algorithms

MDPI

*Article*

# A Novel Short-Memory Sequence-Based Model for Variable-Length Reading Recognition of Multi-Type Digital Instruments in Industrial Scenarios

**Shenghan Wei [1], Xiang Li [2], Yong Yao [1,2,*] and Suixian Yang [1,*]**

[1] School of Mechanical Engineering, Sichuan University, Chengdu 610041, China
[2] National institute of Measurement and Testing Technology, Chengdu 610021, China
* Correspondence: yaoyong@stu.scu.edu.cn (Y.Y.); yangsuixian@scu.edu.cn (S.Y.)

**Abstract:** As a practical application of Optical Character Recognition (OCR) for the digital situation, the digital instrument recognition is significant to achieve automatic information management in real-industrial scenarios. However, different from the normal digital recognition task such as license plate recognition, CAPTCHA recognition and handwritten digit recognition, the recognition task of multi-type digital instruments faces greater challenges due to the reading strings are variable-length with different fonts, different spacing and aspect ratios. In order to overcome this shortcoming, we propose a novel short-memory sequence-based model for variable-length reading recognition. First, we involve shortcut connection strategy into traditional convolutional structure to form a feature extractor for capturing effective features from characters with different fonts of multi-type digital instruments images. Then, we apply an RNN-based sequence module, which strengthens short-distance dependencies while reducing the long-distance trending memory of the reading string, to greatly improve the robustness and generalization of the model for invisible data. Finally, a novel short-memory sequence-based model consisting of a feature extractor, an RNN-based sequence module and the CTC, is proposed for variable-length reading recognition of multi-type digital instruments. Experimental results show that this method is effective on variable-length instrument reading recognition task, especially for invisible data, which proves that our method has outstanding generalization and robustness in real-industrial applications.

**Keywords:** multi-type digital instruments; variable-length reading recognition; distance dependencies information; shortcut connections; sequence-based; CTC

**Citation:** Wei, S.; Li, X.; Yao, Y.; Yang, S. A Novel Short-Memory Sequence-Based Model for Variable-Length Reading Recognition of Multi-Type Digital Instruments in Industrial Scenarios. *Algorithms* **2023**, *16*, 192. https://doi.org/10.3390/a16040192

Academic Editor: Friedhelm Schwenker

Received: 10 March 2023
Revised: 28 March 2023
Accepted: 29 March 2023
Published: 31 March 2023



## 1. Introduction

Digital instruments are widely used in the fields of industrial control, device display and testing for their high precision, strong readability and easy maintenance [1]. However, most of digital instruments are lack of communication interface due to information security, cost control and application environment, which unable to realize automatic acquisition and transmission of digital number for subsequent information processing and data management. In this way, the recognition requirement of digital instruments is only relying on artificial reading by worker, which is inconvenience for information management in real-industrial scenarios. Specifically, there are many uncertain factors in artificial reading, and the result is easy to be interfered by subjective factors, resulting in low efficiency and low accuracy [2–4]. In addition, some high-risk environment with high temperature, high pressure, high radiation and strong corrosion is not suitable for manual work [5,6]. Therefore, it is of great practical value to explore an efficient, accurate and robust method to recognize digital instrument reading automatically.

As with the regular process of Optical Character Recognition (OCR), the core steps of digital instrument reading recognition are mainly divided into two stages, including

reading detection and reading recognition. As the terminal step of the automatic recognition system of digital instruments, the quality of the recognition method directly determines the final recognition rate. Therefore, the recognition method has also received a lot of attention from scholars and experts. However, most of the current related studies only focus on a single instrument type, such as the digital multimeter [7,8], the ammeter and voltmeter [5], the methane detecting instrument [9], the electric energy meter [10], and so on. Additionally, some studies are only applicable to specific application scenarios with superior imaging conditions, such as requiring the relative position of the camera and the instrument to be fixed [6,11]. All of the above methods work well under certain circumstances, but they are difficult to cope with the recognition needs of real-industrial scenarios in which multiple instrument types are mixed. In the real world, the reading strings of the captured images are usually non-standardized with variable-length, due to different instrument locations, the difficulty in building a stable image acquisition system, and different types of instrument displays. To be specific, the changes in string length are reflected in the following aspects: (1) different number of characters in character string; (2) different character fonts and styles; (3) different character spacing and aspect ratios. Thus, more and more researchers focus on the problem of recognizing variable-length reading of multi-type digital instruments.

Currently, the approaches of recognizing variable-length reading strings are mainly based on segmentation-based recognition methods by first splitting the string into individual characters, and then recognizing each character separately. Wei Zhou et al. [12] split the characters separately by obtaining the equations of the lines on both sides of the character. However, this method was easily affected by position changes of characters, not suit for the application with muti-type instruments. Studies [13,14] used vertical projection method to cut out the numbers according to the height-width ratio and other information of the image before recognition, which had the advantages of high recognition rate, high real-time performance and good reliability. Nevertheless, the image pre-processing algorithm is used for key operations, which may lead to segmentation failure if its effect is not good. Montazzolli S et al. [15] directly located to each character in the string, thus avoiding obvious segmentation steps, but it required character-level annotation. All of the above methods have explored solutions of recognizing variable-length reading strings to different degrees, and have achieved good results in their respective application scenarios. However, they are all based on a segmented thinking frame, whose generalization and robustness are extremely limited in real-industrial scenarios. Firstly, it is difficult to find a universal segmentation method due to different digital instrument display styles. Secondly, the error introduced by pre-processing and segmentation steps may directly reduce the final recognition accuracy. Finally, this recognition method is still limited to character-level classification, which requires a large amount of character-level annotation data and a huge workload.

In recent years, the continuous development of deep learning has brought new vitality to digital instrument reading recognition. The alternative approaches are based on the segmentation-free idea in which the string is recognized without having to be previously split into isolated characters. Hochuli et al. [16] proposed a segmentation-free recognition algorithm for handwritten digit strings of unknown length based on dynamic selection of classifiers, including a length classifier and other three string classifiers (10 [0...9], 100 [00...99], and 1000 [000...999]). However, for the digital instrument reading recognition task, the existence of special characters and the increase in the number of characters will greatly increase the number of the classifiers (much larger than 1110 classes), leading to an increased training burden. In 2018, Cai Mengqian [17] proposed a digital instrument reading recognition algorithm based on a full convolution network, which directly obtained the result of variable-length string through graph to graph prediction, but additional post-processing algorithms were needed to extract variable-length reading strings from the prediction matrix, resulting in complex network structure, which is not condu-

cive to practical application. In order to solve this problem more economically and effectively, we expect the prediction results of deep neural networks to combine both class information and location information of characters, and finally achieve pixel-level prediction of images, thus eliminating complex segmentation strategies and reducing computational burden at the same time. In the industrial scenario with multi-type digital instruments, a string in a digital instrument image can be regarded as an ordered string of characters with variable styles and special small characters, therefore, we consider trying the sequence-based segmentation-free method.

Considering that the recurrent neural network (RNN) [18] are very good at handling sequence data and the connectionist temporal classification (CTC) [19] can solve the problem of one-dimensional sequence alignment without character-level annotations, the RNN-CTC framework is particularly suitable for the task of sequence labels with explicit order information, such as license plate recognition [20,21], CAPTCHA recognition [22,23] and handwritten digit recognition [24]. However, although CTC is an effective decoding strategy, it also has some limitations in the reading recognition task of multi-type digital instruments: (1) It needs high requirements for feature extraction. Especially for characters with variable styles or some small characters such as "-" ".", CTC will output "null" for sequences that cannot be recognized, however, its alignment rules of de-duplication and de-blank will easily cause wrong decoding. (2) The short-distance dependencies are important. Since CTC decoding allows repeated characters and blank labels, a label, especially a wide character, can be jointly predicted by more than one sequences. Some neighboring sequences have complementary features and there is some correlation within the feature sequences that need to be captured effectively. For example, if only half of the number "8" is seen, "8" may be misclassified as "E" or "3", as shown in Figure 1. In addition, the characters at each position of the string are relatively random, so only the short-distance dependencies not the long-distance dependencies among the feature sequences are expected. Therefore, in this paper, we propose a novel short-memory sequence-based model for variable-length reading recognition of multi-type digital instruments. In our methods, shortcut connection strategy was involved into traditional convolutional structure to form a feature extractor for capturing effective features from raw digital instrument images. Then, an RNN-based sequence module, which captured the short-distance dependencies among feature sequences by giving up capturing long-distance trend changes and increasing the sensitivity to local changes, was employed to greatly improve the robustness and generalization of the model for invisible data. Finally, a novel short-memory sequence-based model consisting of a feature extractor, an RNN-based sequence module and the CTC, is proposed for variable-length reading recognition of multi-type digital instruments.

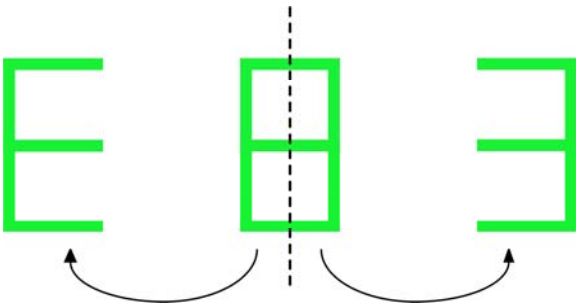

**Figure 1.** An example of misrecognition with incomplete information.

In summary, the contributions of this paper are as follows.

1. Considering the variable font styles and variable aspect ratios of different kinds of digital instrument readings, as well as the difficulty in small characters recognition, we involve shortcut connection strategy into traditional convolutional structure to

form a skip connection structure for extracting more complex and advanced character feature maps, taking advantage of the powerful high-dimensional function fitting ability of residual networks and deep network optimization capability;
2. In order to reduce the connection between the characters of the string, while emphasizing the local connections, we applied an RNN-based sequence module, which reduced the long-distance trending memory of the string and strengthen the short-distance dependencies among adjacent sequences, obviously improving the recognition accuracy and generalization of the model;
3. Based on the above two innovations, we propose a novel short-memory sequence-based model, consisting of a feature extractor, an RNN-based sequence module and the CTC, which achieves promising results in the task of multi-type digital instrument reading recognition and performs robustly for invisible data.

## 2. Model Building

In this section, we will briefly introduce the mathematical model of our multi-type digital instrument reading recognition method, which can be roughly divided into three parts. In the feature extractor, a CNN-based structure with shortcut connections is for extracting effective features from raw images. In the sequence modeling module, a sequence module based on RNN is used to obtain the short-distance dependencies information. In the decoding module, the CTC algorithm is used to calculate the loss and derive the final recognition results.

### *2.1. The Feature Extractor*

CNN have led to a series of breakthroughs for image classification [25–27], due to the characteristic of spatial subsampling, shared weights, and local receptive fields. Plain convolutional layers generally have pooling layers or multi-step convolution operations, which will subsample the feature map and may cause the loss of small object features, resulting in bad recognition performance of small characters such as "-" "."and so on. Motivated by the super performance of the residual network (ResNet) [28], in this paper we involve shortcut connection strategy into traditional convolutional structure to form a feature extractor to extract efficient features. The core idea of ResNet is to introduce a shortcut connection in the network that can skip one or more layers and add the input of the upper layer directly to the output of the bottom layer. Figure 2 shows the shortcut connection structure in a residual block. Consider a reading string image $I$ that is passed through a plain convolutional network with convolutional layers, each layer $\zeta$ conducts a non-linear transformation $F_\zeta(\bullet)$. For traditional plain network, the output of $\zeta^{th}$convolutional layer is:

$$I_\zeta = F_\zeta(I_{\zeta\text{-}1}), \tag{1}$$

where $I_{\zeta-1}$ is the output of the $(\zeta-1)^{th}$ layer. For residual convolutional network, the corresponding $I_\zeta$ is:

$$I_\zeta = F_\zeta[F_{\zeta-1}(I_{\zeta\text{-}2})] + I_{\zeta\text{-}2}, \tag{2}$$

if there is a residual connection between$(\zeta-2)^{th}$ and $\zeta^{th}$layers, where $I_{\zeta-2}$ is the output of the$(\zeta-2)^{th}$layer. Figure 2 shows the shortcut connection structure in a residual block.

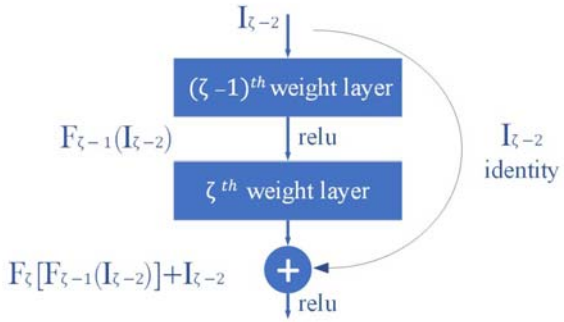


**Figure 2.** The residual block structure [28].

This kind of connection can increase the information transmission between adjacent network layers, which can better fit high-dimensional classification functions and greatly enable the backbone network to learn more complex and accurate feature representation. In practice, the non-linear function $F_\zeta(\bullet)$is always composed of batch normalization [29], ReLU [30] and a convolution layer. In addition, the number of the shortcut connection layers can be flexibly chosen according to the specific task. In general, the feature maps $X$ produced by the feature extractor in our study can be represented as:

$$X = F_{\theta_1}(I), \tag{3}$$

where $\theta_1$ is the parameters of the feature extractor and $F_{\theta_1}(\bullet)$ denotes the designed extractor, which is a CNN-based structure with shortcut connections to obtain high-level feature representation of multi-type characters. Before feeding into the sequence modeling module, the feature map $X$ is converted into a feature sequence. Assuming that the length of the sequence output from the feature extractor is $T$, it can be denoted as:

$$X = \{x_1, \ldots, x_j, \ldots, x_T\}, \tag{4}$$

where $x_j (j = 1,2,\ldots,T)$ is a vector, in the same order to the corresponding local rectangle region of the raw image from left to right.

### *2.2. The Sequence Modeling Module*

RNN is an important branch of deep neural network, which is very good at processing sequence-like data. The most significant feature of RNN is that the current output of a node in the hidden layer is not only related to the current input, but also to the output of the hidden layer at the previous moment, and its node structure is schematically shown in Figure 3.

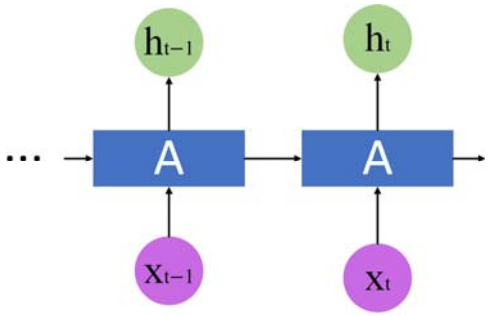


**Figure 3.** The structure of RNN nodes .

Neural network $A$ is regarded as a function $g$, the weight matrix is $\omega$, and the RNN recursive function can be expressed as:

$$h_t = g(h_{t-1}, x_t, w), \tag{5}$$

In this way, it can ensure that each moment of the RNN contains information from the previous moment, and contextual links between feature sequences can be established. The current output of CNN is only related to the current input, leading the dependencies of samples on time or space sequences cannot be modeled, so it is necessary to introduce RNN for the following reasons: Firstly, RNN has a strong ability to capture contextual information in sequences. Secondly, in the digital instrument reading recognition task, the characters at each position of the reading string are relatively random while some neighboring sequences have complementary features, so it is important to establish the short-distance dependencies not the long-distance among the preceding and following sequences. Thirdly, RNN can traverse sequences of arbitrary lengths, and effectively deal with the recognition of variable-length readings.

Traditional RNN unit (BasicRNN) [18] and Long-Short Term Memory (LSTM) [31] unit are two representative types of RNN units, whose structures are shown in Figure 4. From the figure, we can see that the repeated cell structure of BasicRNN is simple, and there is only one neural network layer. Due to the chain law, the preliminary information of BasicRNN is easier to be lost, and the information it keeps is usually short-distance. The repeated cell structure of LSTM is complex with not only one neural network layer but also three gating units, and these gating structures interact with each other to control cell selects the information of interest, as shown by the horizontal line running through the top of the cell in the figure. The cell states in LSTM only need to perform linear summation operations to pass the hidden layer, so it is good at conveying long-term information but lacks the ability to focus on local regions. On account of the previous analysis, we expect the characters of the string to be relatively independent from each other, but to preserve the short-distance dependencies among the feature sequences, in the digital instrument reading recognition task, so we select the BasicRNN network for sequence modeling.

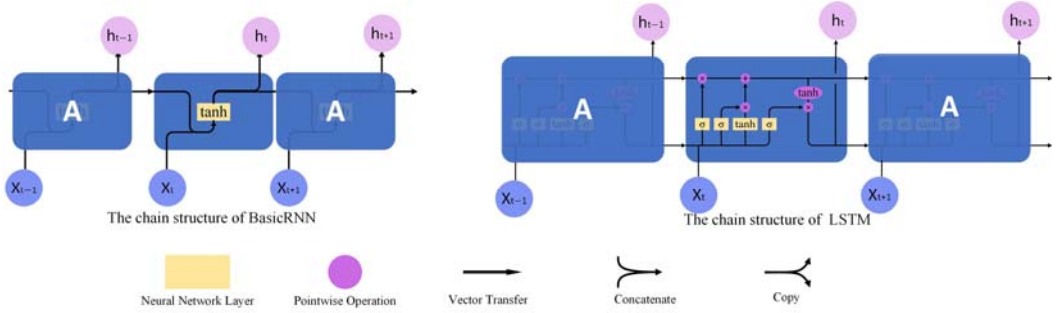


**Figure 4.** The chain structures of two typical recurrent neural networks. The arrow represents passing a complete vector, and the yellow box represents the neural network layer; Purple circles represent point-by-point operations; A merge line means that two inputs are in series, and a forked line means that content is sent to different locations.

To be specific, given the feature sequence $X = \{x_1, \ldots, x_j, \ldots, x_T\}$from the extractor, in order to get the contextual information, the recurrent neural network $G(\bullet)$ generates the contextual representati $C = \{c_1, \ldots, c_j, \ldots, c_T\}$ through the recurrent connection $c_t = G(c_{t-1}, x_t)$. Finally, for the generated sequence, we obtain the posteriori probability distribution over the label space for per-frame in the sequence via a linear layer:

$$y_j = soft\,max(c_j), j = 1,2,...,T, \tag{6}$$

Then the whole label distribution $Y = \{y_1, \ldots, y_j, \ldots, y_T\}$ are made based on $C = \{c_1, \ldots, c_j, \ldots, c_T\}$. The entire process of the sequence modeling module can be represented as:

$$Y = R_{\theta_2}(X), \tag{7}$$

where $\theta_2$is the parameters of the feature extractor $R_{\theta_2}(\bullet)$ denotes the whole sequence module based on Bi-BasicRNN, which capture the short-distance sequential dependencies with both directions.

### *2.3. The Decoding Module: CTC*

Connectionist temporal classification (CTC) [19] is a kind of output layer. It has two main functions. One is to calculate the loss, the other is to decode the output of RNN. CTC introduces an identifier "blank" ($\Phi$) to indicate that there is no character predicted at that position, thus achieving the purpose of occupancy and alignment. The "blank" can only repeat or go to the next non-empty sequence. With the introduction of "blank", the model does not need to struggle with which label should be aligned to a certain time step, and all paths conforming to the CTC topology are a legal alignment path, and the purpose of network training is to find the optimal path.

Using $N$ to denote all possible output characters in the recognition task, the total character set of the reading string recognition task can be expressed as:

$$N' = N \cup \{\Phi\}, \tag{8}$$

Taking the digital instrument reading string recognition in this paper as an example, $N$ contains 10 numbers and "E", "-", ".", a total of 13 characters, while there are 14 characters in $N'$.

For the predicted sequence $Y = \{y_1, \ldots, y_j, \ldots, y_T\}$ output from the sequence modeling module, each $y_j$ of predictions corresponds to a character in $N'$, then there is a total of $|N'|^T$ prediction sequences, so the probability of each possible path (sequence) $\boldsymbol{\pi}$ is:

$$p(\boldsymbol{\pi}|Y) = \prod_{t=1}^{T} p(\pi_t, t|Y), \tag{9}$$

where $\pi_t$ is the label in path $\boldsymbol{\pi}$ at time step $t$.

After removing the repeated labels and blanks of the sequence $\boldsymbol{\pi}$, the final predicted string is obtained, and this mapping process can be expressed as a many-to-one function $\Omega(\boldsymbol{\pi})$. For example, the sequences $\boldsymbol{\pi}_a = "00\Phi EE\Phi 555\Phi\Phi 8"$ and $\boldsymbol{\pi}_b = "00\Phi\Phi EE\Phi\Phi 55\Phi\Phi 88"$ all satisfy $"\Omega(\boldsymbol{\pi}_a) = \Omega(\boldsymbol{\pi}_b) = 0E58"$. Assuming that the corresponding label is denoted as $Z$ over $N$, a conditional probability is defined as the sum of probabilities of all $\boldsymbol{\pi}$, which are mapped by $\Omega(\boldsymbol{\pi})$ onto Z:

$$p(Z|Y) = \sum_{\boldsymbol{\pi}:\Omega(\boldsymbol{\pi})=Z} p(\boldsymbol{\pi}|Y), \tag{10}$$

Generally, for a given sequence Z, there are a huge number of possible paths. Direct computation of the summation in Equation (10) will cost too much time, thus, we choose the forward–backward algorithm described in [32] to calculate it in a reasonable time.

In the inference phase, the sequence $Z^*$ defined in Equation (10) with the highest probability is chosen as the prediction. However, it is extremely difficult to find the optimal solution precisely. In this paper, we choose the Greedy Algorithm adopted in [33] to approximate the result:

$$Z^* \approx \Omega(\arg\max_{\pi} p(\pi \mid Y)), \tag{11}$$

At each time step $t$, we just choose the most probable label, and obtain the final prediction output after $\Omega(\boldsymbol{\pi})$ mapping.

## 3. The Proposed Network

### *3.1. Architecture and Parameters of the Novel Short-Memory Sequence-Based Model*

The detailed information of the architecture is shown in Figure 5 Table. The architecture of the novel short-memory sequence-based model consists of three parts: (1) the feature extractor; (2) the sequence modeling module; (3) the decoding module.

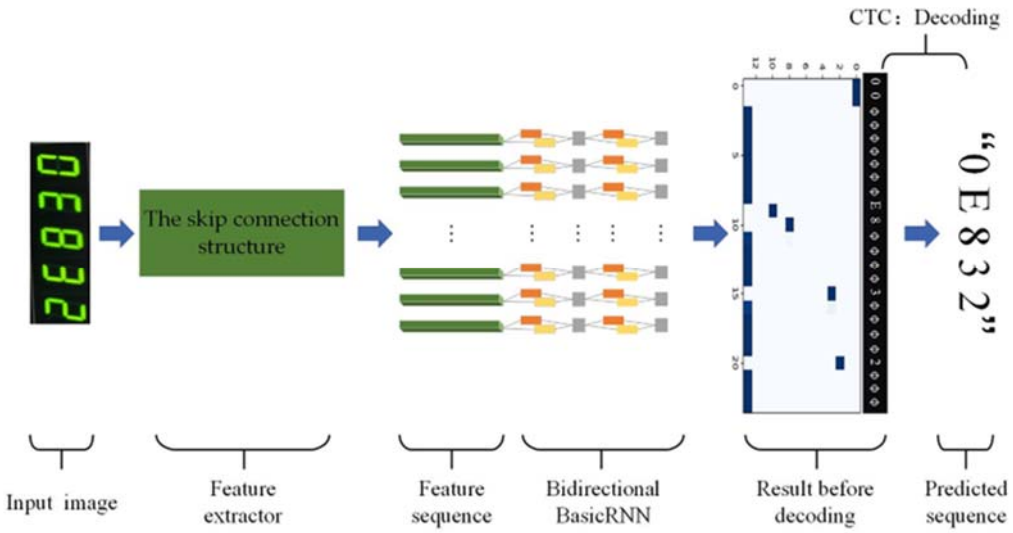


**Figure 5.** The network architecture of the novel short-memory sequence-based model.

In order to extract the best possible representation of feature sequences from images and retain useful information at different levels, we design a skip connection structure based on CNN, consisting a plain CNN network and three shortcut connections, as shown in Figure 6. To be specific, a convolutional layer and a pooling layer are first designed to obtain low-level features and down-sample the feature dimension. The intermediate convolution part has six convolutional layers with three shortcut connections, which are implemented to reuse the activations of the previous layers. In addition, we perform downsampling directly by strided convolutional layers instead of the max pooling layers, which increases the number of network-trainable parameters and improves the expressiveness of the model. The main purpose of this part is to achieve high-level information extraction through the stacked convolutional layers. The network ends with two convolutional layers with appropriate parameters to achieve cross-channel information interaction and strengthen the connection among adjacent frames in the feature map.

After the convolutional layers, in order to capture local dependencies from both the front and back directions and obtain a higher and more abstract feature level, we employ a deep bi-directional BasicRNN (Bi-BasicRNN) network with 2 layers for sequence modeling, whose output sequences are the probability distributions over 14 classes in $N'$. Finally, CTC is adopted to train the whole network and translate the per-frame prediction by the recurrent layers into a label sequence.

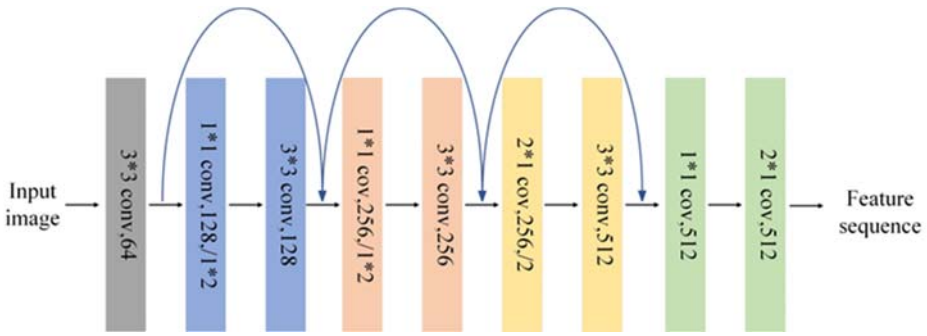

**Figure 6.** The architecture of the feature extractor.

*3.2. Network Training*

If the training set is denoted as $\boldsymbol{S} = \{\boldsymbol{I}, \boldsymbol{Z}\}$ and $(I, Z)$ is one of the samples in $\boldsymbol{S}$, where ( $I \in \boldsymbol{I}$ ) is the training image and ( $Z \in \boldsymbol{Z}$ ) is the ground truth of label sequence. Then the loss function $l(\boldsymbol{S})$ can be calculated as the negative log probability of ground truth on all training examples in training set $\boldsymbol{S}$:

$$l(\boldsymbol{S}) = -\sum_{(I,Z)\in \boldsymbol{S}} log\ p(Z \mid Y), \tag{12}$$

$$= -\sum_{(I,Z)\in \boldsymbol{S}} log\left(\sum_{\boldsymbol{\pi}:\Omega(\boldsymbol{\pi})=Z} p(\boldsymbol{\pi}|Y)\right), \tag{13}$$

The framework of the proposed method consists of three parts. The sequence modeling module can back-propagate the error differentials to the feature extractor and the CTC is a cost function independent of the neural network architecture. Therefore, we can train the whole network simultaneously by adding a CTC layer. To be specific, Equation (13) can calculate a cost value directly from an image $I \in \boldsymbol{I}$ and its ground truth label sequence $Z \in \boldsymbol{Z}$. Thus, the network can be end-to-end trained on the training set $\boldsymbol{S}$. The network is trained with stochastic gradient descent (SGD) with the back-propagation through time (BPTT) [34] algorithm and the network parameters $\theta$ including $\theta_1$ and $\theta_2$, which can be denoted as follows:

$$\theta \leftarrow \theta - \mu \frac{\partial l(\boldsymbol{S})}{\partial \theta}, \tag{14}$$

where $\mu$ is the learning rate.

Once the network has been trained, the best decoding path is applied to the output sequence predictions of the sequence modeling module. The decoder concatenates the most probable labels at each time step and obtains the final reading string after removing the duplicate characters and all the blanks.

## *4.* Experimental Analysis

*4.1. Datasets and Implementation Details*

4.1.1. Datasets

Considering that there are relatively few such studies at present, in order to carry out this work, 1723 images of different types of digital instruments were collected with CDC industrial camera in real working scenarios. Then, in order to improve the detection performance, 10,520 images were crawled through web crawling techniques in various websites as the initial data samples. On this basis, the standard rectangular enclosing box was used to crop the images of small reading area with accurate positioning and less noise interference by conventional detection algorithm. Then we manually label the cropped images to construct a basic dataset containing multi-type digital instrument reading images of variable length from 1–6 digits, and Figure 7 shows some raw images and some samples of the basic dataset.


**Figure 7.** Some raw images and some samples in the basic dataset.

In our experiments, we constructed two different datasets A and B depending on the base dataset, both of them containing pictures with all length of 1–6 digits. For A, the training set($A_1$) and testing set($A_2$) are constructed based on the principle of independent and identically distributed, where the testing data and training data are in equilibrium distribution on the kinds of strings, character fonts, character spacing and aspect ratios, according to the training test ratio of 10:1. To further verify the capability to combat memorial errors and generalization of our model, we construct dataset B. In B, 23% kinds of strings in testing set($B_2$) do not appear in the training set($B_1$), while the other 77% pictures in testing data are quite different from training data on character fonts or character spacing and aspect ratios. Similarly, the ratio of the training set to the testing set is still about 10:1. In addition, the training set $A_1$ and training set $B_1$ contain a considerable number of 0–9 and "E" "-" ".", totaling 13 characters. The distribution of the datasets is shown in Table 1.

**Table 1.** The datasets distribution used in our experiment.

| Dataset | Number of Training Samples | Number of Testing Samples | Notes |
|---|---|---|---|
| A | 4213 | 412 | Containing 1–6 digits images, the testing data and training data are in equilibrium distribution on the kinds of strings, character fonts, character spacing and aspect ratios. |
| B | 4150 | 421 | Containing 1–6 digits images, about 23% of the reading strings in the test set were not found in the training set, while 77% pictures in testing data are quite different from training data for character fonts or character spacing and aspect ratios. |

4.1.2. Implementation Details

The network configuration of the proposed method is summarized in Table 2. The architecture of the feature extractor is inspired from the VGG architecture [35] and the residual network. Some improvement is made in order to make it suitable for recognizing the variable-length reading of muti-type digital instruments. To reduce the computational burden, the first convolutional layer uses 64 filters with a (3,3) kernel size and stride of (2,2). The kernel size and stride for the pooling layer are set to (2,2) and (1,2) respectively, for two reasons. On the one hand, there will be overlap and coverage between the output

features in the horizontal direction, which can enhance the feature richness, on the other hand, the stride set to (1,2) can produce a rectangular perceptual field with narrow width, and it is easier to recognize small characters with narrow width, such as "." "1", etc. To ensure that the height dimension of the output feature sequence is 1 and that the number of feature sequences is appropriate (much larger than the number of characters in the reading string), while retaining more spatial information. The first two residual modules use convolutional layers with stride of (1,2) to down-sample only the height of the feature map, while keeping the width unchanged. Then, the last module performs downsampling in both the height and width directions by convolutional layer with stride of (2,2). It should be noted that the downsampling only happen in their first convolutional layer. In the same residual module, each convolutional layer has the same number of filters. The number of filters is doubled one by one for each of the three residual blocks. In addition, we use a convolutional layer with a (1,1) kernel size and stride of (1,2) to reduce the number of parameters while achieving cross-channel interaction and information integration. In the end, between two feature sequences, the last a convolutional layer with a (2,1) kernel size and stride of (1,1) is adopted to strengthen the adjacent connection in the feature map. Finally, the new 512-dimensional feature sequences with a height of 1 are generated as the input of the recurrent layer.

The network is implemented within the tensorflow1.14 deep learning framework under Python 3.7. Experiments are carried out on a workstation with Intel(R) i7 processor, RTX1070 Ti discrete graphics card, 32G RAM. The AdaDelta is chosen to update the network parameters, while the learning rate is set to 0.0001. ReLu is used as the activation function of each layer, and CTC_loss is used as the loss function to achieve end-to-end network training with a total of 200 epochs. In order to speed up the training process, each convolutional layer is followed by a BN layer. It is found that the value of batch_size has a great impact on the results. If batch_size is too small, training process is time-consuming and bad for convergence. If batch_size is too large, it will easily fall into the local minimum. Therefore, batch_size was finally set to 32, after several training comparisons.

**Table 2.** Parameters of the proposed model. An input instrument image of 100 × 32 will generate a feature sequence with the dimensions 1 × 24 × 512 after the feature extractor. The outputs from the sequence modeling part have the same dimensions as the inputs.

| Layer | Input Shape | Kerner Size | Filter | Stride | Output Shape |
|---|---|---|---|---|---|
| Conv1 | (batch,100,32,1) | 3 × 3 | 64 | 2 × 2 | (batch,49,15,64) |
| MaxPool1 | (batch,49,15,64) | 2 × 2 | - | 1 × 2 | (batch,48,7,64) |
| Conv2_x | (batch,48,7,64) | | 1 × 1, 128, 1 × 2<br>3 × 3, 128, 1 × 1 | | (batch,48,4,128) |
| Conv3_x | (batch,48,4,64) | | 1 × 1, 256, 1 × 2<br>3 × 3, 256, 1 × 1 | | (batch,48,4,128) |
| Conv4_x | (batch,48,2,128) | | 2 × 1, 512, 2 × 2<br>3 × 3, 512, 1 × 1 | | (batch,48,4,128) |
| Conv5 | (batch,24,1,512) | 1 × 1 | 512 | 1 × 1 | (batch,24,1,512) |
| Conv6 | (batch,24,1,512) | 2 × 1 | 512 | 1 × 1 | (batch,24,1,512) |
| Bi-BasicRNN | (batch,24,1,512) | | Hidden units:256 | | (batch,24,1,512) |
| Bi-BasicRNN | (batch,24,1,512) | | Hidden units:256 | | (batch,24,1,512) |
| CTC Layer | - | | - | | - |

### *4.2. Data Pre-Processing and Evaluation Metrics*

#### 4.2.1. Data Pre-Processing

The sizes of images vary in these datasets. For efficient training and the requirement of downsampling operations, the input dimension of models should be fixed. In this paper, we resize all images to a fixed height of 32 with remaining the image respect ratio. In addition, we gray-scale the obtained picture to facilitate the calculation.

#### 4.2.2. Evaluation Metrics

Inspired by the evaluation metrics of OCR, the proposed network model in this experiment can be also measured by hard metric and soft metric. Hard metric, also known as string-level recognition accuracy (*SRA*), is more rigorous, as it recognizes the predicted reading string as a correct when all the characters in the ground truth label are identified. The *SRA* is calculated as:

$$SRA = M_S / M, \tag{15}$$

where $M$ is the total number of testing images and $M_S$ is the number of correctly recognized images.

Soft metric, also known as character-level recognition accuracy (*CRA*), is defined based on the edit distance, which is the minimum number of edit operations required to convert from the predicted string to the ground truth string. There are three types of editing operations allowed: substituting one character with another, inserting a character or deleting a character. The smaller the edit distance, the more similar the two reading strings are. In other words, a smaller sum of the edit distance indicates a higher *CRA*, which is calculated as:

$$CRA = -\sum d_e, \tag{16}$$

where $d_e$ is the edit distance of each incorrectly recognized character.

In addition, according to the characteristics of incorrect test results, we divide the mis-recognition into two categories: memorial errors and the other recognition errors. The memorial error refers to a situation where a reading string in the test set does not appear in the training set, and the model recognizes it as a reading string similar to the one in the training set. Some examples of memory errors are shown in Figure 8. The other recognition errors mainly refer to over-recognition, under-recognition and mis-recognition of some characters in the reading string without obvious patterns other than memory errors.


**Figure 8.** Some examples of memorial errors.

### *4.3. Experimental Result and Analysis*

#### 4.3.1. The Effectiveness of the Feature Extractor in the Proposed Method

To verify that our feature extractor can extract effective feature representations than a plain CNN structure without shortcut connections, we first compared the performance of our feature extractor combined with CTC and a plain CNN structure with CTC, which both eliminated sequential modeling. In order to verify our hypothesis more comprehensively, we conducted experiments on both simple dataset A and complex dataset B and measured the experimental results both on *SRA* and *CRA*. The test results of the two models are shown in Table 3.

From the result, we saw that the *SRA* of the skip connection structure model reached 96.9% and 83.1% for dataset A and dataset B, respectively. This was an improvement of 1.1% and 9.1% compared with the plain CNN structure model for the two datasets, respectively. In order to validate our hypothesis, the skip connection structure model can achieve a better performance more comprehensively, we calculated the *CRA* as a supplement for the *SRA*. The *CRA* of skip connection structure model reached −13 and −95 for dataset A and dataset B, respectively. This was an improvement of 4 and 48 compared with the plain CNN structure model for the two datasets, respectively. As we discuss above, dataset A is simpler compared with B, resulting in easy recognition of other characters in A except for the small character recognition errors, while dataset B is more complex. In this way, the 9.1% accuracy improvement on B, a small portion of which is attributable to avoiding the loss of small object feature, is mostly attributable to the effectiveness of the residual net-work feature extraction advantage on the overall recognition performance improvement. As a result, there is a minor accuracy impact obtained from dataset A, while a greater accuracy impact obtained from dataset B. In conclusion, the improvement from the plain CNN structure model to the skip connection structure model proved that our novel feature extractor was capable of learning more effective features from characters with complex and diverse forms, which indicates our novel feature extractor is more suitable for the variable-length reading recognition task of multi-type digital instruments.

**Table 3.** Recognition accuracy of the skip connection structure + CTC and the plain CNN structure + CTC.

| Network Model | Train on $A_1$ Test on $A_2$ | | Train on $B_1$ Test on $B_2$ | |
|---|---|---|---|---|
| | ***SRA*** | ***CRA*** | ***SRA*** | ***CRA*** |
| **The skip connection structure + CTC** | **0.969** | **−13** | **0.831** | **−95** |
| The plain CNN structure + CTC | 0.958 | −17 | 0.740 | −143 |

#### 4.3.2. The Necessity of the Sequence Modeling

To verify the necessity of the sequence modeling, we applied a sequence module to the framework the skip connection structure + CTC for sequence modeling, constructing two models: the model without sequence module, the model with sequence module, then trained and tested on dataset A. The parameters of recurrent layers are the same as in Table 2 and the rest of the two models stay the same for a fair comparison. The performances of the models are summarized in Table 4.

As shown in Table 4, the test accuracy (*SRA*) of the model without sequence module reached 96.9%, which was 2.1% lower than that of the model with sequence module. It shows that the network model with a sequence modeling part is tested with high accuracy on the dataset with balanced distribution of the training and test sets, which indicates that

there is a certain dependency between the feature sequences of instrument readings and the role of sequence modeling of the recurrent layer is necessary.

To further understand how RNN help to improve the model performance, we visualized the feature sequence of three samples in the test set $A_2$ output directly from the CNN layers or from recurrent layers in Figure 9. As indicated in this figure, we observed that the feature sequences from recurrent layers are more distinctive than that directly from the CNN layers, which is more dispersive. This means that RNN can enhance the contextual information among feature sequences, which is very beneficial to model better feature sequences.

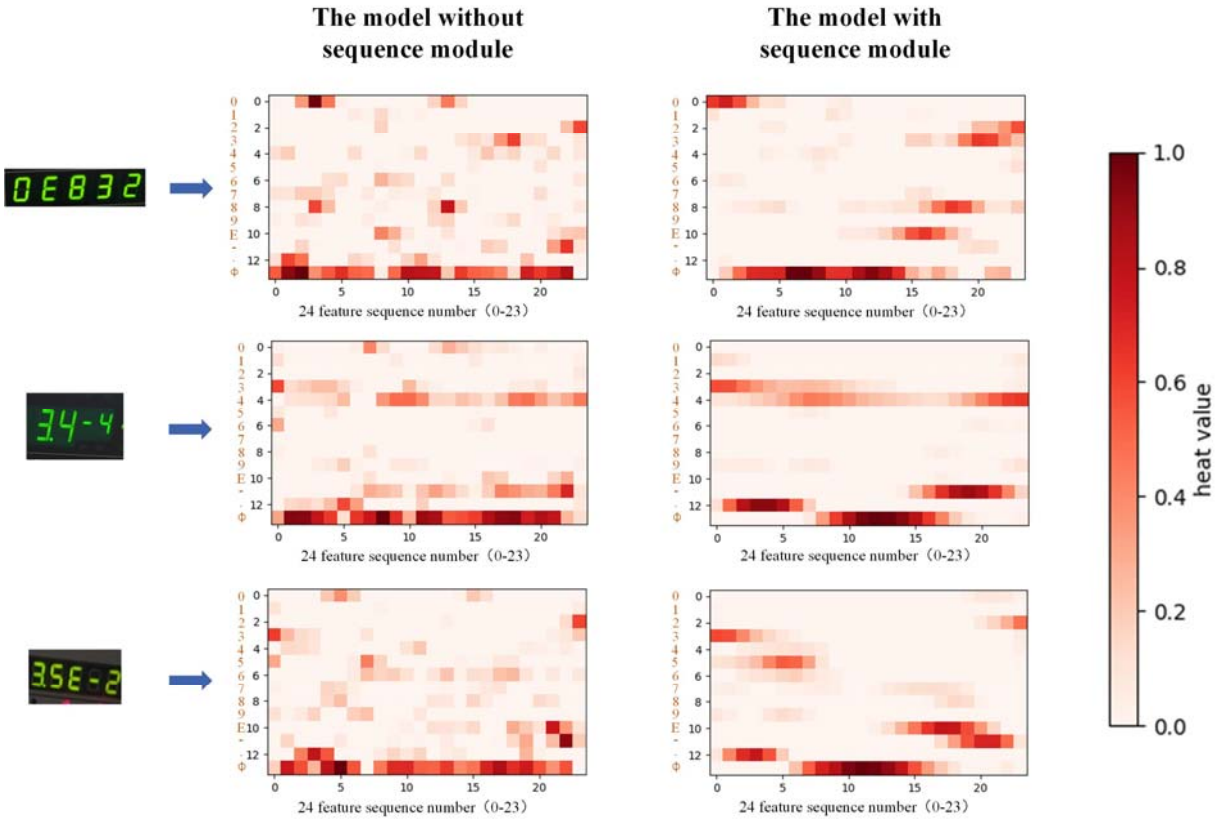


**Figure 9.** Heat map of the feature sequence output from sequence module or not. Left shows the heat map of the feature sequence directly output from the feature extractor without sequence module, and right shows the feature sequence output from the sequence module.

**Table 4.** *SRA* on dataset A when applying a sequence module or not.

| Network Model | Train on $A_1$ Test on $A_2$ |
|---|---|
| | ***SRA*** |
| The model without sequence module | 0.969 |
| **The model with sequence module** | **0.990** |

4.3.3. The Proposed Sequence Module Focuses on Short-Distance Dependencies to Improve Model Generalization

Except for BasicRNN and LSTM, there are many other RNN evolutionary variants to choose from the RNNs family. For example, Gated Recurrent Unit (GRU) [36] structure removes the cell states and uses hidden states to transfer information with only two gate structures simpler than LSTM, while Simple Recurrent Unit (SRU) [37] reduces front-to-back computational dependencies and enables parallel computation of RNNs. However,

they are all good at capturing long-distance dependencies. In order to verify the hypothesis that short-distance dependencies is better than long-distance dependencies in the multi-type digital instrument reading recognition task, where the characters at each position of the reading string are relatively independent from each other. We applied LSTM, GRU and SRU for sequence modeling to compare with our BasicRNN-adopted sequence modeling module. Then we trained and tested the BasicRNN-adopted model, the LSTM-adopted model, the GRU-adopted model and the SRU-adopted model on dataset B, which is designed for invisible data. The test results of the four models for the *SRA* and the number of memorial errors are shown in Table 5.

**Table 5.** The test results of the BasicRNN-adopted model, the LSTM-adopted model, the GRU-adopted model and the SRU-adopted model on B

| Network Model | Train on $B_1$ Test on $B_2$ | |
|---|---|---|
| | *SRA* | The Number of Memorial Errors |
| **The BasicRNN-adopted model** | **0.897** | **6** |
| The LSTM-adopted model | 0.815 | 44 |
| The GRU-adopted model | 0.839 | 36 |
| The SRU-adopted model | 0.808 | 35 |

In Table 5, it was shown that the performance of the BasicRNN-adopted model presented a significant improvement compared with the other three models. The BasicRNN-adopted model achieved an 89.7% accuracy. This was 8.2%, 5.8% and 8.9% higher than the accuracy of the LSTM-adopted model, the GRU-adopted model and the SRU-adopted model, respectively. Looking at it another way, there were 44, 36, and 35 memorial errors for the three models respectively, much more than that of the BasicRNN-adopted model. This fully proved that LSTM, GRU and SRU were good at establishing long-distance dependencies and prone to endorsement style memorial errors. The test results indicate that BasicRNN has good generalization and obtain high classification accuracy for invisible data and is more suitable than LSTM, GRU and SRU for digital instrument variable-length reading recognition where the correlation between characters is relatively weak and local area features need to be focused.

To clearly describe how sequence modeling with BasicRNN not LSTM and so on can help improve the performance of digital instrument reading recognition, Figure 10a,b show the label distribution after softmax layer of two representative examples in $B_2$, whose reading strings didn't appear in training set $B_1$. Since LSTM, GRU and SRU are good at capturing long-distance dependencies, they are very prone to memorial errors. For example, in Figure 10a,b, the test results of the two samples both could be predicted correctly trough BasicRNN-adopted model, while decoded to the similar reading strings in training set $B_1$ trough the other three models.

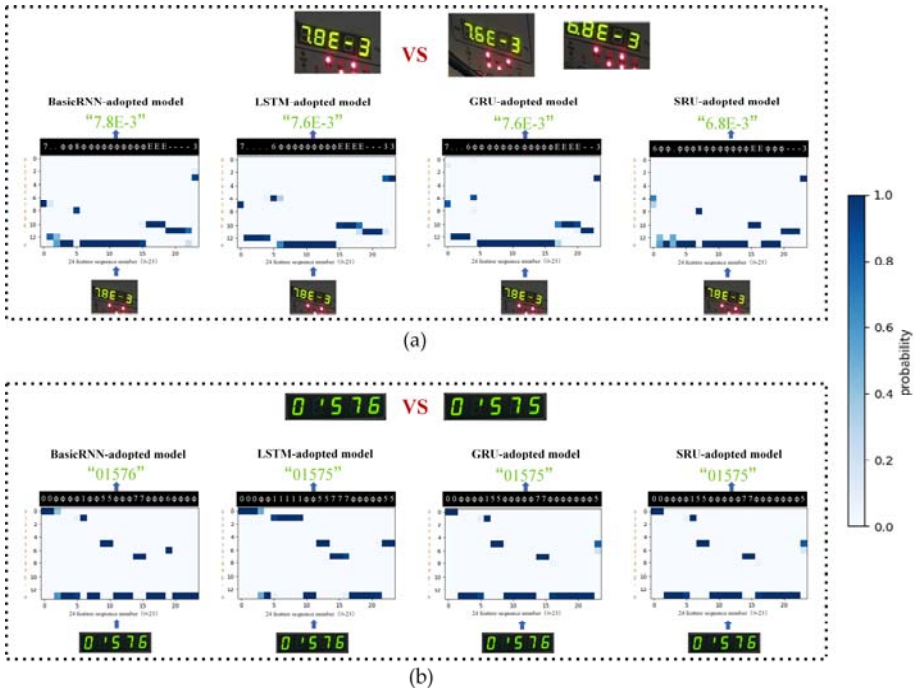


**Figure 10.** Correct prediction of BasicRNN-adopted model. (**a**) The label distribution of "7.8E-3" through the four models. "7.8E-3" didn't appear in $B_1$, while "7.6E-3"and "6.8E-3" appeared in $B_1$. (**b**) The label distribution of "01576" through the four models. "01576" didn't appear in $B_1$, while "01575" appeared in $B_1$.

## 5. Conclusions

In this paper, we proposed a novel short-memory sequence-based model for variable-length reading recognition of multi-type digital instruments. By using a skip connection structure, our proposed method was able to automatically extract more effective feature representations from complex and diverse characters, which outperformed the plain CNN structure. On the basis of the feature extractor, we applied an RNN-based sequence modeling module, which abandoned capturing long-distance trend memory while focusing on the correlation of local sequences. Experiments showed that our model had good generalization to invisible data and obtains high recognition accuracy in the variable-length reading recognition task of multi-type digital instruments. In addition, we analyze the feature extraction capability of our feature extractor by using *SRA* combined with *CRA*, and visualize the feature sequences output from the recurrent layer and the label distribution after softmax layer to gain insight into the reason why modeling local sequence dependencies can improve model performance.

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

**Funding:** this research received no external funding. The APC was funded privately.

**Data Availability Statement:** Not applicable.

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

## References


1. Shan, F.; Sun, H.; Tang, X.; Shi, W.; Wang, X.; Li, X.; Zhang, X.; Zhang, H. Investigation on Intelligent Recognition System of Instrument Based on Multi-step Convolution Neural Network. *Int. J. Comput. Commun. Eng.* **2020**, *9*, 185–192.
2. Wang, R.; Wang, R.; Kang, Y. Decimal recognition based on the LED machine vision. In Proceedings of the 2019 6th International Conference on Systems and Informatics (ICSAI), Shanghai, China, 2019.
3. Sun, H.; Shan, F.; Tang, X.; Shi, W.; Wang, X.; Li, X.; Cheng, Y.; Zhang, H. Intelligent Digital Recognition System Based on Vernier Caliper. *IJCCE* **2021**, *10*, 1–8.
4. Zhang, H.; Duan, H.; Zhang, S. A fast recognition method for display value of digital instrument. *Comput. Eng. Appl.* **2005**, *41*, 223–226.
5. Wang, X.; Wang, J.; Wang, H. Research on Intelligent and Digital Recognition System and Character Recognition of Electrical Instruments. In Proceedings of the 2018 IEEE International Conference on Mechatronics and Automation (ICMA), Changchun, China, 5–8 August 2018.
6. Mo, W.; Pei, L.; Huang, Q.; Liao, W. Digital Multimeter Reading Recognition for Automation Verification. In Proceedings of the AIAM2020: 2nd International Conference on Artificial Intelligence and Advanced Manufacture, Manchester, UK, 15–17 October 2020.
7. Zhang, J.; Zuo, L.; Gao, J.; Zhao, S. Digital instruments recognition based on PCA-BP neural network. In Proceedings of the Technology, Networking, Electronic & Automation Control Conference (ITNEC), Chengdu, China, 15–17 December 2017.
8. Wu, C.; Wu, Q.; Yuan, C.; Li, P.; Zhang, Y.; Xiao, Y. Multimeter digital recognition based on feature coding detection. In Proceedings of the 2017 10th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI), Shanghai, China, 14–16 October 2017.
9. Liu, Z.; Luo, Z.; Gong, P.; Guo, M. The research of character recognition algorithm for the automatic verification of digital instrument. In Proceedings of the 2013 2nd International Conference on Measurement, Information and Control (ICMIC), Harbin, China, 16–18 August 2013.
10. Zhang, Z.; Chen, G.; Li, J.; Ma, Y.; Ju, N. The research on digit recognition algorithm for automatic meter reading system. In Proceedings of the Intelligent Control & Automation, Jinan, China, 7–9 July 2010.
11. Ma, B.; Meng, X.; Ma, X.; Li, W.; Hao, L.; Jiang, D. Digital Recognition Based on Image Device Meters. In Proceedings of the 2010 Second WRI Global Congress on Intelligent Systems, Wuhan, China, 16–17 December 2010.
12. Zhou, W.; Peng, J.; Han, Y. Deep Learning-based Intelligent Reading Recognition Method of the Digital Multimeter. In Proceedings of the 2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Melbourne, Australia, 17–20 October 2021; pp. 3272–3277.
13. Chen, H.; Xin, W.; Bo, X. Study on digital display instrument recognition for substation based on pulse coupled neural network. In Proceedings of the 2016 IEEE International Conference on Information and Automation (ICIA), Melbourne, Australia, 17–20 October 2021.
14. Peng, G.; Du, B.; Li, Z.; He, D. Machine vision-based, digital display instrument positioning and recognition. *Int. J. Ind. Eng.* **2022**, *29*, 230–243.
15. Montazzolli, S.; Jung, C.R. Real-Time Brazilian License Plate Detection and Recognition Using Deep Convolutional Neural Networks. In Proceedings of the 2017 30th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI), Niteroi, Brazil, 17–20 October 2017.
16. Hochuli, A.G.; Oliveira, L.S.; de Souza Britto, A.; Sabourin, R. Segmentation-Free Approaches for Handwritten Numeral String Recognition. In Proceedings of the 2018 International Joint Conference on Neural Networks (IJCNN), Rio de Janeiro, Brazil, 8–13 July 2018.
17. Cai, M.G.; Zhang, L.; Wang, Y. Digital instrument character recognition method based on full convolution network. In *Modern Computer (Professional Edition)*; 2018; pp. 38–43.
18. Bourbakis, N.G.; Koutsougeras, C.; Jameel, A. Handwriting recognition using a reduced character method and neural nets. In Proceedings of the Nonlinear Image Processing VI, San Jose, CA, USA, 28 March 1995; Volume 2424, pp. 592–601.
19. Graves, A.; SFernández Gomez, F. Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. In Proceedings of the 23rd International Conference on Machine Learning, Pittsburgh, PA, USA, 25–29 June 2006; Association for Computing Machinery: New York, NY, USA, 2006.
20. Sun, H.; Fu, M.; Abdussalam, A.; Huang, Z.; Sun, S.; Wang, W. License Plate Detection and Recognition Based on the YOLO Detector and CRNN-12. In Proceedings of the 4th International Conference on Signal and Information Processing, Networking and Computers (ICSINC), Qingdao, China, 23–25 May 2018; Volume 494, pp. 66–74. https://doi.org/10.1007/978-981-13-1733-0_9.
21. Li, X.; Wen, Z.; Hua, Q. Vehicle License Plate Recognition Using Shufflenetv2 Dilated Convolution for Intelligent Transportation Applications in Urban Internet of Things. *Wirel. Commun. Mob. Comput.* **2022**, *2022*, 3627246.
22. Wei, D.U.; Zhuo, W.N. Uncategorized Text Verification Code Recognition Based on CTC Model. Computer and Modernization, **2018**, *9*, 48.
23. Ma, J. Neural CAPTCHA networks. *Appl. Soft Comput.* **2020**, *97 Pt 1*, 106769.

24. Zhan, H.; Wang, Q.; Lu, Y. *Handwritten Digit String Recognition by Combination of Residual Network and RNN-CTC*; Springer: Cham, Switzerland, 2017.
25. Messina, R.; Louradour, J. Segmentation-free handwritten Chinese text recognition with LSTM-RNN. In Proceedings of the International Conference on Document Analysis & Recognition (ICDAR), Tunis, Tunisia, 23–26 August 2015.
26. Krizhevsky, A.; Sutskever, I.; Hinton, G. ImageNet Classification with Deep Convolutional Neural Networks. *Adv. Neural Inf. Process. Syst.* **2017**, *60*, 84–90.
27. Sermanet, P.; Eigen, D.; Zhang, X.; Mathieu, M.; Fergus, R.; LeCun, Y. Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv* **2013**, arXiv:1312.6229.
28. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. *arXiv* **2016**. https://doi.org/10.48550/arXiv.1512.03385.
29. Ioffe, S.; Szegedy, C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *arXiv* **2015**, arXiv:1502.03167.
30. Glorot, X.; Bordes, A.; Bengio, Y. Deep Sparse Rectifier Neural Networks. *J. Mach. Learn. Res.* **2011**, *15*, 315–323.
31. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780.
32. Alsharif, O.; Pineau, J. End-to-End Text Recognition with Hybrid HMM Maxout Models. *arXiv* **2013**, arXiv:1310.1811.
33. Shi, B.; Bai, X.; Yao, C. An End-to-End Trainable Neural Network for Image-Based Sequence Recognition and Its Application to Scene Text Recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 2298–2304.
34. Werbos, P.J. Backpropagation through time: What it does and how to do it. *Proc. IEEE* **1990**, *78*, 1550–1560.
35. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
36. Cho, K.; Merrienboer, B.V.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. *arXiv* **2014**, arXiv:1406.1078.
37. Tao, L.; Yu, Z.; Yoav, A. Training RNNs as Fast as CNNs. *arXiv* **2017**, arXiv:1709.02755.