DeepAD: A Deep Learning Based Approach to Stroke-Level Abnormality Detection in Handwritten Chinese Character Recognition

Tie-Qiang Wang[1,2], Cheng-Lin Liu[1,2,3]
[1]National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, 95 Zhongguancun East Road, Beijing 100190, P.R. China
[2]University of Chinese Academy of Sciences, Beijing, P.R. China
[3]CAS Center for Excellence of Brain Science and Intelligence Technology, Beijing, 100049, P.R. China
Email: {tieqiang.wang, liucl}@nlpr.ia.ac.cn

***Abstract*—Writing abnormality detection is very important in education applications, but has received little attention by the community. Considering that abnormally written strokes (writing error or largely distorted stroke) affect the decision confidence of classifier, we propose an approach named DeepAD to detect stroke-level abnormalities in handwritten Chinese characters by analyzing the decision process of deep neural network (DNN). Firstly, to minimize the effect of stroke width variation of handwritten characters, we propose a skeletonization method based on fully convolutional network (FCN) with cross detection. With a convolutional neural network (CNN) for character classification, we evaluate the role of each skeleton pixel by calculating its impact on the prediction of classifier, and detect abnormal strokes by connecting pixels of negative impact. For quantitative evaluation of performance, we build a template-free dataset named SA-CASIA-HW containing 3696 handwritten Chinese characters with various stroke-level abnormalities, and spanning 3000+ different classes written by 60 individual writers. We validate the usefulness of the proposed DeepAD with comparison to related methods.**

***Index Terms*—handwritten Chinese character recognition (HCCR), stroke-level handwriting abnormality detection, skeletonization, decision making process, conditional sampling**

I. Introduction

Over the last few years, handwritten Chinese character recognition (HCCR) [1] has achieved high accuracies as over 97% on up to 3755 categories by using convolutional neural networks (CNNs) in the writer-independent recognition task [2]. But the interpretability of deep HCCR remains an open issue. Actually, some applications depend heavily on this issue, for example, analyzing how handwritten strokes influence the decision making of deep classifiers is a significant step to the adoption of deep learning in the handwriting abnormality detection, which is needed in education applications.

For instance, Fig. 1 shows 12 test samples from 3 randomly chosen classes in a descending order of confidence predicted by a CNN classifier named HCCR-CNN9Layer [2]. Though they are all recognized correctly, regularly-written samples tend to be given higher scores obviously. Stroke-level deformations, sloppy writing and irregular configurations of handwritten strokes decline the confidence scores dramatically. These

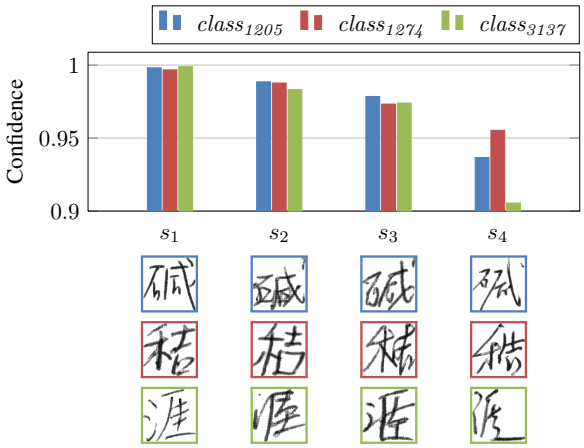


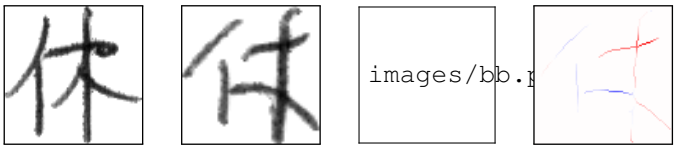
Figure 1: HCCR-CNN9Layer predicts the classes of test samples according to the maximal outputs of softmax function. These samples are from three classes, boxed in three different colors.

cases reveal that written abnormalities influence against the confidence of desirable classes given by classifiers. Broadly speaking, the goal of DeepAD is to find the pixels which cause negative effects on the correct predictions made by DNNs via interpretability based method.

Recently, there have been some works toward the interpretability of deep model [3]–[9]. One popular way is visualizing the knowledge learned by models in the input space. There are mainly two groups of algorithms for the visualization: backpropagation based [3]–[7] and forward propagation based [8], [9]. Backpropagating usually tells how sensitive the loss are to each input pixel. For example, Shrikumar et al. [4] and Simonyan et al. [3] compare the real output of each neuron with its ideal output, then backpropagate the difference to the input space. Bach et al. [5] propose effective approximations to the derivations for different layers in CNNs, while Zeiler et al. [7] simply operate the inverse processes of CNNs to visualize what they have learnt.

Backpropagating operations highlight the most important features for the correct predictions [3]. But since the layer-by-layer propagation magnifies deviations hierarchically as

shown in Fig. 2(b) & 2(c), it fails to generate positive or negative regions, while forward propagation based methods exhibit more meaningful results. Class activation mapping (CAM) [8] uses a linear weighted sum of feature maps from the last convolutional layer to detect the most salient regions. Furthermore, the method in [9] outputs those negative regions which are against the correct predictions. Because stroke-level abnormalities in handwritten characters will decline the scores of correct predictions, our proposed DeepAD is to detect stroke-level abnormalities by finding negative regions in input images (shown in Fig. 2(b) & 2(d)), based on the forward propagation based analysis in [9].



(a) (b) (c) (d)

Figure 2: Results of back/forward-propagation based methods. (a) is the correct writing of $class_{3068}$. (b) is a correctly recognized sample of $class_{3068}$, but contains stroke-level handwritten abnormalities. (c) is produced by the algorithm in [3] on (b). (d) shows the connected (red) positive/(blue) negative regions generated by DeepAD.

As far as we know, our work is the first to detect stroke-level handwriting abnormalities from deep classifiers: DeepAD first performs skeletonization on the input image using a fully convolutional network (FCN) combined with overlap detection for better preserving stroke shapes; Next, the HCCR-CNN9Layer [2] is used to predict the character class label, and the role of each foreground pixel in the skeleton map is measured from the network response with respect to the predicted class; Finally, connected pixels in negative roles are identified as abnormally written strokes.

The rest of this paper is organized as follows: Section II briefly reviews related works. Section III details the proposed DeepAD, Section IV presents experimental results, and Section V draws the conclusion.

II. Related Work

Model-Based Structural Matching. Structural matching methods [10]–[13] parse characters by stroke extraction & template matching. Kim and Kim [11] represent characters by the joint distributions of feature points extracted from their strokes, then a template-driven matching algorithm outputs an optimal path stroke correspondence between the template and the input character [13]. Zeng and Liu [12] model characters via Markov random fields (MRFs), and understand character image by MRF-guided stroke matching. Liu et al. [13] design stroke templates in writer order and inter-stroke relationship types for each character, and stroke correspondence between a template character and the input character is obtained by heuristic search when matching. Afterwards, those strokes with large deformation can be detected.

In this paper, we deploy the model-based structural matching algorithm [13] in our task for comparison, where we assume that the matching results with maximal distances indicate that abnormalities occur.

Skeletonization for Handwritten Characters. Skeletonization overcomes the stroke width variation and grasps the shape. For a certain stroke, analysis on its skeleton reduces the calculation and tends to produce robust outputs [14]. Most thinning methods base on local visual rules [15]. But when facing complex and unsmooth shapes, these methods distort lines at the crosses of strokes and produce redundant branches easily. Recently, [16] extracts pure and unbroken skeletons via a supervised FCN. Here, we improve the rule-based post-processing module in [16] by a cross detection network [17].

Pixel-Level Contributions in CNNs. In [18], to explain predictions for instances, Robnik et al. propose a general method for all classifiers with probabilistic outputs. Specifically, they eliminate each pixel via probabilistic methodology and observe the changes in the predictions. Inspired by [18], Zintgraf et al. [9] mine the fine-grained differences between similar categories in a similar way. Our work is also based on the instance-specific technologies in [9], [18].

III. Methodology

The pipeline of DeepAD is shown in Fig. 3, where the image I is recognized as class c by CNN, and S denotes the skeleton map of I. When we measure the effect of each pixel x in S, the basic idea is that the role of x can be estimated by measuring how the predicted score of classifiers changes if x "disappears" in I. Finally, as shown in Fig. 2(d), blue pixels cause negative effects on predicting I into class c, while red pixels contribute to the correct prediction. Thus a blue connected component is identified to a abnormal stroke. For evaluation, we only extract the detected strokes with the largest areas.

A. Skeletonization Module

Currently, FCN based skeletonization [16] has outperform other methods. However, extra efforts are still needed to thinning the crossing regions. To deal with this, we extract crossing regions in advance and thin them subsequently.

1) Cross/Skeleton Maps Generation: Supervised learning in cross detection [17] need the annotations for a large training set [16]. Fortunately, the online handwritten samples [1] can record strokes in the form of (x, y)-coordinate sequences, thus we generate the cross regions by dilating the intersection areas of strokes [17]. As shown in Fig. 3, our dual nets are the same, thus we only introduce skeleton net. The net is the lightweight version of [16]. The parameters are listed in the following: (1) convolution filters from HCCR-CNN9Layer; (2) weights for giving side outputs from multiple scales; (3) learnable upsampling for enlarging feature maps; (4) convolution kernels for generating candidate maps and the final fusion.

2) Skeletonization Algorithm for Cross Regions: After obtaining cross/skeleton maps, we conduct the final skeletonization process. The details of our algorithm are in **Algorithm 1**. To evaluate the performance, we synthesize training data in {offline image, skeleton target, cross target}-format from CASIA-OLHWDB1.1 [1] ($\sim$1.121 millions) for training and

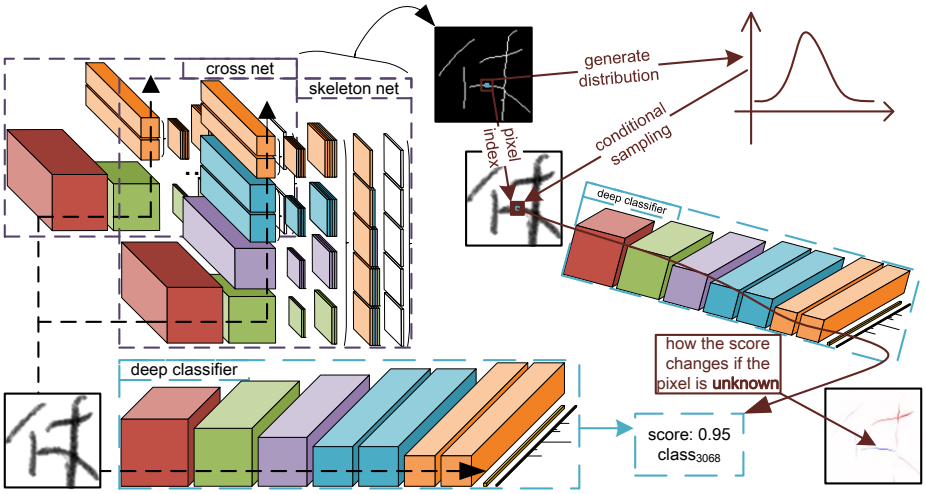


Figure 3: The pipeline of the proposed stroke-level abnormality detection method: DeepAD. Recognition and skeletonization can be conducted simultaneously, and we assume that the deep model gives correct prediction c ($class_{3068}$) with score p_c (0.95). Afterwards, for each foreground pixel in skeleton map, we make it unknown to classifiers, then determine its role by observing the ascending or descending of p_c. Here, the calculations labeled in brown color tell how to make the blue pixel unknown (detailed in Section III-B).

test our models on the synthesized data (∼224 thousands) from ICDAR-2013 Online HCCR Competition Database [16], [19]. The detailed results are listed in **TABLE** I, where our dual FCNs outperform the state-of-the-art model [16].

Algorithm 1 Skeletonization for crossing regions

```
INPUT: skeleton map S_p (i.e., the thinning result of [16]; cross map S_o
get the set of crossing regions: R_o
for each r_o in R_o do
    if each pixel in r_o is background point in S_p then
        continue
    else
        get the central point p_o^c of r_o
        for each skeleton point p_s in S_p do
            if p_s ∉ r_o and p_s is contiguous to r_o then
                link p_s to p_o^c in S_p
            else
                continue
            end if
        end for
    end if
end for
RETURN: S_p
```

Table I: Comparison between skeletonization models on synthesized data from ICDAR-2013 Online HCCR Competition Database. Image size: 96×96. Average minimal distance (AMD) [16] describes the visual distance between the prediction and ground truth maps.

Method	F-measure	AMD
[16]	0.610	1.29
Dual net (proposed)	**0.615**	**1.26**

B. Measuring the Roles of Input Pixels

Since the abnormality parts of a handwritten Chinese character should include those pixels which can make the largest contributions to the **mis**classification in HCCR, we design a pixel-wised contribution quantization method to measure the role that each pixel plays [9] in HCCR. Here, for ith pixel x_i in input image I, its abnormality contribution $C^a_{x_i}$ is regarded as the performance change of the classifier caused by the cases with and "**without**" x_i:

$$C^a_{x_i} = \text{odds}(c|I) - \text{odds}(c|I_{\setminus x_i}),\ \text{odds}(c|I) = \log\left(\tfrac{p(c|I)}{1-p(c|I)}\right), \quad (1)$$

where c represents the class of I, $I_{\setminus x_i}$ denotes the input image "**without**" pixel x_i [9], [18], and $p(c|I)$ and $p(c|I_{\setminus x_i})$ indicate the correct predictions made by HCCR-CNN9Layer on input images I and $I_{\setminus x_i}$, respectively. In statistics, odds$(\cdot)$ represents the expression of relative probabilities. In Equation (1), odds$(\cdot)$ eliminates the numerical difference between $p(c|I)$ and $p(c|I_{\setminus x_i})$ and normalizes them effectively.

Here, $p(c|I)$ simply describes the classifier. As for $p(c|I_{\setminus x_i})$, since each pixel is related to its adjacent pixels, therefore, it is not reasonable to simply remove or set a special value to that pixel to calculate $p(c|I_{\setminus x_i})$. In order to deal with this problem, we simulate the absence of x_i by marginalizing methods [9]. Then $p(c|I_{\setminus x_i})$ can be written as

$$p(c|I_{\setminus x_i}) = \textstyle\sum_j^{N_s} p(c|I_{x_i^j})\ p(I_{x_i^j}|I_{\setminus x_i}),\ I_{x_i^j} = I_{\setminus x_i} + x_i^j, \quad (2)$$

where I_{x_j} indicates the jth possible case of the input image with replaced pixel value x_j and N_s denotes the number of different possible values of the ith pixel x_i.

However, modeling $p(I_{x_i^j}|I_{\setminus x_i})$ can easily become infeasible with a huge number of pixels. Therefore, we approximate Equation 2, by assuming that the value at x_i is dependent among those pixels located in a x_i-centric $l \times l$-region l_{x_i}. The region l_{x_i} has been illustrated in Fig. 4, and the approximation of Equation (2) can be rewritten as the following:

$$p(c|I_{\setminus x_i}) = \textstyle\sum_j^{N_s} p(c|I_{x_i^j})\ p(I_{x_i^j}|l_{x_i}). \quad (3)$$

This approximation is based on the following two observations in Fig. 4: ***Rule A:*** The ith pixel in input image x_i (the blue pixels in Fig. 4) mainly depends on its neighborhood l_{x_i} (pixels in the brown boxes in Fig. 4) around x_i; ***Rule B:*** l_{x_i}

does not depend on its position, i.e., all patches in different positions with the identical size $l \times l$ in I are independently and identically distributed.

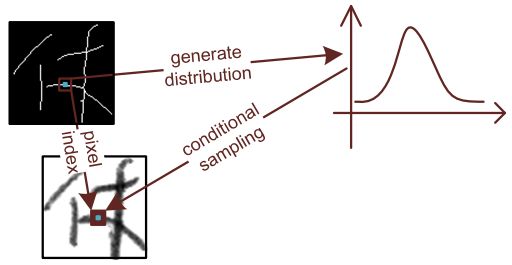


Figure 4: Conditional approximation of $p(I_{x_i^j}|I_{\setminus x_i})$: from the whole image to local neighbors.

Under ***Rule B***, we assume that both l_{x_i} and $p(I_{x_i^j}|I_{\setminus x_i})$ will obey Gaussian distributions. Therefore, we can flatten l_{x_i} into f_{x_i}, and figure out the mean vector μ_f and the covariance matrix cov_f of f_{x_i}. Then we generalize x_i^j into a $k \times k$-size region k_{x_i}, which is surrounded by l_{x_i}. Finally, we need to make k_{x_i} "**disappear**" from f_{x_i}. Now we have:

$$\mu_f = \text{concat}(\mu_k, \mu_{f_{\setminus \mu_k}}),\ f_{x_i} = \text{concat}(k_{x_i}, f_{x_i \setminus k_{x_i}}), \tag{4}$$

$$p(x_i^j|l_{x_i}) = p(k_{x_i}|f_{x_i}). \tag{5}$$

Here, we know that $p(k_{x_i}|f_{x_i})$ obeys Gaussian distribution, and the conditionally sampled mean vector μ_{cs} as well as covariance matrix $\mathbf{cov}_{cs}$ are given by [9] and Fig. 5:

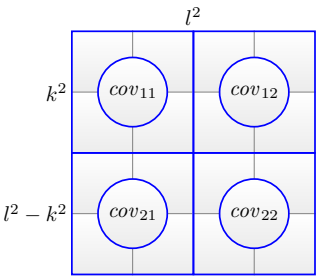


Figure 5: The covariance matrix $\mathbf{cov}_f$ of f_{x_i}.

$$\mu_{cs} = \mu_k + (\mathbf{cov}_{12}\mathbf{cov}_{22}^{-1})(f_{x_i \setminus k_{x_i}} - \mu_{f_{\setminus \mu_k}}), \tag{6}$$

$$\mathbf{cov}_{cs} = \mathbf{cov}_{11} - (\mathbf{cov}_{12}\mathbf{cov}_{22}^{-1})\mathbf{cov}_{21}. \tag{7}$$

With these two parameters (i.e., mean vector μ_{cs} and covariance matrix $\mathbf{cov}_{cs}$), we can get the distribution of all possible values of the ith pixel x_i and sample N_s values from the distribution to calculate $p(c|I_{\setminus x_i})$ using Equation (3).

C. Teacher-Student Learning for HCCR-CNN9Layer

DNNs trained with one-hot supervisory targets [2] are likely to output extreme confidences (approaching 0 or 1). But DeepAD measures the effects caused to the output of deep classifier by pixels, which requires the classifier to respond to fine-grained differences among those similarly handwritten characters. Thus we apply the teacher-student learning [20] to yield fine-grained discrimination. Firstly, we choose a trained HCCR-CNN9Layer T_N as teacher, and a newly-constructed HCCR-CNN9Layer S_N as student. Then, we utilize the loss L_{KD} to update S_N iteratively. The objective function of teacher-student framework can be written in the following:

$$L_{KD} = \mathcal{H}(y, P_S) + \lambda \mathcal{H}(P_T^\tau, P_S^\tau), \tag{8}$$

$$P_T^\tau = \text{softmax}\left(\tfrac{a_T}{\tau}\right),\ P_S^\tau = \text{softmax}\left(\tfrac{a_S}{\tau}\right), \tag{9}$$

where a_T and a_S represent the input of teacher network and student network respectively, P_T and P_S indicate the output of teacher network and student network respectively, λ and τ are hyper-parameters, and $\mathcal{H}$ denotes various candidate loss functions in classification tasks. In our experiments, T_N and S_N share the same configuration. Though S_N reports a little lower classification accuracy compared with T_N (i.e., 96.08% $<$ 97.65% on ICDAR13-OFFLINE [19]), it outperforms T_N on our abnormality detection task.

IV. Experiments

We compiled two datasets from HCCR datasets HANJA [13] and CASIA-HWDB [1] for evaluation. In the former one, stroke templates are available, which enables the comparison between the DeepAD and stroke matching based method. For convenience, we give only one bounding box to the most evident written abnormality of each sample, and evaluate different methods following the IoU (**I**ntersection **o**ver **U**nion: $\text{IoU} = \frac{\text{Area of Cross}}{\text{Area of Union}}$) criteria. When an IoU is $\geq$ a given threshold, we believe that a written abnormality is detected[1].

A. Datasets

1) SA-HANJA: HANJA-DB is a database of handwritten Chinese characters collected by KAIST [13]. The database includes 783 frequently-used Chinese characters, among which we select 355 different classes which belong to the first plane of GB2312. SA-HANJA (**S**troke-level **A**bnormality-HANJA) contains 375 samples extracted from the testing part of HANJA-DB with obvious written abnormalities, and occupies 355 pre-defined templates for the selected 355 classes, which is large enough for us to compare our method with the model-based stroke matching method [13].

2) SA-CASIA-HW: In CASIA-HWDB mentioned in Section III-B, the datasets HWDB1.0 & 1.1 contain all the 3755 classes in GB2312 level-1 set, and were collected from ∼600 writers. These two datasets will serve as the training set of HCCR-CNN9Layer. Meanwhile, ICDAR13-OFFLINE (3755 classes, ∼60 different writers with HWDB1.0 & 1.1) can be used as the testing set of HCCR-CNN9Layer. Moreover, we also construct a subset SA-CASIA-HW (**S**troke-level **A**bnormality-HWDB) containing 3696 samples (3087 classes, 60 writers) with stroke-level abnormalities from ICDAR13-OFFLINE to evaluate our model.

3) SA-CASIA-HW$_T$: SA-CASIA-HW$_T$ can is a subset of SA-CASIA-HW, sharing the same handwritten Chinese character classes with SA-HANJA and inheriting those templates

[1]Referred deep models, all datasets collected by the authors, evaluation toolkits and experimental results are available at https://www.dropbox.com/sh/2979y3gsom8mkvy/AAD0XIxaFCTKfQ0TXuGPL7Fwa?dl=0.

from SA-HANJA. SA-CASIA-HW$_T$ is designed for the comparison between the model-based stroke matching method [13] and our method originally.

B. Experiments on SA-HANJA and SA-CASIA-HW$_T$

First, we conduct experiments on two datasets: SA-HANJA and SA-CASIA-HW$_T$. **TABLE** II shows that our method outperforms stroke-matching based approach. In SA-HANJA, samples are written more standardly with more stable stroke widths, while samples in SA-CASIA-HW$_T$ are written more freely in gray format with more complex stroke-level deformations. Therefore, both of those two methods report higher accuracies on SA-HANJA significantly.

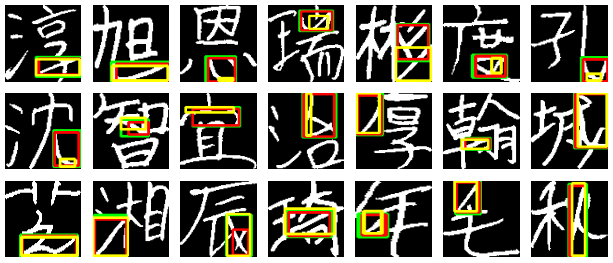


Figure 6: Some good detection results on SA-HANJA. Ground-truth bounding boxes are labeled in green, DeepAD outputs red bounding boxes, and stroke matching method outputs yellow bounding boxes.

Figure 7: Some bad results given by stroke-matching method.

Some randomly chosen abnormality detection results are shown in Fig. 6. In fact, with a given IoU-threshold, though some results are judge as false or missed detections, they are still explainable. Stroke-matching method approximates all strokes by single straight lines, parameterizes stroke-level templates and limits the tolerance of each template for writting distortion. Thus we can see that in Fig. 6, the stroke-matching method finds more abnormalities in straight strokes, while tends to output unreasonable results when facing curved strokes as shown in Fig. 7. From Fig. 7, we can see that there are two cases of miss detections by using stroke-matching method: (1) some normal strokes in curved shape are sentenced to be abnormal; (2) some curved written strokes contained abnormalities but are missed.

C. Experiments on SA-CASIA-HW

This is to evaluate the effectiveness of teacher-student learning framework on the SA-CASIA-HW in **TABLE** III. When CNNs drive the outputs to match one-hot vectors, the negative influences caused by stoke-level written abnormalities will be reduced layer-by-layer. However, in our task, the stroke-level abnormalities are the most crucial causes to the dropping of the predicted scores. Thus, when detecting those abnormalities, we encourage S_N to predict the true targets as well as match the soft targets provided by T_N, and S_N outperforms T_N on our task indeed.

Table II: Detection accuracies on SA-HANJA & SA-CASIA-HW$_T$.

IoU	Stroke Matching [13]		DeepAD	
	SA-HANJA	SA-CASIA-HW$_T$	SA-HANJA	SA-CASIA-HW$_T$
0.20	0.576	0.498	0.839	0.860
0.25	0.544	0.443	0.826	0.833
0.30	0.520	0.407	0.802	0.712
0.35	0.490	0.386	0.776	0.697
0.40	0.458	0.313	0.754	0.619
0.45	0.434	0.302	0.725	0.553
0.50	0.386	0.274	0.714	0.466
0.55	0.352	0.239	0.677	0.401
0.60	0.304	0.151	0.640	0.309
0.65	0.234	0.123	0.616	0.270
0.70	0.186	0.071	0.570	0.179
0.75	0.130	0.055	0.472	0.127

Table III: Comparisons on teacher/student models.

IoU	T_N	S_N	IoU	T_N	S_N
0.20	0.875	0.899	0.50	0.467	0.484
0.25	0.843	0.850	0.55	0.395	0.406
0.30	0.771	0.793	0.60	0.294	0.318
0.35	0.719	0.721	0.65	0.233	0.253
0.40	0.630	0.646	0.70	0.157	0.174
0.45	0.551	0.561	0.75	0.101	0.118

D. Influence of Input Image Format

In **TABLE** IV, we can prove that our method work better when the input is original gray image. There are some reasons to explain this: (1). deep models always show the best performance when processing the raw images [2]; (2). local patches in raw images are more likely to be aligned with Gaussian distribution than those in binary images; (3). the edge points of strokes are not credible enough because they are connected with both foreground and background points.

Table IV: Comparison on different types of input images for HCCR-CNN9Layer. "Raw", "Binary", "Skeleton" indicate the raw images, binarized images and skeleton maps. All these abnormality detection results are evaluated on SA-CASIA-HW.

IoU	Input	Accuracy	IoU	Input	Accuracy
0.50	Raw	**0.467**	0.55	Raw	**0.395**
	Binary	0.454		Binary	0.357
	Skeleton	0.410		Skeleton	0.302
0.60	Raw	**0.294**	0.65	Raw	**0.233**
	Binary	0.261		Binary	0.226
	Skeleton	0.245		Skeleton	0.196
0.70	Raw	**0.157**	0.75	Raw	**0.101**
	Binary	0.142		Binary	0.087
	Skeleton	0.113		Skeleton	0.056

E. Error Analysis

We present some successful detection results on SA-CASIA-HW in Fig. 8, where the explicit handwriting abnormalities are localized accurately. But the accuracies in above-mentioned tables still seem not very satisfactory. Therefore we focus on analyzing the unsuccessful samples in this part.

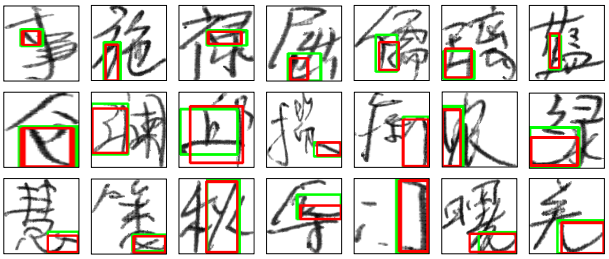


Figure 8: Some detection results given by our DeepAD model on SA-CASIA-HW with ground-truth bounding boxes in green and our detection results in red.

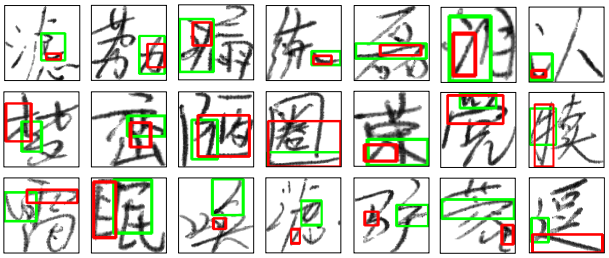


Figure 9: Some unsatisfactory results given by our method on SA-CASIA-HW with ground-truth bounding boxes in green and our results in red.

In Fig. 9, we list three types (in 3 rows) of typical failures in our experiments. ***Row-1:*** the ground-truth bounding boxes cover larger connected components, but our method boxes smaller areas while excludes some normally-written parts in ground-truth boxes; ***Row-2:*** our method detects larger areas than the regions boxed by ground truths; ***Row-3:*** False detections. In these cases, a fair amount of detected strokes have right shapes but appear in unconventional positions or abnormal widths.

Based on the above discussion, we can conclude that the proposed DeepAD can better handle the handwriting abnormality detection problem in HCCR compared with other matching methods, and produce meaningful outputs in general.

V. Conclusions

Character recognition should not only give the class decision, but also detects irregularly/wrongly written strokes, which is especially important for writing quality assessment and education. Therefore, we propose a stroke-level handwriting abnormality detection method named DeepAD for offline handwritten Chinese characters. By combining with cross detection, DeepAD achieve best F-measure in skeleton extraction, and the skeleton maps provide us all the strokes and pixels we need. Afterwards, we analyze how the value at each pixel of skeletons influences the predictions of deep classifiers. Finally, connected pixels which are against the correct predictions are connected to give abnormally written strokes. We also release a dataset for the stroke-level abnormality detection task and build a baseline benchmark. Overall, the proposed method shows promise and potential.

Acknowledgment

This work has been supported by the National Natural Science Foundation of China (NSFC) Grants 61721004 and 61411136002.

References

[1] C.-L. Liu, F. Yin, D.-H. Wang, and Q.-F. Wang, "CASIA online and offline Chinese handwriting databases," in *International Conference on Document Analysis and Recognition*, 2011, pp. 37–41.

[2] X. Xiao, L. Jin, Y. Yang, W. Yang, J. Sun, and T. Chang, "Building fast and compact convolutional neural networks for offline handwritten Chinese character recognition," *Pattern Recognition*, vol. 72, pp. 72–81, 2017.

[3] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps," *arXiv preprint arXiv:1312.6034*, 2013.

[4] A. Shrikumar, P. Greenside, A. Shcherbina, and A. Kundaje, "Not just a black box: Learning important features through propagating activation differences," *arXiv preprint arXiv:1605.01713*, 2016.

[5] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, and W. Samek, "On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation," *PloS one*, vol. 10, no. 7, pp. 1–46, 2015.

[6] G. Montavon, S. Lapuschkin, A. Binder, W. Samek, and K.-R. Müller, "Explaining nonlinear classification decisions with deep taylor decomposition," *Pattern Recognition*, vol. 65, pp. 211–222, 2017.

[7] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *European Conference on Computer Vision*, 2014, pp. 818–833.

[8] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2921–2929.

[9] L. M. Zintgraf, T. S. Cohen, T. Adel, and M. Welling, "Visualizing deep neural network decisions: Prediction difference analysis," in *International Conference on Learning Representations*, 2017.

[10] C. Leung, Y. Cheung, and Y. Wong, "A knowledge-based stroke-matching method for Chinese character recognition," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 17, no. 6, pp. 993–1003, 1987.

[11] I.-J. Kim and J.-H. Kim, "Statistical character structure modeling and its application to handwritten Chinese character recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 11, pp. 1422–1436, 2003.

[12] J. Zeng and Z.-Q. Liu, "Markov random field-based statistical character structure modeling for handwritten chinese character recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 5, pp. 767–780, 2008.

[13] C.-L. Liu, I.-J. Kim, and J. H. Kim, "Model-based stroke extraction and matching for handwritten Chinese character recognition," *Pattern Recognition*, vol. 34, no. 12, pp. 2339–2352, 2001.

[14] W. Shen, K. Zhao, Y. Jiang, Y. Wang, Z. Zhang, and X. Bai, "Object skeleton extraction in natural images by fusing scale-associated deep side outputs," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 222–230.

[15] T. Zhang and C. Y. Suen, "A fast parallel algorithm for thinning digital patterns," *Communications of the ACM*, vol. 27, no. 3, pp. 236–239, 1984.

[16] T.-Q. Wang and C.-L. Liu, "Fully convolutional network based skeletonization for handwritten Chinese characters," in *AAAI Conference on Artificial Intelligence*, 2018.

[17] B. Kim, O. Wang, A. C. Öztireli, and M. Gross, "Semantic segmentation for line drawing vectorization using neural networks," *Computer Graphics Forum*, vol. 37, no. 2, pp. 329–338, 2018.

[18] M. Robnik-Šikonja and I. Kononenko, "Explaining classifications for individual instances," *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, no. 5, pp. 589–600, 2008.

[19] F. Yin, Q.-F. Wang, X.-Y. Zhang, and C.-L. Liu, "ICDAR 2013 Chinese handwriting recognition competition," in *International Conference on Document Analysis and Recognition*, 2013, pp. 1464–1470.

[20] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," in *NIPS Deep Learning and Representation Learning Workshop*, 2015.