Chinese Character Recognition with Augmented Character Profile Matching

Xinyan Zu
Shanghai Key Laboratory of IIP
School of Computer Science, Fudan University
Shanghai, China
xyzu20@fudan.edu.cn

Haiyang Yu
Shanghai Key Laboratory of IIP
School of Computer Science, Fudan University
Shanghai, China
hyyu20@fudan.edu.cn

Bin Li*
Shanghai Key Laboratory of IIP
School of Computer Science, Fudan University
Shanghai, China
libin@fudan.edu.cn

Xiangyang Xue*
Shanghai Key Laboratory of IIP
School of Computer Science, Fudan University
Shanghai, China
xyxue@fudan.edu.cn

ABSTRACT

Chinese character recognition (CCR) has drawn continuous research interest due to its wide applications. After decades of study, there still exist several challenges, *e.g.*, different characters with similar appearance and the one-to-many problem. There is no unified solution to the above challenges as previous methods tend to address these problems separately. In this paper, we propose a Chinese character recognition method named Augmented Character Profile Matching (ACPM), which utilizes a collection of character knowledge from three decomposition levels to recognize Chinese characters. Specifically, the feature maps of each character image are utilized as the character-level knowledge. In addition, we introduce a radical-stroke counting module (RSC) to help produce augmented character profiles, including the number of radicals, the number of strokes, and the total length of strokes, which characterize the character more comprehensively. The feature maps of the character image and the outputs of the RSC module are collected to constitute a character profile for selecting the closest candidate character through joint matching. The experimental results show that the proposed method outperforms the state-of-the-art methods on both the ICDAR 2013 and CTW datasets by 0.35% and 2.23%, respectively. Moreover, it also clearly outperforms the compared methods in the zero-shot settings. Code is available at https://github.com/FudanVI/FudanOCR/character-profile-matching.

CCS CONCEPTS

• **Computing methodologies → Computer vision tasks**; **Object recognition**.

KEYWORDS

Chinese character recognition, OCR, character profile matching, Chinese character knowledge

ACM Reference Format:
Xinyan Zu, Haiyang Yu, Bin Li, and Xiangyang Xue. 2022. Chinese Character Recognition with Augmented Character Profile Matching. In *Proceedings of the 30th ACM International Conference on Multimedia (MM '22), Oct. 10–14, 2022, Lisboa, Portugal.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3503161.3547827

1 INTRODUCTION

As one of the most widespread languages, Chinese has approximately 16% of the world's population as its native speakers. Acting as an essential role in downstream vision applications (*e.g.*, vehicle license plate recognition [23], autonomous driving [27], etc), Chinese character recognition (CCR) enjoys extensive research interest in the last decades. Traditional CCR methods usually rely on handcrafted features (*e.g.*, HOG features and Gabor features) to match the Chinese character in a pixel-wise manner, which is vulnerable to challenging situations such as scribbled handwriting characters, varying fonts, and complicated real-world backgrounds.

With the rapid development of deep learning, modern CCR methods have been proposed to solve the above challenges by extracting more robust representations with convolutional neural networks (CNNs). According to the decomposition level of a Chinese character, these deep learning based methods can be roughly categorized into three approaches: character-based [6, 13, 20, 22], radical-based [1, 18, 19, 21], and stroke-based [3, 4]. Despite that these CCR methods have achieved considerable accuracy improvement on challenging handwritten and scene text datasets, there still exist several challenges harming the performance. Generally speaking, the character-based methods are very likely to suffer from the problem of similar characters while the radical-based and stroke-based methods have difficulties in dealing with the one-to-many problem and predicting the number of radicals or strokes. We observe that some of the existing approaches tend to address the aforementioned challenges separately, which motivates us to explore a unified solution for dealing with these challenges.

In this paper, we propose a Chinese character recognition method named Augmented Character Profile Matching (ACPM), which considers combining character knowledge from three decomposition levels (*i.e.*, character-level, radical-level, and stroke-level) as comprehensive character profiles to predict Chinese characters. As

[*] Corresponding author

MM '22, October 10–14, 2022, Lisboa, Portugal.

ACM ISBN 978-1-4503-9203-7/22/10…$15.00
https://doi.org/10.1145/3503161.3547827

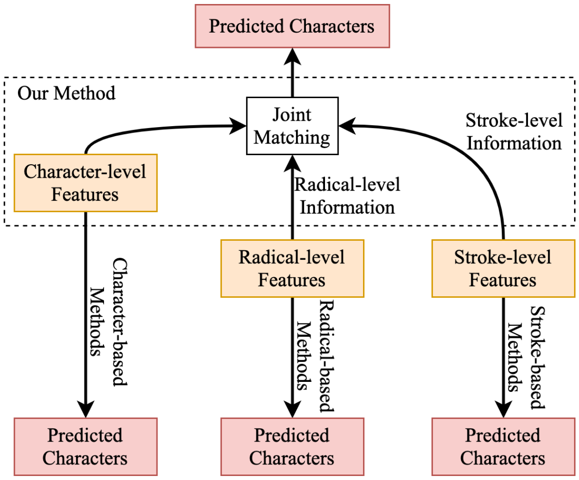

Figure 1: Illustration of the proposed CCR method with augmented character profile matching.

shown in Figure 1, at the character level, the feature maps of the input character images are viewed as the basic character profile. For radical and stroke levels, we introduce a radical-stroke counting module (RSC) into the radical-based framework to help produce an augmented character profile, including the number of radicals, the number of strokes, and the total length of strokes. In inference, all the predictions, including the feature maps of the input character image and the outputs of the RSC module, are collected as a comprehensive character profile to characterize the input image for selecting the closest candidate character through joint matching. Extensive experiments are conducted to validate the effectiveness of the proposed CCR method. The experimental results demonstrate that ACPM outperforms the state-of-the-art methods on both the ICDAR 2013 and CTW datasets. Moreover, ACPM also clearly outperforms the compared methods in the zero-shot settings. To summarize, the contributions of this paper are three-fold:

- We propose a CCR method with augmented character profile matching to simultaneously tackle the similar-appearance and one-to-many problems.
- We propose a radical-stroke counting (RSC) module to produce an augmented Chinese character profile for selecting the closest candidate character.
- The proposed method outperforms the state-of-the-art methods by a margin (0.35%) on the handwritten dataset ICDAR 2013 and by a clear margin (2.23%) on the scene text dataset CTW.

2 RELATED WORKS

In this section, we provide a brief review of CCR methods focusing on different decomposition levels, *i.e.*, character-based, radical-based, and stroke-based methods.

2.1 Character-based Methods

Early methods [2, 9, 15] usually adopt manually crafted features for CCR. However, the performance of these traditional methods is believed to reach their limitation due to the low capacity of representation spaces [5].

The era of deep learning allows for extracting robust image features with CNNs. MCDNN [6] is the first attempt to solve CCR with deep learning, which ensembles eight models trained separately and achieve human-level performance on CCR tasks for the first time. It is followed by ART-CNN [20], which leverages relaxation CNNs to further enhance feature representations. Considering the domain-specific knowledge, DirectMap [13] combines various traditional features with modern methods. Recently, [22] proposes a template-instance loss to alleviate the imbalance between easy and difficult character samples in datasets. However, the character-based methods are easily confused by similar characters.

2.2 Radical-based Methods

Before deep learning era, some methods use traditional strategies to extract character radicals. In [17], the authors adopt a recursive hierarchical scheme to recognize the radicals based on the strokes extracted in advance, which is inaccessible in handwritten scenarios. To bypass the stroke-to-radical protocol, [14] develops a pixel-wise shape-matching-based algorithm to produce radicals directly.

Recently, massive training data and the development of feature extraction networks also benefit the radical-level CCR methods. DenseRAN [19] first puts forward the paradigm of predicting radicals as a sequence, then translating the radical sequence to a character. Subsequently, inspired by the solution of irregular text recognition in generic text recognition field, STN-DenseRAN [21] employs a spatial transform module to tackle distorted Chinese characters in scene text datasets. FewshotRAN [18] ensembles the radical aggregation network with a character-level classifier. HDE [1] proposes to map the character-embedding space to the embedding of both radicals and radical structures. In this paper, the proposed method follows the fashion of HDE in radical decomposition.

2.3 Stroke-based Methods

Most previous stroke-based methods are old-fashioned. In [10], the authors perform pixel-wise matching for strokes to address the inaccuracy in stroke extraction. In [11], a mathematical morphology is proposed to model the Chinese strokes. Matching and filtering techniques are used in the subsequent methods [12, 15] for better perceiving strokes of Chinese characters.

Recently, a deep learning-based method [3] treats a Chinese character as a decomposition of five categories of strokes, where the encoder-decoder architecture produces the prediction of the stroke sequence for the subsequent matching-based metric to select the candidate character. It is worth mentioning that, different from the standard definition of strokes in [3], the strokes adopted in this paper are 'nonstandard' orientation-based such that only four types of strokes are considered in the proposed method.

3 PRELIMINARIES

According to the national standard GB18030-2005, Chinese characters can be decomposed at the radical or stroke level. At the radical level, twelve basic structures of Chinese characters defined in Unicode are shown in Figure 2 (a). Accordingly, one character can be decomposed into a tree structure as in Figure 2 (b). For the stroke

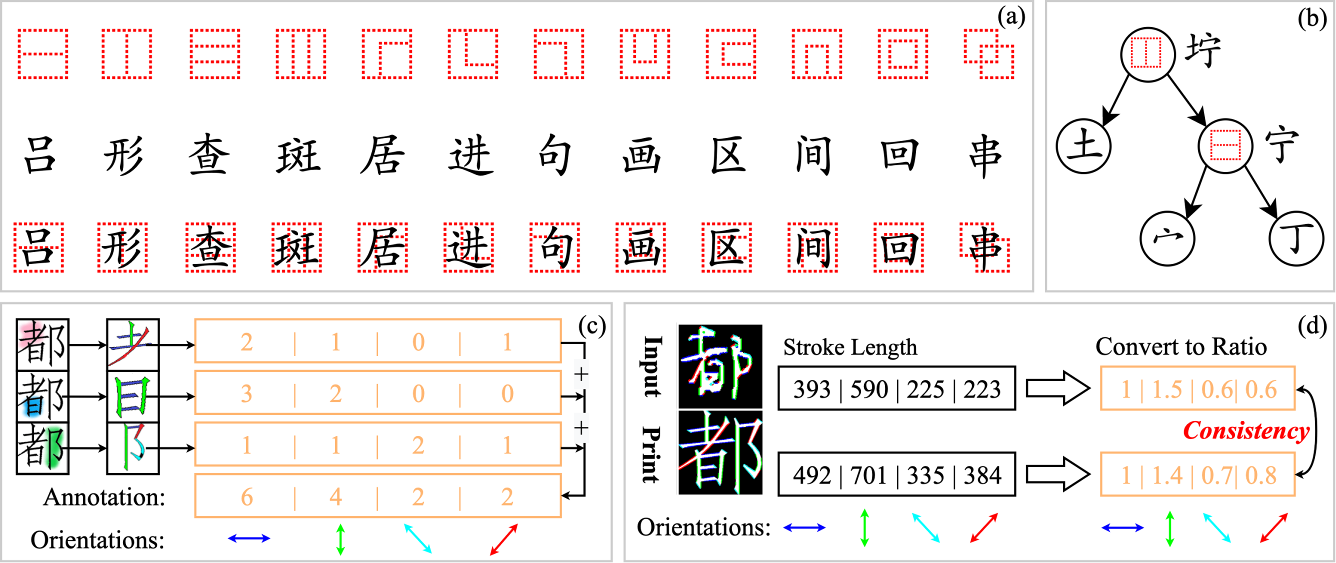

Figure 2: (a) Twelve basic structures of Chinese characters; (b) Tree-structure decomposition; (c) Labeling strokes in four orientations; (d) Observation on the stroke-length ratio consistency between different writing styles of the same character.

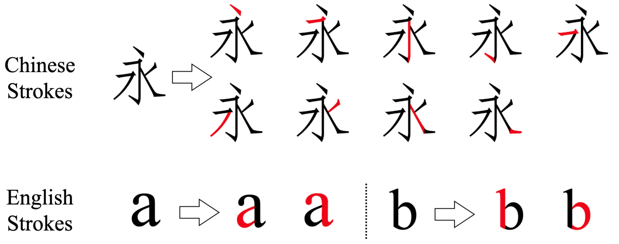

Figure 3: Different from Latin characters which have many arcs, most of strokes in Chinese characters can be simply categorized into four orientations.

level, according to the national standard GB18030-2005, there are five basic strokes, *i.e.*, horizontal, vertical, left-falling, right-falling, and turning. However, different from Latin characters, the strokes in Chinese characters do not contain many arcs as shown in Figure 3, thus the strokes in Chinese characters may be simplified. In this paper, we instead define four types of strokes according to their orientations. Specifically, we simply treat each Chinese character as the composition of strokes in four orientations (*i.e.*, horizontal, vertical, top-left to bottom-right, and bottom-left to top-right) as shown in the bottom row of Figure 2 (c).

In this paper, for comprehensively characterizing a Chinese character, we define the Chinese character profile, including the following four items:

- **Feature Maps**. In some existing methods, the feature maps are used to match the template images to calculate the closest candidate character. Following this manner, we also regard the feature maps of the input character image as the character-level knowledge and treat it as character-level information in the profile.
- **Number of Radicals**. Existing radical-based methods usually directly predict the radical sequence. However, this approach suffers from incorrectly predicted length of sequence. It is beneficial to explicitly employ such knowledge for matching and we thus take the number of radicals into account when constituting the character profile.
- **Number of Strokes**. Strokes are the smallest units for decomposing Chinese characters. We also adopt the number of orientation-based strokes as one aspect of the stroke-level information, which can be obtained by accumulating the numbers of the orientation-based strokes contained in individual radicals of a Chinese character. The mapping between a radical and the corresponding number of orientation-based strokes can be manually defined at a very low labeling cost.
- **Total Length of Strokes**. Serving as another aspect of the stroke-level information, for a Chinese character, the total lengths of strokes in four orientations are integrated into the character profile based on the observation as follows.

An interesting observation is that the ratio of stroke length in four orientations only changes slightly between different writing styles of the same character, as demonstrated in Figure 2(d) – We term this observation *ratio consistency* in this paper. However, the ratio consistency is only used for matching the character profile; in training, the outputs of the RSC module is still the stroke length in each orientation. To supervise the stroke length, the printed version of each character, generated in SIMFANG font, is processed using an off-the-shelf edge-detection-based algorithm to calculate the corresponding ground-truth stroke length in four orientations, illustrated in Figure 5.

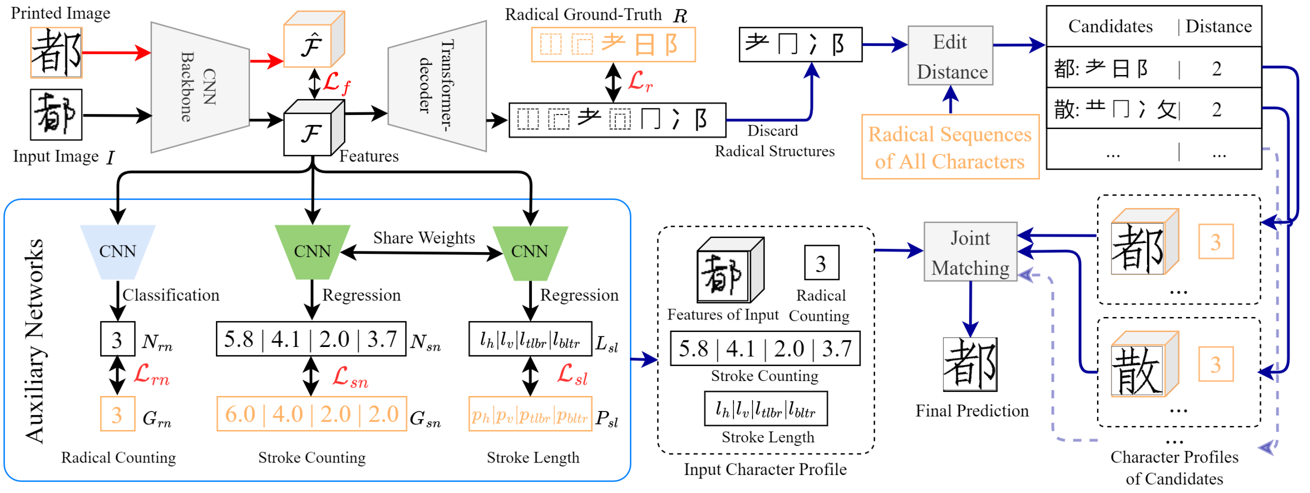

Figure 4: The overall architecture of the proposed ACPM. The data flow in red lines and blue lines is only for training and testing, respectively. The orange boxes represent the ground truth data of the corresponding predictions. The CNNs in green have the same configuration and share the weights.

With the augmented character profile, more supervision information from different levels can be used to enhance the robustness of the character recognizer. In this paper, we propose to utilize the radical-level information (*i.e.*, the number of radicals) and the stroke-level information (*i.e.*, the number of strokes and the total length of strokes) to provide more information to determine the closest candidate character in the testing phase. It is worth mentioning that we obtain the radical-level information for each character from the existing radical-level labels, and the stroke-level information from the printed character images, which barely require additional labeling cost.

4 METHODOLOGY

The proposed CCR method comprises three modules: the radical prediction module, the radical-stroke counting module, and the profile matching module. The overall architecture is shown in Figure 4. Firstly, the input character image is fed into the CNN backbone to extract the feature maps, which will be sequentially fed into a transformer decoder to produce the radical sequence and the proposed RSC module to predict the number of radicals, the number of strokes, and the total length of strokes. Finally, all these predictions viewed as the character profile are utilized to search for the closest candidate character through joint matching. Detailed methodology will be introduced subsequently.

4.1 Radical Prediction Module

The radical prediction module adopts the encoder-decoder architecture. We choose the ResNet [7] as the encoder and Transformer [16] as the decoder to predict the radical sequence. Specifically, the feature maps $\mathcal{F}$ are extracted from the input image $\mathbf{I} \in \mathbb{R}^{H \times W \times 3}$ by the encoder. Through the Transformer decoder, the feature maps $\mathcal{F}$ are decoded into the corresponding radical sequence, which is supervised by the ground truth $\mathbf{R} = \{R_1, R_2, ..., R_N\}$. We employ the cross-entropy loss to optimize the radical prediction module as follows:

$$\mathcal{L}_r = -\Sigma_{i=1}^{N} \log p(R_i) \tag{1}$$

where R_i represents the i-th radical or radical structure of the radical sequence and N is the length of $\mathbf{R}$.

During training, to alleviate the similar-appearance problem, we employ the L2 loss to constrain the feature maps between the input image and the corresponding printed image to assure robust feature representation:

$$\mathcal{L}_f = ||\mathcal{F} - \hat{\mathcal{F}}||_2^2 \tag{2}$$

where $\hat{\mathcal{F}}$ denotes the feature maps of the printed image.

4.2 Radical-Stroke Counting Module

The proposed radical-stroke counting module aims at establishing an auxiliary network that is used to perceive the numerical values of both radicals and strokes. As discussed in Section 3, three types of supervision information at the radical and stroke levels are introduced in this module. Technically, we design three sub-modules (*i.e.*, the radical counting sub-module, the stroke counting sub-module, and the stroke length-aware sub-module) to predict more attributes of Chinese characters, which can provide additional clues to determine the final predictions in inference.

The radical counting sub-module, aiming to predict the number of radicals in the input character image $\mathbf{I}$, takes the feature maps $\mathcal{F}$ as input and stacks several CNN layers to expand the receptive field for better perceiving the number of radicals. Considering that the number of radicals in a character falls in a narrow range, we treat the problem of radical counting as a classification task. Finally, we use a fully connected layer to predict the number of radicals, thus the cross-entropy loss is adopted to supervise this sub-module:

$$\mathcal{L}_{rn} = -\log p(G_{rn}) \tag{3}$$

where G_{rn} is the ground truth of the number of radicals.

Different from the radical counting sub-module, in the stroke counting sub-module and the stroke length-aware sub-module, we treat the problems of counting strokes and predicting stroke length as regression tasks because their predicted values are continuous or in a large range. We also stack several CNN layers to extract the task-specific features in these two sub-modules. Since both these two sub-modules are designed to perceive the stroke-level information, they share the weights of CNNs but have separate regression heads. For the stroke counting sub-module, it predicts the number of strokes in four orientations, which is denoted as $N_{sn} = \{n_h, n_v, n_{tlbr}, n_{bltr}\}$. Through extensive experiments, we choose the MSE loss to supervise the prediction of N_{sn}:

$$\mathcal{L}_{sn} = MSE(N_{sn}, G_{sn}) \tag{4}$$

where G_{sn} denotes the ground truth of N_{sn}. As shown in Figure 2(c), G_{sn} can be directly obtained through low-cost stroke annotation for each radical. For the stroke length-aware sub-module, we employ a two-stage training strategy to regress the total length of strokes in four orientations. At the pre-training stage, the prediction of stroke length L_{sl} is directly supervised by the pre-defined ground truth P_{sl}, which makes this sub-module aware of the stroke length. After pre-training, we use the proportionally scaled ground truth for supervision to avoid overfitting. Specifically, given the prediction $L_{sl} = \{l_h, l_v, l_{tlbr}, l_{bltr}\}$ and the original ground truth $P_{sl} = \{p_h, p_v, p_{tlbr}, p_{bltr}\}$, the proportionally scaled ground truth P'_{sl} is calculated as follows:

$$P'_{sl} = \{p_h, p_v, p_{tlbr}, p_{bltr}\} \times \frac{Sum_L}{Sum_P} \tag{5}$$

where Sum_L and Sum_P are the total lengths of L_{sl} and P_{sl}. In both stages, we employ the MSE loss to supervise the stroke length-aware sub-module:

$$\mathcal{L}_{sl} = \begin{cases} MSE(L_{sl}, P_{sl}), & \text{pre-training} \\ MSE(L_{sl}, P'_{sl}), & \text{training} \end{cases} \tag{6}$$

The algorithm utilized to produce the ground truth P_{sl} is illustrated in Figure 5. Firstly, the printed image of the character is processed by an off-the-shelf edge detection algorithm to generate a binary map which contains edge points. Subsequently, a Hough-Transform algorithm is employed to vote for straight lines among the edge points. The lines are classified into four orientations based on their slopes, denoted as k:

$$O = \begin{cases} horizontal, & k \in (-0.41, 0.41) \\ vertical, & k \in (-\infty, -2.41) \cup (2.41, +\infty) \\ tl - br, & k \in (-2.41, -0.41) \\ bl - tr, & k \in (0.41, 2.41) \end{cases} \tag{7}$$

where O denotes the orientation, 0.41 and 2.41 are the slopes of lines that cut the round angle into eight equal parts. By adding up the length of the lines, we can obtain the pixel-wise lengths of strokes in four orientations, which are regarded as labels in the stroke length-aware sub-module.

4.3 Augmented Profile Matching Module

In the testing stage, the predicted radical sequence may not match the target Chinese character. To tackle this mismatching problem, we select multiple candidate characters that have the shortest edit distance with the predicted radical sequence. Thus the augmented profile matching module is designed to calculate the similarity between the profiles of candidates and that of the input, and select the most similar candidate as the final prediction. Specifically, we calculate the similarity of four types of information (*i.e.*, the feature maps $\mathcal{F}$, the number of radicals N_{rn}, the number of strokes N_{sn}, and the stroke length L_{sl}) in the augmented profile separately and use the weighted sum of these four types of similarity as the similarity between characters. Given feature maps $\mathcal{F}'$ and $\mathcal{F}''$, we employ the cosine similarity to evaluate the resemblance:

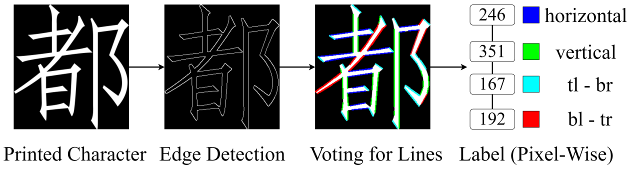

Figure 5: Annotation of the stroke length in four orientations, where tl-br and bl-tr denote top-left to bottom-right and bottom-left to top-right, respectively.

$$S_f(\mathcal{F}', \mathcal{F}'') = \frac{\mathcal{F}'^{\top}\mathcal{F}''}{||\mathcal{F}'|| \times ||\mathcal{F}''||} \tag{8}$$

The similarity between the number of radicals N'_{rn} and N''_{rn} is computed by:

$$S_{rn}(N'_{rn}, N''_{rn}) = -|N'_{rn} - N''_{rn}| \tag{9}$$

while the similarity between the number of strokes N'_{sn} and N''_{sn} is calculated by:

$$S_{sn}(N'_{sn}, N''_{sn}) = -||N'_{sn} - N''_{sn}||_2^2 \tag{10}$$

where N'_{sn} and N''_{sn} are both 4-dimensional vectors. For the stroke length L'_{sl} and L''_{sl} , we follow the observation of ratio consistency demonstrated in Figure 2(d) to calculate the similarity, which is denoted as $S_{sl}(L'_{sl}, L''_{sl})$. The computation of $S_{sl}(L'_{sl}, L''_{sl})$ is detailed in the Supplementary Material.

Finally, we select the i^*-th candidate character with the least weighted sum of the above four types of similarity as the final prediction, which is presented as follows (for simplicity, we use r_i, s_i, and l_i for denoting the i-th N_{rn}, N_{sr}, and L_{sl}, respectively):

$$i^* = \arg\min_i (S_f(\mathcal{F}, \mathcal{F}_i) + \lambda_1 S_{rn}(r, r_i) + \lambda_2 S_{sn}(s, s_i) + \lambda_3 S_{sl}(l, l_i)) \tag{11}$$

where $\mathcal{F}, r, s, l$ are from the character profile of the input image; λ_1, λ_2, and λ_3 are balancing coefficients.

4.4 Overall Training Objective

As a multi-task architecture, the overall training loss can be described as follows:

$$\mathcal{L} = \mathcal{L}_r + \lambda_{rn}\mathcal{L}_{rn} + \lambda_{sn}\mathcal{L}_{sn} + \lambda_{sl}\mathcal{L}_{sl} + \lambda_f\mathcal{L}_f \tag{12}$$

where λ_{rn}, λ_{sn}, λ_{sl}, and λ_f are trade-off coefficients for balancing the loss functions.

5 EXPERIMENTS

In this section, we first introduce the experimental configurations, including datasets and implementation details. Subsequently, we

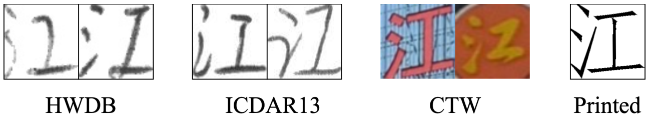

Figure 6: Some examples in the adopted datasets. HWDB1.0-1.1 and ICDAR13 are handwritten datasets. The scene text images in the CTW dataset are captured from street views. The printed image are constructed in SIMFANG font.

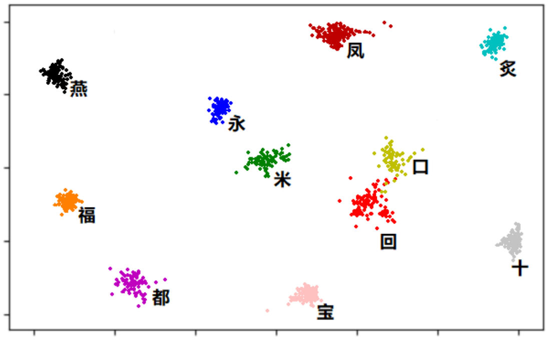

Figure 7: Visualization of the ratio consistency of orientation-based strokes (4-D vectors embedded using t-SNE). Each character class is shown in one color. In each class, the stroke-length ratios of 60 examples in the handwritten benchmark ICDAR2013 are plotted.

conduct extensive experiments to evaluate the proposed ACPM. At last, ablation studies and analysis are carried out to further validate the effectiveness of the proposed CCR method.

5.1 Datasets

All datasets used to conduct experiments are introduced in the following. Some examples are visualized in Figure 6.

- **HWDB1.0-1.1** [13] collects 2,730,885 offline handwritten Chinese character images from 720 writers. The number of class are 3755 and 3866 in HWDB1.1 and HWDB1.0, respectively.
- **ICDAR 2013** [24] contains 224,419 offline handwritten Chinese character images from 60 writers with 3755 classes. It is a popular benchmark for performance comparison between modern handwritten CCR methods.
- **CTW** [25] contains 812,872 Chinese character images with 3650 classes. The images in this dataset, captured from street views, are especially challenging due to complex backgrounds, blurring, varying fonts, and occlusions.
- **Printed Standard Character Images** are produced in SIMFANG font, which is the most widely used Chinese font. it is worth noting that this dataset is only used for constructing the character profiles of the candidate characters.

5.2 Implementation Details

The proposed ACPM is implemented with Pytorch and all experiments are conducted on a NVIDIA RTX 3090 GPU with 24GB memory. We use the Adadelta [26] as the optimizer with an initial learning rate 1.0. The batch size is set to 32. The input images are resized to 32 × 32 and normalized to [-1,1]. We empirically set λ_{rn}, λ_{sn}, λ_{sl}, and λ_f to be 1, 1, 1, and 0.01, respectively.

Table 1: Performance comparison on handwritten benchmark ICDAR 2013.

Method	Decomposition Level	CACC
HCCR-GoogLeNet [28]	Character	96.35%
DirectMap+ConvNet [13]	Character	97.37%
DenseRAN [19]	Radical	96.66%
FewShotRAN [18]	Radical	96.97%
HDE [1]	Radical	97.14%
template+instance[22]	Character	97.45%
Chen et al. [3]	Stroke	96.74%
ACPM (ours)	All	**97.80%**

5.3 Comparison with Existing Methods

The proposed ACPM outperforms the state-of-the-art methods on both the handwritten dataset ICDAR 2013 and scene text dataset CTW. Furthermore, ACPM achieves better performance than the compared methods in the setting of character zero-shot. Following the previous works, Character Accuracy (CACC) is used as the evaluation metric.

Performance on Handwritten Benchmarks. Following the fashion of previous works, ACPM is trained on HWDB1.0-1.1 and evaluated on the handwritten Chinese character benchmark ICDAR 2013. For fair comparison, only 3755 Level-1 Chinese characters in HWDB1.0-1.1 are used for training. As shown in Table 1, the experimental results demonstrate that the proposed ACPM outperforms the best state-of-the-art method (from 97.45% to 97.80%), which benefits from the proposed augmented character profile. Through the analysis of recognition results, we observe that the failure cases (around 2.20% samples in ICDAR 2013) are mostly challenging ones, *e.g.*, extremely scribbled writing styles that violate the aforementioned ratio consistency, which may bring difficulties to the proposed stroke length-aware sub-module.

Performance on Scene Text Benchmarks. Compared with the handwritten datasets, the scene text benchmark CTW is more challenging due to complex real-world backgrounds and noises (*e.g.*, blurs and occlusions). As shown in Table 2, ACPM achieves a significant improvement by 2.23% on CTW compared with the state-of-the-art method HDE [1]. A possible reason is that the proposed ACPM utilizes a collection of Chinese character knowledge from three decomposition levels, which is more robust to complex backgrounds and noises, while existing methods are only based on a single-level representation.

Performance in Zero-shot Settings. Recently, some studies pay interest on the zero-shot problems in handwritten CCR, since the character classes contained in the training set may not cover all the classes in the test set in real-world applications. Instead of directly recognize unseen character classes, previous methods

Table 2: Performance comparison on scene text benchmark CTW.

Method	Decomposition Level	CACC
ResNet-50 [7]	Character	79.46%
ResNet-152 [7]	Character	80.94%
DenseNet [8]	Character	79.88%
DenseRAN [19]	Radical	85.56%
FewShotRAN [18]	Radical	86.78%
HDE [1]	Radical	89.25%
Chen et al. [3]	Stroke	85.90%
ACPM (ours)	All	**91.48%**

predict a radical sequence to infer unseen characters, where all radicals are seen in the training set. Although the purpose of the proposed method is not to tackle zero-shot problems, the predictions of the proposed RSC module consist of radical-level and stroke-level knowledge, which can also be used to predict unseen characters. As shown in Table 3, the proposed ACPM achieves the state-of-the-art performance on character zero-shot settings. However, the proposed ACPM still suffers from the radical zero-shot problem due to the utilization of radical-level information in inference. Therefore, the performance of ACPM is inferior to the stroke-based method [3] in radical zero-shot settings.

5.4 Ablation Study

Radical Counting and Stroke Counting. We conduct an ablation study on the radical-stroke counting module to further specify the contribution of information from different levels in the augmented character profile. As shown in Table 4, compared with the baseline (without radical and stroke counting), the performance is boosted by 0.41% and 1.33% on ICDAR 2013 and CTW respectively with the radical counting sub-module, and by 1.06% and 4.94% with the stroke counting sub-module. When both the radical and stroke counting sub-modules are equipped, the performance is improved by 1.22% and 5.35%, respectively. In summary, the stroke counting sub-module seems to act as the main contributor while the performance improvement brought by the radical counting sub-module is not significant. This is not surprising since the radical counting aims at abating errors caused by the incorrectly predicted radical sequence. Moreover, the counting of radicals has been implicitly expressed in the radical sequence decoding and the accuracy of radical counting can reach 84.94% (shown in Table 5).

Joint Training and Feature Matching. Since the proposed ACPM adopts a multi-task framework, it is necessary to investigate whether the joint training of the radical prediction module and the RSC module can improve the ability of feature representation. As shown in Table 5, by blocking the gradient flow between the RSC module and the CNN backbone, the accuracy of radical prediction drops by 0.16% (from 84.94% to 84.78%) while the overall accuracy hardly decreases. Therefore, we can come to a conclusion that the joint training with the RSC module can enhance the feature representation thus leading to a slight improvement of radical prediction accuracy. In addition, we also conduct an ablation study on the feature matching loss $\mathcal{L}_f$ to evaluate its ability to distinguish similar Chinese characters. By removing $\mathcal{L}_f$ from Eq.(12) in the training stage, the accuracy of radical prediction drops by 0.74% (from 84.94% to 84.20%), and the overall accuracy also drops by 0.75% (from 91.48% to 90.73%). The experimental results demonstrate that feature matching is the main contributor to stronger feature-representation ability, which benefits the radical prediction module. Finally, when both the feature matching loss $\mathcal{L}_f$ and joint training strategy are employed, the accuracy of radical prediction and overall prediction can be boosted by 0.99% (from 83.95% to 84.94%) and 0.97% (from 90.51% to 91.48%), respectively; the performance improvement further validates their effectiveness.

5.5 Additional Discussions

Effectiveness of the RSC Module. Since the proposed ACPM depends on character profile matching which includes the outputs of the RSC module, the performance of the RSC module is crucial to the final performance. Here, we conduct experiments to investigate the performance of three sub-modules in the RSC module. As shown in Table 6, the performance of these three sub-modules on the HWDB1.0-1.1 and ICDAR 2013 is better than that on the CTW dataset since the character images in HWDB1.0-1.1 and ICDAR 2013, which tend to have clean backgrounds, are more recognizable than those in the CTW dataset. Moreover, we observe that compared with the number of strokes, the fine-grained information of Chinese character (*i.e.*, the length of strokes) is more difficult to predict.

Ratio Consistency of Orientation-based Strokes. To further validate the above-mentioned ratio consistency of orientation-based strokes, we sample 660 examples of 11 characters and visualize the 4-D vectors embedded in a 2-D space using t-SNE, where each character class is shown in one color. As shown in Figure 7, the stroke-length ratios of each character are gathered as a cluster and it is easy to find a boundary to separate them from one another, which further confirms our observation of ratio consistency. Moreover, even the characters, "kou" (marked in yellow) and "hui" (marked in red), which have almost the same stroke-length ratio in the printed version, are separated in the embedding space.

One-to-Many and Similar-Appearance Problems. As described in Section 1, one-to-many and similar-appearance problems still bother the existing methods. In the proposed ACPM, the two challenging problems can be handled through the character profile matching, where the information from three decomposition levels (*i.e.*, character-level, radical-level, and stroke-level) is utilized. For the one-to-many problem, we observe that those characters sharing the same stroke sequence tend to have significant visual difference. Therefore, the character-level feature maps in the character profile play an important role in distinguishing those one-to-many characters. In addition, the similar-appearance problem can be solved by comparing the number or length of strokes in the character profile. Detailed case studies about these two challenges are performed in the Supplementary Material.

Size of Candidate Collection. In the test phase, the character profiles of 3755 classes in SIMFANG font can be saved on the local disk and loaded in advance. Therefore, the time cost of the joint matching algorithm depends on the size of the candidate collection. To balance accuracy and efficiency, we conduct an experiment

Table 3: Performance comparison in the character zero-shot and radical zero-shot settings on the handwritten benchmark HWDB and scene text benchmark CTW.

HWDB	Character Zero-Shot Setting					**HWDB**	Radical Zero-Shot Setting				
	500	1000	1500	2000	2755		50	40	30	20	10
DenseRAN [19]	1.70%	8.44%	14.71%	19.51%	30.68%	DenseRAN [19]	0.21%	0.29%	0.25%	0.42%	0.69%
HDE [1]	4.90%	12.77%	19.25%	25.13%	33.49%	HDE [1]	3.26%	4.29%	6.33%	7.64%	9.33%
Chen et al. [3]	5.60%	13.85%	22.88%	25.73%	37.91%	Chen et al. [3]	**5.28%**	**6.87%**	**9.02%**	**14.67%**	**15.83%**
ACPM (ours)	**9.72%**	**18.50%**	**27.74%**	**34.00%**	**42.43%**	ACPM (ours)	4.29%	6.20%	7.85%	10.36%	12.51%
CTW	Character Zero-Shot Setting					**CTW**	Radical Zero-Shot Setting				
	500	1000	1500	2000	3150		50	40	30	20	10
DenseRAN [19]	0.15%	0.54%	1.60%	1.95%	5.39%	DenseRAN [19]	0%	0%	0%	0%	0.04%
HDE [1]	0.82%	2.11%	3.11%	6.96%	7.75%	HDE [1]	0.18%	0.27%	0.61%	0.63%	0.90%
Chen et al. [3]	1.54%	2.54%	4.32%	6.82%	8.61%	Chen et al. [3]	**0.66%**	**0.75%**	**0.81%**	**0.94%**	**2.25%**
ACPM (ours)	**3.44%**	**6.18%**	**10.65%**	**15.40%**	**21.29%**	ACPM (ours)	0.54%	0.70%	0.74%	0.78%	0.89%

Table 4: Ablation study on radical counting (RC) and stroke counting (SC) in the testing phase.

RC	SC	ICDAR 2013	CTW
		96.58%	86.13%
	✓	97.64%	91.07%
✓		96.89%	87.46%
✓	✓	**97.80%**	**91.48%**

Table 5: Ablation study on the joint training and feature matching loss $\mathcal{L}_f$.

Joint Training	$\mathcal{L}_f$	Radical ACC	CACC
		83.95%	90.51%
	✓	84.78%	91.45%
✓		84.20%	90.73%
✓	✓	**84.94%**	**91.48%**

Table 6: Evaluation of each sub-module in the RSC module (CACC is used for evaluating the radical counting submodule and MSE is used for evaluating the stroke counting and stroke length-aware sub-modules).

Sub-module	HWDB1.0-1.1	ICDAR 2013	CTW
Radical Counting	99.34%	97.42%	89.08%
Stroke Counting	0.091	0.523	1.492
Stroke Length-aware	0.168	0.642	0.855

Table 7: Performance on the handwritten dataset ICDAR 2013 and scene text dataset CTW with different size of candidate collection.

Edit Distance	ICDAR 2013	CTW	Time
≤ 1	97.76%	91.37%	0.03s
≤ 2	**97.80%**	**91.48%**	0.08s
≤ 3	97.80%	91.47%	0.69s

shown in Table 7 to select the appropriate size of candidate collection (controlled by edit-distance), where the time cost is the average matching time of 10 iterations. By changing the edit-distance, we find that the candidate collection with radical edit-distance within 2 is likely to produce the best matching result, since a larger candidate collection boosts time-cost rapidly and is not necessary to result in better CACC (the CACC on CTW drops from 91.48% to 91.47% instead).

6 CONCLUSION

This paper proposes a Chinese character recognition method, named ACPM, through jointly matching multi-level (*i.e.*, character-level, radical-level, and stroke-level) Chinese character information with an augmented character profile. To this end, we propose a radical-stroke counting (RSC) module to extract the number of radicals, the number of strokes, and the total length of strokes for each character. All the predictions of the RSC module are employed as the augmented character profile for the input image and used to search for the closest candidate character in inference. The experimental results show that the proposed ACPM outperforms the state-of-the-art methods in both well-known handwritten and scene text benchmarks and also outperforms the compared methods in the zero-shot settings by a clear margin.

7 ACKNOWLEDGEMENTS

This work was supported in part by the National Natural Science Foundation of China (No.62176060), STCSM project (No.20511100400), Shanghai Municipal Science and Technology Major Project (No.2018SHZDZX01) and ZJLab, Shanghai Research and Innovation Functional Program (No.17DZ2260900), and the Program for Professor of Special Appointment (Eastern Scholar) at Shanghai Institutions of Higher Learning.

REFERENCES

[1] Zhong Cao, Jiang Lu, Sen Cui, and Changshui Zhang. 2020. Zero-Shot Handwritten Chinese Character Recognition with Hierarchical Decomposition Embedding. *Pattern Recognition* 107 (2020), 107488.

[2] Fu Chang. 2006. Techniques for Solving the Large-Scale Classification Problem in chinese handwriting recognition. In *SACHR*. 161–169.

[3] Jingye Chen, Bin Li, and Xiangyang Xue. 2021. Zero-Shot Chinese Character Recognition with Stroke-Level Decomposition. In *IJCAI*. 615–621.

[4] Jingye Chen, Haiyang Yu, Jianqi Ma, Bin Li, and Xiangyang Xue. 2022. Text Gestalt: Stroke-Aware Scene Text Image Super-Resolution. In *AAAI*, Vol. 36. 285–293.

[5] Xiaoxue Chen, Lianwen Jin, Yuanzhi Zhu, Canjie Luo, and Tianwei Wang. 2021. Text Recognition in the Wild: A Survey. *Comput. Surveys* 54, 2 (2021), 1–35.

[6] Dan Cireşan and Ueli Meier. 2015. Multi-Column Deep Neural Networks for Offline Handwritten Chinese Character Classification. In *IJCNN*. 1–6.

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *CVPR*. 770–778.

[8] Gao Huang, Zhuang Liu, Van Der Maaten Laurens, and Weinberger Kilian Q. 2017. Densely Connected Convolutional Networks. In *CVPR*. 4700–4708.

[9] Lianwen Jin, Junxun Yin, Xue Gao, and Jiangcheng Huang. 2001. Study of Several Directional Feature Extraction Methods with Local Elastic Meshing Technology for HCCR. In *the Sixth Int. Conference for Young Computer Scientist*. 232–236.

[10] In Jung Kim, Chenglin Liu, and Jin Hyung Kim. 1999. Stroke-Guided Pixel Matching for Handwritten Chinese Character Recognition. In *ICDAR*. 665–668.

[11] Jin Wook Kim, Kwang In Kim, Bong Joon Choi, and Hang Joon Kim. 1999. Decomposition of Chinese Character into Strokes Using Mathematical Morphology. *Pattern Recognition Letters* 20, 3 (1999), 285–292.

[12] Chenglin Liu, In Jung Kim, and Jin H. Kim. 2001. Model-Based Stroke Extraction and Matching for Handwritten Chinese Character Recognition. *Pattern Recognition* 34, 12 (2001), 2339–2352.

[13] Chenglin Liu, Fei Yin, Dahan Wang, and Qiufeng Wang. 2013. Online and Offline Handwritten Chinese Character Recognition: Benchmarking on New Databases. *Pattern Recognition* 46, 1 (2013), 155–162.

[14] Daming Shi, Steve R. Gunn, and Robert I. Damper. 2003. Handwritten Chinese Radical Recognition Using Nonlinear Active Shape Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 25, 2 (2003), 277–280.

[15] Yih Ming Su and Jhing Fa Wang. 2003. A Novel Stroke Extraction Method for Chinese Characters Using Gabor Filters. *Pattern Recognition* 36, 3 (2003), 635–647.

[16] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is All You Need. *NeurIPS* 30 (2017).

[17] Anbang Wang, Kuo Chin Fan, and Wei Hsien Wu. 1996. A Recursive Hierarchical Scheme for Radical Extraction of Handwritten Chinese Characters. In *ICPR*, Vol. 3. 240–244.

[18] Tianwei Wang, Zecheng Xie, Zhe Li, Lianwen Jin, and Xiangle Chen. 2019. Radical Aggregation Network for Few-Shot Offline Handwritten Chinese Character Recognition. *Pattern Recognition Letters* 125 (2019), 821–827.

[19] Wenchao Wang, Jianshu Zhang, Jun Du, Zirui Wang, and Yixing Zhu. 2018. Denseran for Offline Handwritten Chinese Character Recognition. In *ICFHR*. 104–109.

[20] Chunpeng Wu, Wei Fan, Yuan He, Jun Sun, and Satoshi Naoi. 2014. Handwritten Character Recognition by Alternately Trained Relaxation Convolutional Neural Network. In *ICFHR*. 291–296.

[21] Changjie Wu, Zirui Wang, Jun Du, Jianshu Zhang, and Jiaming Wang. 2019. Joint Spatial and Radical Analysis Network for Distorted Chinese Character Recognition. In *ICDARW*, Vol. 5. 122–127.

[22] Yao Xiao, Dan Meng, Cewu Lu, and Chi Keung Tang. 2019. Template-Instance Loss for Offline Handwritten Chinese Character Recognition. In *ICDAR*. 315–322.

[23] Yun Yang, Donghai Li, and Zongtao Duan. 2018. Chinese Vehicle License Plate Recognition Using Kernel-Based Extreme Learning Machine with Deep Convolutional Features. *IET Intelligent Transport Systems* 12, 3 (2018), 213–219.

[24] Fei Yin, Qiufeng Wang, Xuyao Zhang, and Chenglin Liu. 2013. ICDAR 2013 Chinese Handwriting Recognition Competition. In *ICDAR*. 1464–1470.

[25] Tailing Yuan, Zhe Zhu, Kun Xu, Chengjun Li, Taijiang Mu, and ShiMin Hu. 2019. A Large Chinese Text Dataset in the Wild. *Journal of Computer Science and Technology* 34, 3 (2019), 509–521.

[26] Matthew D. Zeiler. 2012. Adadelta: an Adaptive Learning Rate Method. *arXiv:1212.5701* (2012).

[27] Chongsheng Zhang, Weiping Ding, Guowen Peng, Feifei Fu, and Wei Wang. 2020. Street View Text Recognition with Deep Learning for Urban Scene Understanding in Intelligent Transportation Systems. *IEEE Transactions on Intelligent Transportation Systems* 22, 7 (2020), 4727–4743.

[28] Zhuoyao Zhong, Lianwen Jin, and Zecheng Xie. 2015. High Performance Offline Handwritten Chinese Character Recognition Using Googlenet and Directional Feature Maps. In *ICDAR*. 846–850.