The Thirty-Second AAAI Conference
on Artificial Intelligence (AAAI-18)

Fully Convolutional Network Based Skeletonization for Handwritten Chinese Characters

Tie-Qiang Wang,[1,2] **Cheng-Lin Liu**[1,2]
[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
Email: {tieqiang.wang, liucl}@nlpr.ia.ac.cn

Abstract

Structural analysis of handwritten characters relies heavily on robust skeletonization of strokes, which has not been solved well by previous thinning methods. This paper presents an effective fully convolutional network (FCN) to extract stroke skeletons for handwritten Chinese characters. We combine the holistically-nested architecture with regressive dense upsampling convolution (rDUC) and recently proposed hybrid dilated convolution (HDC) to generate pixel-level prediction for skeleton extraction. We evaluate our method on character images synthesized from the online handwritten dataset CASIA-OLHWDB and achieve higher accuracy of skeleton pixel detection than traditional thinning algorithms. We also conduct skeleton based character recognition experiments using convolutional neural network (CNN) classifiers on offline/online handwritten datasets, and obtained comparable accuracies with recognition on original character images. This implies the skeletonization loses little shape information.

Introduction

Recently, deep neural networks have promoted the handwritten character recognition performance significantly. Even the large category set problem, handwritten Chinese character recognition (HCCR) (Zhang, Bengio, and Liu 2017), has achieved high accuracies as over 97% by using convolutional neural networks (CNNs). Despite the superior feature learning and classification capability of CNNs, they do not offer structural interpretation of characters, say, the composition of strokes and radicals and their inter-relationship. Structural analysis of handwritten characters has been studied since 1970s (Pritchard and Sondak 1973) but until now, it is unsolved, partially because of the difficulty of stroke extraction and structural model learning. Structural analysis remains an important issue because in many applications, such as education (Hsiung et al. 2017), human interaction, and personalized font generation, the interpretation of strokes and radicals, and the detection of stroke errors are necessary.

Character skeleton conveys key information for shape recognition, and is particularly important for extracting the structure of strokes. So, skeletonization or thinning has been studied intensively and many algorithms have been proposed (Zhang and Suen 1984; Arcelli and Di Baja 1985; Dong, Lin, and Huang 2016).

There are mainly two types of thinning algorithms: neighbor-based algorithms and distance-based ones. The former methods execute iteratively to delete pixels on the boundary strokes until centered lines remain, and the deletion or retention of stroke pixels depends on the connectivity in the neighborhood, such as the ZhangSuen algorithm (Zhang and Suen 1984). An improved ZhangSuen algorithm was designed for Odia characters, combining with stroke correction (Pujari, Mitra, and Mishra 2014). In (Dong et al. 2017), stroke continuity detection serves as a preprocessing step for thinning. Recently, (Alghamdi and Teahan 2017) proposes a novel algorithm based on the boundary deletion with colour coding. The latter ways yield skeletons straightforwardly by using distance transforms to extract the medial axis of a stroke. Variant methods differ in the distance functions: city block distance (Arcelli and Di Baja 1989), Euclidean distance or constrained Delaunay triangulation technique (Zou and Yan 2001).

These methods are likely to yield unsatisfactory results when facing: (1) complex shapes, (2) variable stroke widths and (3) unsmooth edges. Particularly, the extracted lines are often distorted at the crosses or intersections of strokes (Dong et al. 2017). These outcomes make stroke extraction and structural analysis difficult, while fully convolutional networks (FCNs) (Long, Shelhamer, and Darrell 2015) provide an pixel-to-pixel manner to solve these problems. For example, holistically-nested networks (Xie and Tu 2015) and scale-associated networks (Shen et al. 2016) successfully predict the contour maps and skeleton maps of generic objects in natural images, respectively. But these models do not assure that it is one-pixel width in the output contour or skeleton.

This paper proposes a FCN-based skeleton extraction method for handwritten Chinese characters, which are typical of complex structures. The network fuses holistically-nested features at multiple scales (Xie and Tu 2015), to reduce the information loss caused by upsampling operations. With fewer computational overhead, our method is particularly beneficial for better structure preservation at crossing areas. For supervised training, it is infeasible to label skeleton pixels for large number of offline handwritten samples.

So, we synthesize training samples from online handwritten data. Experiments show that the model trained with synthesized data work well on real offline data.

The major contributions of this work are as follows: (1) we propose an effective FCN-based method for skeletonization of handwritten characters; (2) we adopt a regressive version of dense upsampling convolution (DUC) (Wang et al. 2017) to bridge breakpoints; (3) in the fusion phase, we design multi-rate dilated convolution to make full use of contextual information from different scales, and attain recognizable skeletons for machine recognition.

The rest of this paper is organized as follows: Section 2 briefly reviews the multi-loss FCNs for relative tasks, Section 3 details the proposed method, Section 4 presents experimental results of skeleton extraction and character recognition, and Section 5 offers concluding remarks.

Related Work

Deep Side Outputs

Most existing methods in character skeleton extraction focus on either local visual rules (Zhang and Suen 1984; Pujari, Mitra, and Mishra 2014; Dong et al. 2017) or distance measurements (Zou and Yan 2001). These methods focus on low-level features in local regions, but when reading, humans turn to concern the skeletons of characters subconsciously and ignore the colors or widths of strokes.

Deep side outputs (Xie and Tu 2015) emulate the aforementioned human behavior at multiple scales with different receptive field (RF) sizes. A standard architecture with layer-by-layer side outputs in a simple 1-stream network (Shen et al. 2016) has announced the progress in extracting skeletons of generic objects, such as quadrupeds or airplanes under natural scenes. Though fusing scale-specific features from different stages and producing more sensible results, deep side outputs generate skeleton lines with nonuniform widths and bring out quite a number of breakpoints inevitably.

Multi-Loss Learning under Multi-Scale Skeleton

Since the side outputs are attached to different convolution layers on the identical 1-stream net, (Shen et al. 2016) incorporate a weighted-fusion output layer that connects to all side-output layers. The key characteristic of the standard multi-loss architecture is that each side-output map should drive a loss function to optimize the networks.

Here are some reasons why researchers adopt multi-loss learning in similar tasks: (1) different side outputs trace back to different receptive field sizes and scales, thus each scale generates a relevant skeleton map. Furthermore, the fusion operation of the single-scale skeletons are also designed to be a learnable convolutional operation; (2) different side outputs are learned from features with different levels. Our goal is learning effective features from which it is easy to capture skeletons. Low-level features from shallow convolution layers usually bring out skeletons of smooth strokes without bending segmentations, i.e., the trends of these strokes change weakly. High-level features from deep convolution layers work on extracting skeletons at the places where the trends of strokes change dramatically, such as junctions, intersections and inflections in strokes.

Domain-Relative Initialization

The performances of almost all FCNs are limited by the lackness of training data (Dai et al. 2016). So, researchers cast pre-trained deep classifiers into FCNs (Long, Shelhamer, and Darrell 2015) and fine-tune them. Due to the usage of the fully connected layers, deep classifiers only accept input samples with fixed-sizes and calculate confidences for each category. Thus, casting CNNs into FCNs by removing fully connected layers provides a effective method for fine-tuning networks without the limit of input size (Long, Shelhamer, and Darrell 2015).

Domain-relative initialization indicates that the FCN-based tasks share the same feature spaces with their homologous CNNs. For segmenting generic objects out of natural images, (Long, Shelhamer, and Darrell 2015) initializes FCNs with VGG net (Simonyan and Zisserman 2014) which was previously trained on large-scale generic object images. Pre-trained VGG net also works well in extracting contours (Xie and Tu 2015) or skeletons (Shen et al. 2016) of generic objects. Similarly, we pre-train deep character recognition models for our character skeleton extraction task.

Methodology

The proposed character skeleton extraction method diagrammed in Fig. 1 consists of 4 major parts: the 1-stream convolutional layers act as feature extractor; regressive dense upsampling convolution (rDUC) extends feature maps without interpolation; scale-associated side outputs contribute to predicting skeletons at multiple scales; and multi-rate dilated fusion (MDF) fuse all candidate skeleton maps into a final results. Besides, we design a concise and effective postprocessing method to obtain pure skeletons.

Network Architecture

The training and testing stages of the proposed network are detailed in Fig. 1, where the convolutional feature extractor are initialized by the pre-trained model in Fig. 7. The feature extractor stretches into 4 groups of side outputs (in the solid box, derived from conv2, conv3, conv4, and conv5) connected to subsequent layers. Side outputs go through two branches: (1) rDUC applies convolutional operations directly on each side output to get a pixel-wise prediction at the corresponding scale. (2) Regular upsampling expands the size of side-output feature maps to the same as input images.

Then the slicing and concatenating operations divide all available feature maps into 5 groups. When features at different scales contribute to the final performance all alone like (Xie and Tu 2015), the skeletonization performance is relatively poor. Therefore all groups should have concatenated features from all scales and generate candidate skeleton maps respectively (Shen et al. 2016). Dilated convolution plays a key role in fusing candidate skeleton maps and predictions of rDUC. In order to drive the side outputs to capture better features at all scales, as the dashed box shows

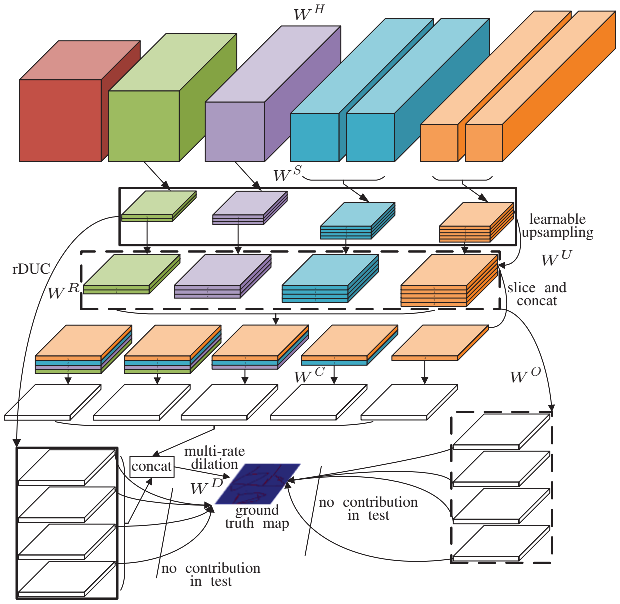


Figure 1: The proposed character skeleton extraction network.

in Fig. 1, the results of regular upsampling are also exploited to generate the predictions of character skeletons.

Regressive Dense Upsampling Convolution (rDUC) DUC was described in (Wang et al. 2017) for semantic segmentation. Suppose that the size of a input image is $C \times H \times W$ ($channels \times height \times width$), and we have the feature maps F_M with size $c \times h \times w$ from the final convolution layer (the downsampling factor $r = \frac{H}{h} = \frac{W}{w}$). F_M is used to draw a label map with size $H \times W$ where each pixel is predicted with a category label. N_c is the total number of different categories. DUC performs convolution on F_M to get a feature map F_M^D with size $((r^2 \times N_c) \times h \times w)$. Finally, F_M^D is reshaped to predicted map with size $(N_c \times H \times W)$. Our final goal is classifying pixels into 2 categories (skeleton/non-skeleton points), thus we propose the regressive dense upsampling convolution as shown in Fig. 2.

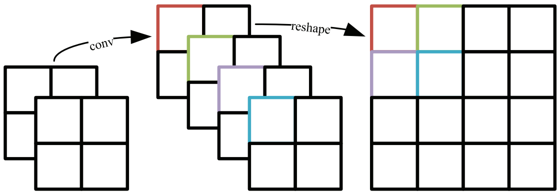
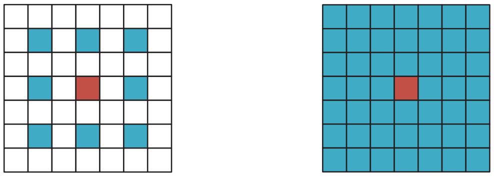


Figure 2: The computation of regressive dense upsampling convolution (rDUC).

The left-most feature maps F_l with the size $(2 \times 2 \times 2)$ in Fig. 2 are the outputs of a convolutional feature extractor, and they are used to generate a $(1 \times 4 \times 4)$ prediction. In FCNs, this process is done by bilinear upsampling (Long, Shelhamer, and Darrell 2015). Due to the unlearnable property and insufficient amount of parameters, bilinear upsampling loses lots of valuable information. In rDUC, we directly conduct convolution operation on F_l to attain the middle feature maps F_m with size $(4 \times 2 \times 2)$. Following the rules indicated by different colors in Fig. 2, we reshape F_m to the final prediction P_r with size $(1 \times 4 \times 4)$.

Though named "sampling", rDUC is a learnable process without interpolation. Moreover, it is capable of capturing fine-detailed features, which are easily missed in the bilinear interpolation. In comparison with DUC, rDUC reduces the number of parameters and has better generalization. The reason is that it has no concern with variant classes. In our task, rDUC contributes a lot to eliminating the breakpoints in character skeletons.

Multi-Rate Dilated Fusion (MDF) Dilated convolution is used to maintain high resolution of feature maps in FCNs through replacing the max-pooling operation or convolution layer (Wang et al. 2017), and it can also extend the receptive field. But single-rate dilated convolution always cause "gridding" problems as Fig. 3(a) (Wang et al. 2017) shows.



(a) Single rate fusion.

(b) Multi-rate dilated fusion.

Figure 3: Comparison of multi/single-rate dilation.

Fig. 3 displays 2 feature maps: the blue pixels mean that they have contribution to the calculation of the central pixel (marked in red). In Fig. 3(a), all kernels are 3×3 with single dilate rate, and many pixels at fixed positions contribute nothing. The pixel-wise utilization of HDC is shown in Fig. 3(b), where each pixel is in full use because of multiple dilated rates $1, 2, 3$.

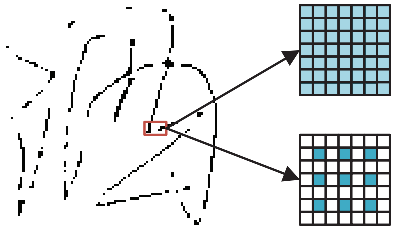


Figure 4: The role of HDC. HDC covers larger context. In the red box, 3×3 kernel with single-rate dilation may cover 9 background points but 0 foreground ones. But the multi-rate kernels concern much more pixels, and are more likely to connect broken lines.

In our task, large convolutional kernels ($\geq 5 \times 5$) neglect details, while the respective fields of 3×3 kernels are narrow. Especially when extracting the character skeletons, we need small kernels to predict detailed skeleton points, and large kernels to capture the holistic shapes of handwritten strokes.

Dilated convolutions provide respective fields with different sizes without extra parameters. Furthermore, the foreground points of predicted skeleton maps are far less than the number of background points, HDC fully utilizes all foreground pixels as shown in Fig. 4. The HDC module in our task are shown in Fig. 5, where dilated convolutions contain multiple rates from 1 to 4. The summation of 4 feature maps from dilated convolution kernels is delivered to a fusing convolution layer for the final prediction.

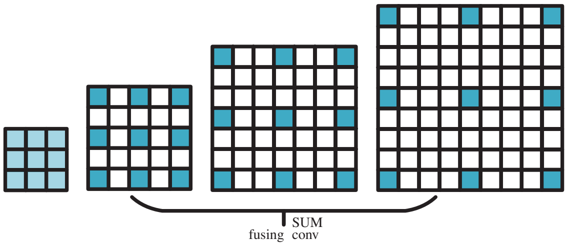


Figure 5: The HDC module in our task. Blue points mean the non-zero parameters of one kernel, and we insert blanks between them.

Multi-Loss Learning

(X_i, Y_i) denotes one training pair, where X_i refers the input sample and Y_i expresses the ground truth skeleton of X_i. A typical instance of (X_i, Y_i) is (Fig. 6(b), Fig. 6(a)). Predictions and ground truth maps are used to calculate **S**igmoid **C**ross **E**ntropy loss L^{SCE}.

Besides fine-tuning the weights W^H inherited from the pre-trained HCCR-CNN9Layer, we need to learn following parameters: (1) Side outputs come from K different scales, and the weights for working out k-th side output are denoted as W_k^S. (2) Unlike the unlearnable bilinear filters in (Long, Shelhamer, and Darrell 2015; Shen et al. 2016), our kernels W_k^U of k-th upsampling layer are learnable, which we will prove to be superior to the unlearnable case in Fig. 10. (3) Suppose after the slicing and concatenating, there are G groups of feature maps. The convolutional kernel W_g^C works out the g-th candidate skeleton map, and the predictions boxed by dashed lines in Fig. 1 are related to W_k^O. (4) The proposed rDUC also contains convolutional calculations, and W_k^R denotes the weights for the k-th side output. (5) For convenience, we denote all kernels in multi-rate dilated convolution by W^D.

From the perspective of the final outputs, P_k^U, P_k^R, and P^F represents the k-th prediction generated by the regular upsampling results, the k-th prediction generated by the rDUC results, and the final fused prediction, respectively. Given the above, we define the fusion loss as

$$L_f = L^{SCE}(P^F, Y_i | W^H, W^S, W^U, W^C, W^O, W^R, W^D), \tag{1}$$

where $W^H \sim W^D$ are the weights which will be updated by minimizing L_f. As for the k-th prediction generated by the regular upsampling results, the objective function caused by P_k^U is

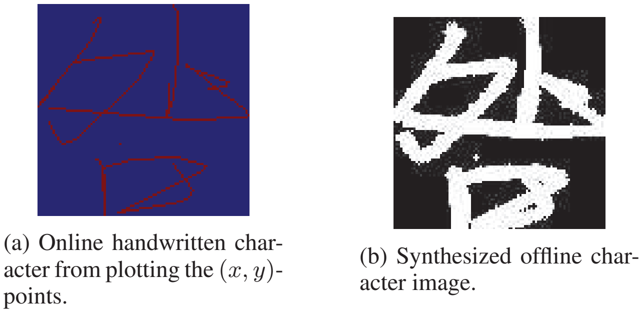
$$L_s = L^{SCE}(P_k^U, Y_i | W_k^H, W_k^S, W_k^U, W_k^O). \tag{2}$$

Similarly, we consider the loss refers to P_k^R as

$$L_r = L^{SCE}(P_k^R, Y_i | W_k^H, W_k^S, W_k^R). \tag{3}$$

Training Data Preparation Supervised deep learning demands plenty of ground-truthed skeletons of offline handwritten Chinese characters. It is impossible to human-annotate ground truth skeletons for a large number of samples. Fortunately, the online handwritten samples (Liu et al. 2011a) record strokes by (x, y)-coordinate sequences, which can be viewed as the ideal skeletons of synthesized images (generated by dilating the stroke skeletons). Thus we generate a large number of synthesized character images from online handwritten characters for training.

As shown in Fig. 6(a), the online character is plotted by connecting the sequential (x, y)-points recorded for pen trajectory in writing. The offline image Fig. 6(b) is generated by dilating the strokes of plotted online character image with appropriate control of stroke width, edge smoothness and foreground gray scale. Specifically, we obtain realistic gray-scale images by controlling the image qualities of the whole images in JPG format. It is clear that our synthetic image is much better than the one which is processed by Gaussian noises or pseudo-gray means.



(a) Online handwritten character from plotting the (x, y)-points.

(b) Synthesized offline character image.

Figure 6: Conversion from online handwritten character to offline image.

We train deep HCCR model by samples like Fig. 6(b) to provide foundational parameters for our FCNs, and we will illustrate the necessity of pre-training in Fig. 12. Obviously, Fig. 6(a) is quite qualified for serving as the ground truth skeleton of Fig. 6(b), and provides accurate supervision.

Pre-trained Model As shown in Fig. 7, the HCCR-CNN9Layer (Xiao et al. 2017) reaches excellent performance, and its similar architecture with VGG net (Simonyan and Zisserman 2014) makes it easy to cast into FCNs. Therefore we use the convolutional parameters of HCCR-CNN9Layer to initialize our FCNs and fine-tune all layers.

Because the character skeleton extraction network should share the same inputs with its corresponding deep domain-relative HCCR model, we use the universal set of online Handwritten database CASIA-OLHWDB1.1 (Liu et al. 2011a) to synthesize training data (∼1.121 millions)

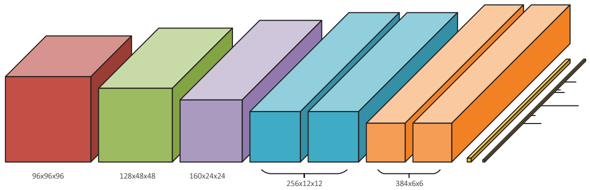


Figure 7: The Architecture of HCCR-CNN9Layer. Different colors indicate different groups of conv layers. Each convolutional layer works with 3×3-kernels, 1-stride and 1-pad. We employ SGD with learning rate 10^{-1}, momentum 0.9, weight decay 2×10^{-4}, and the learning rate is dropped by $\times 0.1$ per 3.5 epochs. Here we omit the 2×2-max-pooling and batch normalization layers after each convolutional group.

like Fig. 6(b). We test the pre-trained model on a random subset (~230 thousands) of the training part of CASIA-OLHWDB1.0 (Liu et al. 2011a).

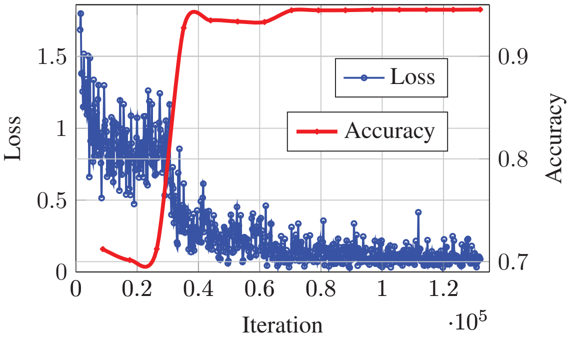
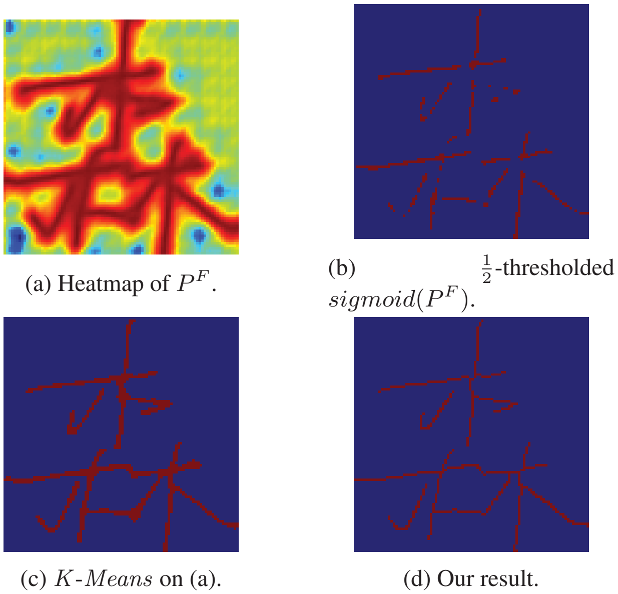


Figure 8: The pre-training process of HCCR-CNN9Layer.

As shown in Fig. 8, during the early stages of training, the model exhibits unstable state. With the periodic drop of learning rate, there is a significant accuracy improvement for testing set. Therefore we stop training after 15 epochs (94.53% accuracy) instead of running more epochs to achieve higher performance.

Postprocessing

The postprocessing in similar tasks (Xie and Tu 2015; Shen et al. 2016) is very simple: employing $\frac{1}{2}$-thresholded binarization on $sigmoid(P^F)$. However, this method only captures a coarse result like Fig. 9(b), where there are lots of breakpoints, and most segmentations are not one-pixel width. In the heatmap of P^F in Fig. 9(a), it is easy to distinguish the mainlines of objective skeletons (in deep red) from non-skeleton pixels (in bright color). Therefore we conduct K-$Means$ on Fig. 9(a) (set K as 2) and generate a binary image Fig. 9(c), where there are no unsatisfactory breakpoints. Finally, rules from (Zhang and Suen 1984) clear the redundant foreground points and generate ideal skeleton as Fig. 9(d) shows.



(a) Heatmap of P^F.

(b) $\frac{1}{2}$-thresholded $sigmoid(P^F)$.

(c) K-$Means$ on (a).

(d) Our result.

Figure 9: The results of our postprocessing method.

Experiments and Results

Experiments on Skeleton Extraction Tasks

We use the framework in Fig. 1 without rDUC as our baseline model, which obeys the widely accepted construction standards introduced by (Shen et al. 2016). We replace the MDF with the general convolutional fusion in the baseline model. Besides, the bilinear upsampling operations in the baseline model are unlearnable.

We synthesize the pair-wise training data as (Fig. 6(a), Fig. 6(b)) from CASIA-OLHWDB1.1 (~1.121 millions) for the training and test our models on the synthesized data (~224 thousands) from ICDAR-2013 Online HCCR Competition Database (Yin et al. 2013). The performances of skeleton extraction methods are measured by the F-measure ($\frac{2 \times Precision \times Recall}{Precision + Recall}$) (Shen et al. 2016). Because the F-measures cannot give visual descriptions about the performances of different models, we propose a new metric named Average Minimal Distance (AMD) to evaluate our methods. P_s and P_g express the sets of skeleton points in the predicted maps and ground truth map, respectively. Each item d_{ij} in matrix $D^{|P_s| \times |P_g|}$ indicates the distance between P_s^i and P_g^j. Therefore, we have:

$$AMD = average(H(D)), \tag{4}$$

where H is the Hungarian algorithm[1] designed to solve the linear sum assignment problems.

We conduct our experiments on 4 models: baseline, baseline + learnable upsampling, baseline + learnable upsampling + MDF, and baseline + learnable upsampling + MDF + rDUC, and 3 different input sizes: 96×96, 64×64, and 48×48. The L_f curves with different configurations and the consistent input size 96×96 are present in Fig. 10 and

[1] https://en.wikipedia.org/wiki/Hungarian_algorithm

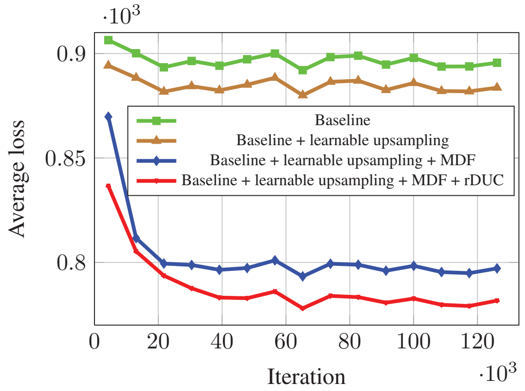


Figure 10: The average loss of each epoch.

Fig. 11. The average losses of each epoch in Fig. 10 shows the combination of baseline + learnable upsampling + MDF + rDUC exceeds all other models. Moreover, the curves in Fig. 10 indicate that all of the learnable upsampling, MDF, and rDUC are conducive to optimizing our networks.

In Fig. 11, the black curve tells the history of L_f when we train the baseline net without inheriting W^H from the pre-trained model, and the green one means the fine-tuning phase on the basis of W^H. Obviously, even if we have > 1 million pairs of training samples, domain-relative initialization by pre-trained HCCR-CNN9Layer is indispensable.

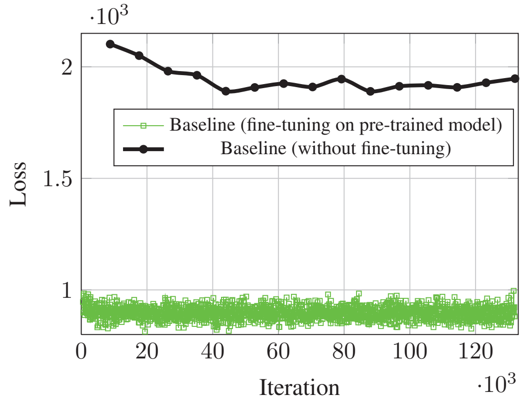


Figure 11: The necessity of pre-training.

We report the F-measure in Table. 1, where our models outperforms others significantly. In our experiments, ZhangSuen algorithm works better than other traditional methods, such as stroke correction (Pujari, Mitra, and Mishra 2014) and stroke continuity (Dong et al. 2017). Distance-based methods are suitable for patterns with simple shapes and smooth contours, but fail to present comparable outputs in our task. HED network reports a much better AMD than traditional methods, but its recall is not ideal. Though the skeletons extracted by our models are not fit the ground truth perfectly, they look very natural, that means they can be read by humans and recognized by machines. AMDs in Table. 1 strongly demonstrate that in our methods, the predicted skeleton points are closer to the ground truth points.

Table 1: Comparison between different models on synthesized data from ICDAR-2013 Online HCCR Competition Database. Image size: 96 × 96.

Method	F-measure	AMD
Stroke Correction (Pujari, Mitra, and Mishra 2014)	0.215	3.87
Stroke Continuity (Dong et al. 2017)	0.349	3.45
ZhangSuen (Zhang and Suen 1984)	0.381	3.23
HED (Xie and Tu 2015)	0.373	1.78
Baseline	0.575	1.51
Baseline + learnable upsampling	0.592	1.44
Baseline + learnable upsampling + MDF	0.597	1.46
Baseline + learnable upsampling + MDF + rDUC	**0.610**	**1.29**

Experiments on Recognition Tasks

We illustrate the character skeleton extraction results obtained by different methods in Fig. 12. These examples show that our method has detected the most ground truth points. Obviously, our results are more smooth, and contain fewer noisy points. Above all, our results are as human-readable as the raw images and ground truths. Though our models are trained on synthetic data, they can handle the real offline handwritten Chinese character samples in CASIA-OFFHWDB1.1 (Liu et al. 2011a), and extract resultful skeletons in Fig. 13.

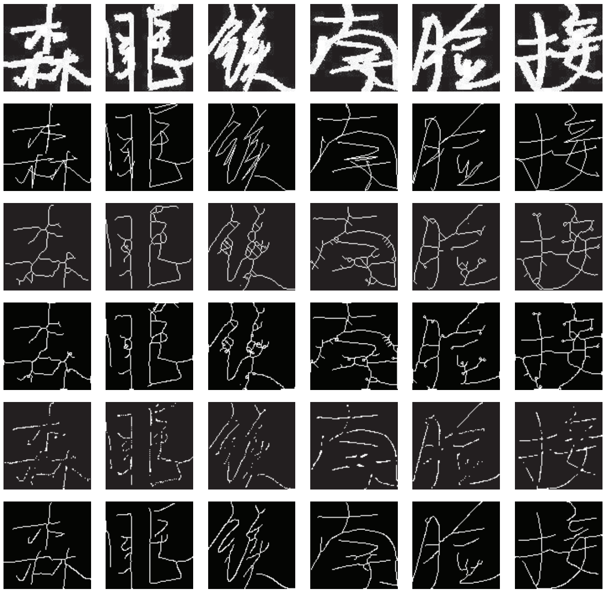


Figure 12: The character skeletons attained by different methods on synthesized data from ICDAR-2013 Online HCCR Competition Database. From up to down: input images; ground truths; stroke continuity based algorithm; ZhangSuen method; HED network, and ours. Image size: 96 × 96.

It is clear that skeletons of real/synthesized offline characters are human-readable. Thus, we recognize skeletons directly in classification tasks. We use HCCR-CNN9Layer as our classifier and conduct experiments on two sides: (1) Training models directly on character skeletons. (2) Training models on the 2-channel inputs, where the raw images and skeletons occupy one input channels separately. From Table. 2, we can see that when only recognizing skeletons, the best accuracy 95.53% is 1.28% lower than the state of the art 96.81% achieved by training on the 96×96 raw images. Nevertheless, combining raw image and skeleton as classifier input reports the best performance 96.90% when only training on CASIA-OFFHWDB1.1, this result is comparable with the best accuracy 96.95%. Moreover, in Table. 2, the low accuracy 69.94% for skeletons generated by Zhang-Suen algorithm indicates that traditional thinning algorithms do not preserve character shapes well.

Table 2: Testing accuracies on ICDAR-2013 Offline HCCR Competition Database. All models are trained on offline dataset CASIA-OFFHWDB1.1.

Method	Training Data	Input	Image Size	Accuracy
(Liu et al. 2011b)	1.1	Raw	48×48	92.18%
(Yin et al. 2013)	1.1	Raw	48×48	94.77%
(Cireşan and Meier 2015)	1.1	Raw	48×48	95.79%
(Wu et al. 2014)	1.1	Raw	64×64	96.06%
HCCR-CNN9Layer	1.1	Raw	96×96	**96.81%**
	1.1	Skeleton (Ours)	96×96	94.34%
	1.1	Skeleton (Ours)	64×64	**95.53%**
	1.1	Skeleton (Ours)	48×48	94.66%
	1.1	Skeleton (ZhangSuen)	64×64	69.94%
	1.1	Raw + Skeleton (Ours)	96×96	**96.90%**
	1.1	Raw + Skeleton (Ours)	64×64	96.63%
	1.1	Raw + Skeleton (Ours)	48×48	96.10%
(Zhang, Bengio, and Liu 2017)	1.1 + 1.0	DirectMap	$8 \times 32 \times 32$	**96.95%**

Though only recognizing skeletons cannot reach the best performance, we can not ignore the structural invariance contained in skeletons. The changes of foreground/background colors, stroke widths, and image qualities will cause the performances degradation on models trained on the raw images, but the recognitions of skeleton are robust to these kinds of variants. As Fig. 6(b) shows, we synthesize two offline datasets from two online datasets: ICDAR-2013 Online HCCR Competition database and CASIA-OLHWDB1.1TST (the testing set of CASIA-OLHWDB1.1 (Liu et al. 2011a)), which are named ON2OFF-Comp2013 and ON2OFF-HWDB1.1TST.

In Table. 3, we present the accuracies under 4 kinds of different inputs: raw images, binary images, directMaps (Liu 2007), and skeletons. On the synthesized datasets, the classifiers trained on the raw data of CASIA-OFFHWDB1.1 drop by >6%. When training and testing on binary images or directMaps, models perform better. However, models trained on binary samples or directMaps cannot reach higher accuracies because the stroke widths in testing sets are various while the stroke width in CASIA-OFFHWDB1.1 have smaller variance.

Table 3: Testing accuracies on synthesized datasets. All models are trained on the real offline datasets CASIA-OFFHWDB1.1.

Method	Input	Image Size	ON2OFF-Comp2013	ON2OFF-HWDB1.1TST
HCCR-CNN9Layer	Raw	96×96	89.27%	89.41%
	Binary	96×96	88.37%	89.65%
(Zhang, Bengio, and Liu 2017)	DirectMap (Raw)	$8 \times 32 \times 32$	90.14%	90.73%
	DirectMap (Binary)	$8 \times 32 \times 32$	90.40%	91.23%
HCCR-CNN9Layer	Skeleton (ZhangSuen)	64×64	70.26%	70.44%
	Skeleton (Ours)	64×64	**94.42%**	**94.53%**

In Table. 3 the models trained on skeleton still report comparable performances with the result in Table. 2, and this can be easily explained. A real offline sample is shown in Fig. 14(a), and a typical synthesized sample is just like Fig. 14(c).

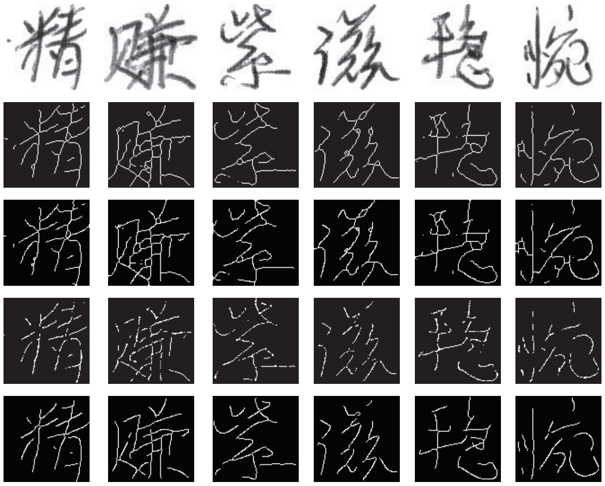
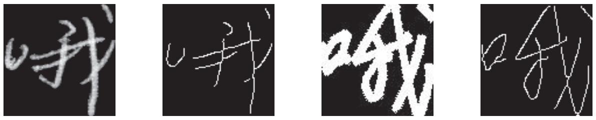


Figure 13: The character skeleton extraction results on the real offline data. From up to down: input images; stroke continuity based algorithm; ZhangSuen method; HED network, and ours. Image size: 96×96.



(a) Offline sample.

(b) Skeleton of (a).

(c) Synthesized data.

(d) Skeleton of (c).

Figure 14: The different forms of offline and synthesized data.

Though they are machine-readable, the pixel-wise distributions of synthesized data are totally different from the real offline one. Hence, even if under the gray normalization, the models trained on the real offline data fall down evidently on synthesized data. By contrast, the directMaps can capture more invariant features for recognition tasks. In the skeleton based recognition task, we recognize the extracted skeletons of input samples instead of the raw images. Though there exists different distributions in the raw data and the synthesized data, the skeletons hold enough discriminative and invariant clues for robust recognition, i.e., Fig. 14(b) and Fig. 14(d) are the skeletons of Fig. 14(a) and Fig. 14(c) respectively, and models trained on samples like 14(b) are capable to understand samples like Fig. 14(d).

To show the usefulness of skeletons extracted from character images synthesized from online handwritten characters, we also evaluate the recognition accuracies on real offline test data using models trained with synthesized samples in ON2OFF-HWDB1.1 (synthesized by the universal set of the online data CASIA-OLHWDB1.1). The results are shown in Table. 4, where we can see that the accuracies are comparable with the ones in Table. 3. This implies means the online samples can fulfill data argumentation in offline recognition tasks.

Table 4: Testing accuracy on ICDAR-2013 Offline HCCR Competition Database. All models are trained on the synthesized offline data ON2OFF-HWDB1.1.

Method	Input	Image Size	Accuracy
HCCR-CNN9Layer	Raw	96 × 96	89.05%
	Binary	96 × 96	89.42%
(Zhang, Bengio, and Liu 2017)	DirectMap (Raw)	8 × 32 × 32	90.56%
	DirectMap (Binary)	8 × 32 × 32	91.20%
HCCR-CNN9Layer	Skeleton (ZhangSuen)	64 × 64	69.77%
	Skeleton (Ours)	64 × 64	**94.51%**

Conclusion

We propose a fully convolutional network with multi-loss learning to extract skeletons for handwritten Chinese characters. By combining the standard side-output architecture with the regressive dense upsampling convolution (rDUC) and multi-rate dilated fusion (MDF), we achieve high F-measure in skeleton pixel detection. Our experimental results of skeleton-based character recognition using CNNs demonstrate that the skeletons extracted using the proposed method preserves character shapes very well. In the future, we will study into structural shape analysis, matching and interpretation based on character skeletonization.

Acknowledgments

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

References

Alghamdi, M. A., and Teahan, W. J. 2017. A new thinning algorithm for Arabic script. *International Journal of Computer Science and Information Security* 15(1):204.

Arcelli, C., and Di Baja, G. S. 1985. A width-independent fast thinning algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 7(4):463–474.

Arcelli, C., and Di Baja, G. S. 1989. A one-pass two-operation process to detect the skeletal pixels on the 4-distance transform. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 11(4):411–414.

Cireşan, D., and Meier, U. 2015. Multi-column deep neural networks for offline handwritten Chinese character classification. In *International Joint Conference on Neural Networks*, 1–6.

Dai, J.; He, K.; Li, Y.; Ren, S.; and Sun, J. 2016. Instance-sensitive fully convolutional networks. In *European Conference on Computer Vision*, 534–549.

Dong, J.; Chen, Y.; Yang, Z.; and Ling, B. W.-K. 2017. A parallel thinning algorithm based on stroke continuity detection. *Signal, Image and Video Processing* 11(5):873–879.

Dong, J.; Lin, W.; and Huang, C. 2016. An improved parallel thinning algorithm. In *International Conference on Wavelet Analysis and Pattern Recognition*, 162–167.

Hsiung, H.-Y.; Chang, Y.-L.; Chen, H.-C.; and Sung, Y.-T. 2017. Effect of stroke-order learning and handwriting exercises on recognizing and writing Chinese characters by Chinese as a foreign language learners. *Computers in Human Behavior* 74:303–310.

Liu, C.-L.; Yin, F.; Wang, D.-H.; and Wang, Q.-F. 2011a. CASIA online and offline Chinese handwriting databases. In *International Conference on Document Analysis and Recognition*, 37–41.

Liu, C.-L.; Yin, F.; Wang, Q.-F.; and Wang, D.-H. 2011b. ICDAR 2011 Chinese handwriting recognition competition. In *International Conference on Document Analysis and Recognition*, 1464–1469.

Liu, C.-L. 2007. Normalization-cooperated gradient feature extraction for handwritten character recognition. *IEEE transactions on Pattern Analysis and Machine Intelligence* 29(8):1465–1469.

Long, J.; Shelhamer, E.; and Darrell, T. 2015. Fully convolutional networks for semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 3431–3440.

Pritchard, J. D., and Sondak, N. E. 1973. Automatic recognition of handwritten characters using structural features. In *ACM Annual Conference*, 442–2.

Pujari, A. K.; Mitra, C.; and Mishra, S. 2014. A new parallel thinning algorithm with stroke correction for Odia characters. In *Advanced Computing, Networking and Informatics*, volume 1. Springer. 413–419.

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

Simonyan, K., and Zisserman, A. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Wang, P.; Chen, P.; Yuan, Y.; Liu, D.; Huang, Z.; Hou, X.; and Cottrell, G. 2017. Understanding convolution for semantic segmentation. *arXiv preprint arXiv:1702.08502*.

Wu, C.; Fan, W.; He, Y.; Sun, J.; and Naoi, S. 2014. Handwritten character recognition by alternately trained relaxation convolutional neural network. In *International Conference on Frontiers in Handwriting Recognition*, 291–296.

Xiao, X.; Jin, L.; Yang, Y.; Yang, W.; Sun, J.; and Chang, T. 2017. Building fast and compact convolutional neural networks for offline handwritten Chinese character recognition. *arXiv preprint arXiv:1702.07975*.

Xie, S., and Tu, Z. 2015. Holistically-nested edge detection. In *IEEE International Conference on Computer Vision*, 1395–1403.

Yin, F.; Wang, Q.-F.; Zhang, X.-Y.; and Liu, C.-L. 2013. ICDAR 2013 Chinese handwriting recognition competition. In *International Conference on Document Analysis and Recognition*, 1464–1470.

Zhang, T., and Suen, C. Y. 1984. A fast parallel algorithm for thinning digital patterns. *Communications of the ACM* 27(3):236–239.

Zhang, X.-Y.; Bengio, Y.; and Liu, C.-L. 2017. Online and offline handwritten Chinese character recognition: A comprehensive study and new benchmark. *Pattern Recognition* 61:348–360.

Zou, J. J., and Yan, H. 2001. Skeletonization of ribbon-like shapes based on regularity and singularity analyses. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 31(3):401–407.