electronics

MDPI

Article

SGooTY: A Scheme Combining the GoogLeNet-Tiny and YOLOv5-CBAM Models for Nüshu Recognition

Yan Zhang and Liumei Zhang *

School of Computer Science, Xi'an Shiyou University, Xi'an 710065, China; yan_zhang0409@163.com
* Correspondence: zhangliumei@xsyu.edu.cn; Tel.: +86-180-9233-0186

Abstract: With the development of society, the intangible cultural heritage of Chinese Nüshu is in danger of extinction. To promote the research and popularization of traditional Chinese culture, we use deep learning to automatically detect and recognize handwritten Nüshu characters. To address difficulties such as the creation of a Nüshu character dataset, uneven samples, and difficulties in character recognition, we first build a large-scale handwritten Nüshu character dataset, HWNS2023, by using various data augmentation methods. This dataset contains 5500 Nüshu images and 1364 labeled character samples. Second, in this paper, we propose a two-stage scheme model combining GoogLeNet-tiny and YOLOv5-CBAM (SGooTY) for Nüshu recognition. In the first stage, five basic deep learning models including AlexNet, VGGNet16, GoogLeNet, MobileNetV3, and ResNet are trained and tested on the dataset, and the model structure is improved to enhance the accuracy of recognising handwritten Nüshu characters. In the second stage, we combine an object detection model to re-recognize misidentified handwritten Nüshu characters to ensure the accuracy of the overall system. Experimental results show that in the first stage, the improved model achieves the highest accuracy of 99.3% in recognising Nüshu characters, which significantly improves the recognition rate of handwritten Nüshu characters. After integrating the object recognition model, the overall recognition accuracy of the model reached 99.9%.

Keywords: Nüshu recognition; deep learning; computer vision; CNN; SGooTY

check for updates

Citation: Zhang, Y.; Zhang, L. SGooTY: A Scheme Combining the GoogLeNet-Tiny and YOLOv5-CBAM Models for Nüshu Recognition. *Electronics* **2023**, *12*, 2819. https://doi.org/10.3390/electronics12132819

Academic Editor: Hamed Mozaffari

Received: 12 May 2023
Revised: 22 June 2023
Accepted: 23 June 2023
Published: 26 June 2023

Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Nüshu [1,2] is a writing system unique to Jiangyong County, Hunan Province, China, and is also known as "women's script" or "Nüshu characters". It originates from folk literature and women's traditions, expresses women's emotions and psychology, and has unique artistic and cultural values. Nüshu characters are concise, beautiful, and highly artistic, and are used for communication and inheritance among women. The style and content of Nüshu are also regarded as symbols and expressions of women's culture [3]. In 2006, Nüshu culture was included in the first batch of the National Intangible Cultural Heritage List approved by the State Council. With the development of and changes in society, and at the same time influenced by foreign cultures, Nüshu faces the threat of gradual marginalization and even extinction [4]. Therefore, it is necessary to use information and digital technology to protect the cultural heritage of Nüshu. Figure 1 shows Nüshu-related works, including calligraphy, fans, and cloth patches.

Due to the fact that Nüshu is usually only learned and passed down by women, very few people are able to recognize Nüshu characters, resulting in enormous pressure to protect and pass on Nüshu. In recent years, researchers of Nüshu have studied and organized the basic Nüshu characters, which amount to less than 500, through the application of character position theory. Currently, there are few achievements in the recognition of handwritten Nüshu characters and no publicly available Nüshu dataset for use in information digitization technology. In recent years, with the rapid development of deep learning technology, deep learning models have achieved outstanding results in image classification and recognition. Increasingly, researchers have attempted to apply convolutional neural networks (CNNs) to text recognition tasks. Jeow [5] proposed a text classification algorithm that uses dual-modal information extraction and long short-term memory recurrent neural networks. Liu Min et al. [6] applied deep learning models to the recognition of oracle bone inscriptions and achieved good results. Aneja et al. [7] proposed the use of deep convolutional neural networks in transfer learning for the recognition of handwritten Sanskrit characters, achieving an accuracy rate of 98%. It can be seen that deep learning algorithms have been widely used in the field of text recognition, and in order to better preserve the culture of Nüshu, this paper will also explore the research of deep learning in the recognition of handwritten characters in Nüshu.

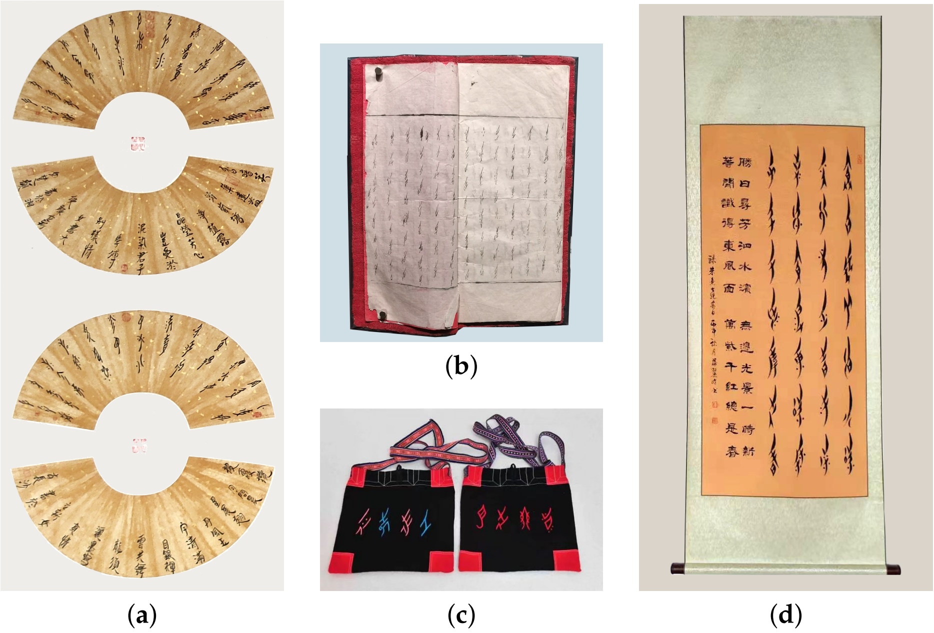

(**a**) (**b**) (**c**) (**d**)

Figure 1. Nüshu artifacts. (**a**) Calligraphy fan cover. (**b**) "Third day missive" book. (**c**) Bag stitch. (**d**) Calligraphy scroll. Where (**a**,**b**,**d**) were photographed by Yan Zhang from the Hunan Museum, and (**c**) comes from the Yong Zhou City Government official website.

In this paper, we explore the application of CNNs in the recognition of handwritten Nüshu characters. Our main contributions are summarized as follows:

(1) In order to solve the problem of scarce resources for Nüshu datasets, in this paper, we first established a reliable, clear image and large-scale handwritten Nüshu dataset, called HWNS2023 (handwritten Nüshu 2023). This dataset contains ten characters from the Nüshu international coding standard character set 1B180 to 1B189, with a total of 2364 samples. Considering the diversity of handwriting styles and different lighting conditions of handwritten Nüshu characters, we also use rotation, brightness adjustment, and noise addition to expand the dataset to approximately 6800 images. The HWNS2023 dataset can be found on https://www.kaggle.com/datasets/yanz0409/hwns2023 (accessed on 1 June 2023).

(2) A scheme for handwritten Nüshu recognition (SGooTY) is proposed in order to aid in the preservation and organization of cultural heritage. The scheme includes two deep learning models to achieve better Nüshu character detection and recognition. In the first stage, an improved GoogLeNet-tiny model is applied to identify and classify handwritten Nüshu character. However, there may still be some Nüshu images that are not recognized. In the second stage, YOLOv5-CBAM is used for image recognition of undetected handwritten Nüshu character images, and the recognition results are outputted in the end. The proposed system achieves a recognition accuracy of 99.9%

on the WHNS2023 dataset and effectively solves the problem of handwritten Nüshu character recognition.

(3) Five classic recognition methods, including AlexNet, VGGNet16, GoogLeNet, MobileNetV3, and ResNet were compared in experiments, and the two models with the best results, AlexNet and GoogLeNet, were selected. New models AlexNet-BN, AlexNet-SC, AlexNet-LR, and GoogLeNet-tiny were generated by adjusting network parameters and modifying network layers. Various optimization strategies were used to train and test these new models, and the new models were compared and analyzed to determine the first-stage recognition model. In the second stage, YOLOv3 [8], YOLOv5, and YOLOv7 [9] were trained and tested, and the results were compared to determine that the second-stage network model is YOLOv5. An attention mechanism was introduced into the YOLOv5 backbone network to improve the recognition accuracy of the model.

This paper proposes a two-stage recognition scheme based on convolutional neural networks for recognizing handwritten Nüshu characters, and improves the recognition models in both stages to effectively recognize handwritten Nüshu characters. The structure of this paper is organized as follows. The first part introduces the research background of this paper. The second part discusses related work on character recognition based on convolutional neural networks. The third part introduces the activation function, BN function, and CBAM attention mechanism in the model improvement. The fourth part introduces the dataset used in the experiment, related work on model improvement, and model training optimization methods. The fifth part describes the experimental design, experimental environment, and model training process. The sixth part presents the experimental results. The seventh part concludes and summarizes the research of this paper.

2. Related Work

The recognition technology for Nüshu characters is traditionally divided into holistic text recognition and single character recognition. While holistic techniques in characters such as Roman letters refer to word recognition, in Nüshu or other cases, these techniques refer to partial characters or ligatures (as character boundaries are difficult to determine in Nüshu text). On the other hand, single character recognition refers to character-level recognition in language text, and this classification is commonly used for both printed and handwritten text.

Handwriting recognition has always been one of the most researched topics in computer vision. The problem has been studied extensively for several decades, and many handwriting recognition systems have gradually matured. Handwriting recognition is usually limited by the accuracy of the preceding text recognition and segmentation steps. Inspired by this, Wigington et al. [10] proposed a deep learning model that jointly learns text detection, segmentation, and recognition using images without detection or segmentation annotations. Ptucha et al. [11] proposed a fully convolutional network architecture that outputs symbol streams of arbitrary length from handwritten text. Although promising, there are several limitations to the current results, including computational complexity, dependence on the classifier used, and difficulty in assessing interactions between features. Cilia et al. [12] attempted to overcome some of these drawbacks by adopting a feature-ranking-based technique, considering different univariate measures to generate feature rankings, and proposing a greedy search method to select a subset of features that can maximize classification results. In addition, Choudhury et al. [13] proposed an online handwriting representation method using a multicomponent sine wave model. The multicomponent sine wave model was proposed as a model-based representation for online pen strokes. Pashine et al. [14] performed handwritten digit recognition using support vector machine (SVM), multilayer perceptron (MLP), and convolutional neural network (CNN) models using the MNIST dataset.

With the continued development of deep learning, researchers have applied it to various handwriting recognition tasks. Ly et al. [15] proposed an end-to-end deep convolutional recurrent network (DCRN) model for offline handwritten Japanese text line recognition. Majid et al. [16] proposed an offline Bengali handwriting recognition system using Faster R-CNN for sequential recognition of characters and diacritics. Ali et al. [17] studied intelligent handwriting recognition using a mixed SVM classifier based on CNN architecture and dropout. The authors modeled deep learning architectures that can effectively recognize Arabic handwriting. In addition, Carbune et al. [18] described an online handwriting system that can support 102 languages using a deep neural network architecture.

Despite the development of many sophisticated handwriting recognition systems, Nüshu character recognition remains a challenging problem because humans can write the same information in almost infinite ways. Currently, researchers mainly rely on traditional machine learning methods to learn the features of Nüshu characters by artificially extracting them. For example, Hei et al. [19] proposed an offline handwritten Nüshu character image multi-directional text line extraction method to extract text lines in different directions from Nüshu character images on different media such as fans and paper. However, this method cannot recognize individual Nüshu characters. To obtain individual Nüshu character images, Sun et al. [20] proposed a Nüshu character segmentation algorithm based on a local adaptive thresholding technique. This method can automatically obtain local thresholds, avoid the loss of character image information, and improve the accuracy of Nüshu character image segmentation. In practical image segmentation, this method can effectively reduce the effect of background noise. Wang et al. [21] proposed a statistical–structural character learning algorithm based on hidden Markov models, which considers the stroke relationship of handwritten Nüshu characters to recognize their features, and achieved impressive performance in handwritten Nüshu character recognition.

In recent years, artificial intelligence technologies such as machine learning and deep learning have made breakthroughs, benefiting from their powerful feature extraction capabilities. Deep learning has been widely applied to text recognition, resulting in the development of many outstanding algorithms. However, no researchers have attempted to apply deep learning to Nüshu character recognition tasks. Therefore, this paper attempts to explore the application of convolutional neural networks to the recognition and study of Nüshu characters.

3. Preliminaries

3.1. ReLU and Leaky ReLU

Many deep learning models use the rectified linear unit [22] (ReLU) as their activation function. When ReLU is activated above 0, its partial derivative is always 1, which solves the gradient vanishing problem when the sigmoid activation function input is too large. However, if the input is negative, the derivative of ReLU is always 0, leading to complete saturation, which may result in neurons never being activated. This drawback could slow down the training speed of the network and even reduce the generalization performance of the network when using gradient-based optimization algorithms [23]. The function graph of ReLU is shown in Figure 2.

The difference between the Leaky ReLU [24] and ReLU activation functions is that the partial derivative of Leaky ReLU is not 0 when the input is negative. This solves the problem of neurons not being activated in ReLU, allowing the network to train faster and produce better results. In this paper, we use Leaky ReLU instead of ReLU as the activation function in the improved model. The function graph of Leaky ReLU is shown in Figure 2.

The Leaky ReLU function is formulated as follows:

$$LeakyReLU(x) = \begin{cases} x, x > 0 \\ \alpha x, x < 0 \end{cases} \tag{1}$$

Characteristics of the Leaky ReLU function:

- The Leaky ReLU function adjusts the zero-gradient problem for negative values by giving a very small linear component of x to a negative input of 0.01 x;
- Leaky ReLU helps to extend the range of the ReLU function, which usually has a value of about 0.01 for α;
- The function range of Leaky ReLU is from negative infinity to positive infinity.

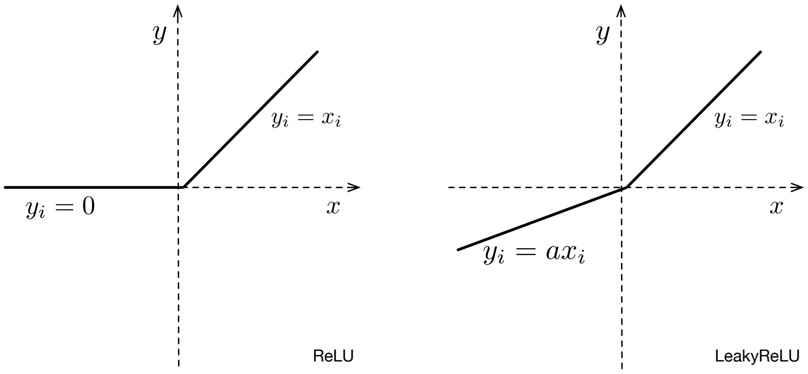


Figure 2. Function images of ReLU and Leaky ReLU. The image of the ReLU function is shown on the left side of this figure, and the image of the Leaky ReLU function is shown on the right side

3.2. Batch Normalization

Batch normalization (BN) [25] introduces learnable parameters γ and β, which perform a second transformation and reconstruction on the normalized values. The following is the key algorithm of the BN layer.

$$y^{(k)} = \gamma^{(k)} \hat{x}^{(k)} + \beta^{(k)} \tag{2}$$

there is a γ and β parameter for each layer and each channel.

$$\gamma^{(k)} = \sqrt{Var[x^{(k)}]} \tag{3}$$

$$\beta^{(k)} = E[x^{(k)}] \tag{4}$$

Thus, we can reconstruct the feature distribution that the original network needs to learn by allowing the network to learn the γ and β parameters on its own. Out of these parameters, only γ and β are trainable, while the mean and variance are calculated from the input data.

3.3. CBAM

CBAM [26,27] is a simple yet effective attention module used for feedforward convolutional neural networks. CBAM sequentially infers a 1D channel attention map Mc with a size of $C \times 1 \times 1$ and a 2D spatial attention map Ms with a size of $1 \times H \times W$, which can be placed in parallel or in sequence.

$$F' = M_c(F) \otimes F \tag{5}$$

$$F'' = M_s(F) \otimes F' \tag{6}$$

where $\otimes$ denotes element multiplication and F'' is the final refinement output.

3.3.1. Channel Attention Module

Channel attention focuses on the meaningful parts of the input image. To efficiently calculate channel attention, compression of the spatial dimensions of the input feature map is required, and average pooling is commonly used to aggregate spatial information. However, maximum pooling collects clues about unique object features, enabling it to infer attention on finer channels. Therefore, both average and max pooled features can be used simultaneously.

$$\begin{aligned} M_c(F) &= \sigma(MLP(AvgPool(F)) + MLP(MaxPool(F))) \\ &= \sigma(W_1(W_0(F^c_{avg})) + W_1(W_0(F^c_{max}))) \end{aligned} \tag{7}$$

F^c_{avg} and F^c_{max} represent the average pooled features and max pooled features, respectively. These two descriptors are forwarded to a shared network to generate the channel attention map M_c. The shared network consists of a multilayer perceptron (MLP) with a hidden layer. To reduce parameter cost, the activation size of the hidden layer is set to $R/C = r \times 1 \times 1$, where R is the dropout rate. After applying the shared network to each descriptor, the output feature vectors are merged using element-wise addition. σ denotes the sigmoid function. This $M_c(F)$ is then element-wise multiplied with F to obtain F'.

3.3.2. Spatial Attention Module

Spatial attention focuses on the most informative parts in a specific region. In calculating spatial attention, the channel axis is pooled using average and max pooling operations and concatenated to generate an effective feature descriptor. Then, a convolutional layer is applied to generate a spatial attention map $M_s(F)$ of size $R \times H \times W$, which encodes the positions that need to be attended or suppressed. The location information of this specific region is encoded as features in the spatial attention map.

$$\begin{aligned} M_s(F) &= \sigma(f^{7\times7}([AvgPool(F); MaxPool(F)])) \\ &= \sigma(f^{7\times7}([(F^s_{avg}; F^s_{max}])) \end{aligned} \tag{8}$$

Specifically, the channel information of the feature map is aggregated using two pooling operations, generating two 2D maps: F^s_{avg} of size 1 × H × W and F^s_{max} of size $1 \times H \times W$. σ represents the sigmoid function, and $f^{7\times7}$ represents a convolution operation with a filter size of 7 × 7.

4. Materials and Methods

4.1. Dataset Preparation

4.1.1. Sample Acquisition and Processing

The handwritten Nüshu character images used in this paper were created by team members. We referred to the Nüshu International Code Standard Character Set 1B180-1B189 to create 2364 handwritten Nüshu character images, named HWNS2023 (handwritten Nüshu 2023), under natural light and night environments. Among them, 1000 images are used for the stage one model training dataset and 1364 images are used for the stage two end-to-end recognition model training dataset. The initial dataset is shown in Figure 3 and the relevant explanatory notes are shown in Table 1. Due to the inevitable occurrence of noise during image acquisition and the certain requirements of image size in the model training process, after completing the original data acquisition, the dataset is preprocessed through denoising and normalization. The process is shown in Figure 4.

4.1.2. Data Augmentation

The limited number of samples in the dataset resulted in low recognition accuracy, so we expanded the dataset to improve the effectiveness of the recognition task. We performed operations such as rotation, adjustment of brightness, and adding Gaussian noise to the 1000 images used in the first phase of the HWNS2023 dataset. The total number of samples reached 5500 images, as shown in Figure 5.

4.2. Basic Framework of Handwritten Nüshu Recognition System

In order to ensure accurate recognition of handwritten Nüshu characters, this paper proposes a two-stage handwritten Nüshu character recognition scheme. As shown in Figure 6, in the first stage, the improved GoogLeNet-tiny network serves as the backbone model. First, the input images are preprocessed, and then the images are input into the model for recognition. At the same time, a threshold is set as a judgment criterion and the difference between the predicted output of the model and the true label is calculated. When the calculated value is less than the threshold, the Nüshu characters in the image are identified as incorrectly recognized. When the calculated value is greater than the preset threshold, an image with correct recognition is output.he incorrectly recognized images in the first stage are transferred to the second stage, where the end-to-end object recognition YOLOv5-CBAM is used as the main network, which detects and recognizes the inputted handwritten Nüshu characters, and finally outputs the recognition results.

The formula for calculating the variance value is as follows:

$$g(x,y) = \begin{cases} 1, & x = y \\ 0, & x \neq y \end{cases} \tag{9}$$

where x denotes the real word meaning of the Nüshu character and y the word meaning predicted by the model.

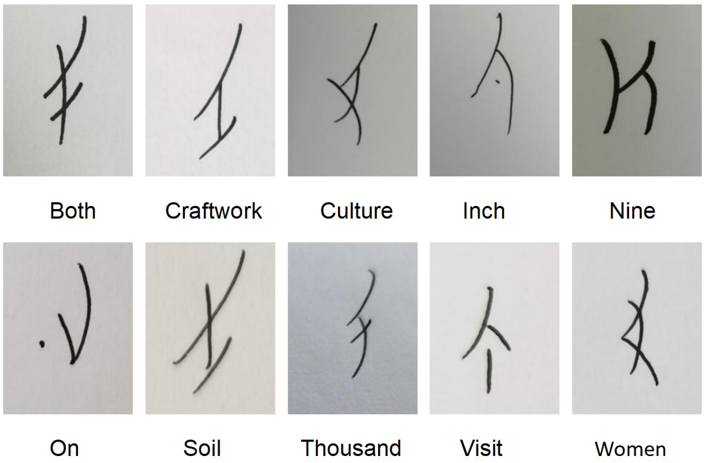
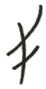
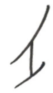
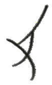
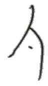
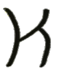
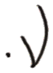
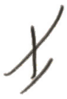
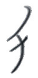
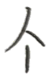
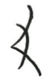


Figure 3. HWNS2023 dataset display. Where the English meaning is labeled below each of the images containing a Nüshu character.

Table 1. HWNS2023 dataset image meanings and translation in Chinese associated with Figure 3.

Categories	**Both**	**Craftwork**	**Culture**	**Inch**	**Nine**	**On**	**Soil**	**Thousand**	**Visit**	**Women**
Lettering										
Chinese meaning	两	工	文	寸	九	上	土	千	顾	女
Number	100	100	100	100	100	100	100	100	100	100

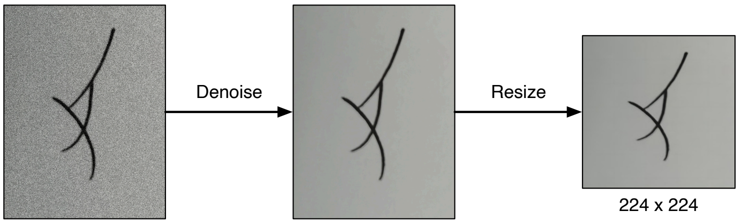


Figure 4. Pre-processing process of handwritten Nüshu character images.

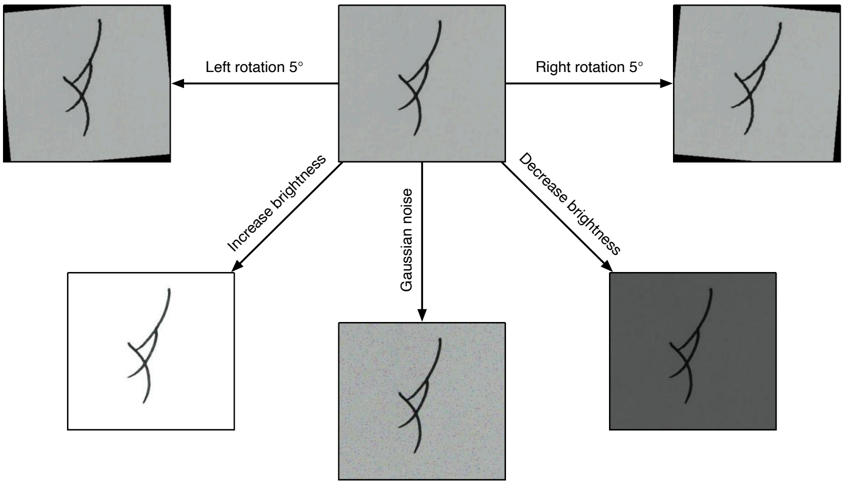


Figure 5. An example of data augmentation. Here, the image is enhanced by rotating it ±5°, adjusting the brightness, and adding Gaussian noise.

During the process of testing, the threshold of the experiment was set to 1. The formula for determining misidentification in the first stage is as follows:

$$Output = \begin{cases} True\ \ image,\ \ g(x,y) \geq threshold \\ Wrong\ \ image,\ \ g(x,y) < threshold \end{cases} \tag{10}$$

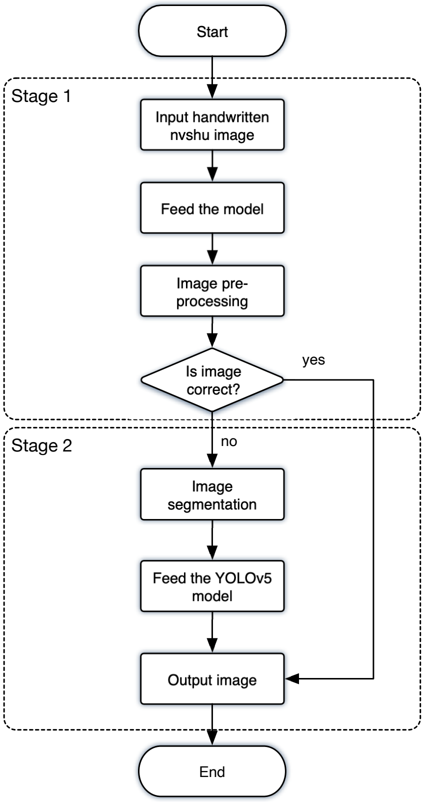


Figure 6. Handwritten Nüshu character recognition process.

4.3. Preparation of the Model

4.3.1. CNN

Convolutional neural networks (CNNs) [28,29] are widely used and highly effective deep learning algorithms for processing data with lattice-like structures, such as images and sounds. A CNN extracts features from data using multiple layers of convolution, pooling operations, and non-linear activation functions, making it capable of classification, recognition, and other tasks.

As shown in Figure 7, the basic structure of a CNN includes an input layer, several convolution and pooling layers, several fully connected layers, and an output layer. The core of a CNN is the convolutional layer, which extracts features from the data using convolutional operations. Based on the input data, the convolutional layer uses a set of learnable filters to convolve each region and output it to the next layer. This is a feature extraction and information compression process that extracts meaningful data features from complex raw data for better classification and recognition tasks. The pooling layer is used to reduce the size of the feature map, reduce the number of parameters in the model, reduce the complexity of the model, and alleviate overfitting problems. It slides a small convolutional kernel over the feature map, computes the aggregated feature value for each window, and outputs it as a representative value for the region to the next layer.

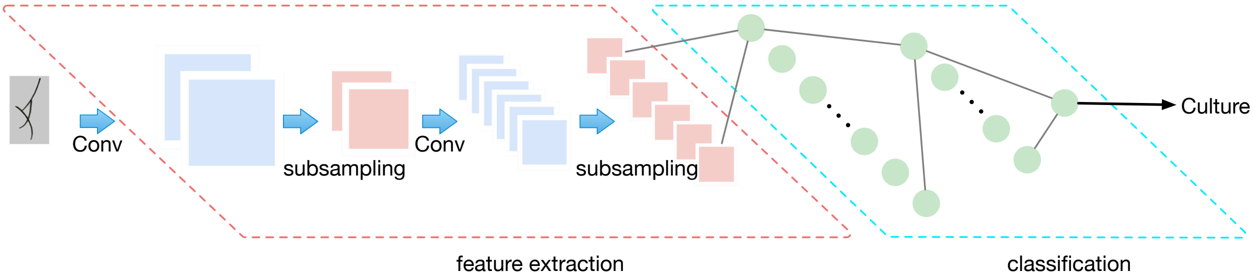


Figure 7. Basic structure of a CNN.

4.3.2. Introduction to the Basic Model

This paper selected five basic models, AlexNet, VGGNet16 [30], GoogLeNet, MobileNetV3 [31], and ResNet [32] to study the recognition of Nüshu handwritten characters.

AlexNet [33–35] is a deep convolutional neural network model that features a deep network structure and a large number of parameters, making it capable of automatically extracting high-level features from input images for high-precision image classification and object detection. The AlexNet model consists of 5 convolutional layers and 3 fully connected layers, with convolutional and fully connected layers alternating. A pooling layer follows each convolutional layer to reduce the size and number of feature maps. Meanwhile, AlexNet also uses common regularization techniques such as dropout and local response normalization (LRN) to prevent overfitting.

GoogLeNet [36,37] is a convolutional neural network classification model proposed by the Google team in 2014. This model extracts image features by combining various convolutional kernels of different scales and uses global average pooling to reduce the number of parameters. Furthermore, it uses 1 × 1 convolutional kernels to reduce and increase the channels, thus reducing the computational complexity and achieving faster training and higher accuracy.

4.3.3. The Improved AlexNet Models

By comparing the accuracy of five classical classification models in testing, the AlexNet model was selected as the basis for improvement, and three improved models, AlexNet-BN, AlexNet-SC, and AlexNet-LR, were proposed.

The original AlexNet used the local response normalization (LRN) layer for normalization to improve the generalization performance of the model, but the actual improvement effect is limited and the training time of the model is greatly increased. The AlexNet-BN model introduces a BN layer similar to the LRN layer based on the original network to speed up the model fitting speed, and at the same time adds two 3 × 3 convolutional layers to improve the model's feature extraction ability.

A convolution layer with a small kernel size has more non-linear features than a convolution layer with a larger kernel, which strengthens the decisiveness of the decision function and has fewer parameters. The AlexNet-SC model reduces the size of the convolution kernel based on AlexNet, reducing the 11 × 11 and 5 × 5 convolution kernels to 5 × 5 and 3 × 3 convolution kernels, respectively, and adding two convolution layers to ensure the receptive field.

AlexNet-LR is based on the original model, which not only reduces the size of the convolutional kernel, but also introduces a BN layer to speed up network convergence. In addition, in the feature extraction part, the ReLU activation function is replaced by the Leaky ReLU function to solve the problem of the gradient being 0 when the input is negative. The improved structure of the AlexNet network model is shown in Table 2.

Table 2. Convolutional neural network configuration. Conv $x \times x$ indicates that the convolutional layer core size is $x \times x$. $FC - y$ is the fully connected layer of the feature map with output dimension y. Maxpool $m \times m$ indicates that the maximum pooling layer core size is $m \times m$. Below the network name is the activation function used by the network.

ConvNet Configuration			
Proposed Models	**AlexNet-BN**	**AlexNet-SC**	**AlexNet-LR**
Network structures	Conv 11 × 11 + BN	Conv 5 × 5	Conv 5 × 5 + BN
		Maxpool 3 × 3	
	Conv 5 × 5 + BN	Conv 3 × 3	Conv 3 × 3 + BN
	Conv 3 ×3 + BN	Conv 3 × 3	
		Maxpool 3 × 3	
	Conv 3 × 3 + BN	Conv 3 × 3	Conv 3 × 3 + BN
	Conv 3 × 3 + BN	Conv 3 × 3	Conv 3 × 3 + BN
	Conv 3 × 3 + BN	Conv 3 × 3	Conv 3 × 3 + BN
	Conv 3 × 3 + BN	Conv 3 × 3	
		Maxpool 3 × 3	
FC-4096			
FC-4096			
FC-5			
softmax			

4.3.4. The Improved GoogLeNet Model

The GoogLeNet network comprises 3 convolutional layers, 9 inception modules, 2 auxiliary classifiers, and a fully connected layer. As shown in Figure 8, the improved new model GoogLeNet-tiny has improved the inception structure by adding a BN layer after the convolutional layer to solve the gradient saturation problem and accelerate the convergence speed of the model during training.he inception structure branch has been simplified and the 5 × 5 convolutional kernel has been replaced with a 3 × 3 convolutional kernel to reduce the size of the network model and improve recognition accuracy Finally, the activation function of the feature extraction stage has been replaced by the Leaky ReLU function to solve the gradient = 0 problem when the input is negative.

4.3.5. The Improved YOLOv5 Model

YOLOv5 [38–40] is currently a mainstream object detection model, and its network structure mainly consists of four parts: input, backbone, neck, and prediction. The input section utilizes data augmentation techniques to improve the detection effect of small targets, and also uses adaptive bounding box calculation and adaptive image scaling techniques to enhance the speed of object detection. In the backbone section, YOLOv5 adds the focus structure compared to YOLOv4. In the neck section, YOLOv5 adopts the FPN+PAN structure and strengthens the network feature fusion capability by using the CSP2 structure inspired by CSPnet design. The CIOU_Loss is used as the bounding box loss function in the output section. There are four versions of the YOLOv5 algorithm, and among them, the YOLOv5s network model is the smallest and fastest in detection speed with high accuracy. It can be used in embedded devices and is suitable for Nüshu character recognition. The structure of YOLOv5s is shown in Figure 9.

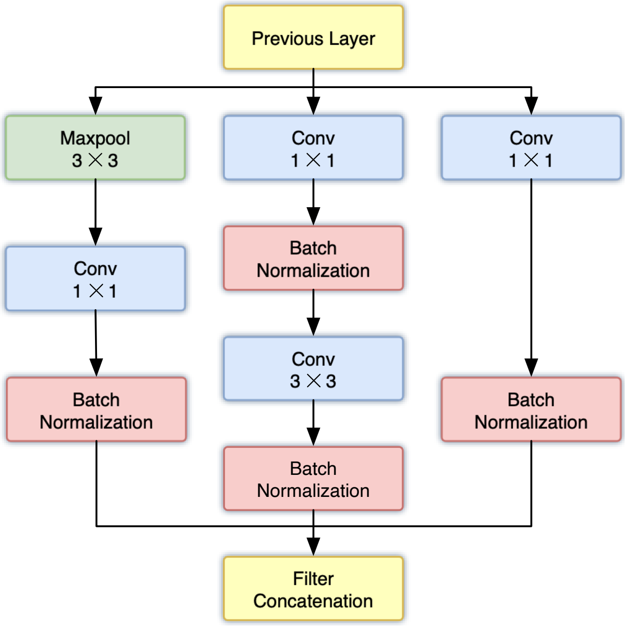


Figure 8. Improved inception module.

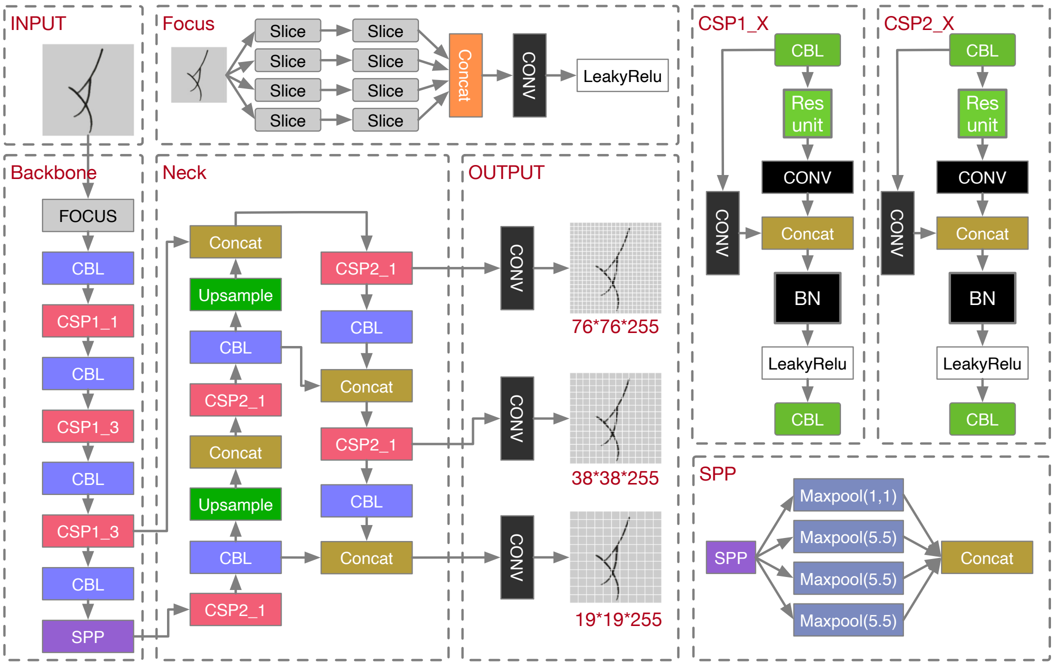


Figure 9. Model structure of YOLOv5s.

Nüshu characters have complex glyph shapes, and there is a certain gap between handwritten Nüshu characters and fonts, which poses certain difficulties in recognition. To improve the ability of the backbone network to extract features of handwritten Nüshu characters, we introduced the CBAMC3 attention mechanism into the YOLOv5 backbone network. CBAM (convolutional block attention module) is an attention mechanism used for image classification and object recognition. The basic idea is to adaptively learn the importance of features in the feature map to improve model performance and accuracy. The CBAM module includes two parts: channel attention and spatial attention. Channel attention and spatial attention learn and extract Nüshu features through two neural networks, and finally obtain an integrated attention feature map. YOLOv5 combined with the CBAM attention mechanism can adaptively learn the importance of features in images, allowing the model to better capture key information.

In addition, we need to focus on the loss values of YOLOv5, which mainly consist of three parts: classification loss, box loss, and confidence loss. In the YOLOv5 model, the binary cross-entropy (BCE) [41] loss function is used to calculate classification loss and confidence loss. The BCE function is defined as follows:

$$C = -\frac{1}{n}\sum_{x}[y \ln a + (1-y)\ln(1-a)] \tag{11}$$

where x is the sample, y is the label, a is the predicted output, and n is the total number of samples.

YOLOv5 uses the binary cross-entropy loss function to calculate the loss in class probability and target confidence score, where the different labels are not mutually exclusive. YOLOv5 replaces the softmax function with multiple independent logistic classifiers to calculate the probability that the input belongs to a particular label. When training the classification loss, the binary cross-entropy loss is used for each label. This also avoids the use of a softmax function and reduces computational complexity.

The box loss calculation uses the IoU function to measure the overlap between the predicted bounding box and the ground truth box in object detection. Assuming the predicted box is A and the ground truth box is B, the expression of IoU is:

$$IoU = \frac{A \cap B}{A \cup B} \tag{12}$$

This expression shows that the intersection ratio is equal to 1 if the two boxes overlap completely and equal to 0 if they do not overlap at all.

4.4. Methods

4.4.1. Dataset Division

In the model training process, the data from both stages were divided into training, validation, and test sets in an 8:1:1 ratio. In the first stage, there were a total of 1000 training samples, with 800 images in the training set and 100 images in each of the validation and test sets. In the second stage, the training set contained 1110 images, and there were 127 images in both the validation and test sets. The training set was used to fit the model for prediction or classification, while the data from the validation set helped to find the optimal combination of hyperparameters. The test set was used to evaluate the performance of the selected model.

4.4.2. Parameter Setting

In deep learning models, parameter settings are crucial for the performance and accuracy of the model. Parameter settings include two main aspects: the network weight initialization strategy and the hyperparameter configuration for model training. Choosing appropriate initial configurations has a crucial impact on the whole training process. In our experiments, we used zero initialization to set the initial weights for all networks. Regarding the training hyperparameters, we set the training epochs to 100 for all networks and used a discrete staircase method for model training. Discretizing gradient estimation into step functions can overcome noise issues in gradient estimation, which helps to speed up the training process of convolutional neural networks and improve the generalization performance of the model. It also enables algorithms to quickly converge to local minima, thus enhancing the accuracy of the model. The initial learning rate was set to 0.1, which decreased by half every 20 epochs. We also set the batch size for the training dataset to 32.

5. Experiments

5.1. Design of the Experiments

In line with the design of the two-stage Nüshu handwriting recognition scheme, our experiments were divided into two parts. In the first part, two datasets were used for comparative experiments to investigate the effect of data augmentation on the recognition performance of Nüshu characters, as discussed in Section 4.1.2. One set used the original data for training and testing the recognition accuracy of the model, while the other set used the data augmented data for training the model and testing the recognition accuracy. The effect of data augmentation was evaluated by comparing the recognition accuracy of the model before and after data augmentation. Furthermore, the recognition performance of the five basic models was compared horizontally, considering both speed and accuracy, and the models were improved and trained. The model improvement was tested and the first stage recognition model was determined by comparing the test results of the improved models. In the second part, a series of comparative experiments was designed to determine the second-stage recognition model by training and testing three mainstream object recognition models, YOLOv3, YOLOv5, and YOLOv7. The results of the comparative experiments were used to determine the second-stage model. The contents of each experiment are shown in Table 3.

Table 3. Experimental scheme.

Experimental Stage	Experiment	Contents	Training Images
I	1	raw data + basic model	900
	2	augmented data + basic model	5400
	3	augmented data + improved model	5400
II	4	raw data + YOLOv3/YOLOv5/YOLOv7	1364

5.2. Experimental Environment and Configuration

All experiments in this paper were conducted on a computer with a hardware configuration of NVIDIA GeForce GTX 1070ti, 8 GB RAM, Windows operating system, and Pytorch deep learning framework. The compilation tool used was Pycharm, and the programming language used was Python 3.9.

5.3. Training

The training progress of the three improved AlexNet models is shown in Figures 10 and 11. It can be seen that AlexNet-SC showed a sharp decrease in training loss and a sharp increase in training accuracy during the first 10 epochs, indicating that the model was learning effectively, and then stabilized after 20 epochs. In contrast, AlexNet-BN and AlexNet-LR showed no changes in training loss and validation accuracy during the first 10 epochs, and then gradually decreased their training loss and increased their validation accuracy from the 10th to the 30th epoch until stabilizing after 40 epochs, which is consistent with the changes in training accuracy. Around the 100th epoch, the validation accuracy approached 1, indicating that the model could fully converge after only 100 epochs of training.

The training process of the GoogLeNet-tiny model, which is an improved version of the GoogLeNet network, is shown in Figures 10 and 11. By adding BN layers to the GoogLeNet-tiny model, the training loss decreased rapidly during the first 20 epochs and then gradually stabilized. At the same time, the validation accuracy increased rapidly during the first 20 epochs and then showed some fluctuations, but remained above 0.95 overall. Compared to the GoogLeNet network, the speed of convergence of the GoogLeNet-tiny network has been significantly improved. Therefore, we conclude that the GoogLeNet-tiny model, due to the reduction of the overall network and the introduction of BN layers, greatly improves the training speed, converges in only about 20 epochs, and has good recognition performance.

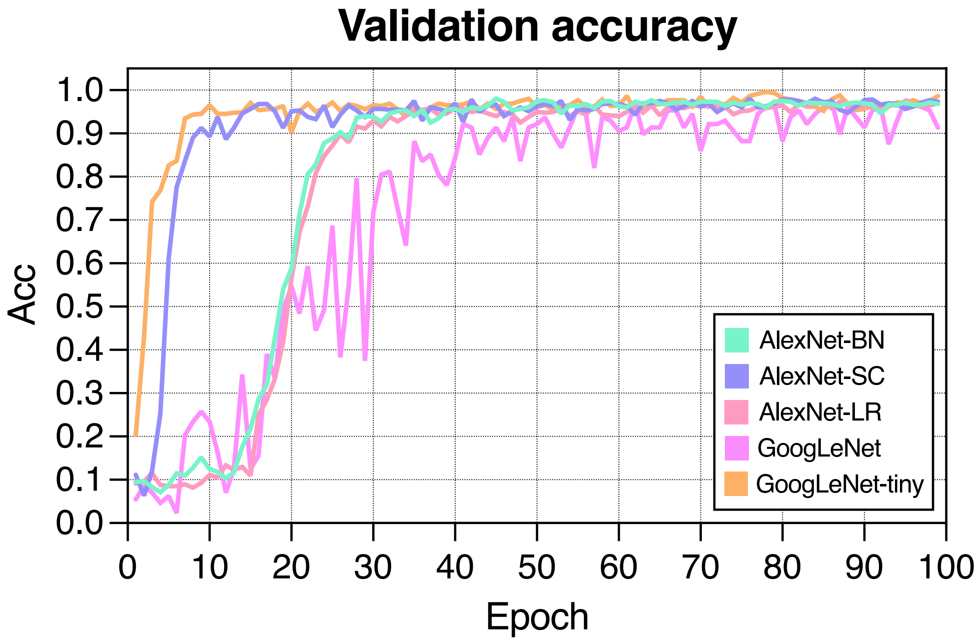


Figure 10. A new training curve for the model has been added, validation_acc.

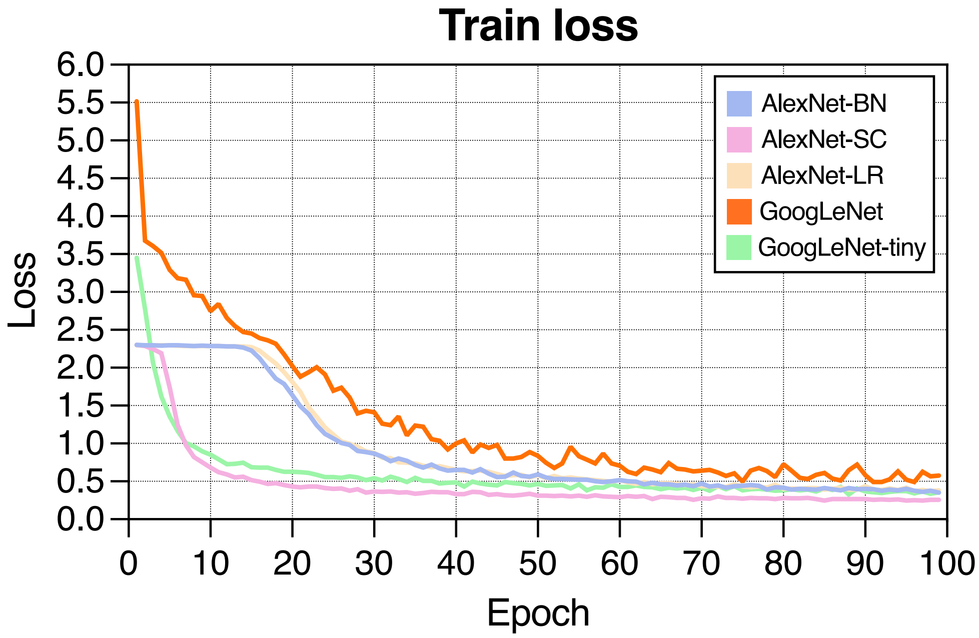


Figure 11. Training loss.

The training process of the improved YOLOv5 model is shown in Figure 12. We can clearly see that the loss of YOLOv5 is mainly composed of three parts: classification loss, object loss, and location loss. These losses continuously decrease during the first 30 epochs and gradually converge as the number of iterations increases. After 30 epochs, the object and location losses gradually plateau, and the classification loss stabilizes around 0 after 40 epochs, indicating that the network has basically converged. The mAP is used to measure the performance of the model for all categories, and the higher this value, the higher the average detection accuracy and the better the performance. Figure 12 shows that after about 30 iterations, the mAP of the YOLOv5 model reaches about 93% and gradually stabilizes to a maximum of 97%, indicating that the overall performance of the model is as expected.

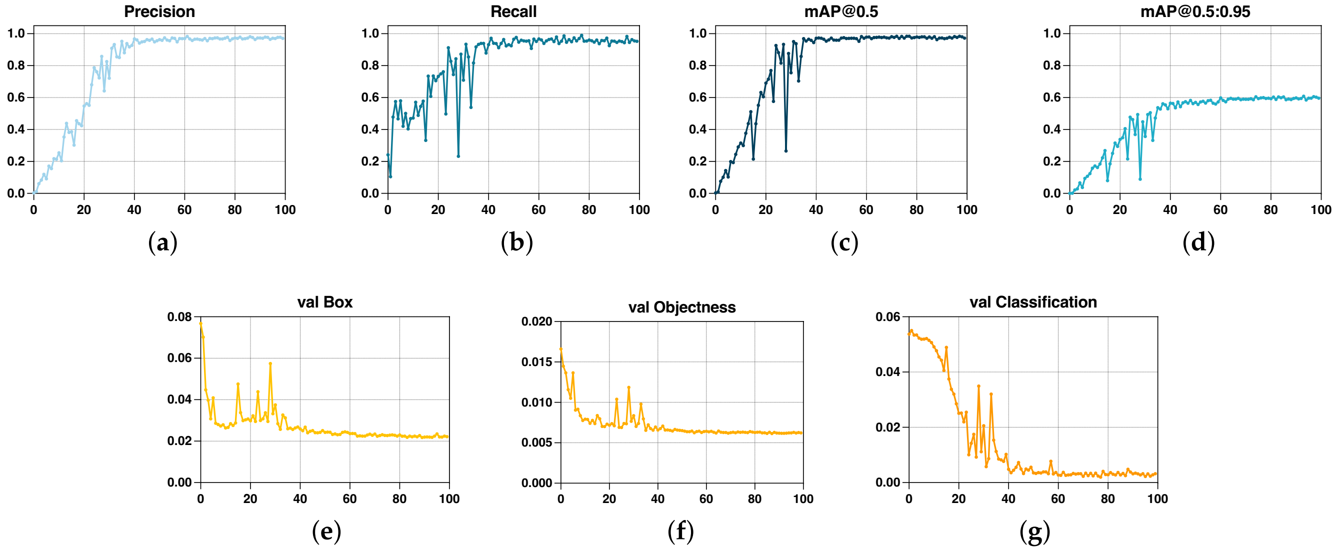


Figure 12. YOLOv5 training. (**a**) Precision plot. (**b**) Recall plot. (**c**) mAP@0.5: mean average precision. (**d**) mAP@0.5:0.95. (**e**) val Box: validation set bounding box loss. (**f**) val Objectness: mean validation set object detection loss. (**g**) val Classification: mean validation set classification loss.

6. Results

6.1. Evaluation Indexes

In this paper, accuracy is used to evaluate the recognition performance of the first-stage model, and mAP is used in the second stage to assess the model's precision. Accuracy is the most widely used evaluation metric in classification models. It refers to the proportion of samples that the model correctly predicts in all classified samples. Accuracy is an important metric for evaluating the prediction performance of models, and it can be used to assess the model's performance. It is represented as:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{13}$$

Mean average precision (mAP) is a metric used in the field of object detection to measure algorithm performance. It is commonly used to evaluate detection accuracy and consistency. The YOLO series also uses mAP as its primary performance evaluation metric. mAP measures the balance between precision and recall for detecting objects by calculating the accuracy of prediction results and true results. The calculation process of mAP classifies different categories of objects and calculates the average precision of algorithms for each category. Finally, the overall model mAP is calculated based on the average precision and IoU value of each symmetric object.

$$mAP = \frac{\sum_{i=1}^{k} AP_i}{K} \tag{14}$$

where AP is the average precision and k is the number of sample categories, and its formula is:

$$AP = \sum_{i=1}^{n-1} (r_{i+1} - r_i) P_{inter}(r_i + 1) \tag{15}$$

6.2. Baseline Model Test Results

In the first stage, we selected five convolutional neural network models, namely, AlexNet, VGGNet16, GoogLeNet, ResNet, and MobileNetV3, for training, and tested the trained models. Table 4 shows the test results of the models trained without using augmented data, where the best performing model is the GoogLeNet network, with a recognition accuracy of 93.3% on the test dataset, followed by the AlexNet network, with an accuracy of 87.3%, and the worst performing model is the lightweight MobileNetV3, with a test accuracy of only 36.6%. The experimental results show that the selected models can be well applied to the recognition of Nüshu handwritten character images.

According to the experimental arrangement in Table 3, in experiment 2, the data was further augmented using data augmentation methods based on experiment 1, which further increased the number of Nüshu character samples. After data augmentation, the highest recognition accuracy obtained for the 10 identified categories of Nüshu characters in experiment 2 was 98%, and the lowest was 49.3%. The significant improvement in recognition rate was due to the substantial increase in data obtained by adding 4500 Nüshu character samples through various techniques such as skewing and brightness adjustment. In addition, the images generated by the data augmentation simulated various practical application scenarios, which improved the generalization ability of the model and was another reason for the improvement in recognition accuracy. The experimental results show that the GoogLeNet network performed best on the test dataset, followed by the AlexNet network. Therefore, these two models were selected for improvement in experiment 3 to further improve the recognition accuracy of Nüshu character images. The detailed results are shown in Table 5.

Table 4. Accuracy results of the base model test set of experiment 1.

Network	Test (%)	Test Time (s)
AlexNet	87.3	3.54
VGGNet16	49.3	5.91
GoogLeNet	**93.3**	4.53
ResNet	74.0	4.65
MobileNetV3	36.6	**1.02**

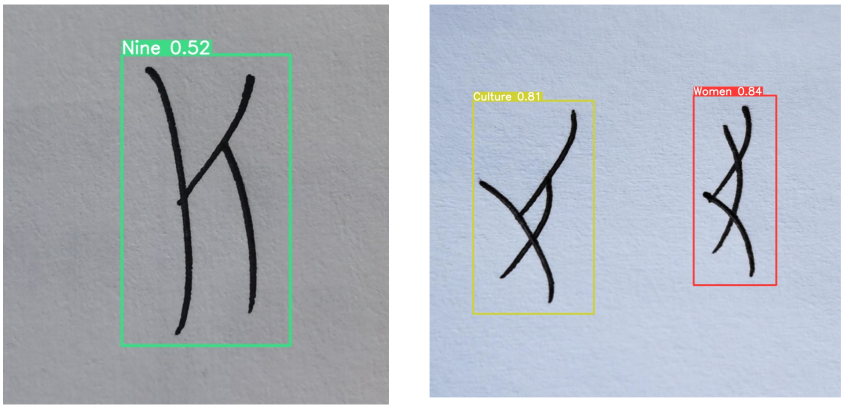
Table 5. Accuracy results of the base model test set of experiment 2.

Network	Test (%)	Test Time (s)
AlexNet	96.6	3.55
VGGNet16	96.3	5.79
GoogLeNet	**98.0**	4.45
ResNet	93.3	4.61
MobileNetV3	49.3	**1.01**

In the second stage, to determine the recognition model for the second stage, we selected three object recognition models, YOLOv3, YOLOv5, and YOLOv7, for training based on the experimental arrangement in Table 3. We tested the trained models and selected the one with the best test results as the recognition network for the second stage. As shown in Table 6, YOLOv5 had the best recognition performance among the three models, with a mAP of 97.7% and a precision of 96.4%. Therefore, we decided to use YOLOv5 as the recognition model for the second stage. In order to improve the recognition accuracy of this model for Nüshu handwritten characters, we introduced the CBAM attention mechanism module into the backbone network of YOLOv5 to enhance its ability to extract Nüshu characters' features. The experimental results showed that the CBAM attention mechanism can effectively improve the recognition accuracy of the model. Compared to the original model, the mAP increased by 0.4%, and the model could accurately recognize handwritten Nüshu characters. The recognition results of YOLOv5-CBAM for handwritten Nüshu characters are shown in Figure 13, with single character recognition results on the left and multi-character image recognition results on the right, both of which can be accurately recognized.

Table 6. Accuracy results of the base model test set of experiment 4.

Network	mAP (%)	Precision (%)
YOLOv3	97.2	96.3
YOLOv5	97.7	96.4
YOLOv7	97.3	95.9
YOLOv5-CBAM	**98.1**	**97.4**



(a)

(b)

Figure 13. Recognition results of the second-stage model YOLOv5-CBAM. The image (**a**) shows the recognition results for a single Nüshu character, while the image (**b**) shows the recognition results for multiple Nüshu characters.

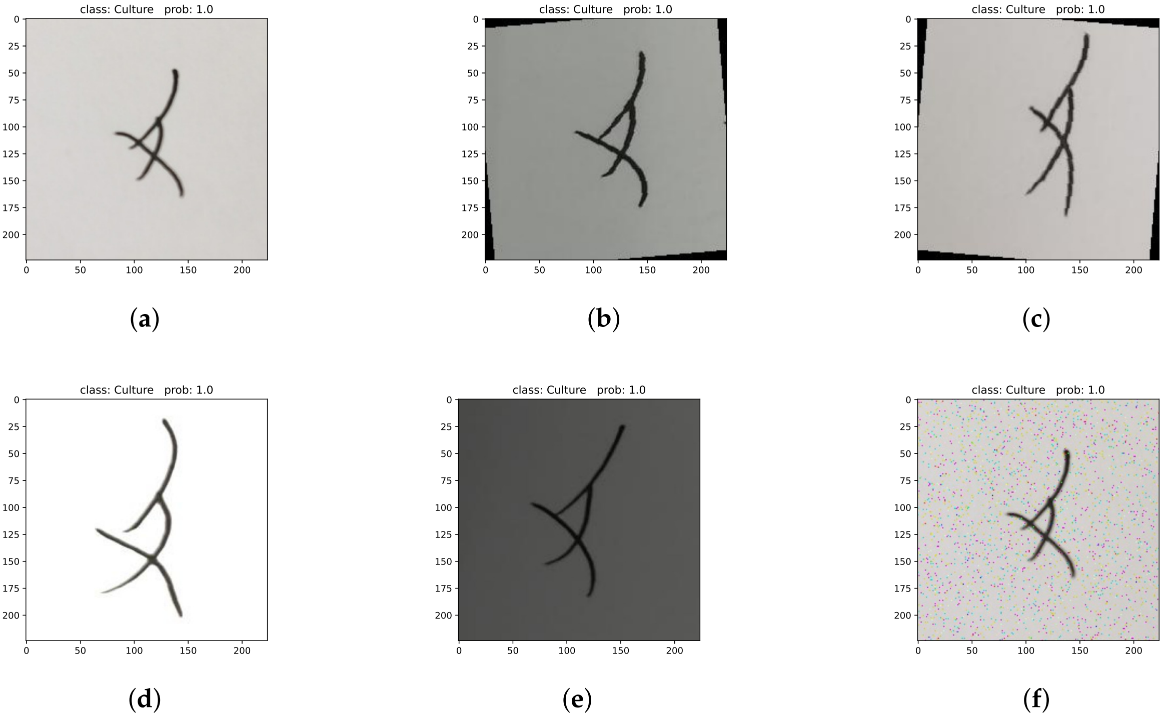
6.3. Improved Model Test Results

According to the experimental design, in the first stage, this paper compared the test accuracy of five basic models and selected AlexNet and GoogLeNet for model improvement. Table 7 shows the test accuracy and detection times of the AlexNet-BN, AlexNet-SC, AlexNet-LR, and GoogLeNet-tiny models on the test set. The results show that the accuracy of AlexNet-SC and AlexNet-LR has been greatly improved compared to the AlexNet network. Among them, the accuracy of AlexNet-SC is the highest, reaching 98%, which is a 1.4% improvement over the baseline AlexNet model. The accuracy of AlexNet-LR is 97.3%, which is only a 0.7% improvement over the baseline model, indicating that the latter two enhancement methods can improve the recognition accuracy of the model. Compared to the original model, the recognition accuracy of AlexNet-BN did not change, indicating that the addition of convolutional and BN layers alone cannot improve the accuracy of the model. AlexNet-SC, AlexNet-LR, and GoogLeNet-tiny all reduced the size of the convolution kernels in the convolution process, and the recognition accuracy increased to some extent, indicating that reducing the size of the convolution kernels can effectively improve the recognition of Nüshu characters by the model. In addition, compared with the original model, the improved models still maintain good speed in terms of recognition speed.

Secondly, the accuracy of the GoogLeNet-tiny model on the test set is 99.3%, which is 1.3% higher than the original model. This is the best performance of the improved models. After a comprehensive comparison, it was decided to use the GoogLeNet-tiny model in the first stage of the Nüshu character recognition scheme. The recognition results of GoogLeNet-tiny for handwritten Nüshu characters are shown in Figure 14; the handwritten Nüshu character in the image can be correctly recognized in a variety of situations.

Table 7. Accuracy results of the improved model test set of experiment 3.

Network	Test (%)	Test Time (s)
AlexNet-BN	96.6	**3.25**
AlexNet-SC	98.0	3.28
AlexNet-LR	97.3	3.31
GoogLeNet-tiny	**99.3**	4.81



(**a**) (**b**) (**c**)

(**d**) (**e**) (**f**)

Figure 14. GoogLeNet-tinyrecognition results. (**a**) Recognition of the Nüshu character meaning "culture". (**b**) Results of the recognition of the Nüshu character rotated 5° to the left. (**c**) Results of the recognition of the Nüshu character rotated 5° to the right. (**d**) Enhanced brightness of Nüshu character recognition results. (**e**) Results of the recognition of the Nüshu character with reduced brightness. (**f**) Recognition results for the Gaussian blurred Nüshu character.

7. Conclusions

The complex character shapes and high stroke count of handwritten Nüshu characters make recognition difficult. In this paper, we first established the HWNS2023 dataset for handwritten Nüshu characters and explored several data augmentation methods. We then proposed a two-stage handwritten Nüshu recognition scheme that utilizes two deep learning models. In the first stage, to ensure recognition accuracy, we evaluated five different models to determine which one was best suited for Nüshu recognition and improved and assessed the selected model, ultimately determining that the GoogLeNet-tiny model was the best for stage one recognition. However, due to the limitations of classification models, not all characters can be correctly detected and recognized. Therefore, in the second stage, we used YOLOv5-CBAM as the backbone network to perform secondary recognition on misidentified Nüshu handwritten characters. However, the small size of our dataset currently limits the effectiveness of the proposed method, and expanding its scale will be a key focus of future work. The recognition method proposed in this paper lays the foundation for the construction of a mobile Nüshu recognition system and provides new ideas for the protection and inheritance of Nüshu culture, which is of great significance for solving practical problems in Nüshu character recognition in real-world scenarios.

Author Contributions: Conceptualization, Y.Z.; methodology, Y.Z.; software, Y.Z.; validation, Y.Z.; formal analysis, Y.Z. and L.Z.; investigation, L.Z.; resources, L.Z.; data curation, L.Z. and Y.Z.; writing—original draft preparation, Y.Z.; writing—review and editing, L.Z. and Y.Z.; visualization, Y.Z.; supervision, L.Z.; project administration, L.Z.; funding acquisition, L.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Shaanxi Key Laboratory for Network Computing and Security Technology, grant number NCST2021YB-04.

Data Availability Statement: The data in this article were collected by ourselves. Volunteers consulted a dictionary and created the dataset by handwriting. If the reader would like to access the data in this paper, they may visit https://www.kaggle.com/datasets/yanz0409/hwns2023, accessed on 1 June 2023.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Yaxia, L.Q. Cultural Deconstruction and Reconstruction:Transformation of Nüshu's Inheritance Field and Its Main Practice. *Media Obs.* **2023**, *471*, 55–63.
2. Zhang, Y. *The Nüshu and Its Cultural Heritage*; XinJiang Art: Ürümqi, China, 2023; No.170. (In Mandarin)
3. Liu, F.W. Practice and Cultural Politics of "Women's Script" nüshu as an endangered heritage in contemporary china. *Angelaki* **2017**, *22*, 231–246. [CrossRef]
4. Luo, W.; Lu, Y.; Timothy, D.J.; Zang, X. Tourism and conserving intangible cultural heritage:Residents' perspectives on protecting the nüshu female script. *J. China Tour. Res.* **2022**, *18*, 1305–1329. [CrossRef]
5. Huan, J.L.; Sekh, A.A.; Quek, C.; Prasad, D.K. Emotionally charged text classification with deep learning and sentiment semantic. *Neural Comput. Appl.* **2022**, *34*, 2341–2351. [CrossRef]
6. Liu, M.; Liu, G.; Liu, Y.; Jiao, Q. Oracle Bone Inscriptions Recognition Based on Deep Convolutional Neural Network. *J. Image Graph.* **2020**, *8*, 114–119. [CrossRef]
7. Aneja, N.; Aneja, S. Transfer learning using CNN for handwritten devanagari character recognition. In Proceedings of the 2019 1st International Conference on Advances in Information Technology (ICAIT), Chikmagalur, India, 25–27 July 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 293–296.
8. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016.
9. Wang, C.Y.; Bochkovskiy, A.; Liao, H.Y.M. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors, In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022.
10. Wigington, C.; Tensmeyer, C.; Davis, B.; Barrett, W.; Price, B.; Cohen, S. Start, follow, read: End-to-end full-page handwriting recognition. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 367–383.
11. Ptucha, R.; Petroski Such, F.; Pillai, S.; Brockler, F.; Singh, V.; Hutkowski, P. Intelligent character recognition using fully convolutional neural networks. *Pattern Recognit.* **2019**, *88*, 604–613. [CrossRef]
12. Cilia, N.D.; De Stefano, C.; Fontanella, F.; Scotto di Freca, A. A ranking-based feature selection approach for handwritten character recognition. *Pattern Recognit. Lett.* **2019**, *121*, 77–86. [CrossRef]
13. Choudhury, H.; Prasanna, S.R.M. Representation of online handwriting using multi-component sinusoidal model. *Pattern Recognit.* **2019**, *91*, 200–215. [CrossRef]
14. Pashine, S.; Dixit, R.; Kushwah, R. Handwritten Digit Recognition using Machine and Deep Learning Algorithms. *Int. J. Comput. Appl.* **2020**, *176*, 27–33.
15. Ly, N.T.; Nguyen, C.T.; Nguyen, K.C.; Nakagawa, M. Deep convolutional recurrent network for segmentation-free offline handwritten Japanese text recognition. In Proceedings of the 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), Kyoto, Japan, 9–15 November 2017; IEEE: Piscataway, NJ, USA, 2017; Volume 7, pp. 5–9.
16. Majid, N.; Smith, E.H.B. Segmentation-free bangla offline handwriting recognition using sequential detection of characters and diacritics with a Faster R-CNN. In Proceedings of the 2019 International Conference on Document Analysis and Recognition (ICDAR), Sydney, Australia, 20–25 September 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 228–233.
17. Ali, A.A.A.; Mallaiah, S. Intelligent handwritten recognition using hybrid CNN architectures based-SVM classifier with dropout. *J. King Saud Univ.-Comput. Inf. Sci.* **2022**, *34*, 3294–3300. [CrossRef]
18. Carbune, V.; Gonnet, P.; Deselaers, T.; Rowley, H.A.; Daryin, A.; Calvo, M.; Wang, L.L.; Keysers, D.; Feuz, S.; Gervais, P. Fast multi-language LSTM-based online handwriting recognition. *Int. J. Doc. Anal. Recognit. (IJDAR)* **2020**, *23*, 89–102. [CrossRef]
19. Hei, G.Y.; Wang, J.Q.; Sun, Y.G. Multi-oriented text lines extraction from offline Nüshu characters image. *Appl. Res. Comput.* **2013**, *30*, 627–630.

20. Sun, Y.; Cai, Z. An improved character segmentation algorithm based on local adaptive thresholding technique for Chinese NvShu documents. *J. Netw.* **2014**, *9*, 1496. [CrossRef]
21. Wang, J.; Zhu, R. Handwritten Nushu Character Recognition Based on Hidden Markov Model. *J. Comput.* **2010**, *5*, 663–670. [CrossRef]
22. Glorot, X.; Bordes, A.; Bengio, Y. Deep sparse rectifier neural networks. In Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, JMLR Workshop and Conference Proceedings, Fort Lauderdale, FL, USA, 11–13 April 2011; pp. 315–323.
23. Chen, Y.; Dai, X.; Liu, M.; Chen, D.; Yuan, L.; Liu, Z. Dynamic ReLU. In *Computer Vision—ECCV 2020*; Vedaldi, A., Bischof, H., Brox, T., Frahm, J.M., Eds.; Springer International Publishing: Berlin/Heidelberg, Germany, 2020; pp. 351–367.
24. Xu, B.; Wang, N.; Chen, T.; Li, M. Empirical Evaluation of Rectified Activations in Convolutional Network. *arXiv* **2015**, arXiv:1505.00853.
25. Ioffe, S.; Szegedy, C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In Proceedings of the 32nd International Conference on Machine Learning, PMLR, Lille, France, 6–11 July 2015; pp. 448–456.
26. Woo, S.; Park, J.; Lee, J.Y.; Kweon, I.S. CBAM: Convolutional Block Attention Module. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 3–19.
27. Fu, H.; Song, G.; Wang, Y. Improved YOLOv4 Marine Target Detection Combined with CBAM. *Symmetry* **2021**, *13*, 623. [CrossRef]
28. Bhatt, D.; Patel, C.; Talsania, H.; Patel, J.; Vaghela, R.; Pandya, S.; Modi, K.; Ghayvat, H. CNN Variants for Computer Vision:History, Architecture, Application, Challenges and Future Scope. *Electronics* **2021**, *10*, 2470. [CrossRef]
29. Chen, L.; Wang, S.; Fan, W.; Sun, J.; Naoi, S. Beyond human recognition: A CNN-based framework for handwritten character recognition. In Proceedings of the 2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR), Kuala Lumpur, Malaysia, 3–6 November 2015; pp. 695–699.
30. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition, *arXiv* **2014**, arXiv:1409.1556.
31. Howard, A.; Sandler, M.; Chu, G.; Chen, L.C.; Chen, B.; Tan, M.; Wang, W.; Zhu, Y.; Pang, R.; Vasudevan, V.; et al. Searching for MobileNetV3. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019.
32. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015.
33. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems*; Curran Associates, Inc.: Red Hook, NY, USA, 2012; Volume 25.
34. Lee, S.G.; Sung, Y.; Kim, Y.G.; Cha, E.Y. Variations of AlexNet and GoogLeNet to Improve Korean Character Recognition Performance. *J. Inf. Process. Syst.* **2018**, *14*, 205–217.
35. Rasheed, A.; Ali, N.; Zafar, B.; Shabbir, A.; Sajid, M.; Mahmood, M.T. Handwritten Urdu characters and digits recognition using transfer learning and augmentation with AlexNet. *IEEE Access* **2022**, *10*, 102629–102645. [CrossRef]
36. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going Deeper with Convolutions. *arXiv* **2014**, arXiv:1409.4842; Version: 1.
37. Li, J.; Song, G.; Zhang, M. Occluded offline handwritten Chinese character recognition using deep convolutional generative adversarial network and improved GoogLeNet. *Neural Comput. Appl.* **2020**, *32*, 4805–4819. [CrossRef]
38. Zhang, Y.; Li, Z.; Yang, Z.; Yuan, B.; Liu, X. Air-GR: An Over-the-Air Handwritten Character Recognition System Based on Coordinate Correction YOLOv5 Algorithm and LGR-CNN. *Sensors* **2023**, *23*, 1464. [CrossRef]
39. Bakhri, I.A.; Sidik, H.P. Realtime Recognition of Handwritten Lontara Makassar Characters Using Yolov5 Algorithm. In Proceedings of the 2022 6th International Conference on Information Technology, Information Systems and Electrical Engineering (ICITISEE), Virtual, 13–14 December 2022; pp. 309–313.
40. Lin, F.; Hou, T.; Jin, Q.; You, A. Improved YOLO Based Detection Algorithm for Floating Debris in Waterway. *Entropy* **2021**, *23*, 1111. [CrossRef]
41. Shannon, C.E. A mathematical theory of communication. *Bell Syst. Tech. J.* **1948**, *27*, 379–423. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.