, no. 10, pp. 1229–1238, 2003.

[9] X. Ma, D. Schonfeld, and A. Khokhar, "A general two-dimensional hidden markov model and its application in image classification," in *2007 IEEE International Conference on Image Processing*, 2007, vol. 6, pp. VI – 41–VI – 44.

[10] J. Baumgartner, A. G. Flesia, J. Gimenez, and J. Pucheta, "A new approach to image segmentation with two-dimensional hidden markov models," in *2013 BRICS Congress on Computational Intelligence and 11th Brazilian Congress on Computational Intelligence*, 2013, pp. 213–222.

[11] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, Jan Silovsky, Georg Stemmer, and Karel Vesely, "The kaldi speech recognition toolkit," in *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*. Dec. 2011, IEEE Signal Processing Society, IEEE Catalog No.: CFP11SRW-USB.

[12] P. Ghahremani, B. BabaAli, D. Povey, K. Riedhammer, J. Trmal, and S. Khudanpur, "A pitch extraction algorithm tuned for automatic speech recognition," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 2494–2498.

[13] T. Hayashi, R. Yamamoto, K. Inoue, T. Yoshimura, S. Watanabe, T. Toda, K. Takeda, Y. Zhang, and X. Tan, "Espnet-tts: Unified, reproducible, and integratable open source end-to-end text-to-speech toolkit," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 7654–7658.

[14] Heba Othman and T. Aboulnasr, "A simplified second-order hmm with application to face recognition," pp. 161 – 164 vol. 2, 06 2001.

[15] Mehryar Mohri, Fernando Pereira, and Michael Riley, "Weighted finite-state transducers in speech recognition," *Computer Speech & Language*, vol. 16, pp. 69–88, 01 2002.

[16] Ruwei Dai, Cheng-Lin Liu, and Baihua Xiao, "Chinese character recognition: History, status and prospects," *Frontiers of Computer Science in China*, vol. 1, pp. 126–136, 05 2007.

[17] Dan Ciresan and Ueli Meier, "Multi-column deep neural networks for offline handwritten chinese character classification," 07 2015, pp. 1–6.

[18] Zhao Zhong, Xu-Yao Zhang, Fei Yin, and Cheng-Lin Liu, "Handwritten chinese character recognition with spatial transformer and deep residual networks," 12 2016, pp. 3440–3445.

[19] Z R Wang, J Du, W C Wang, et al., "A comprehensive study of hybrid neural network hidden markov model for offline handwritten chinese text recognition," *International Journal on Document Analysis and Recognition*, vol. 21, no. 4, pp. 241–251, 2018.

[20] Cheng Lin Liu, Fei Yin, Da Han Wang, and Qiu Feng Wang, "Casia online and offline chinese handwriting databases," *International Conference on Document Analysis & Recognition*, 2011.

[21] Cheng Lin Liu, "Normalization-cooperated gradient feature extraction for handwritten character recognition," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 29, no. 8, pp. 1465–1469, 2007.

[22] Z Bai and Q Huo, "A study on the use of 8-directional features for online handwritten chinese character recognition," *8th International Conference on Document Analysis & Recognition*, 2006.