Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2024.046449

ARTICLE

A Method for Detecting and Recognizing Yi Character Based on Deep Learning

Haipeng Sun[1,2], Xueyan Ding[1,2,*], Jian Sun[1,2], Hua Yu[3] and Jianxin Zhang[1,2,*]

[1]School of Computer Science and Engineering, Dalian Minzu University, Dalian, 116600, China

[2]Institute of Machine Intelligence and Bio-Computing, Dalian Minzu University, Dalian, 116600, China

[3]Yi Language Research Room, China Ethnic Languages Translation Centre, Beijing, 100080, China

*Corresponding Authors: Xueyan Ding. Email: dingxueyan@dlnu.edu.cn; Jianxin Zhang. Email: jxzhang0411@163.com

Received: 01 October 2023 Accepted: 26 December 2023 Published: 27 February 2024

ABSTRACT

Aiming at the challenges associated with the absence of a labeled dataset for Yi characters and the complexity of Yi character detection and recognition, we present a deep learning-based approach for Yi character detection and recognition. In the detection stage, an improved Differentiable Binarization Network (DBNet) framework is introduced to detect Yi characters, in which the Omni-dimensional Dynamic Convolution (ODConv) is combined with the ResNet-18 feature extraction module to obtain multi-dimensional complementary features, thereby improving the accuracy of Yi character detection. Then, the feature pyramid network fusion module is used to further extract Yi character image features, improving target recognition at different scales. Further, the previously generated feature map is passed through a head network to produce two maps: a probability map and an adaptive threshold map of the same size as the original map. These maps are then subjected to a differentiable binarization process, resulting in an approximate binarization map. This map helps to identify the boundaries of the text boxes. Finally, the text detection box is generated after the post-processing stage. In the recognition stage, an improved lightweight MobileNetV3 framework is used to recognize the detect character regions, where the original Squeeze-and-Excitation (SE) block is replaced by the efficient Shuffle Attention (SA) that integrates spatial and channel attention, improving the accuracy of Yi characters recognition. Meanwhile, the use of depth separable convolution and reversible residual structure can reduce the number of parameters and computation of the model, so that the model can better understand the contextual information and improve the accuracy of text recognition. The experimental results illustrate that the proposed method achieves good results in detecting and recognizing Yi characters, with detection and recognition accuracy rates of 97.5% and 96.8%, respectively. And also, we have compared the detection and recognition algorithms proposed in this paper with other typical algorithms. In these comparisons, the proposed model achieves better detection and recognition results with a certain reliability.

KEYWORDS

Yi characters; text detection; text recognition; attention mechanism; deep neural network

1 Introduction

The Yi ethnic minority is the sixth-largest ethnic minority in China and boasts a rich literary heritage covering various fields such as medicine, astronomy, religion, and more. To further the advancement of related disciplines, it can serve as a valuable source of reference and insight. Although research into Yi literature is ongoing, it is currently mainly carried out using traditional methods such as scanning, transcription, and manual translation. Unfortunately, much of this literature has been lost to the ravages of time. Consequently, there is an urgent need for the digital preservation of Yi literature, which has the potential to unearth the profound cultural treasures and spiritual essence embedded in Yi character's writings. Accurate recognition of Yi characters serves as a crucial foundation for advancing the digitization of Yi literature, the quality of which directly affects subsequent identification and analysis efforts [1,2].

However, current text recognition technology is mainly applied to universal languages such as Chinese and English [3–5]. When dealing with Yi characters, the existence of similar shapes, complex fonts, and the fact that their characters often appear in complex backgrounds makes it difficult to apply traditional text recognition techniques to Yi character recognition. With the continuous development of deep learning, researchers are committed to using deep learning for training, eliminating the cumbersome manual rules and intermediate processing steps to significantly improve model performance, which has become a hot spot in current research [6,7]. However, there is still a lack of research on Yi character detection and recognition by using deep learning theory, so it is necessary to design specific detection and recognition methods, to improve the accuracy and robustness of model recognition.

Therefore, this paper proposes a deep learning-based approach for detecting and recognizing Yi characters. The main contributions of this article are as follows:

1. Introducing an improved segmentation-based Differentiable Binarization Network (DBNet) [8] serves as the detection model, where the ResNet-18 [9] backbone network incorporates the Omni-dimensional Dynamic Convolution (ODConv) [10]. ODConv is employed to extract fine-grained Yi character features, thereby effectively bolstering the network's ability to detect local features.
2. Designing an improved lightweight MobileNetV3 [11] framework for recognition models. Instead of the original SE block, the Shuffle Attention module [12] is employed to enhance the network's ability to represent global information and effectively model features within Yi character images.
3. Constructing a Yi character dataset and enriching the number of samples to improve the accuracy of Yi character detection and recognition in the context of the printed background.
4. The experimental results demonstrate that the proposed method achieves an impressive accuracy rate of 97.5% on the test datasets. This marks a notable improvement of 1.4% over the standard character detection model DBNet. Lastly, the recognition model attains an outstanding accuracy of 96.8% when recognizing characters from the detected regions. This represents a significant improvement of 2.3% compared to the standard MobileNetV3 character recognition model.

The rest of the manuscript is structured as follows: Section 2 reviews related works on other ethnic minorities character, Section 3 examines research that is Yi character detection based on Omni-dimensional Dynamic Convolution, Section 4 describes the Yi character recognition based on shuffle attention module in detail, Section 5 outlines the experiment methodology, and Section 6 presents with the conclusion and future work.

2 Related Works

2.1 Chinese and English Text Detection and Recognition Methods

In recent years, the detection and recognition of Chinese, English, and other mainstream characters has made a breakthrough, which can provide a reference for the research of Yi character detection and recognition. In the text detection stage, Wang et al. [13] proposed a shape-robust text detection method based on Progressive Scale Expansion Network (PSENet), which uses adaptive polygon shapes to represent text areas. This enables more accurate detection of text areas in the scene. In 2020, Bai et al. [8] proposed a real-time scene text detection method based on a Differentiable Binarization Network (DBNet) to achieve fast text area detection and text line segmentation tasks. In 2021, Zhu et al. [14] proposed a Fourier Contour Embedding scene text detection method (FCENet), which focuses on modeling the representation of an arbitrary shape of a text-enclosing box and can accurately approximate any closed curve. In the text recognition stage, Shi et al. [15] proposed a Convolutional Recurrent Neural Network (CRNN) that includes three parts: feature extraction, sequence analysis, and sequence decoding. The network uses bidirectional Long-Short Term Memory (LSTM) and Convolutional Neural Networks (CNN) to extract image features and introduces Connectionist Temporal Classification (CTC) in the speech recognition field to the image processing problem of sequence alignment of indefinite length. Liao et al. [16] have proposed a two-dimensional scene text recognition based on Fully Convolutional Network (FCN). Wang et al. [17] proposed a network with a typical coding-decoding structure to solve the problem that the structure information and pixel spatial correlation information of two-dimensional images are destroyed when two-dimensional feature maps are converted into one-dimensional feature vectors. First, the Visual Geometry Group (VGG) embedded in deformable convolutional networks is used as the backbone network, and then the extracted feature map is mapped to the character output by the full connection layer. This algorithm significantly improves the accuracy of text recognition.

2.2 Tibetan Character Detection and Recognition

However, some research has been dedicated to detecting and recognizing other ethnic characters. In the detection and recognition of the Tibetan character, Zhang et al. [18] proposed an innovative Tibetan character segmentation approach grounded in critical feature information. The method effectively tackled challenges such as text lines with varying degrees of tilt and distortion, overlapping and intersecting character strokes, and the diversity in stroke styles. The researchers utilized projection techniques and syllable point location information to condense the text lines, creating a dataset of word blocks. Subsequently, they partitioned these word blocks into upper and lower regions, employing distinct methods for segmentation. Ma et al. [19] introduced a comprehensive segmentation framework for historical Tibetan document images. To address issues like unbalanced noise, the initial step involves pre-processing the data and converting it into binarized form. Following this, they proposed a block projection-based layout segmentation algorithm to divide the image into three main sections: text, lines, and frames. Finally, they offered a graph data model text line segmentation method to overcome challenges related to overlapping strokes within text lines, ultimately achieving the segmentation of Tibetan strings and the identification of Tibetan characters.

2.3 Manchu Character Detection and Recognition

In the Manchu characters detection and recognition, Wang et al. [20] observed that prevailing approaches for machine-printed digitized Manchu documents predominantly rely on coarse image segmentation methods, potentially leading to recognition biases and other issues. They introduced a

segmentation-free Manchu character recognition method to overcome these challenges and developed the framework. This framework leverages a convolutional recurrent neural network with an attention mechanism, offering a more precise and effective solution for Manchu recognition. Due to the intricacies of Manchu characters' spelling rules and various Manchu scripts, achieving accurate segmentation in segmentation-based Manchu character recognition methods has proven challenging. Facing these challenges, Zheng et al. [21] introduced the concept of segmentation-free recognition to circumvent the issues associated with incorrect segmentation. Zhang et al. [22] proposed a Manchu-based character recognition system that comprises two distinct components: character recognition and retrieval. In this system, character recognition and segmentation are performed by a CNN using a sliding window approach. Additionally, character retrieval is employed to identify similar points within the image, allowing for the localization of character positions within datasets.

2.4 Mongolian Character Detection and Recognition

In the detection and recognition of the Mongolian character, as suggested by Su et al. [23], Employing two distinct approaches for different Mongolian character lines is recommended. These approaches encompass the overall character line recognition and the segmentation-based method. The selection between these two recognition schemes is contingent on the ratio of the text line height to the width of the baseline, and this decision is made using a CNN during the training process.

2.5 Uyghur Character Detection and Recognition

In the Uyghur characters detection and recognition, Mahmut et al. [24] introduced an approach that involves obtaining disambiguation points during the disambiguation phase. They employed a three-character deep learning classifier as the recognition unit during the classification phase, primarily due to the Uyghur script's tendency to exhibit relatively strong character adhesion. Xu et al. [25] proposed a segmentation-driven recognition algorithm incorporating a feedback model and linguistic analysis. Firstly, a handwritten Uighur word is over-segmented into a two-queue grapheme sequence using a Main Segmentation and Additional Clustering (MSAC). Subsequently, a feedback-based grapheme merging strategy is devised to determine the most favorable sequence of segmented characters, yielding word recognition results. Fang et al. [26] introduced three distinct network architectures that leverage baseline features from Uyghur characters as text features. These architectures encompass a feature extraction network designed to extract low-level CNN features from the original image, a region proposal network responsible for generating text candidate regions at three different scales from different convolutional layers, and a character detection network that predicts confidence scores and offsets for these candidate regions.

2.6 Yi Character Detection and Recognition

Compared with advanced Chinese and English character recognition technologies and other character recognition technologies, the technology of Yi character detection and recognition based on deep learning is still at an early stage of development. Yi et al. [27] proposed a convolutional neural network-based Yi character recognition algorithm and constructed a multi-layer deep convolutional neural network to extract the features of Yi character images, which improved the accuracy of Yi character recognition compared with other Yi character recognition algorithms. Chen et al. [28] introduced a deep learning-based approach to the recognition of ancient Yi characters. The researchers first built five models using a four-layer convolutional neural network. They then encoded the output neurons of these five models and added a distributed penalty term. Finally, they re-evaluated the Yi characters, derived their probability distribution, and identified the characters with the highest

probability of recognition. Yin et al. [29] pointed out that Yi strokes present the problem of writing irregularities and other factors that lead to random changes in the writing order and shifting of stroke positions, constructed an online handwriting recognition database for Yi and proposed two models combining deep learning: RnnNet-Yi and ParallelRnnNet-Yi, which are effective in improving the recognition accuracy. The Yi script, with its logographic nature, exhibits a high degree of complexity. Unlike alphabetic scripts that have a limited set of characters, Yi script consists of a vast array of unique symbols, each representing a word or a syllable. The characters can vary significantly in shape and size, leading to high intraclass variability and low interclass variability, making the task of character detection particularly challenging. Moreover, the Yi script has connected characters with subtle differences, further complicating the recognition process. Additionally, the scarcity of large, annotated datasets for Yi script restricts the effectiveness of machine learning algorithms that depend on voluminous and diverse training data.

3 Yi Character Detection Based on Omni-Dimensional Dynamic Convolution

Since Yi characters have complex shapes and high similarity, we adopt the DBNet algorithm based on real-time segmentation. Compared with the detection method based on regression, the DBNet algorithm has the advantage that it is not limited by shapes and can achieve better detection results for texts with different shapes. This paper uses DBNet as the basic framework, which has the disadvantages of large computation and too many parameters, resulting in low detection performance of the DBNet algorithm. Therefore, we improve the feature extraction stage of DBNet and propose a Yi character detection method based on Omni-dimensional Dynamic Convolution (ODC_DBNet). The ODConv module is adopted for feature extraction, and the multidimensional attention mechanism is used to learn the four attention types of the convolution kernel in parallel along all dimensions of the kernel space. By applying this method to the corresponding convolution kernel, the feature extraction capability of the CNN convolution operation is improved. The overall structure is shown in Fig. 1. ODC_DBNet is composed of three basic components:

- The ODConv Feature Extraction Module.
- The Feature Pyramid Network Fusion Module.
- The Differentiable Binarization Prediction Module.

By integrating these three essential components, the ODC_DBNet method achieves exceptional performance in Yi character detection and recognition. This outstanding performance not only strengthens the recognition of the Yi character but also serves as a solid foundation for further refinement. It is important to emphasize that the specific technical intricacies and parameter configurations will be extensively explored and fine-tuned in subsequent experiments.

Initially, the feature extraction stage utilizes ResNet-18 as the backbone network. In this stage, the standard convolution operations from the original network are replaced with an ODConv module with adaptive capabilities. The parameters of the convolution kernel are dynamically adjusted based on the input data, thereby enhancing the model feature detection capabilities. Subsequently, the feature fusion stage uses the FPN to fuse 1/4, 1/8, 1/16, and 1/32 feature maps of the original image size to better fuse the features of different sizes of Yi character images to improve detection accuracy. Then, the fused 1/4 size feature maps prediction, generating threshold maps and probability maps through the Differentiable Binarization (DB) prediction module. These two resulting images are subjected to the DB operation, yielding approximate binarization maps. Finally, the approximate binarization maps undergo post-processing to derive the text bounding box.

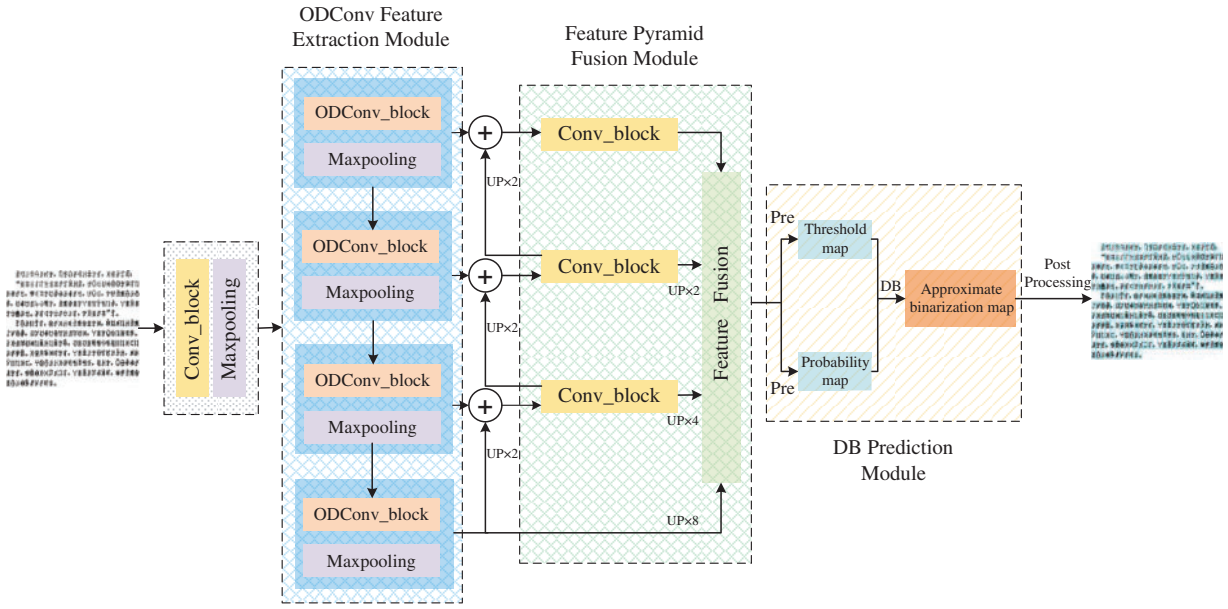


Figure 1: Overall structure diagram of ODC_DBNet

3.1 ODConv Feature Extraction Module

Compared to the DBNet, a dynamic convolutional module is introduced to solve the problem of fixed size and shape of the convolutional kernel of traditional convolutional neural networks in this paper. Our model dynamically adjusts convolution kernels of different sizes at different levels to adaptively adjust feature scales and capture richer features of Yi characters. Specifically, we use ResNet-18 as the backbone network and replace Conv with ODConv. In terms of spatial size, number of convolutional kernels, number of input channels, and number of output channels, different attention mechanisms are beneficial for different degrees of learning.

Compared to traditional convolution techniques, researchers integrate the attention mechanism into the convolution kernel, introducing the concept of dynamic convolution. For example, Yang et al. [30] introduced conditional parametric convolution, while Chen et al. [31] used dynamic convolution as part of their respective methods. Dynamic convolution differs from conventional convolution, where a single convolution kernel is applied uniformly to all samples. In contrast, dynamic convolution adds an attention mechanism to the number of convolution kernels, allowing each kernel parameter to be dynamically determined based on the input data. It is important to note that dynamic convolution only focuses on one dimension and ignores the other three dimensions related to the size of the space, the number of input channels, and the number of output channels. Li et al. [10] designed a full-dimensional and efficient attention mechanism within dynamic convolution to solve the above problems, thus creating a comprehensive ODConv module.

The detailed implementation of the ODConv is illustrated in Fig. 2. For clarity, $X_{ODConv} \in R^{C_{in} \times d \times h \times w}$ represents the input feature map, $Y \in R^{C_{out} \times d' \times h' \times w'}$ represents the output feature map, and W_i $(i = 0, 1, \cdots, n)$ denotes the i th convolution kernel with a convolution kernel size of k. Initially, the spatial size of the input feature map X_{ODConv} is compressed to match the dimensions of the feature map $C_{in} \times 1 \times 1 \times 1$ using Global Average Pooling (GAP). Next, a Fully Connected (FC) layer is employed to reduce the number of channels in the feature map C_{in} down to 16. Following this, the mechanism

branches into four paths, each dedicated to computing attention across different dimensions. Branch 1 involves altering the number of channels to three times the size of the convolution kernel and applying attention $\partial_{si}\,(i = 0, 1, \cdots, n)$ to the spatial extent of the convolution kernel based on the Sigmoid function. Branch 2 upscales the number of channels to the number of input channels C_{in}. After Sigmoid, attention $\partial_{ci}\,(i = 0, 1, \cdots, n)$ is applied to the number of input channels C_{in}. Branch 3 reduces the dimension of the number of channels to align with the number of convolution kernels in the feature map n. For convolution kernels in the feature map n, attention $\partial_{wi}\,(i = 0, 1, \cdots, n)$ is computed using the Softmax function and subsequently applied to the dimension of the number of convolution kernels. Branch 4 involves increasing the number of channels to match the number of output channels, employing the Sigmoid function for this scaling process. Subsequently, attention $\partial_{fi}\,(i = 0, 1, \cdots, n)$ is applied to the output channel C_{out}. The ultimate fusion of the four distinct attention modules yields an output represented as $Y \in R^{C_{out} \times d' \times h' \times w'}$, and the weight Y is calculated as shown in Eq. (1).

$$Y = \left(W_1 \odot \partial_{s1} \odot \partial_{c1} \odot \partial_{w1} \odot \partial_{f1} + \cdots + W_n \odot \partial_{sn} \odot \partial_{cn} \odot \partial_{wn} \odot \partial_{fn}\right) \times X_{ODConv} \tag{1}$$

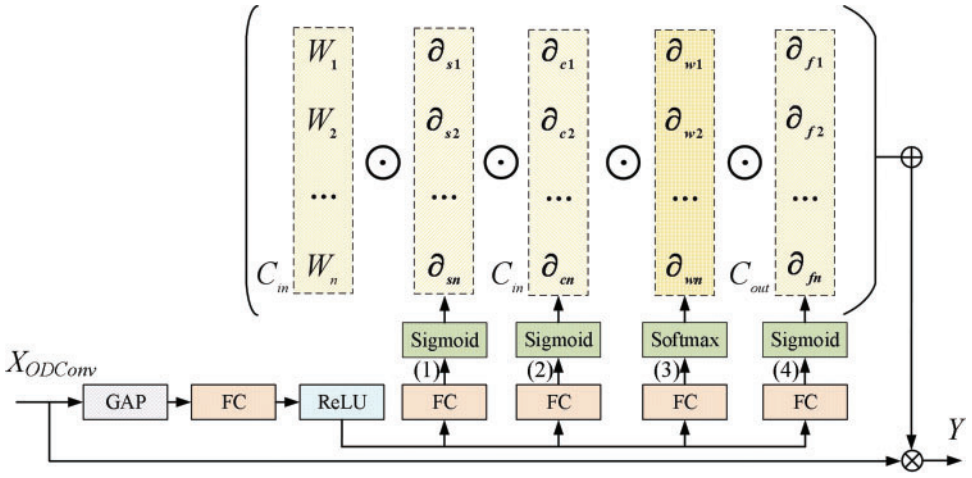


Figure 2: ODConv structure diagram

By amalgamating four distinct attention dimensions, the ODConv enhances its feature extraction capabilities in Yi characters detection. This dynamic approach allows it to adaptively generate convolution kernels based on different inputs, effectively addressing variations among other Yi characters. Furthermore, it is worth noting that the ODConv exhibits a lower computational cost than standard convolution, as deduced from the computation of $dhwk^3 C_{in} C_{out}$ according to Eq. (2).

$$M_{ODConv} = dhwk^3 C_{in} C_{out} + dhwC_{in} + \frac{C_{in}\left(2C_{in} + C_{out} + k^3 + n\right)}{r} + k^3 C_{in}\left(1 + C_{out} + 2nC_{out}\right) \tag{2}$$

where the number of convolution kernels n is configured to be 4 and r is set to 1/16.

3.2 Feature Pyramid Network Fusion Module

This module serves as a feature extraction and fusion tool for image processing and computer vision tasks. In particular, it addresses the challenge of fusing Yi character images at different scales. The fundamental concept revolves around merging feature pyramids derived from multiple levels to capture semantic information across different scales, ultimately improving the overall performance of the model. The network architecture uses a combination of bottom-up, top-down, and horizontally

linked multi-level fusion techniques to improve the feature representation of Yi character images. The FPN structure is visually represented in Fig. 3.

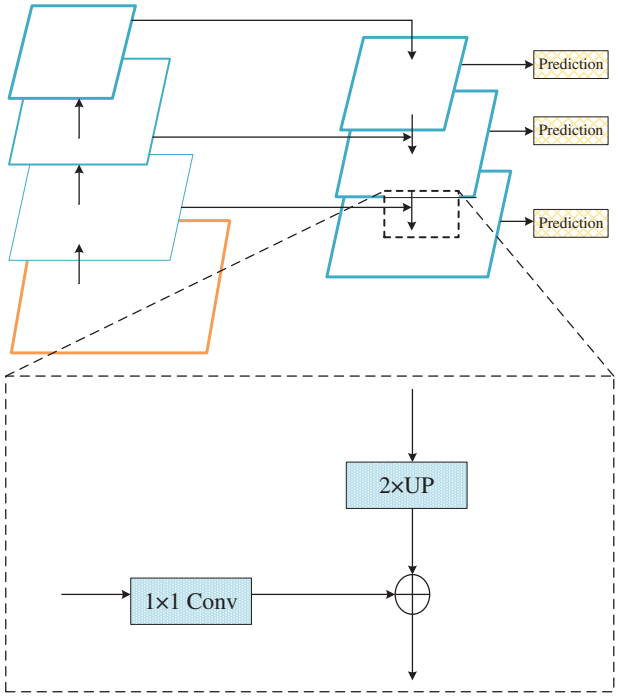


Figure 3: FPN structure diagram

In the bottom-up stage, we first use a convolution kernel to extract image features of Yi characters. Second, the image size is gradually reduced to half of the original size by operations such as convolution and pooling, and the generated image features of Yi characters are fused into feature maps. In the top-down phase, the top-level feature map is up-sampled to match the size of the feature map in the previous phase. It is worth noting that this method uses high-level semantics and bottom-up spatial positioning information to achieve bidirectional information fusion. In the whole process, horizontal join and convolutional fusion are used for feature fusion and prediction.

3.3 DB Prediction Module

An end-to-end learning approach is used in the prediction module and the core network structure, which is the DB used for character recognition. The structure is shown in Fig. 4, the output features from the feature pyramid network fusion module serve as inputs to the prediction module, generating both a prediction map and a threshold map. These resulting maps are subsequently subjected to a DB operation, culminating in the creation of an approximate binarization map. Within the prediction module, the input feature maps are subjected to a series of operations, including convolutional layers, pooling layers, layer normalization, and more. These operations collectively extract high-level feature expressiveness, which aids in the detection of text regions within Yi character images.

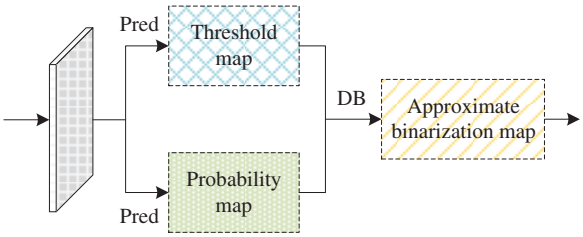


Figure 4: DB prediction module

Since the standard binarization step function is not differentiable, direct optimization is not possible. To enable differentiable end-to-end training, a considerably approximate DB function replaces the original standard binarization function. Subsequently, the text bounding box is derived from the approximate binarization map using the box formulation module. At the same time, this approximated step function is fine-tuned and optimized during the training process. This enables the network to acquire an improved binarization strategy through Eq. (3) as follows:

$$\widehat{B}_{i,j} = \frac{1}{1 + e^{-k\left(P_{i,j} - T_{i,j}\right)}} \tag{3}$$

where $\widehat{B}_{i,j}$ denotes the approximate binarization map pixels, $P_{i,j}$ represents the probability map pixels, and $T_{i,j}$ stands for the threshold map pixels. The DB method is implemented as a Sigmoid function with coefficients k, which takes values within the range (0, 1).

3.4 Loss Function

The loss function L in of the ODC_DBNet model is comprised of the weighted sum of three components: the probability map loss L_s, the threshold map loss L_t, and the approximate binarization map loss L_b, as illustrated in Eq. (4). In Eqs. (5) and (6), the probabilistic map loss L_s, the threshold map loss L_t, and the approximate binarization map loss L_b are detailed as follows:

$$L = L_s + \alpha \times L_b + \beta \times L_t \tag{4}$$

$$L_s = L_b = \sum_{i \in S_l} y_i log x_i + (1 - y_i)\, log\, (1 - x_i) \tag{5}$$

$$L_t = \sum_{i \in R_d} \left| y_i^* - x_i^* \right| \tag{6}$$

where the binary cross-entropy loss function is used for calculating the probability map loss L_s and the approximate binary map loss L_b. The threshold map loss L_t utilizes the L1 loss to compute the distance between the predicted values and the labels. The weighting factors α and β are assigned values of 1 and 10, respectively. Here, S_l represents the set of positive and negative samples with a proportion of 1:3.

4 Yi Character Recognition Based on Shuffle Attention

MobileNetV3 is a lightweight deep learning model characterised by a simple network structure and relatively few parameters. This model is based on the reversible residual structure of the deep separable convolution MobileNetV2 of MobileNetV1. Based on MobileNetV3, we propose a Yi character recognition method with shuffle attention, called SA_MobileNetV3. Shuffle attention

solves the problem of information loss caused by insufficient feature extraction ability of the SE module. The addition of the shuffle attention module of our method shows a positive impact on the task of Yi character recognition, such as improving the network's perception ability of Yi character features, helping to model the relationship between different spatial positions in the input images, and improving the performance of Yi characters recognition. The overall structure of SA_MobileNetV3 is shown in Fig. 5.

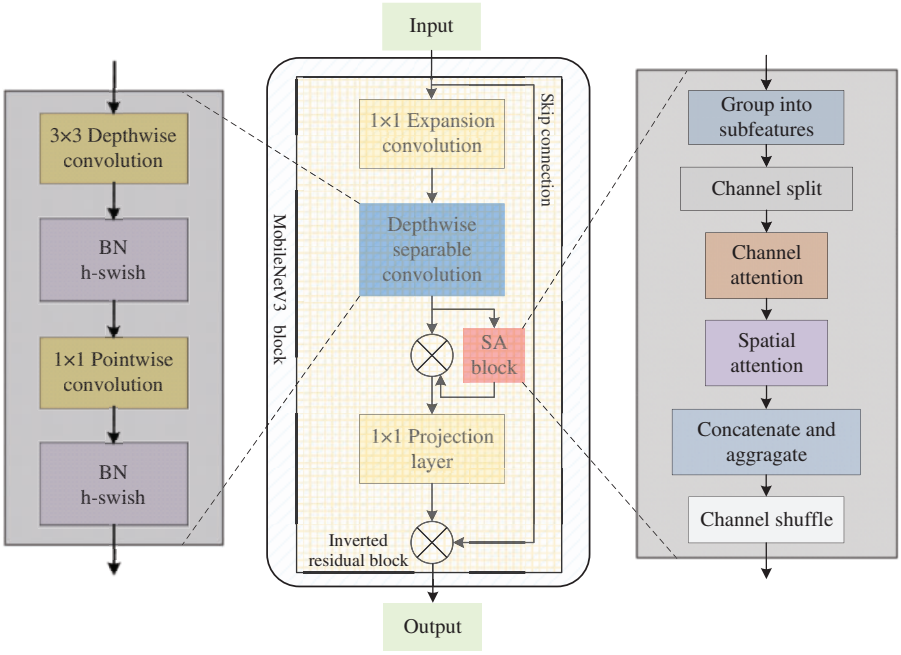


Figure 5: Overall structure diagram of SA_MobileNetV3

4.1 Reversible Residual Structure

As illustrated in Fig. 5, the SA_MobileNetV3 block comprises a central module called the reversible residual structure. This structure encompasses both the depth separable convolution and the SA module. The concept of the reversible residual block draws inspiration from the bottleneck block, where reversible residual concatenation is employed to link input and output features within the same channel. The approach enhances feature representation and reduces memory consumption. Reversible residuals are a lightweight network structure designed to maintain complete reversibility of information flow during both forward and backward propagation in neural networks The efficiency of training and model parameter optimization is optimized by this design. The reversible residual block conducts convolution operations on input features through channel-by-channel depth separable convolution. Subsequently, it combines the convolved elements with the original input to enable reversible information transfer. This architecture allows the network to capture high-level semantic features while preserving the original low-level feature information. As a result, there is an increase in the stability of the model and the speed of convergence during training.

4.2 Depth Separable Convolution

Depth separable convolution is a pivotal component in SA_MobileNetV3 and primarily contributes to its lightweight design. It can be divided into two distinct processes: (1) Deep Convolution: It convolves each channel of the input data with separate convolution kernels, thus capturing the spatial features of the input data while maintaining inter-channel independence. The convolution kernel exclusively processes information from that specific channel for each channel without any cross-channel information transfer. The approach effectively extracts local features within each channel. The outcome of deep convolution is a collection of deep feature maps, where each feature map corresponds to an individual channel in the input data. These feature maps provide richer local information for subsequent pointwise convolution. This approach improves performance by reducing the number of inter-channel references and computational costs while maintaining effectiveness. (2) Pointwise Convolution: A convolution kernel is employed on the feature map following deep convolution in this phase. It linearly combines the feature information from each channel. This operation enables the transfer and fusion of information between channels and generates a new feature map for each channel through pointwise convolution. The outcome of pointwise convolution boosts inter-channel interaction. Consequently, it supplies more informative inputs to the network operations, effectively achieving efficient feature fusion. The structure of depth separable convolution is depicted in Fig. 6.

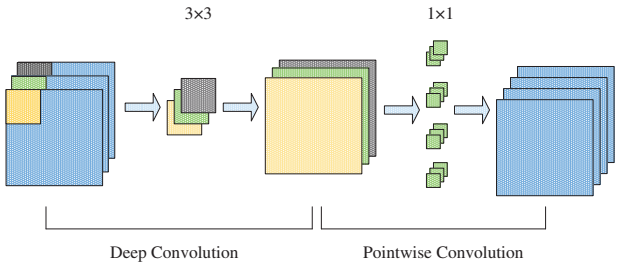


Figure 6: Structure diagram of depth separable convolution

4.3 Shuffle Attention

The standard convolutional neural network receptive field is limited and may miss specific local details. In the context of Yi characters recognition, the extraction of local detailed features from Yi characters is of particular importance. Adopting the shuffle-based attention module to capture the dependencies between spatial and channel numbers becomes crucial to significantly improve the network's feature extraction capabilities when processing Yi characters. The SA module emphasizes local features at a spatial level and efficiently facilitates information interaction across channels. Consequently, it comprehensively captures crucial details of the Yi characters. Combined with the standard attention module, it is expected to lead to improved performance in the Yi character recognition task.

In summary, the introduction of an SA module in Yi character recognition can improve the network's ability to capture local and complex features. While combining both attention mechanisms can improve performance, it is important to consider the increased computational burden. To tackle this challenge, the SA module utilizes a shuffle unit to effectively integrate the two attention mechanisms. The structure of the shuffle attention module is depicted in Fig. 7.

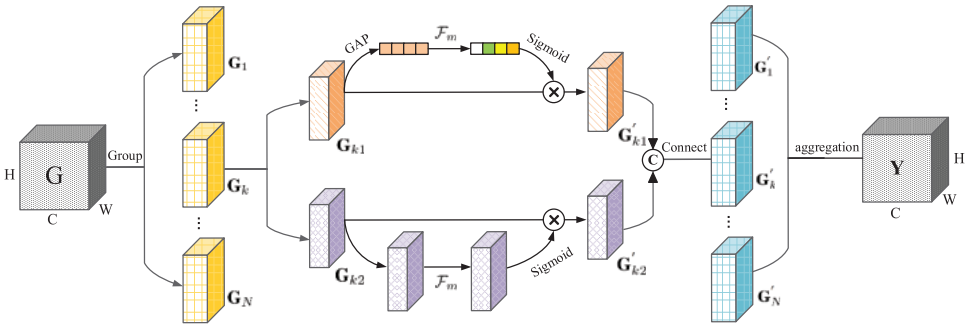


Figure 7: Shuffle attention module

In particular, to capture pixel-level representations and spatial channel dependencies, permutation attention employs a spatial channel attention mechanism for each sub-feature. The feature map G is initially partitioned into groups along the channel number, resulting in $G = [G_1, \ldots, G_N]$, $G \in \mathbb{R}^{C\times H\times W}$, and $G_k \in \mathbb{R}^{C/N\times H\times W}$as the input vectors and the partial feature tensor. Here, C, H, and W respectively denote the number of channels, height, and width. Subsequently, G_k is divided into two branches, $G_{k1}, G_{k2} \in \mathbb{R}^{C/2N\times H\times W}$, along the channel number. As depicted in Fig. 6, the upper branch employs the channel attention mechanism to capture the channel dependency relationships. At this stage, to maintain a lightweight model and prevent an excessive increase in computational parameters, the SA module utilizes global average pooling to calculate the channel-level statistics by considering all elements within each channel. Furthermore, the gated attention mechanism with an activation function generates compact features. The output of the channel attention can be derived using Eq. (7).

$$G_{k1}{}' = \sigma\left(W_1 \cdot Gap\left(G_{k1}\right) + b_1\right) \cdot G_{k1} \tag{7}$$

where W_1 and b_1 represent the parameters of the feature matrix $\mathbb{R}^{C/2N\times 1\times 1}$, while W_1 is used to scale h, and b_1 introduces a particular shift h. $\sigma(\cdot)$ corresponds to the Sigmoid activation function. Additionally, the spatial attention mechanism complements the channel attention mechanism. Another branch employs the spatial attention mechanism to identify vital informative regions with finer granularity, and it is computed as shown in Eq. (8):

$$G_{k2}{}' = \sigma\left(W_2 \cdot GN\left(G_{k2}\right) + b_2\right) \cdot G_{k2} \tag{8}$$

where W_2, $b_2 \in \mathbb{R}^{C/2N\times 1\times 1}$, and Group Norm (GN) for the calculation of spatial statistics.

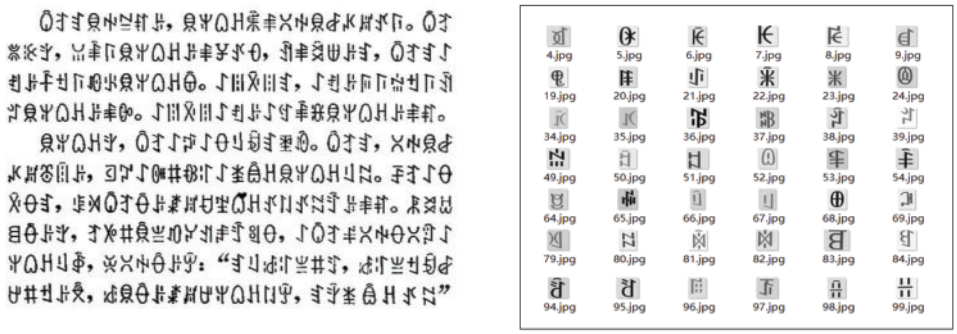
The subsequent step consists of merging the two branches, denoted as $G_k{}' \in \mathbb{R}^{C/N\times H\times W}$. Subsequently, all the sub-features are aggregated, and channel substitution operations are employed to facilitate information flow among different groups. The ultimate output of the SA module maintains the same size as the input. This operation has the advantage of easy integration of the SA module into other networks, thus improving network performance.

5 Evaluation and Comparison of Experimental Results

5.1 Detection and Recognition Datasets

For the Yi character detection dataset, the Yi literature image data we used came from the scanned Yi classics such as "Mamutei", "Leoteyi", "Guiding Meridian" and other books. A total of 200 original data images were collected, with approximately 500 Yi characters per page. The annotation method is semi-automatic annotation and manual annotation. The training set, validation set and test

set are divided in a ratio of 8:1:1, each image is annotated with a single character area, and the size of the annotation area is 32 × 32. A sample of detection dataset is shown in Fig. 8a.



(a) Yi character detection sample (b) Yi character recognition samples

Figure 8: Yi character detection and recognition samples

The Yi character recognition dataset comprises a total of 100,000 datasets, primarily consisting of 1165 automatically generated Yi character datasets and semi-automatically annotated real datasets. The ratio of the training set, validation set and test set are divided in a ratio of 8:1:1, and the size of the Yi character image in the dataset is 32 × 32. The recognition samples are shown in Fig. 8b.

5.2 Assessment Metrics

5.2.1 Detection Indicators

The most frequently utilized evaluation metrics in detection tasks include Precision, Recall, and the composite metric F-measure. Their predictions and labels are presented in Table 1.

Table 1: Confusion matrix

Tagged value	Definition of veracious	Definition of fallacious
Veracious label	TP	FN
Fallacious label	FP	TN

Precision is the ratio of correctly recognized text boxes to all ones recognized. At the same time, recall is the ratio of all correctly labelled text boxes accurately predicted by the model. The accuracy and recall expressions are provided in Eqs. (9) and (10):

$$Precision = \frac{TP}{TP + FP} \tag{9}$$

$$Recall = \frac{TP}{TP + FN} \tag{10}$$

In detection tasks, assessing a model solely based on accuracy and recall metrics is restrictive. To more comprehensively evaluate the model's effectiveness, a composite metric F-measure is employed to gauge the algorithm's performance. The expression Eq. (11) for F-measure is as follows:

$$F - measure = 2\frac{Precision * Recall}{Precision + Recall} \tag{11}$$

5.2.2 Recognition Indicators

Two metrics are employed in the recognition phase to assess the effectiveness of Yi character recognition: accuracy and processing time. The accuracy formula is provided in Eq. (12).

$$ACC = \frac{TP + TN}{TP + TN + FN + FP} \tag{12}$$

5.3 Experimental Environment and Network Parameters

The method described in this paper is implemented using the efficient PaddlePaddle library, all running on Python 3.8.0 with the specific experimental setup detailed in Table 2.

Table 2: Experimental environment

Experimental details	Detailed information
Operating system	Ubuntu 20.04.5
Processing unit	Intel (R) Xeon (R) Silver 4210R CPU @ 2.40 GHz
Random access memory	64 GB
Graphics processing unit	Nvidia RTX3090Ti (24 GB)

5.4 Experimental Results and Analysis

5.4.1 Detection Experiment

To validate the method outlined in this paper, we employ various text detection models for comparison, including CNN [32], Faster Region-CNN (Faster R-CNN) [33], PSENet [13], FCENet [14], and DBNet [8]. Their corresponding experimental results are presented in Table 3.

Table 3: Comparison experiment

Model name	Precision	Recall	F-measure
CNN [32]	0.891	0.771	0.827
Faster R-CNN [33]	0.923	0.835	0.877
PSENet [13]	0.924	0.881	0.902
FCENet [14]	0.927	0.880	0.903
DBNet [8]	0.961	0.891	0.924
ODC_DBNet	**0.975**	**0.902**	**0.937**

Table 3 demonstrates that when using the same datasets and parameters for detecting Yi characters, this paper's method, ODC_DBNet, exhibits improvements across accuracy Precision, Recall, and F-measure. Due to the utilization of traditional machine learning algorithms with manual feature selection, the detection outcomes are often subject to human adjustments, and interference from background images can result in variations in the detection performance. As a result, we have employed various deep-learning algorithms for comparison with the method proposed in this paper. The enhanced ODC_DBNet detection model presented in this paper exhibits improvements in Precision,

Recall, and F-measure by 1.4%, 1.1%, and 1.3%, respectively, compared to the DBNet model, which ranks second in effectiveness.

Table 4 shows an ablation study demonstrating the performance of different modules. ODC_DB Net-woODC, ODC_DBNet-woFP, and ODC_DBNet-woDB indicate instances where the proposed ODC_DBNet excludes the ODConv feature extraction module, the feature pyramid network fusion module, and the DB prediction module, respectively. It can be observed that in these cases, the F-measure decreased by 1.3%, 0.8%, and 1.1%, respectively. Among them, the ODConv feature extraction module can improve the model's feature representation ability for Yi character images, the feature pyramid network fusion module can improve the model's ability to detect text at different scales, and the DB prediction module acts to generate the position and shape of text boxes to provide more accurate text regions for subsequent text recognition tasks.

Table 4: Ablation experiments under different modules

Methods	Precision	Recall	F-measure
ODC_DBNet-woODC	0.961	0.891	0.924
ODC_DBNet-woFP	0.969	0.892	0.929
ODC_DBNet-woDB	0.965	0.890	0.926
ODC_DBNet	0.975	0.902	0.937

The various backbone networks have been compared with the ODC feature extraction network (ResNet-18+ODC) outlined in this paper, and the results are presented in Table 5. In contrast to ResNet-50, ResNet-45, and the original ResNet-18 network, ResNet-18+ODC improves Precision, Recall, and F-measure. Notably, the F-measure of ResNet-18+ODC in this paper demonstrates a significant improvement of 1.6% compared to the next best-performing ResNet-18 model.

Table 5: Ablation experiments under different backbone

Backbone	Precision	Recall	F-measure
ResNet-50	0.890	0.843	0.866
ResNet-45	0.942	0.883	0.912
ResNet-18	0.961	0.891	0.921
ResNet-18+ODC	**0.975**	**0.902**	**0.937**

For the Yi character detection task, the adoption of different backbone networks can effectively verify that our proposed method has some reliability. Table 5 shows that the lower the number of ResNet layers, the higher the F-measure. Due to the low complexity of the Yi characters detection dataset, ResNet-50 and ResNet-45 cannot detect Yi characters well. As can be seen from Table 5, the deeper the layer number of the network, the worse the effect. Therefore, we chose ResNet-18 as the feature extraction network to improve the detection accuracy. Fig. 9 illustrates the detection results under four different backbone network models. Specifically, Figs. 9a–9d employ the feature extraction networks ResNet-50, ResNet-45, ResNet-18, and ResNet-18+ODC, respectively. Yi characters exhibit complex shapes and varying width-to-height ratios. The method presented in this paper, ResNet-18+ODC, outperforms other methods regarding both reasonable computational cost and improved

detection performance. In summary, the ResNet-18+ODC approach employed in this paper enhances detection performance while maintaining a reasonable computational cost compared to alternative methods.

(a) ResNet-50 (b) ResNet-45

(c) ResNet-18 (d) ResNet-18+ODC

Figure 9: Visualisation of Yi characters detection

5.4.2 Recognition Experiment

To confirm the superiority of the SA_MobileNetV3 model proposed in this paper for the Yi characters recognition task, we use CNN [32] as the baseline model and compare it with ResNet-18 [9], NRTR (No Recurrence Sequence-to-Sequence Text Recognition) [34], MobileNetV3 [11], and the SA_MobileNeV3 model. Notably, the SA_MobileNetV3 approach in this paper substantially improves accuracy, surpassing the two models with higher accuracy, NRTR and MobileNetV3, by 6.3% and 2.3%, respectively. When evaluating model performance, it is essential to consider factors beyond recognition accuracy, such as the training time. These metrics help measure both model performance and efficiency. The experimental results for different models are presented in Table 6.

Table 6: Comparison experiment

Model name	Accuracy	Training time	Parameter quantity
CNN [32]	0.712	40 h	35.41 M
ResNet-18 [9]	0.846	45 h	11.69 M
NRTR [34]	0.905	36 h	31.56 M
MobileNetV3 [11]	0.945	**35 h**	**3.31 M**
SA_MobileNetV3	**0.968**	**35 h**	3.62 M

Evaluated on the same test set, Table 7 demonstrates that the SA_MobileNetV3 model introduced in this paper significantly outperforms the first three comparison models regarding test accuracy. However, its training time is comparable to MobileNetV1 [35]. Furthermore, compared to MobileNetV2

[36] and MobileNetV3 [11], it achieves a noteworthy improvement in accuracy by 4.6% and 2.3%, respectively, while maintaining similar time consumption.

Table 7: Ablation experiment

Model name	Accuracy	Training time
MobileNetV1 [35]	0.896	**30 h**
MobileNetV2 [36]	0.922	35 h
MobileNetV3 [11]	0.945	35 h
SA_MobileNetV3	**0.968**	35 h

6 Conclusion

In response to the difficulty of text detection and recognition in Yi character images, this paper proposes a deep learning-based method for Yi character detection and recognition. In the detection stage, the Omni-dimensional Dynamic Convolution module is embedded into the DBNet model to replace the ordinary convolution in the ResNet-18 backbone network, and this dynamic convolution module dynamically adapts the convolution kernel parameters to improve the accuracy of Yi character detection. During the recognition stage, a combination of channel and spatial alternating attention modules is integrated into the MobileNetV3 recognition framework. Its function is to adjust the attention of features between different levels or channels of the model to improve the model's performance. The experimental results show that the detection and recognition model introduced in this paper has robust capabilities in detecting and recognizing text in practical scenarios. In the future, it is necessary to explore in depth how the Transformer [37] mechanism can be introduced into related tasks. The design of an end-to-end text detection and recognition framework with high efficiency will be the focus of our next major research effort.

Acknowledgement: The authors extend their gratitude to the anonymous reviewers and the editor for their invaluable suggestions, which have significantly enhanced the quality of this article.

Funding Statement: The work was supported by the National Natural Science Foundation of China (61972062, 62306060) and the Basic Research Project of Liaoning Province (2023JH2/101300191) and the Liaoning Doctoral Research Start-Up Fund Project (2023-BS-078) and the Dalian Academy of Social Sciences (2023dlsky028).

Author Contributions: The authors confirm contribution to the paper as follows: study conception and design: H.P. Sun, X.Y. Ding; data collection: H.P. Sun, J. Sun, H. Yu; analysis and interpretation of results: H.P. Sun, X.Y. Ding, J. Sun, J.X. Zhang, H. Yu; draft manuscript preparation: H.P. Sun, X.Y. Ding, J.X. Zhang, H. Yu. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The authors do not have permission to share the data.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] J. Yao, X. Yan, and S. Liu, "Linguistic landscape in Liangshan Yi autonomous prefecture: The case of an ethnic minority region in China," *Int. J. Multiling.*, vol. 20, no. 2, pp. 169–188, 2023. doi: 10.1080/14790718.2020.1800018.

[2] X. Jia, W. Gong and J. Yuan, "Handwritten Yi character recognition with density-based clustering algorithm and convolutional neural network," in *Proc. IEEE Int. Conf. Comput. Sci. Eng. and IEEE Int. Conf. Embed. Ubiquitous Comput.*, Guangzhou, China, 2017, vol. 1, pp. 337–341.

[3] C. L. Liu, S. Jaeger, and M. Nakagawa, "Online recognition of Chinese characters: The state-of-the-art," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 2, pp. 198–213, 2004. doi: 10.1109/TPAMI.2004.1262182.

[4] J. Memon, M. Sami, R. Khan, and M. Uddin, "Handwritten optical character recognition (OCR): A comprehensive systematic literature review (SLR)," *IEEE Access*, vol. 8, pp. 142642–142668, 2020. doi: 10.1109/ACCESS.2020.3012542.

[5] R. Ptucha, F. Such, S. Pillai, F. Brockler, V. Singh and P. Hutkowski, "Intelligent character recognition using fully convolutional neural networks," *Pattern Recognit.*, vol. 88, pp. 604–613, 2019. doi: 10.1016/j.patcog.2018.12.017.

[6] Y. Sun, C. Zhang, Z. Huang, J. Liu, J. Han and E. Ding, "Textnet: Irregular text reading from images with an end-to-end trainable network," in *Proc. 14th Asian Int. Conf. Comput. Vis. (ACCV)*, Perth, Australia, 2019, pp. 83–99.

[7] A. Singh, G. Pang, M. Toh, J. Huang, W. Galuba and T. Hassner, "Textocr: Towards large-scale end-to-end reasoning for arbitrary-shaped scene text," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Pattern Recognit.*, Nashville, TN, USA, 2021, pp. 8802–8812.

[8] M. Liao, Z. Wan, C. Yao, K. Chen, and X. Bai, "Real-time scene text detection with differentiable binarization," *in Proc. AAAI Int. Conf. Artif. Intell.*, 2020, vol. 34, no. 7, pp. 11474–11481. doi: 10.1609/aaai.v34i07.6812.

[9] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, Las Vegas, NV, USA, 2016, pp. 770–778.

[10] C. Li, A. Zhou, and A. Yao, "Omni-dimensional dynamic convolution," arXiv:2209.07947v1, 2022.

[11] A. Howard *et al.*, "Searching for MobileNetV3," in *Proc. IEEE/CVF Int. Conf. on Compt. Vis.*, Seoul, Korea, 2019, pp. 1314–1324.

[12] Q. Zhang and Y. Yang, "SA-Net: Shuffle attention for deep convolutional neural networks," in *Proc. ICASSP Int. Conf. on Acoustics, Speech and Signal Processing*, Toronto, ON, Canada, 2021, pp. 2235–2239.

[13] W. Wang *et al.*, "Shape robust text detection with progressive scale expansion network," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Pattern Recognit.*, Long Beach, CA, USA, 2019, pp. 9336–9345.

[14] Y. Zhu, J. Chen, L. Liang, Z. Kuang, L. Jin and W. Zhang, "Fourier contour embedding for arbitrary-shaped text detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Pattern Recognit.*, Nashville, TN, USA, 2021, pp. 3123–3131.

[15] B. Shi, X. Bai, and C. Yao, "An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 11, pp. 2298–2304, 2016. doi: 10.1109/TPAMI.2016.2646371.

[16] M. Liao *et al.*, "Scene text recognition from two-dimensional perspective," in *Proc. AAAI Int. Conf. Artif. Intell.*, 2019, vol. 33, no. 1, pp. 8714–8721. doi: 10.1609/aaai.v33i01.33018714.

[17] Q. Wang, Y. Huang, W. Jia, X. He, M. Blumenstein and Y. Lu, "FACLSTM: ConvLSTM with focused attention for scene text recognition," *Sci. China Inf. Sci.*, vol. 63, pp. 1–14, 2020. doi: 10.1007/s11432-019-2713-1.

[18] C. Zhang, W. Wang, H. Liu, G. Zhang, and Q. Lin, "Character detection and segmentation of historical Uchen Tibetan documents in complex situations," *IEEE Access*, vol. 10, pp. 25376–25391, 2022. doi: 10.1109/ACCESS.2022.3151886.

[19] L. Ma, C. Long, L. Duan, X. Zhang, Y. Li and Q. Zhao, "Segmentation and recognition for historical Tibetan document images," *IEEE Access*, vol. 8, pp. 52641–52651, 2020. doi: 10.1109/ACCESS.2020.2975023.
[20] Z. Wang, S. Lu, M. Wang, X. Wei, and Y. Qi, "AMRE: An attention-based CRNN for Manchu word recognition on a woodblock-printed dataset," in *Proc. Int. Conf. Neural Inf. Process.*, Cham, Switzerland, 2022, pp. 267–278.
[21] R. Zheng, M. Li, J. He, J. Bi, and B. Wu, "Segmentation-free multi-font printed Manchu word recognition using deep convolutional features and data augmentation," in *Proc. 11th Int. Cong. Image and Signal Process., BioMed. Eng. Inf.*, Beijing, China, 2018, pp. 1–6.
[22] D. Zhang, Y. Liu, Z. Wang, and D. Wang, "OCR with the deep CNN model for ligature script-based languages like Manchu," *Sci. Programming*, vol. 2021, pp. 1–9, 2021. doi: 10.1155/2021/5520338.
[23] X. Su, G. Gao, H. Wei, and F. Bao, "A knowledge-based recognition system for historical Mongolian documents," *Int. J. Doc. Anal. Recognit.*, vol. 19, pp. 221–235, 2016. doi: 10.1007/s10032-016-0267-1.
[24] M. Mahmut and Y. Genc, "A deep-learning approach to optical character recognition for Uighur language," in *Proc. 2019 Int. Conf. Adv. Comput. Commun. Control*, Mumbai, India, 2019, pp. 1–6.
[25] Y. Xu and J. Xue, "A Segmentation-driven handwritten uighur word recognition algorithm based on feedback structure," in *Proc. 10th IEEE Trans. Softw. Eng. Service Sci.*, Beijing, China, 2019, pp. 419–423.
[26] S. Fang, H. Xie, Z. Chen, S. Zhu, X. Gu and X. Gao, "Detecting Uyghur text in complex background images with convolutional neural network," *Multimed. Tools Appl.*, vol. 76, pp. 15083–15103, 2017. doi: 10.1007/s11042-017-4538-8.
[27] Y. H. Jiejue, "The algorithm and implementation of Yi character recognition based on convolutional neural network," in *Proc. 2022 Int. Conf. Netw., Commun. Inf. Tech.*, Manchester, UK, 2022, pp. 346–349.
[28] S. Chen, X. Han, X. Wang, and H. Ma, "A recognition method of ancient Yi script based on deep learning," *Int. J. Comput. Inf. Eng.*, vol. 13, no. 9, pp. 504–511, 2019. doi: 10.5281/zenodo.3462062.
[29] Z. Yin, S. Chen, D. Wang, X. Peng, and J. Zhou, "Yi characters online handwriting recognition models based on recurrent neural network: RnnNet-Yi and ParallelRnnNet-Yi," in *Proc. SPRINGER Int. Conf. Front. in Handwriting Recognit.*, Cham, Switzerland, 2022, pp. 375–388.
[30] B. Yang, G. Bender, Q. V. Le, and J. Ngiam, "CondConv: Conditionally prameterized convolutions for efficient inference," *Adv. Neural Inf. Process. Syst.*, vol. 32, no. 117, pp. 1307–1318, 2019. doi: 10.48550/arXiv.1904.04971.
[31] Y. Chen, X. Dai, M. Liu, D. Chen, L. Yuan and Z. Liu, "Dynamic convolution: Attention over convolution kernels," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Pattern Recognit.*, Seattle, WA, USA, 2020, pp. 11030–11039.
[32] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Adv. Neural Inf. Process. Syst.*, vol. 25, pp. 1097–1105, 2012. doi: 10.1145/3065386.
[33] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *Adv. Neural Inf. Process.*, vol. 28, pp. 91–99, 2015. doi: 10.1109/TPAMI.2016.2577031.
[34] F. Sheng, Z. Chen, and B. Xu, "NRTR: A no-recurrence sequence-to-sequence model for scene text recognition," in *Proc. IEEE Int. Conf. Doc. Anal. Recognit.*, Sydney, NSW, Australia, 2019, pp. 781–786.
[35] A. Howard *et al.*, "MobileNets: Efficient convolutional neural networks for mobile vision applications," arXiv:1704.04861, 2017.
[36] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, Salt Lake, UT, USA, 2018, pp. 4510–4520.
[37] K. Han, A. Xiao, E. Wu, J. Guo, C. Xu and Y. Wang, "Transformer in transformer," *Adv. Neural Inf. Process.*, vol. 34, pp. 15908–15919, 2021.