applied sciences

MDPI

Article

Multimodal Handwritten Exam Text Recognition Based on Deep Learning

Hua Shi *, Zhenhui Zhu, Chenxue Zhang *, Xiaozhou Feng and Yonghang Wang

College of Sciences, Xi'an Technological University, Xi'an 710021, China
* Correspondence: shihua@xatu.edu.cn (H.S.); 19957977250@163.com (C.Z.)

Abstract

To address the complex challenge of recognizing mixed handwritten text in practical scenarios such as examination papers and to overcome the limitations of existing methods that typically focus on a single category, this paper proposes MHTR, a Multimodal Handwritten Text Adaptive Recognition algorithm. The framework comprises two key components, a Handwritten Character Classification Module and a Handwritten Text Adaptive Recognition Module, which work in conjunction. The classification module performs fine-grained analysis of the input image, identifying different types of handwritten content such as Chinese characters, digits, and mathematical formula. Based on these results, the recognition module dynamically selects specialized sub-networks tailored to each category, thereby enhancing recognition accuracy. To further reduce errors caused by similar character shapes and diverse handwriting styles, a Context-aware Recognition Optimization Module is introduced. This module captures local semantic and structural information, improving the model's understanding of character sequences and boosting recognition performance. Recognizing the limitations of existing public handwriting datasets, particularly their lack of diversity in character categories and writing styles, this study constructs a heterogeneous, integrated handwritten text dataset. The dataset combines samples from multiple sources, including Chinese characters, numerals, and mathematical symbols, and features high structural complexity and stylistic variation to better reflect real-world application needs. Experimental results show that MHTR achieves a recognition accuracy of 86.63% on the constructed dataset, significantly outperforming existing methods. Furthermore, the context-aware optimization module demonstrates strong adaptive correction capabilities in various misrecognition scenarios, confirming the effectiveness and practicality of the proposed approach for complex, multi-category handwritten text recognition tasks.

Keywords: deep learning; text recognition; attention mechanism; computer vision

Academic Editor: Pedro Couto

Received: 20 July 2025
Revised: 8 August 2025
Accepted: 9 August 2025
Published: 12 August 2025

Citation: Shi, H.; Zhu, Z.; Zhang, C.; Feng, X.; Wang, Y. Multimodal Handwritten Exam Text Recognition Based on Deep Learning. *Appl. Sci.* **2025**, *15*, 8881. https://doi.org/10.3390/app15168881

1. Introduction

The development of handwritten character recognition technology can be traced back to 1929, when the German scientist Tausheck first proposed the concept of Optical Character Recognition (OCR). In 1996, Casey and Nagy [1] introduced the template matching method, which involved segmenting text images into individual characters and comparing them with predefined templates. This approach enabled the recognition of 1000 printed Chinese characters and marked a significant milestone in the application of OCR technology to Chinese character recognition.

Currently, existing handwritten character recognition methods can be generally categorized into two major types: traditional approaches and deep learning-based approaches.

Traditional handwritten Chinese character recognition typically involves three main steps: image preprocessing, feature extraction, and classification. The goal of preprocessing is to improve image quality, with commonly used methods including median filtering [2], Otsu's method [3], the Bernsen algorithm [4], and the Viterbi algorithm [5]. These techniques effectively enhance data quality, laying a solid foundation for subsequent recognition tasks. Feature extraction is the core component of the recognition pipeline and can be broadly divided into structural features, statistical features, and their hybrid approaches [6]. Representative techniques include elastic mesh features [7], Gabor features [8], and moment-based features [9]. These extracted features serve as the basis for classifier decision-making. Classification methods fall into two broad categories: single classifiers and ensemble classifiers. Common single classifiers include the Modified Quadratic Discriminant Function (MQDF) [10], Support Vector Machines (SVMs) [11], and Hidden Markov Models (HMMs) [12]. Ensemble classification methods integrate multiple classifiers through either serial or parallel architectures, aiming to maximize the advantages of each individual model and compensate for their respective weaknesses, thereby improving overall recognition accuracy.

Deep learning methods, particularly those based on Convolutional Neural Networks (CNNs), have significantly improved the accuracy of handwritten Chinese character recognition (HCCR) and demonstrated clear advantages over traditional methods, especially in complex scenarios. In 2011, CNN-based approaches achieved outstanding results in the ICDAR competition [13], after which the Fujitsu team [14] proposed an improved deep convolutional network that further increased recognition accuracy to 94.77%. Researchers such as Chen Y. et al. [15] proposed a segmentation-based learning approach for online handwritten Chinese text recognition, utilizing a convolutional prototype network to enhance the recognition performance of Chinese text. Bharati P. V. [16] introduced a new method that combines Deblur Generative Adversarial Networks (Deblur GANs) with CNNs to tackle challenges such as blurring and distortion in Handwritten Character Recognition (HCR), significantly improving recognition accuracy. Sunori S. K. et al. [17] proposed a hybrid model integrating CNNs and Recurrent Neural Networks (RNNs) to improve the accuracy of offline Handwritten Text Recognition (HTR).

Handwritten digit recognition has been widely applied in fields such as finance, postal services, and education. Early approaches were primarily based on template matching, evolving in the 1990s into shallow models grounded in statistical learning methods, such as the Maximum Entropy method and the Boosting algorithm. These models advanced character recognition technologies due to their strong generalization capabilities and flexibility. The emergence of CNNs marked a significant leap forward in recognition performance. Kolhe P. S. [18] proposed a method based on CNNs for handwritten digit recognition. This novel approach leverages deep learning techniques to automatically extract handwriting features and enhance recognition performance. Ahmed S. S. et al. [19] introduced a handwritten digit recognition method based on the EfficientDet-D4 model, which utilizes region of interest (ROI) annotation and deep feature extraction to improve recognition accuracy under varying handwriting styles and image artifacts. Mohamed N. et al. [20] presented a comparative multi-model approach for handwritten digit recognition, employing SVMs, MLPs, and CNNs on the MNIST dataset. This method demonstrates effective digit recognition and contributes to improved processing efficiency and accuracy in application scenarios such as banking.

The field of handwritten mathematical formula recognition has advanced rapidly, with most deep learning frameworks adopting an encoder-decoder model structure. Tang J. M. et al. [21] proposed a Graph–Encoder–Transformer–Decoder (GETD) approach for handwritten mathematical expression recognition. In this method, mathematical ex-

pressions are modeled as symbol graphs, where candidate symbols are first detected using an object detector and then organized into a graph structure. A graph neural network is employed to aggregate the spatial relationships among symbols, and a Transformer-based decoder is used to identify symbol classes and structural information [22], significantly improving the model's accuracy. During the evolution of handwritten formula recognition models, researchers proposed the concept of tree-based encoders, replacing traditional sequence decoders with tree-structured encoders to better capture the structural information inherent in mathematical expressions [23]. Zhu J. et al. [24] proposed an innovative model for handwritten mathematical expression recognition called Tree-Aware Transformer (TAMER). By introducing a tree-aware module, TAMER maintains the efficiency and flexibility of Transformer while combining the advantages of both sequence decoding and tree structure decoding. It jointly optimizes LaTeX sequence prediction and tree structure modeling tasks, thereby enhancing the model's understanding and generalization ability for complex mathematical structures. Additionally, Wu et al. [25] enhanced the model's representational power by incorporating subgraph attention mechanisms through Graph Neural Networks (GNNs). Li et al. [26] designed a weakly supervised counting module capable of estimating symbol frequencies without requiring symbol-level positional annotations, and integrating this statistical information into the encoder-decoder architecture to correct prediction bias. Yuan et al. [27] developed a syntax-aware driven architecture that leverages syntactic modeling to improve recognition performance. Zhao et al. [28] replaced traditional RNN components with a Transformer-based pretrained encoder, which not only enhanced the model's decoding capabilities but also reduced time complexity. Finally, Bian et al. [29] proposed a bidirectional mutual learning network based on attention aggregation, consisting of a shared encoder and two parallel inverse decoders that iteratively optimize parameters during training. To address the challenge of multi-scale representation of mathematical symbols, a multi-granularity attention aggregation mechanism was designed, significantly improving the model's ability to capture spatial semantics.

In summary, although handwritten text recognition technology has made significant progress, achieving 100% recognition accuracy remains challenging due to factors such as diverse handwriting styles, character ambiguity, and complex structural layouts. Moreover, most existing methods are designed to recognize a single type of character—such as Chinese characters, digits, or mathematical symbols—making them insufficient for handling complex handwritten texts containing mixed character types commonly found in real-world scenarios. These challenges impose higher demands on the generalization and adaptability of recognition models. As a result, developing a unified recognition framework capable of adaptively processing multimodal handwritten text has become one of the key directions in current research.

This paper proposes a Multimodal Handwritten Text Adaptive Recognition algorithm, aiming to address the complex task of recognizing mixed handwritten content in examination-style scenarios. The main contributions of this study are as follows:

(1) A Multimodal Handwritten Text Adaptive Recognition algorithm, MHTR, is proposed for examination scenarios. By integrating a Handwritten Character Classification Module with a Handwritten Text Adaptive Recognition Module, the method enables effective recognition of mixed handwritten content, including Chinese characters, digits, and mathematical expressions.

(2) A Context-aware Recognition Optimization Module is designed to incorporate local semantic and structural information, effectively mitigating misrecognition issues caused by similar character shapes and diverse handwriting styles.

(3) A heterogeneous integrated handwritten text dataset tailored for examination scenarios is constructed, covering various character types such as Chinese characters, digits, and mathematical symbols, with high structural complexity and stylistic diversity.

The rest of the paper is organized as follows. Section 2 introduces the materials and methods; Section 3 presents the experiments and detailed analysis of the results, and finally, the paper is concluded in Section 4.

2. Materials and Methods

2.1. Dataset Construction

To address the limitations of existing publicly available handwritten text datasets, which often exhibit uniform character styles and struggle to accommodate the diversity of handwriting found in examination papers, this study constructs a heterogeneous fused handwritten dataset tailored for exam scenarios. The dataset comprises two parts: a self-collected dataset and publicly available datasets. The public datasets include classic benchmarks such as MNIST, CASIA-HWDB, and CROHME.

2.1.1. Self-Constructed Handwritten Exam Paper Dataset

To enhance the effectiveness of handwritten text recognition research on examination papers, this study constructed a heterogeneous fused dataset specifically for exam handwriting. The data were manually collected and annotated, sourced from mathematics final exam papers at our institution. High-resolution images were captured using a high-pixel device positioned vertically overhead, with a resolution of 3024 × 4032 pixels, and saved in PNG format. During the image acquisition process, images were taken under well-lit conditions at consistent heights and angles to ensure uniform image quality and high resolution. A total of 1000 original images were collected, encompassing diverse handwriting styles (both regular and irregular) as well as varying writing densities (sparse and dense). Some sample images are shown in Figure 1.

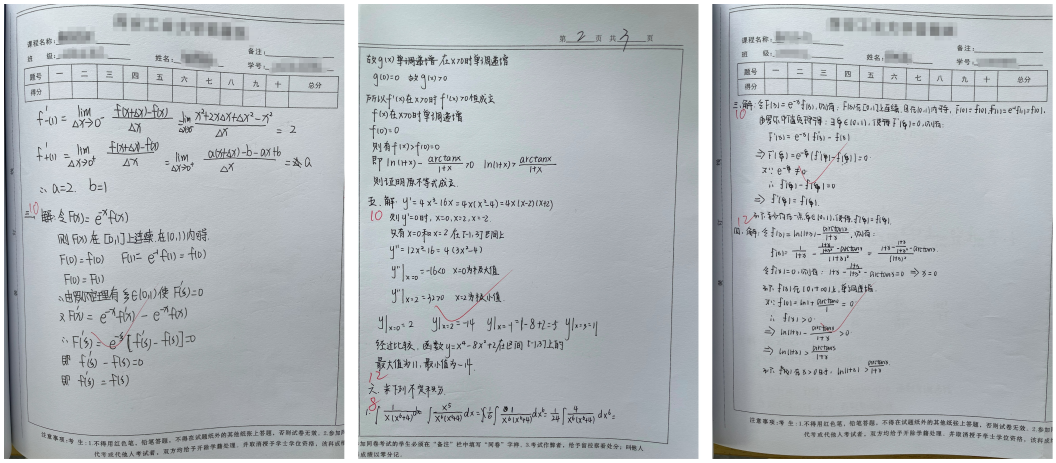

Figure 1. Selected Examination Paper Samples.

The Figure 1 displays handwritten content extracted from actual final mathematics examination papers from school, including numerals, Chinese characters, and mathematical symbols. Given that the exams originate from a Chinese-language educational context, the non-English content primarily consists of mathematics-related Chinese terms.

To enhance the distinguishability of handwritten regions and reduce background interference, this study applies an adaptive thresholding algorithm to binarize the original images. This method dynamically adjusts the threshold based on local grayscale variations, effectively preserving handwriting details and improving image contrast. Compared to

traditional global thresholding methods, it is better suited for scenarios with uneven lighting and diverse handwriting styles. Experimental results demonstrate that this approach significantly enhances the clarity of handwritten regions, providing a solid foundation for subsequent feature extraction and recognition.

To improve the model's generalization ability and mitigate overfitting caused by the limited sample size of a single exam handwriting dataset, data augmentation techniques such as cropping and rotation were applied to the exam handwritten images. First, the original images of 3024 × 4032 pixels were cropped into 640 × 640 pixel patches to emphasize local features and ensure that the model could learn detailed handwriting content. Next, images were rotated by 90°, 180°, and 270° to generate multi-angle samples, effectively expanding the dataset size and enhancing the model's capability to recognize handwriting in various orientations. In total, 12,269 samples were obtained, covering diverse angles and exhibiting significant local feature variations, thereby providing rich and high-quality data support for model training.

Additionally, this study utilizes the LabelImg 1.8.6 toolto perform rectangular bounding box annotations on the exam handwritten images. The annotation order follows a clockwise direction starting from the top-left corner, and the output is saved in YOLO format as .txt files, containing the class labels and the coordinates of the bounding boxes. This annotation approach maintains the original image dimensions while facilitating efficient parsing and accurate recognition by the model. It provides standardized and high-quality labeled data support for handwritten text recognition. The preprocessing workflow is illustrated in Figure 2. The handwritten content shown in the Figure 2 consists of actual written text from mathematics examination papers, with non-English terms primarily being Chinese expressions related to mathematics.

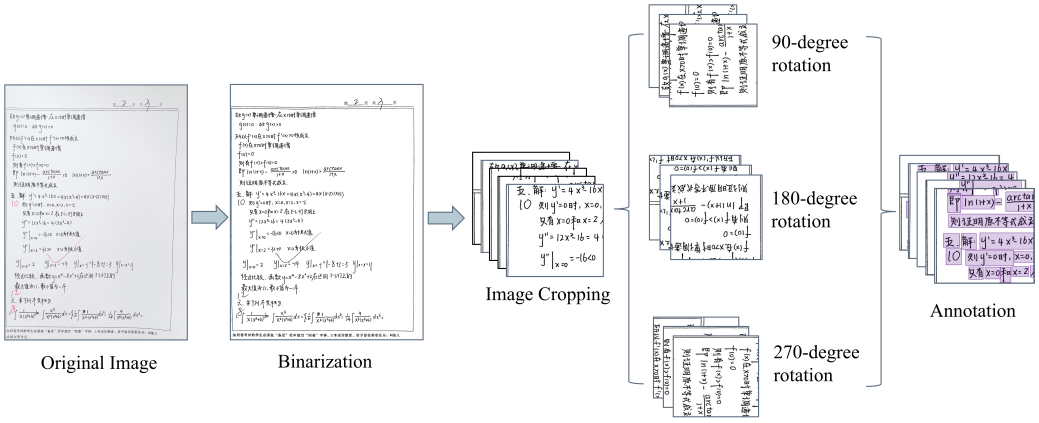

Figure 2. Preprocessing Pipeline for Examination Paper Images.

2.1.2. Public MNIST Dataset

This study selects the MNIST database as the publicly available experimental sample set. The MNIST dataset, created by LeCun et al. in 1998 based on handwritten digit samples from NIST, is a standardized dataset widely used for model training and evaluation in handwritten digit recognition. The dataset contains 70,000 grayscale images of size 28 × 28 pixels, including 60,000 training samples and 10,000 testing samples, covering ten digit classes from 0 to 9. Pixel values range from 0 to 255. Sample handwritten digits from the MNIST dataset are shown in Figure 3.

The MNIST dataset originates from the NIST SD3 and SD1 databases, where the former contains highly standardized handwriting, and the latter features diverse handwriting styles. To enhance the model's generalization capability across different handwriting styles, MNIST combines these two sources and re-divides them into training and testing sets, effectively mitigating the issue of distribution mismatch between training and testing data.

This study employs the MNIST dataset to validate the applicability and effectiveness of the proposed method in a standard handwritten digit recognition task.

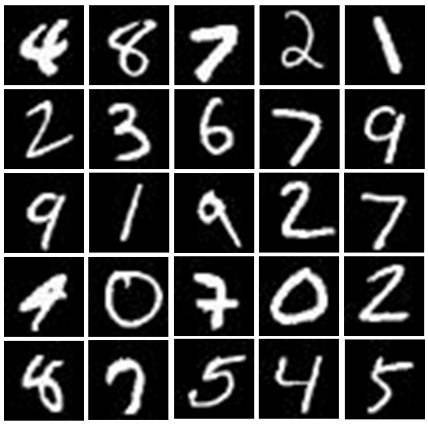

Figure 3. Samples of Handwritten Digit Characters.

2.1.3. Public CASIA-HWDB Dataset

CASIA-HWDB is a classic handwritten Chinese character recognition dataset released by the Institute of Automation, Chinese Academy of Sciences. It is widely used in research related to handwritten Chinese character recognition. The dataset is collected from a large number of different writers, covering diverse handwriting styles, and possesses strong representativeness and practical value. This study selects the CASIA-HWDB1.1 subset as the experimental data source. This subset contains 3755 commonly used Level-1 Chinese characters, with approximately 300 samples per character, totaling about 1.1265 million handwritten image samples. The original data is provided in .gnt file format and requires preprocessing to convert into 64 × 64 pixel images with RGB three channels in .png format for convenient model input and training. Some sample images are shown in Figure 4, which displays examples of Chinese characters from the dataset.

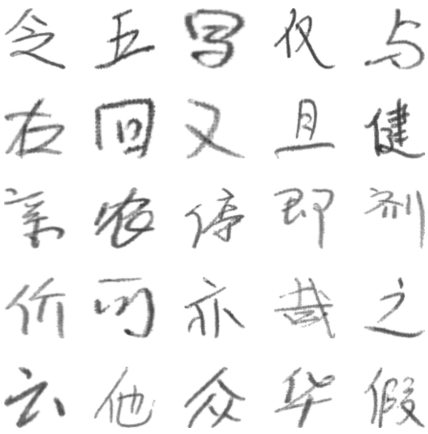

Figure 4. Samples of Handwritten Chinese Characters.

2.1.4. Public CROHME Dataset

The CROHME dataset is a mainstream publicly available resource in the field of handwritten mathematical formula recognition and is widely used for research and evaluation of related algorithms. The dataset originates from past CROHME online handwritten formula recognition competitions and is highly practical and authoritative. The training set contains a total of 8836 handwritten mathematical expression samples, while the test sets comprise data from multiple competition years: CROHME-2013 (671 samples), CROHME-2014 (986 samples), CROHME-2016 (1147 samples), and CROHME-2019 (1199 samples). All

data are stored in InkML file format, which records detailed stroke trajectory information during handwriting. In this study, these trajectory data are converted into image format for model training and testing. Some typical samples are shown in Figure 5.

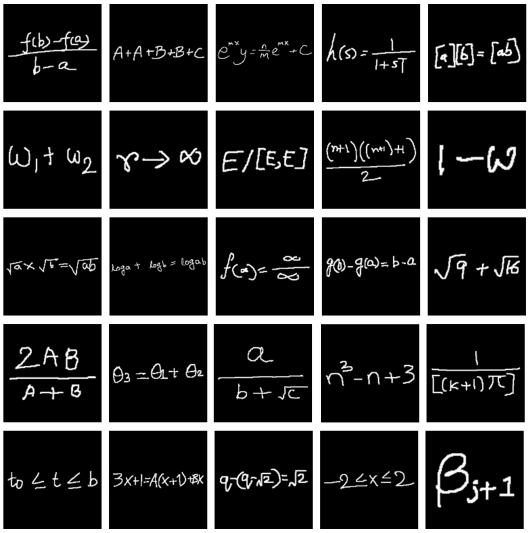

Figure 5. Samples of Handwritten Mathematical Formulas.

2.2. Multimodal Handwritten Text Adaptive Recognition Algorithm

To address the complex recognition challenges of mixed handwritten text in practical scenarios such as examination papers, a Handwritten Character Classification Module is first employed to perform fine-grained classification of different handwritten character categories. Subsequently, based on the classification results, the samples are directed to their respective recognition submodules for further content recognition. Finally, a Context-aware Recognition Module is introduced to improve the accuracy of handwritten Chinese text recognition. The framework diagram of the Multimodal Handwritten Text Adaptive Recognition Algorithm is shown in Figure 6.

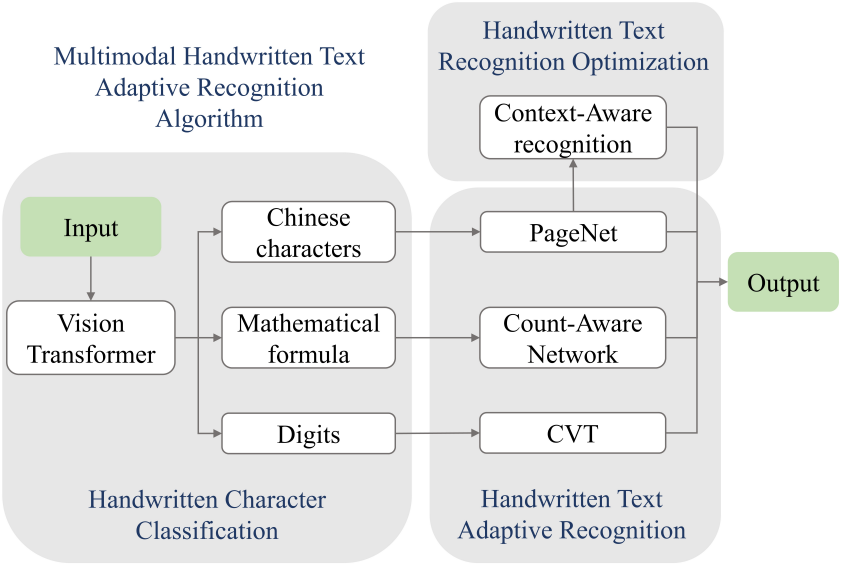

Figure 6. MHTR Model Framework Diagram.

2.2.1. Handwritten Character Classification Module

In practical recognition scenarios of mathematics examination papers, handwritten text typically consists of a mixture of Chinese characters, digits, and mathematical formulas,

with significant differences in structural features among these text categories. Handwritten Chinese characters have numerous strokes and complex constructions, whereas mathematical formulas mainly comprise digits, letters, and symbols. Directly applying a unified recognition model may reduce recognition accuracy. Therefore, this study introduces a handwritten character classification model as a preliminary recognition module to enhance the recognition performance of mixed handwritten text. The architecture of this model is illustrated in Figure 7.

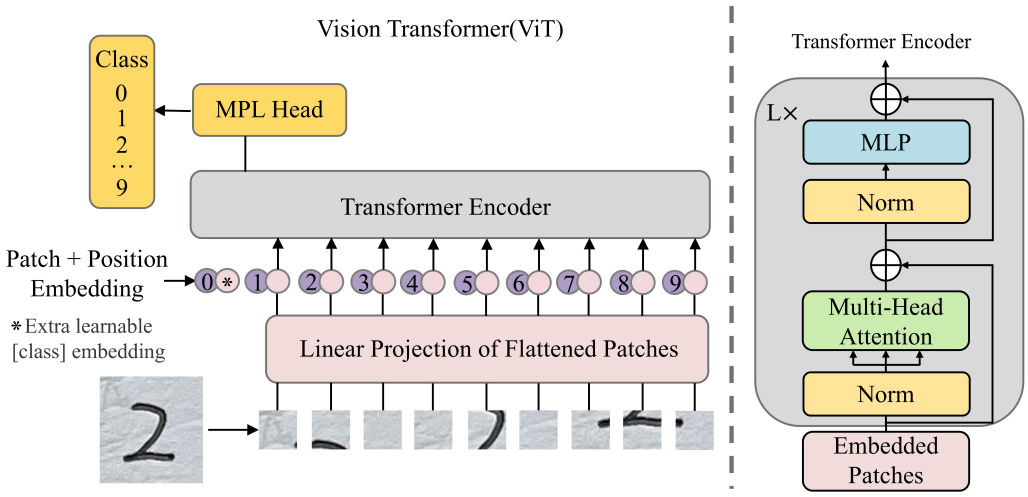

Figure 7. ViT Model Architecture.

The model shown in Figure 7 is based on the Vision Transformer (ViT) architecture, initially proposed by Google in 2020. The ViT model incorporates the self-attention mechanism from the Transformer framework, enabling it to capture global dependencies among long-range features within images. Compared to traditional CNNs, ViT offers superior representational capacity and excels at modeling complex structures, making it particularly well-suited for large-scale image classification tasks. The ViT model primarily consists of the following three components.

(1) Embedding Layer

The image with a resolution of 640 × 640 pixels, obtained after binarization and image enhancement preprocessing, is used as the input to the embedding layer. This module first divides the input image into non-overlapping patches of size 16 × 16 pixels. Each image patch is then flattened and linearly transformed into a feature vector of a unified dimension, forming a set of token vectors. To preserve the spatial order among the image patches, the model incorporates positional embeddings for each token. The resulting sequence of embedded tokens with positional information serves as the input to the Transformer encoder.

(2) Transformer Encoder

In the ViT architecture, the Transformer encoder serves as the core component and is typically composed of multiple stacked identical encoder blocks. Each encoder block primarily contains three key modules: Layer Normalization (LN), Multi-Head Self-Attention (MHSA), and a Multilayer Perceptron (MLP). LN normalizes the input data to reduce feature distribution variance across different samples; MHSA computes global dependencies among image patches through multiple parallel attention heads, thereby extracting contextual information; the MLP, consisting of two fully connected layers and a GELU activation function, facilitates the extraction of complex nonlinear features. Additionally, residual connections are incorporated within the module to alleviate gradient vanishing and explosion

problems in deep networks, improving training stability. Layer Normalization is applied after each submodule to further maintain numerical stability of the outputs.

(3) MLP Head

The MLP Head is responsible for mapping the high-dimensional feature vectors output by the Transformer encoder to the final classification results. This component consists of several fully connected layers and nonlinear activation functions, enabling it to extract deep semantic features while maintaining strong representational capacity and computational efficiency. In mixed-character recognition tasks, the MLP Head plays a crucial role in decision-making. In summary, the ViT model, with its powerful global modeling capability and structural flexibility, can effectively enhance accuracy and robustness in mixed handwritten text classification tasks, providing reliable decision support for the recognition model.

2.2.2. Handwritten Text Adaptive Recognition Module

After processing exam handwritten images through the handwritten character classification model, the character image patches are categorized into handwritten Chinese characters, handwritten digits, and handwritten mathematical formulas, each exhibiting significant differences in structure and writing style. To improve overall recognition accuracy, adaptive recognition models will be developed specifically for handwritten Chinese characters, digits, and mathematical formulas.

Handwritten Chinese Character Recognition Module

To address the characteristics of handwritten Chinese characters, such as dense strokes, complex structures, and multi-line arrangements, this paper adopts a recognition architecture based on PageNet. This model integrates CNNs and Recurrent Neural Networks (RNNs), exhibiting strong multi-scale adaptability and the ability to model text line structures. The PageNet architecture, illustrated in Figure 8, primarily consists of a backbone network, detection and recognition modules, a reading order prediction module, and a graph-structured decoder.

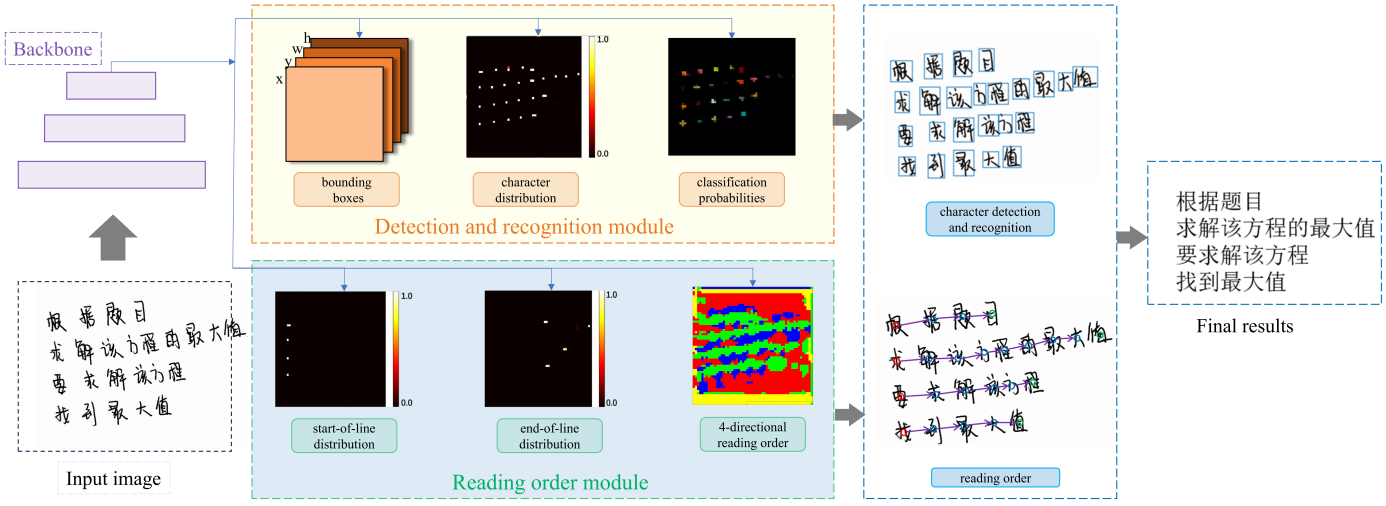

Figure 8. Diagram of the Handwritten Chinese Character Recognition Model Architecture.

(1) Backbone Network

The backbone network is constructed by stacking multiple residual units. For an input image of size $H \times W$, the final output feature map after processing by the backbone network has spatial dimensions of $\frac{H}{16} \times \frac{W}{16} \times 512$. Here, $\frac{W}{16}$ and $\frac{H}{16}$ are denoted as W_g and H_g, respectively.

(2) Detection and Recognition Module

Based on the backbone output, the detection and recognition module is divided into three branches: Character Bounding Box Prediction (CharBox), Character Distribution Estimation (CharDis), and Character Classification (CharCls). First, the input image is divided into a regular grid of size $W_g \times H_g$, where the grid cell in the i-th row and j-th column is denoted as $G^{(i,j)}$. In this structure, the main task of the character bounding box prediction branch is to output a single-character bounding box prediction O_{box} for each $G^{(i,j)}$. Here, the bounding box prediction for grid cell $G^{(i,j)}$ is defined as $O_{\text{box}}^{(i,j)} = \left(x_o^{(i,j)}; y_o^{(i,j)}; \omega_o^{(i,j)}; h_o^{(i,j)}\right)$, and its corresponding coordinates are given by $B_{\text{box}}^{(i,j)} = \left(x_b^{(i,j)}; y_b^{(j)}; \omega_b^{(i,j)}; h_b^{(i,j)}\right)$. The computation formula is given in Equation (1).

$$\begin{aligned} x_b^{(i,j)} &= \left(i - 1 + x_o^{(i,j)}\right) / W_g \times W \\ y_b^{(i,j)} &= \left(j - 1 + y_o^{(i,j)}\right) / H_g \times H \\ \omega_b^{(i,j)} &= \omega_o^{(i,j)} \\ h_b^{(i,j)} &= h_o^{(i,j)} \end{aligned} \tag{1}$$

The character distribution branch outputs a character distribution O_{dis} of shape $W_g \times H_g$, where $O_{dis}^{(i,j)}$ denotes the presence of a character in $G^{(i,j)}$; the character classification branch outputs classification probabilities O_{cls} of shape $W_g \times H_g \times N_{cls}$, where $O_{cls}^{(i,j)}$ represents the classification probabilities over $Ncls$ categories for grid cell $G^{(i,j)}$.

(3) Reading Order Prediction Module

The reading order prediction module is designed to restore the natural arrangement of handwritten characters. It consists of three main steps: first, identifying the starting character of each text line; second, sequentially inferring the next adjacent target character based on spatial and structural relationships; and finally, predicting the ending character of the text line to determine its termination point. Each recognized character is treated as a node in a graph structure and mapped to its corresponding grid cell. The model employs a four-directional reading order prediction method, denoted as O_{rd}, where it searches iteratively along the direction with the highest probability among neighboring grids until locating the next valid node. If the path exceeds boundaries or forms a loop, it is considered to have no subsequent node. By leveraging the distributions O_{sol} (start-of-line) and O_{eol} (end-of-line), the model determines whether a node is the beginning or end of a line, and ultimately constructs a complete reading path from start to end.

(4) Graph-Based Decoding Algorithm

This algorithm integrates the outputs of the detection and recognition module with the reading order prediction module to achieve final recognition at both character and text line levels. First, it obtains the outputs of the three branches, namely B_{box}, O_{dis}, and O_{cls}, and applies Non-Maximum Suppression (NMS) to eliminate redundant bounding boxes, resulting in preliminary recognition results. Each character region is treated as a node in a graph and mapped to the corresponding grid cell based on its center coordinates. Then, using the directional information provided by the four-directional reading order module O_{rd}, the algorithm infers successor nodes within the grid and establishes edge connections among nodes. The distributions O_{sol} (start-of-line) and O_{eol} (end-of-line) are used to determine the beginning and end nodes of each text line, thereby constructing valid reading paths. Finally, the nodes on each path are mapped back to the original detection and classification results to generate the complete recognized text content. The proposed model effectively addresses the challenge of handwritten Chinese character recognition in

complex scenarios. It is particularly well-suited for multi-line arrangements and irregular writing styles, significantly improving the accuracy of Chinese character recognition in such contexts.

Handwritten Math Formula Recognition Module

To address the complex structures such as fractions in handwritten mathematical formulas, this paper proposes a handwritten mathematical formula recognition model based on a Count-Aware Network. The model consists of a backbone network, a Multi-Scale Counting Module (MSCM), and a Counting-Combined Attention Decoder (CCAD), as illustrated in Figure 9. The model employs DenseNet as the backbone network, with the input being a grayscale image $X \in RH' \times W' \times 1$. The backbone extracts a two-dimensional feature map $F \in RH \times W \times 684$ for use by the MSCM and CCAD, where $HH' = WW' = 16$. The MSCM predicts the count of each symbol category, generating a one-dimensional counting vector, which, together with the feature map, is fed into the CCAD decoder to produce the final recognition result.

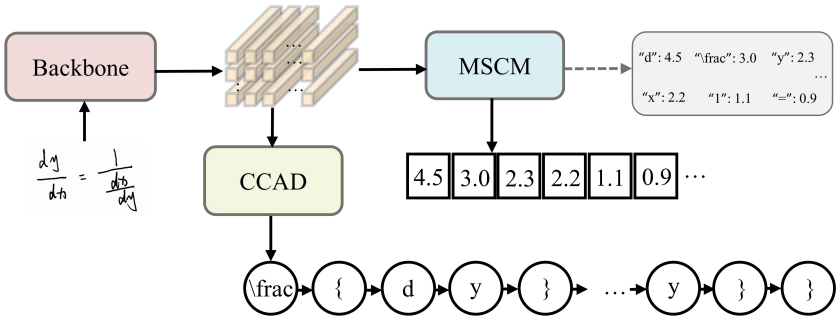

Figure 9. Architecture of the Handwritten Mathematical Formula Recognition Model.

(1) Multi-Scale Counting Module

The core objective of the Multi-Scale Counting Module (MSCM) is to predict the quantity of each symbol category. MSCM is composed of multi-scale feature extraction, channel attention, and pooling operations. To address the issue of symbol size variability caused by differences in handwriting styles, two parallel convolutional branches are introduced, employing 3×3 and 5×5 convolution kernels, respectively, to extract features at different scales. Subsequently, a channel attention mechanism is applied to enhance the features, with the calculation process as follows. The branch feature map $H \in RH \times W \times C$ represents the features extracted from the convolutional layer (either 3×3 or 5×5). The computation process of the enhanced feature S is shown in Equation (2) and (3).

$$O = \sigma(W_1(G(H)) + b_1) \tag{2}$$

$$S = O \otimes g(W_2 O + b_2) \tag{3}$$

Here, G denotes global average pooling, while o and $g(\cdot)$ represent the ReLU and Sigmoid activation functions, respectively. The operator $\otimes$ indicates element-wise channel-wise multiplication. W_1, W_2, b_1, and b_2 are trainable weights.

After obtaining the enhanced feature S, a 1×1 convolution is applied to reduce the number of channels to C, where C corresponds to the number of symbol classes. The symbol counting mainly focuses on the foreground (symbols), so the background responses should be close to zero. Therefore, after the 1×1 convolution, a Sigmoid activation is applied to generate the counting map $H \in \mathbb{R}^{H \times W \times C}$. For each $M_i \in \mathbb{R}^{H \times W}$, it effectively reflects the spatial location of the i-th symbol class. Each M_i serves as a pseudo-density map,

from which the counting vector $V \in \mathbb{R}^{1\times C}$ can be obtained via pooling and summation operators, as shown in Equation (4).

$$V_i = \sum_{p=1}^{H}\sum_{q=1}^{W} M_{i,pq} \tag{4}$$

Here, $V_i \in \mathbb{R}^{1\times 1}$ represents the predicted count of the i-th symbol class by the model. The counting results from different scale branches are fused and averaged to obtain the final predicted counting vector $\mathbf{V}^f \in \mathbb{R}^{1\times C}$, which is then fed into the decoder CCAD for subsequent recognition.

(2) Counting-Combined Attention Decoder

The Counting-Combined Attention Decoder (CCAD) takes as input the feature map $F \in \mathbb{R}^{H\times W\times 684}$ and the counting vector $\mathbf{V}^f$. First, a 1×1 convolution is applied to F, mapping it to a fixed-dimensional feature map $T \in \mathbb{R}^{H\times W\times 512}$.Then, spatial positional encoding P, constructed using sine and cosine functions, is introduced to enhance spatial–location awareness. At decoding step t, the attention weights $\alpha_t \in \mathbb{R}^{H\times W}$ are computed as shown in Equation (5) and (6).

$$e_t = \omega^T tanh(T + P + W_a A + W_h h_t) + b \tag{5}$$

$$\alpha_{t,jj} = \exp(e_{t,jj}) / \sum_{p-1}^{H}\sum_{q-1}^{W} e_{t,pq} \tag{6}$$

Here, ω, b, W_a, and W_h are trainable weights. The coverage attention A represents the cumulative sum of all past attention weights, which helps the model keep track of which parts of the input have been attended to, thereby preventing repeated focus on the same regions.

By combining the attention weights α_t with the feature map F, a context vector $C \in \mathbb{R}^{1\times 256}$ is obtained. In most handwritten mathematical expression recognition methods, the prediction of the current symbol y_t typically relies on the context vector C, the decoder hidden state h_t, and the embedding of the previously predicted symbol $E(y_{t-1})$. However, since C focuses only on local regions of F, and both h_t and $E(y_{t-1})$ lack global context, their effectiveness in capturing comprehensive structural information is limited. To address this, the counting vector V, which reflects the global distribution of symbol categories, is introduced as a global prior. This vector is combined with C, h_t, and $E(y_{t-1})$ to enhance the prediction of y_t, as formulated in Equation (7).

$$p(y_t) = soft\max\left(\omega_o^T(\omega_c C + W_v V + W_t h_t + W_s E)\right) + b_o \tag{7}$$

Here, ω_0, b_o, W_c, W_v, W_t, W_e are all trainable weights

(3) Loss Function

The overall loss function consists of two components, and its formulation is given in Equation (8).

$$L = L_{cls} + L_{counting} \tag{8}$$

Here, L_{cls} is the standard cross-entropy classification loss, which measures the discrepancy between the predicted probability $p(y_t)$ and its corresponding ground truth label. The ground truth for the count of each symbol category is denoted as V. The term $L_{counting}$ represents a smoothed $L1$ regression loss, which is defined in Equation (9).

$$L_{counting} = smooth_{L1}(V, \hat{V}) \tag{9}$$

Handwritten Digit Recognition Module

This section focuses on the recognition of handwritten digits with varying scales, rotations, and distortions, based on the Convolutional Vision Transformer (CVT) model for handwritten digit recognition. The CVT model is a hybrid architecture that combines CNN and ViT, aiming to leverage the strength of CNN in capturing local features and the capability of Transformers in modeling global dependencies. In the task of handwritten digit recognition, the CVT model can effectively enhance recognition performance. The overall structure is illustrated in Figure 10.

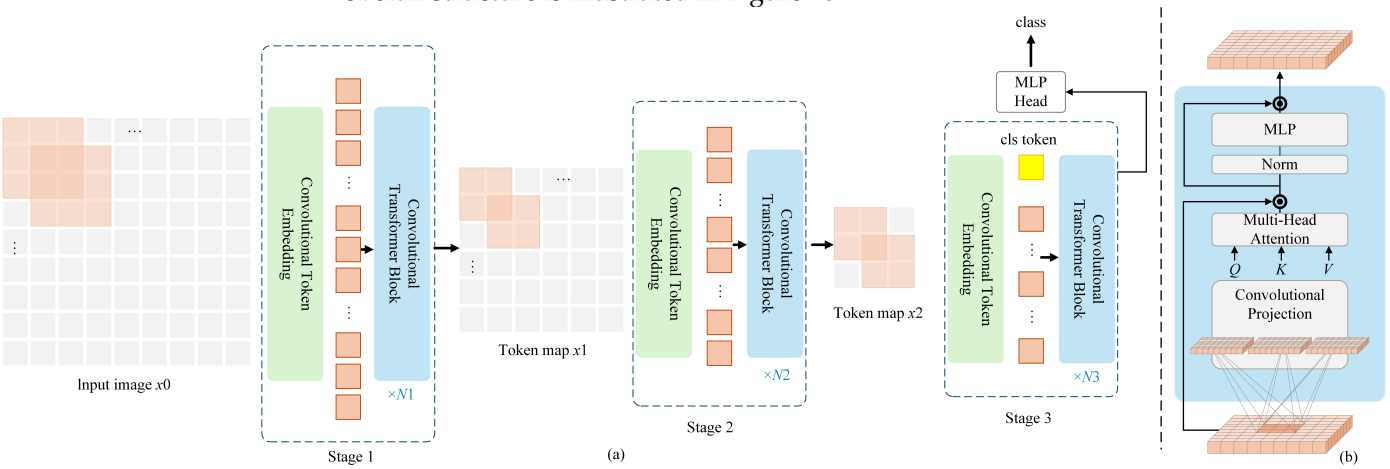

Figure 10. Handwritten Digit Recognition Model Framework. (**a**) Overall architecture, showing the hierarchical multi-stagestructure facilitated by the Convolutional Token Embedding laver. (**b**) Details of the Convolutional Transformer Blockwhich contains the convolution projection as the first layer.

CVT integrates the strengths of convolutional operations and Transformers, aiming to compensate for the limitations of ViT in capturing local features. By incorporating convolutional operations into the Transformer architecture, CVT enhances the model's ability to extract fine-grained local details while preserving the global context modeling capability inherent to Transformers.

(1) Proposed Convolutional Token Embedding Mechanism

Convolutional Token Embedding (CTE) is an improved mechanism introduced in ViT, designed to enhance model performance and efficiency. By removing the traditional position embedding in ViT and replacing it with convolutional layers, CTE effectively captures spatial information in a dynamic manner. This approach not only reduces the number of position-related parameters but also enables more efficient computation, as the convolutional layers inherently encode local spatial dependencies within the input data.

(2) Proposed Convolutional Transformer Block Mechanism

The Convolutional Transformer Block (CTB) integrates the strengths of the CNN and Transformer architectures. In this module, depthwise separable convolutions are incorporated into the Transformer layers to enhance the model's ability to capture local features within image regions. Compared to conventional convolution operations, this design significantly reduces overall computational complexity while maintaining recognition performance, as convolution operations are generally more efficient than self-attention mechanisms. Furthermore, by reducing the reliance on position embeddings and minimizing the number of self-attention layers within the Transformer, the CTB effectively decreases the total number of model parameters.

2.2.3. Context-Aware Handwritten Text Recognition Optimization Module

To enhance the accuracy of handwritten text recognition, this paper introduces a context-aware recognition module that integrates a Bidirectional Long Short-Term Memory (Bi-LSTM) network with an attention-enhanced LSTM structure. The Bi-LSTM captures both forward and backward contextual dependencies, thereby improving the model's ability to perceive sequential relationships. Meanwhile, the attention mechanism dynamically focuses on key information regions during the decoding stage, enabling more effective association between local strokes and the overall character structure, which significantly boosts recognition performance. The overall architecture is illustrated in Figure 11, where the input image is a sample segment taken from a mathematics examination paper, demonstrating the model's capability to handle complex handwritten text within real exam scenarios.

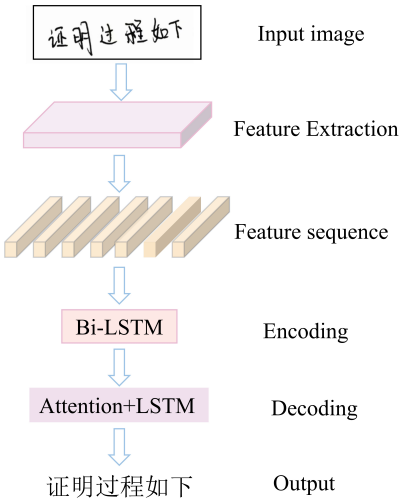

Figure 11. Context-Aware Handwritten Text Recognition Model.

This module employs a Bi-LSTM as the encoder and an attention-based LSTM as the decoder, forming an encoder–decoder architecture that effectively enhances the recognition performance for handwritten text with complex structures and diverse writing styles.

Context Feature Encoding Based on Bi-LSTM

Contextual information plays a crucial role in character recognition. To capture semantic dependencies in both forward and backward directions, this study employs a Bi-LSTM network to encode the input sequence, extracting contextual features from both directions. Compared with traditional RNNs, Bi-LSTM alleviates gradient vanishing issues through its gating mechanism and enables information integration across time steps. To further enhance feature representation capability, two layers of Bi-LSTM are stacked, as illustrated in Figure 12.

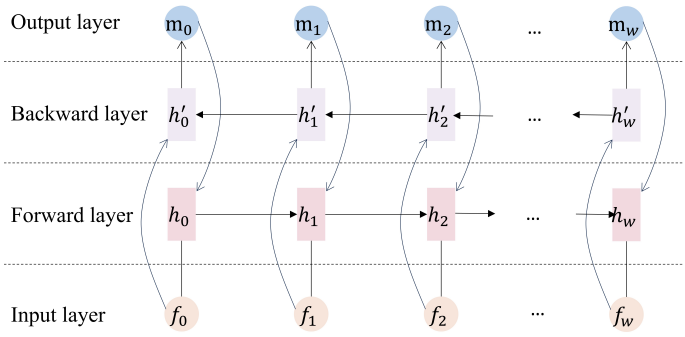

Figure 12. Bi-LSTM architecture diagram.

To achieve context-aware feature encoding, a two-layer Bi-directional Long Short-Term Memory (Bi-LSTM) architecture is stacked to extract more refined features. Let $F_s\{f_0, f_1, f_2, \ldots, f_w\}$ denote the input sequence of feature vectors. The set $\{h_0, h_1, h_2, \ldots, h_w\}$ represents the hidden states at each time step during the forward pass of the Bi-LSTM, while $\{h'_0, h'_1, h'_2, \ldots, h'_w\}$ corresponds to the hidden states obtained from the backward pass. The final context-aware feature vectors, denoted by $\{m_0, m_1, m_2, \ldots, m_w\}$, are formed by combining the forward and backward representations and serve as the encoded output. The variable w represents the width of the input feature map. This process is formally described in Equation (10).

$$F_r = L(F_s) \tag{10}$$

Here, F_r denotes the context feature vectors extracted by the Bi-LSTM, which provide rich semantic support for the subsequent decoding stage.

LSTM Context Feature Decoding Based on Attention Mechanism

To further improve the performance of sequence decoding, this study incorporates an attention mechanism into the LSTM decoder. This allows the model to selectively focus on the most relevant features in the input sequence at each time step, thereby mitigating the limitations of traditional LSTM in modeling long-range dependencies.

The attention mechanism plays a critical role in sequence modeling. Its core idea is to dynamically focus on the input features most relevant to the current decoding state during output generation. Functionally, attention mechanisms can be categorized into spatial attention and temporal attention; in terms of implementation, they are divided into soft attention and hard attention. This study adopts the soft attention mechanism, which assigns varying weights to different positions in the input sequence. During decoding, the model dynamically adjusts its focus to enhance the capture of key information, thereby improving the decoder's ability to model both the local strokes and overall structure of handwritten characters.

The soft attention mechanism applies the Softmax function to normalize the attention weights. Let the sequence of feature vectors extracted by the backward LSTM be denoted as shown in Equation (11).

$$C = \left\{c_1, c_2, \cdots, c_k, \cdots, c_m\right\} \tag{11}$$

Here, C serves as the input to the soft attention mechanism. Let e_i^t denote the attention score computed at time step t for the input feature c_i, which measures the relevance between the decoder state and the feature vector. The attention weights α_i^t are obtained by applying a Softmax normalization, ensuring that the sum of the weights equals 1. The calculation is expressed by Equation (12).

$$\alpha_i^t = \frac{\exp\left(e_i^t\right)}{\sum_{j=1}^{m} \exp\left(e_j^t\right)} \tag{12}$$

The decoder generates the context vector c_t by performing a weighted fusion of relevant information based on the attention distribution weights α_i^t, as defined in Equation (13).

$$c^t = \sum_{i=1}^{m} \exp(\alpha_i c_i) \tag{13}$$

Figure 13 illustrates the detailed process of computing the correlation between feature vector sequences.

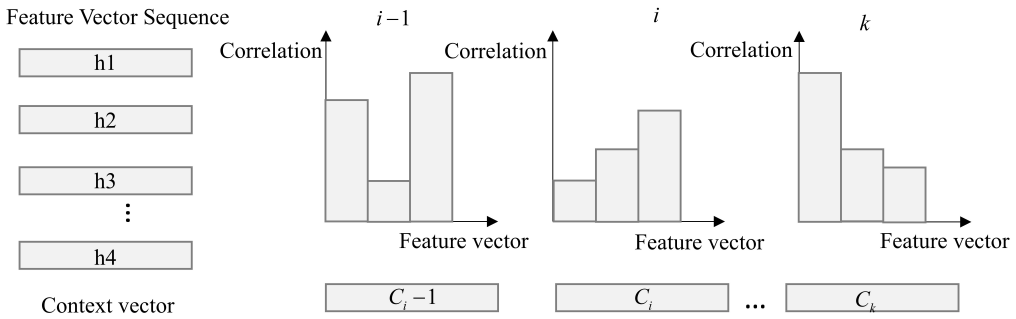

Figure 13. Correlation calculation.

Long Short-Term Memory (LSTM) networks represent a significant improvement over traditional Recurrent Neural Network (RNN) architectures, specifically designed to address the common issues of vanishing and exploding gradients encountered when processing long sequences. Compared to standard RNN models, LSTMs incorporate specialized memory cells and gating mechanisms, which enhance the model's ability to learn and retain long-term dependencies. This results in improved representational capacity and stability in sequence modeling tasks. The internal structure of the LSTM is illustrated in Figure 14.

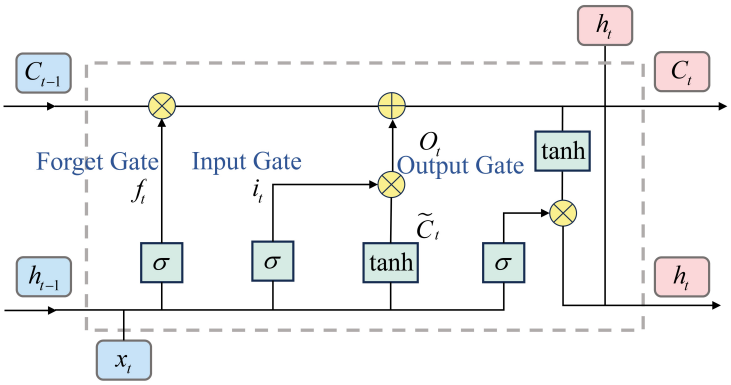

Figure 14. LSTM architecture diagram.

The LSTM network internally contains three key gating structures: the input gate, the forget gate, and the output gate. The gating mechanism essentially acts as a selective channel that controls the flow of information, typically composed of a Sigmoid activation function combined with element-wise multiplication operations. From the graph of the Sigmoid function, it can be observed that its output values are mainly concentrated near 0 and 1. This characteristic enables the neural network to selectively retain or discard certain information. An output value close to 0 implies that almost no information is allowed to pass through, whereas a value near 1 means information can be transmitted without obstruction. C_t denotes the memory cell, which is capable of retaining information from the previous time step.

The Input Gate receives input signals from both the current time step and the previous time step. Its output at the current time step is denoted as i_l^t, and is computed as shown in Equation (14).

$$i_l^t = \sigma\left(\sum_{j=1}^{J} w_u x_j^t + \sum_{c=1}^{C} w_{cl} s_c^{t-1}\right) \tag{14}$$

where o denotes the Sigmoid activation function, x_i^t represents the input vector to the LSTM unit at the current time step t, and s_c^{t-1} denotes the cell state information passed

from the previous time step. w refers to the weight parameter matrix associated with the gating mechanism.

The Forget Gate is primarily used to regulate and update the state information within the memory cell. Its specific computation is defined as shown in Equation (15).

$$f_{\phi}^{t} = \sigma\left(\sum_{j=1}^{J} w_{j\phi} x_{j}^{t} + \sum_{c=1}^{C} w_{c\phi} s_{c}^{t-1}\right) \tag{15}$$

The memory cell C_t lies at the core of the entire memory module and is connected via a loop known as the Constant Error Carousel (CEC), which enables the continuous transmission of information across the time dimension. The state update process generally consists of two stages: first, computing the current hidden layer activation; then, under the joint regulation of the Input Gate and Forget Gate, updating the internal state of the memory cell. This process can be described by Equations (16) and (17).

$$a_{c}^{t} = \tanh\left(\sum_{j=1}^{J} w_{jc} x_{j}^{t} + \sum_{h=1}^{H} w_{hc} b_{h}^{t-1}\right) \tag{16}$$

$$s_{c}^{t} = a_{\phi}^{t} s_{c}^{t-1} + a_{l}^{t} a_{c}^{t} \tag{17}$$

The activation value of the Output Gate is computed based on the current time step's memory cell state. The specific computation method is given by Equation (18).

$$a_{\omega}^{t} = \sigma\left(\sum_{j=1}^{J} w_{j\omega} x_{j}^{t} + \sum_{c=1}^{C} w_{c\omega} s_{c}^{t}\right) \tag{18}$$

The final output of the LSTM is obtained by performing an element-wise multiplication between the activation result of the Output Gate and the current state of the memory cell, as shown in Equation (19).

$$a_{c}^{t} = \tanh\left(a_{\omega}^{t} s_{c}^{t}\right) \tag{19}$$

3. Experiment and Performance Analysis

3.1. Experimental Configuration

To more intuitively demonstrate the effectiveness and comparability of the model training results in this study, a brief description of the hardware environment and relevant software dependencies is provided. Furthermore, to ensure fairness in performance evaluation and consistency in comparative analysis, all baseline methods in the experiments were conducted under the same environment. The devices and parameters used in the experiments are listed in the table below, while the parameter configurations for model implementation and training are detailed in Tables 1 and 2.

Table 1. Experimental Environment.

Name	Configuration
Operating system	Windows 11
CPU	Intel (R) Xeon (R) Platinum 8352 V
GPU	NVIDIA GeForce RTX 4090 (24 GB)
CUDA	11.8
Python	Python 3.10
PyTorch	PyTorch 2.1.2

Table 2. Training parameter settings.

Name	Configuration
Image Size	64×64
Weight Decay	0.005
Batch Size	32
Learning Rate	0.01
Number of iterations	200

3.2. Experimental Indicators

Since the recognition task focuses on semantic correctness, this study uses character recognition accuracy as the evaluation metric. Therefore, accuracy serves as the primary indicator for assessing model performance in the experiments. Accuracy is one of the most widely used evaluation metrics in handwritten recognition tasks, measuring the proportion of correctly predicted samples out of the total samples, thereby reflecting the overall predictive capability of the model. Its calculation formula is presented in Equation (20).

$$Accuracy = \frac{n_c}{N} \times 100\% \tag{20}$$

where n_c denotes the number of correctly predicted samples, and N represents the total number of samples.

3.3. Comparison Experiments of Different Recognition Models

During the handwritten text recognition testing process, to validate the performance of the proposed Multimodal Handwritten Text Adaptive Recognition Algorithm, comparative experiments were conducted between the proposed method and other classical recognition models. The experimental results are presented in Table 3.

Table 3. Comparison of Different Handwriting Recognition Models.

Model	Character Recognition Accuracy
CRNN-CTC	80.53%
Transformer	82.82%
MHTR (Proposed)	86.63%

As shown in Table 3, compared with single recognition models, the proposed Multimodal Handwritten Text Adaptive Recognition algorithm achieves a character recognition accuracy of 86.63% on the validation set, representing a significant improvement over the other models.

Single recognition models exhibit certain limitations in the mixed-text recognition task involving digits, Chinese characters, and mathematical formulas. CRNN-CTC performs well for handwritten digits and simple characters, but it struggles with complex formulas containing two-dimensional structures such as subscripts, superscripts, and fractions, and is prone to misrecognition due to conflicts between class-specific features. Transformer demonstrates strong capability in capturing complex Chinese character features; however, its structural modeling ability for formulas is limited, and it faces challenges in achieving balanced optimization across different categories in mixed data. These single models are unable to perform deep optimization for specific categories, especially formulas, which hinders their overall recognition performance.

In contrast, the proposed Multimodal Handwritten Text Adaptive Recognition algorithm first employs a Handwritten Character Classification Module to determine the category of the input. Handwritten digits, Chinese characters, and formulas are then fed

into dedicated recognition sub-models for targeted optimization. This strategy improves recognition accuracy and robustness across all categories, with notable performance in formula recognition. It also avoids feature conflicts between categories and enhances the model's generalization ability and adaptability.

3.4. Comparison Experiments of Text Recognition Optimization Models

In the training of the context-aware handwritten text recognition optimization model, the publicly available CASIA-HWDB dataset and a self-constructed exam paper dataset were utilized. The combined dataset consists of a total of 257,922 handwritten Chinese character images. During the experiments, approximately 80% of the samples were allocated to the training set for model training, while the remaining 20% were reserved for testing model performance. During testing, the multimodal adaptive handwritten text recognition model was optimized, and the optimized recognition model proposed in this study was compared with other handwritten text recognition models. The results are presented in Table 4.

Table 4. Comparison of Different Handwritten Chinese Character Recognition Models.

Model	Character Recognition Accuracy
CRNN	84.63%
SATR	85.32%
MFCNN	85.58%
MHTR (Proposed)	86.63%
Optimization Model (Proposed)	88.64%

After completing the network training, this study applied the model to the testing phase using real exam papers for handwritten Chinese character recognition. During the experiments, portions of handwritten text in the exam papers were recognized, yielding preliminary results.

As shown in Table 4, a comparison with other model approaches reveals performance differences in offline handwritten Chinese character recognition tasks. The CRNN model achieved a character recognition accuracy of 84.63%. Although this model can capture temporal features in images, it falls short in handling intricate handwriting details and long-term dependencies, limiting its accuracy. The SATR (Sequence Attention) model attained an accuracy of 85.32%. While its introduced sequence attention mechanism enhances focus on key information, its inherent limitations prevent it from fully exploiting contextual semantic information, resulting in modest performance gains. The MFCNN model, leveraging multi-scale feature fusion, achieved an accuracy of 85.58%. This approach improves the extraction of Chinese character features at different scales to some extent, but potential loss of fine details during feature fusion leaves room for accuracy improvement. The multimodal handwritten adaptive recognition algorithm achieved 86.63%, showing a significant improvement over the previous methods, indicating better feature extraction. However, it still falls short of the proposed method in terms of depth in contextual semantic understanding and feature integration.

The Context-Aware Recognition Optimization model proposed in this paper demonstrated superior performance, achieving an accuracy of 88.64%, which is a 2.01 percentage point improvement over the multimodal handwritten text adaptive recognition algorithm. By optimizing the context-awareness mechanism, this model more effectively captures the contextual semantic correlations of handwritten Chinese characters, enhances recognition capability for complex writing styles and ambiguous characters, and fully utilizes contextual information to guide the recognition process, thereby significantly improving character recognition accuracy.

3.5. Visualization of Recognition Results for the MHTR

The trained network architecture was applied to actual exam paper images to recognize the handwritten text contained within. Specifically, the handwritten regions extracted from the exam paper images were input into the adaptive handwritten text recognition model, which processed and generated the corresponding recognition results. Finally, the results are presented in the visualized format shown in Figure 15, intuitively demonstrating the model's recognition performance in real-world scenarios. The recognized content displayed in the figure is an example from an actual exam paper, highlighting the practical effectiveness of the model.

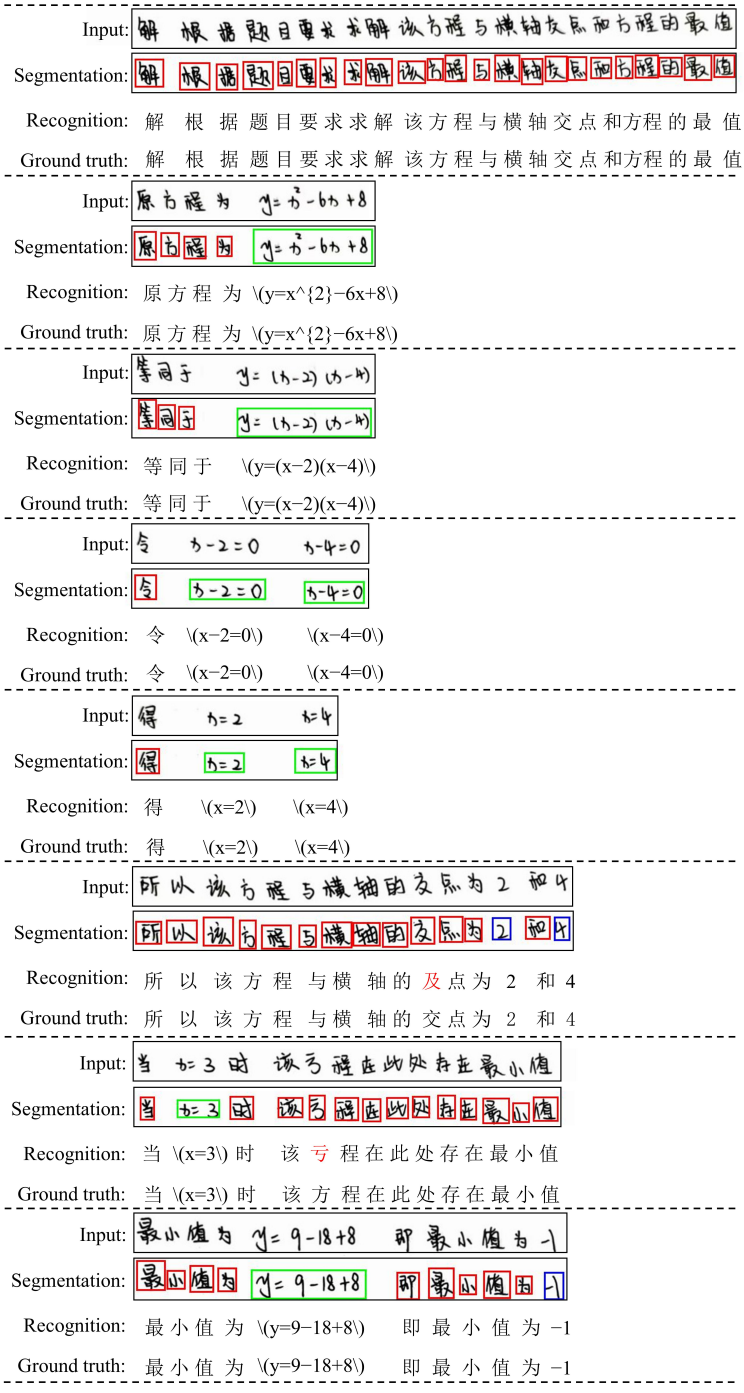

Figure 15. Visualization of MHTR Model Recognition Results.

As shown in Figure 15, the red bounding boxes indicate Chinese characters, the blue bounding boxes indicate digits, and the green bounding boxes indicate mathematical formulas. The red text in the figure highlights the locations where recognition errors occurred. Analysis of the visualization results reveals that the Multimodal Handwritten Text Adaptive Recognition Algorithm proposed in this paper achieves relatively ideal recognition performance on real examination papers, with a high overall recognition accuracy rate. However, there are still misrecognition issues in the recognition of diverse Chinese character writing styles and irregularly written characters, which calls for further optimization of the recognition model.

3.6. Visualization of Recognition Results for the Text Optimization Model

In traditional handwritten character recognition, especially under complex writing conditions, misrecognition often occurs due to the similarity of character shapes or irregular handwriting. In the practical application of the proposed Multimodal Handwritten Text Adaptive Recognition Algorithm, the model mistakenly recognized the Chinese character "方" as "亏" and "交" as "及". Such errors mainly stem from the model's inability to sufficiently capture the contextual relationships and semantic information between characters during training, resulting in deviations in recognizing certain individual characters.

To address this issue, this study proposes a Context-Aware Handwritten Text Recognition Optimization model that enhances the model's understanding of character sequences by capturing local contextual information. When optimizing recognition results, the model comprehensively considers the semantic and structural relationships between characters, enabling it to correct misrecognized characters based on the surrounding context. In experiments, by incorporating the context-aware mechanism, the model successfully corrected the misrecognized characters "亏" and "及" to their true forms "方" and "交", respectively.

To further resolve the aforementioned problems, the Context-Aware Handwritten Text Recognition Optimization model introduces a context modeling mechanism that strengthens the ability to capture semantic and structural correlations within character sequences. During recognition optimization, the model takes into account the contextual dependencies between characters, allowing the system to correct misrecognized characters within the overall semantic environment. Figure 16 illustrates the comparison of recognition results before and after optimization.

Figure 16. Visualization of Text Optimization Model Recognition Results.

Based on the optimized visualization results, the model successfully corrected the characters "亏" and "及" to their correct Chinese characters "方" and "交" at the blue text locations. The algorithm proposed in this study demonstrates a significant improvement in recognition accuracy, validating the effectiveness of the context-aware handwritten text recognition optimization algorithm in enhancing recognition precision. This confirms

the advantage of the algorithm in addressing misrecognition issues within handwritten text recognition.

4. Discussion

4.1. Model Innovations and Advantages

This paper proposes a Multimodal Handwritten Text Recognition (MHTR) algorithm for mixed handwritten text recognition, addressing the limitations of existing methods that are tailored to single character types. The proposed model consists of a Handwritten Character Classification Module and a Handwritten Text Adaptive Recognition Module. The classification module categorizes input images into different types, such as Chinese characters, digits, and mathematical formulas, while the adaptive recognition module dynamically selects and applies the corresponding sub-recognition network for each character type, enabling accurate recognition of multiple categories of handwritten characters. In addition, a Context-Aware Recognition Optimization Module is introduced to further mitigate recognition errors caused by similar character shapes and diverse handwriting styles. Experimental results show that the proposed algorithm achieves an accuracy of 86.63% on the constructed handwritten examination dataset, representing a significant improvement over existing methods. Moreover, the introduced context-aware optimization mechanism demonstrates strong error-correction capability in confusing and misrecognition scenarios, highlighting the method's generalization ability and practical application potential in complex and dynamic environments.

4.2. Limitations and Future Work

4.2.1. Limitations

Although this study achieves accurate recognition of mixed handwritten text in examination papers, certain limitations remain in the research and development of recognition algorithms for such handwriting. These limitations can be summarized as follows:

(1) The experiments were conducted only on horizontally written handwriting and did not address recognition of handwritten content at arbitrary angles, which to some extent limits the applicability of the model. In addition, the model's recognition performance still faces certain limitations in extreme scenarios, such as severe writing distortions or heavy noise interference.

(2) In the recognition of handwritten text on exam papers, current research mainly focuses on single visual input data types from offline exam papers (such as handwritten text in images), while neglecting other modal information present in the papers (such as paper quality, pen pressure, etc.), which may have a certain impact on handwriting recognition.

4.2.2. Future Work

To address the above limitations, future research can be further improved in the following aspects:

(1) Future research should incorporate multi-angle and multi-orientation handwritten text data to enhance the model's adaptability to arbitrary writing directions. A more diverse dataset covering extreme handwriting styles and complex scenarios should be built to improve the model's generalization ability under non-ideal conditions.

(2) In the field of exam paper handwritten text recognition, future work can explore multi-modal data fusion techniques, combining image information with other types of sensor data (such as pressure sensors and tilt sensors) to further improve recognition accuracy.

5. Conclusions

This paper proposes the Multimodal Handwritten Text Adaptive Recognition algorithm, which consists of a Handwritten Character Classification Module and a Handwritten Text Adaptive Recognition Module. First, the Handwritten Character Classification Module performs fine-grained classification of handwritten text into categories such as Chinese characters, mathematical formulas, and digits. Subsequently, the classification results are fed into corresponding recognition sub-network models to achieve recognition of handwritten Chinese characters, mathematical formulas, and digits.

The Multimodal Handwritten Text Adaptive Recognition Algorithm demonstrates significant advantages in recognizing mixed handwritten text in exam papers. Experimental results show that the proposed model achieves a recognition accuracy of 86.63% on the dataset, outperforming other methods in terms of accuracy.

Furthermore, to address misrecognition issues caused by similar character shapes or differences in handwriting styles, this study introduces a Context-Aware Recognition Optimization Module. By capturing local contextual information, the module further enhances the model's understanding of character sequences. During recognition result optimization, the model comprehensively considers the semantic and structural relationships among characters, enabling it to correct misrecognized characters based on the contextual environment. In experiments, by incorporating the context-aware mechanism, the model exhibits strong adaptive correction capabilities across multiple misrecognition cases, successfully correcting the characters "亏"and "及" to their correct forms "方" and "交", respectively.

Although the proposed model demonstrates strong performance in mixed handwritten text recognition, it still exhibits certain limitations when handling extreme handwriting distortions, severe noise interference, and other challenging scenarios. Future work will focus on incorporating more diverse and complex datasets to further enhance the model's recognition accuracy and generalization capability under complex and non-ideal conditions. In summary, the proposed Multimodal Handwritten Text Adaptive Recognition algorithm provides a highly robust and accurate solution for mixed handwritten text recognition, offering significant practical value for advancing intelligent grading systems and other application domains involving the processing of complex handwritten information.

Author Contributions: Conceptualization, H.S. and X.F.; methodology, H.S. and Z.Z.; software, Y.W.; validation, Y.W. and C.Z.; data curation, C.Z.; writing—original draft preparation, Z.Z.; writing—review and editing, H.S.; funding acquisition, H.S. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Xi'an Technological University Graduate Education and Teaching Reform Research Project (Grant No. XAGDYJ240107).

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

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

References

1. Casey, R.; Nagy, G. Recognition of Printed Chinese Characters. *IEEE Trans. Electron. Comput.* **1966**, *15*, 91–101. [CrossRef]
2. Ahmed, S.; Islam, S. Methods in Detection of Median Filtering in Digital Images: A Survey. *Multimed. Tools Appl.* **2023**, *82*, 43945–43965. [CrossRef]
3. Du, Y.; Yuan, H.; Jia, K.; Li, F. Research on Threshold Segmentation Method of Two-Dimensional Otsu Image Based on Improved Sparrow Search Algorithm. *IEEE Access* **2023**, *11*, 70459–70469. [CrossRef]
4. Petlu, M.; Shanmugam, U.; Elangovan, K. Damaged Number Plate Detection to Improve the Accuracy Rate Using Bernsen Algorithm over Genetic Algorithm. *Proc. Adv. Sustain. Constr. Mater.* **2023**, *2655*, 020074. [CrossRef]

5. Di, Y.; Li, R.; Tian, H.; Guo, J.; Shi, B.; Wang, Z.; Yan, K.; Liu, Y. A maneuvering target tracking based on fastIMM-extended Viterbi algorithm. *Neural Comput. Appl.* **2023**, *37*, 7925–7934. [CrossRef]
6. Fan, G.; Han, Y.; Li, J.; Peng, L.; Yeh, Y.; Hong, W. A Hybrid Model for Deep Learning Short-Term Power Load Forecasting Based on Feature Extraction Statistics Techniques. *Expert Syst. Appl.* **2024**, *238*, 122012. [CrossRef]
7. Shen, L.; Jiang, C.J.; Liu, G.J. Satellite Objects Extraction and Classification Based on Similarity Measure. *IEEE Trans. Syst. Man Cybern. Syst.* **2015**, *46*, 1148–1154. [CrossRef]
8. Zhu, L.; Chen, T.; Yin, J.; See, S.; Liu, J. Learning Gabor Texture Features for Fine-Grained Recognition. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Paris, France, 1–6 October 2023; pp. 1621–1631. [CrossRef]
9. Barbhuiya, A.A.; Karsh, R.K.; Jain, R. A Convolutional Neural Network and Classical Moments-Based Feature Fusion Model for Gesture Recognition. *Multimed. Syst.* **2022**, *28*, 1779–1792. [CrossRef]
10. Wei, X.; Lu, S.; Lu, Y. Compact MQDF Classifiers Using Sparse Coding for Handwritten Chinese Character Recognition. *Pattern Recognit.* **2018**, *76*, 679–690. [CrossRef]
11. Valkenborg, D.; Rousseau, A.; Geubbelmans, M.; Burzykowski, T. Support Vector Machines. *Am. J. Orthod. Dentofac. Orthop.* **2023**, *164*, 754–757. [CrossRef] [PubMed]
12. Borucka, A.; Kozłowski, E.R.; Parczewski, R.; Antosz, K.; Gil, L.; Pieniak, D. Supply Sequence Modelling Using Hidden Markov Models. *Appl. Sci.* **2023**, *13*, 231. [CrossRef]
13. Ciresan, D.C.; Meier, U.; Gambardella, L.M. Convolutional Neural Network Committees for Handwritten Character Classification. In Proceedings of the 2011 International Conference on Document Analysis and Recognition, Beijing, China, 18–21 September 2011; pp. 1135–1139. [CrossRef]
14. Wang, S.; Chen, L.; Wu, C.; Fan, W.; Sun, J.; Naoi, S. CNN Based Handwritten Character Recognition. In *Advances in Chinese Document and Text Processing*; World Scientific Publishing: Singapore, 2017; pp. 57–77.
15. Chen, Y.; Zhang, H.; Liu, C. Improved Learning for Online Handwritten Chinese Text Recognition with Convolutional Prototype Network. In Proceedings of the Document Analysis and Recognition—ICDAR 2023, San José, CA, USA, 21–26 August 2023; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2023; pp. 38–53. [CrossRef]
16. Bharati, P. A Hybrid Approach for Denoising and Recognition of Handwritten Characters Using Deblur GAN-CNN. In Proceedings of the 2024 IEEE 4th International Conference on ICT in Business Industry & Government (ICTBIG), Indore, India, 13–14 December 2024; pp. 1–7. [CrossRef]
17. Sunori, S.; Sumithra, S. Enhancing Handwritten Text Identification through a Hybrid CNN-RNN Method. In Proceedings of the 3rd International Conference on Optimization Techniques in the Field of Engineering (ICOFE-2024), Tamil Nadu, India, 11–12 October 2024; pp. 1–13. [CrossRef]
18. Kolhe, P.S. Various Approaches of Convolutional Neural Network-Based Recognition of Handwritten Devanagari Characters. In Proceedings of the 2023 3rd International Conference on Innovative Sustainable Computational Technologies (CISCT), Dehradun, India, 8–9 September 2023; pp. 1–4. [CrossRef]
19. Ahmed, S.; Mehmood, Z.; Awan, I.; Yousaf, R. A Novel Technique for Handwritten Digit Recognition Using Deep Learning. *J. Sens.* **2023**, *2023*, 2753941. [CrossRef]
20. Mohamed, N.; Josphineleela, R.; Madkar, S.; Sena, J.; Alfurhood, B.; Pant, B. The Smart Handwritten Digits Recognition Using Machine Learning Algorithm. In Proceedings of the 2023 3rd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE), Greater Noida, India, 12–13 May 2023; pp. 340–344. [CrossRef]
21. Tang, J.; Guo, H.; Wu, J.; Yin, F.; Huang, L. Offline Handwritten Mathematical Expression Recognition with Graph Encoder and Transformer Decoder. *Pattern Recognit.* **2024**, *148*, 110155. [CrossRef]
22. Zhang, H.; Song, H.; Li, S.; Zhou, M.; Song, D. A Survey of Controllable Text Generation Using Transformer-based Pre-trained Language Models. *ACM Comput. Surv.* **2023**, *56*, 1–37. [CrossRef]
23. Zhang, J.; Du, J.; Yang, Y.; Song, Y.Z.; Wei, S.; Dai, L. A Tree-Structured Decoder for Image-to-Markup Generation. In Proceedings of the 37th International Conference on Machine Learning. PMLR, Virtual, 13–18 July 2020; Proceedings of Machine Learning Research; pp. 11076–11085.
24. Zhu, J.; Zhao, W.; Li, Y.; Hu, X.; Gao, L. TAMER: Tree-Aware Transformer for Handwritten Mathematical Expression Recognition. *Proc. AAAI Conf. Artif. Intell.* **2025**, *39*, 10950–10958. [CrossRef]
25. Wu, J.W.; Yin, F.; Zhang, Y.M.; Zhang, X.Y.; Liu, C.L. Graph-to-Graph: Towards Accurate and Interpretable Online Handwritten Mathematical Expression Recognition. *Proc. AAAI Conf. Artif. Intell.* **2021**, *35*, 2925–2933. [CrossRef]
26. Li, B.; Yuan, Y.; Liang, D.; Liu, X.; Ji, Z.; Bai, J.; Liu, W.; Bai, X. When Counting Meets HMER: Counting-Aware Network for Handwritten Mathematical Expression Recognition. In Proceedings of the European Conference on Computer Vision, Tel Aviv, Israel, 23–27 October 2022; Springer Nature: Cham, Switzerland, 2022; pp. 197–214. [CrossRef]
27. Yuan, Y.; Liu, X.; Dikubab, W.; Liu, H.; Ji, Z.; Wu, Z.; Bai, X. Syntax-Aware Network for Handwritten Mathematical Expression Recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022, pp. 4553–4562. [CrossRef]

28. Zhao, W.; Gao, L.; Yan, Z.; Peng, S.; Du, L.; Zhang, Z. Handwritten Mathematical Expression Recognition with Bidirectionally Trained Transformer. In Proceedings of the International Conference on Document Analysis and Recognition, Lausanne, Switzerland, 5–10 September 2021; Springer International Publishing: Cham, Switzerland, 2021; pp. 570–584. [CrossRef]
29. Bian, X.; Qin, B.; Xin, X.; Li, J.; Su, X.; Wang, Y. Handwritten Mathematical Expression Recognition via Attention Aggregation Based Bi-Directional Mutual Learning. *Proc. AAAI Conf. Artif. Intell.* **2022**, *36*, 113–121. [CrossRef]