Journal of Computers Vol. 36 No. 5, October 2025, pp. 245-259
https://doi.org/10.63367/19911599202510360501 7

Study on Enhancing Object Recognition Accuracy in Contextual Application Scenarios

Xiaohua Liu, Xiaojun Wen, and Qiuxia Chen*

School of Artificial Intelligence, Shenzhen Polytechnic University, Shenzhen, Guangdong, 518055, China
{lxh, wxjun, chenqiuxia}@szpu.edu.cn

Received 18 September 2024; Revised 16 August 2025; Accepted 7 October 2025

Abstract. This paper focuses on enhancing object recognition accuracy within contextual application scenarios, specifically using handwritten digit recognition as a case study. Through comprehensive analysis and demonstration, we outline a technical roadmap and implementation plan that leverage advanced methodologies. Our approach adopts a 'segmentation-first, recognition-second' strategy, beginning with traditional contour-based methods for incremental segmentation to effectively separate digit regions. For complex scenarios involving abnormal image blocks—such as digit adhesion or local overlap—we implement a robust deep learning model based on Mask R-CNN for secondary segmentation. This two-tier segmentation process ensures that even challenging cases are accurately processed. Following segmentation, we employ a compact deep learning model trained on the modified MNIST-J dataset to perform handwritten digit recognition. This dual-phase methodology not only achieves high accuracy in object recognition for real-world images but also demonstrates significant applicability across various fields. Notably, our approach can be extended to applications such as agricultural product production and growth monitoring, illustrating its versatility and potential for widespread impact. By addressing the intricacies of object recognition in diverse contexts, our work contributes valuable insights and methodologies that can enhance future research and practical applications in related domains.

Keywords: handwritten digit segmentation, handwritten digit recognition, Convolutional Neural Network (CNN)

1 Introduction

Object recognition accuracy should not only be evaluated on test sets that are consistent with the training set distribution but should also be applicable to real-world scenarios. In the context of real-world applications, the environment is often far more intricate and complex than what is typically assumed in idealized scenarios. This complexity can be manifested in multiple aspects. For instance, the background of the target objects may be highly cluttered, with numerous interfering elements present. There could be a great deal of noise, whether it is visual noise in the case of image-based recognition tasks, or acoustic noise for audio-related identification. Additionally, environmental factors such as varying lighting conditions (ranging from extremely bright sunlight to dimly lit areas), weather conditions (including rain, fog, or snow which can distort or obscure the target), and the presence of occlusions (where parts of the target are hidden by other objects) all contribute to this complexity. Such complex environmental conditions inevitably lead to more in-depth and elaborate reasoning processes. In order to accurately perform a recognition task, there are often numerous pre-processing steps required. For example, in the case of object recognition within a complex background, segmentation becomes a crucial pre-requisite step. Segmentation aims to isolate the target object from its complex surrounding background, which is no easy feat considering the numerous and diverse elements within the environment. This process of segmentation itself may require complex algorithms and significant computational resources. Moreover, the presence of the above-mentioned environmental factors further complicates the segmentation process, as they can make the boundaries of the object less distinct and more difficult to define. All these factors combined significantly increase the challenges associated with recognition tasks. Without the development of advanced methods and technologies, the actual accuracy will be substantially lower than the accuracy achieved on the test set after model training.

In the field of object detection and recognition, convolutional neural networks (CNNs) are widely used at present. Theoretically, CNNs should exhibit translational invariance [1], meaning that small shifts of the fore-

* Corresponding Author

ground object in the input image should not affect the network's recognition results. However, recent studies have revealed that CNNs do not possess complete translational invariance [2, 3], a shift of just one pixel in the input image has a 30% probability of altering the network's output. CNNs encode a surprising degree of absolute position information [4]. The primary reason for the lack of translational invariance in CNNs is the presence of downsampling operations within the network [3]. In the context of CNNs, downsampling refers to the process of reducing the spatial dimensions of the input volume. This reduction is typically achieved through operations such as max pooling or strided convolutions. Downsampling is a critical component in CNNs as it helps in learning hierarchical features at multiple levels of abstraction while reducing computational cost and controlling overfitting. By downsampling the input feature maps, CNNs can progressively increase their receptive fields and learn more abstract representations of the input data. This process is an essential aspect of CNN architecture that enables effective feature learning and hierarchical pattern recognition. These downsampling methods disregard the sampling theorem, leading to the possibility that small translations or shifts in the input can cause significant changes in the output. One of the key aspects of our work is to effectively address the lack of translational invariance in CNNs. Our research focuses on developing novel strategies and algorithms to enhance the performance of CNNs in this regard. By doing so, we aim to improve the accuracy of object recognition and contribute to more robust and reliable performance in various real-world applications where input data may have significant variations in object positions. This part of our work is an important component of our overall endeavor to advance the state-of-the-art in CNN-based technologies and applications.

The objective of this paper is to improve object recognition accuracy within real-world application contexts, which is a crucial and comprehensive goal. In the field of object recognition, numerous challenges exist throughout the entire process, but our focus is not simply on resolving an individual issue within this complex process. Instead, we are aiming at a more holistic and ultimate objective.

Specifically, we center our attention on enhancing the accuracy of handwritten digit recognition in practical application scenarios. Handwritten digit recognition is a fundamental yet challenging task in the realm of object recognition. In real-world applications, factors such as varying writing styles, different pen pressures, and diverse background noises can significantly impact the recognition accuracy. Our research is dedicated to overcoming these combined challenges and ultimately improving the overall accuracy of handwritten digit recognition, which in turn will contribute to the broader objective of enhancing object recognition accuracy in real-world applications.

2 Related Work

Methods for improving object recognition accuracy should balance both specificity and generality. In the paper, it is essential to discuss the generalizability of the methods while also analyzing their application-specific aspects. Combining these two perspectives constitutes the correct research approach.

2.1 General Methods for Object Recognition

To improve object recognition accuracy, the first approach that comes to mind is to restore the translational invariance of CNNs. There are two main approaches to restoring the translational invariance of CNNs. The first approach involves data augmentation [2]. By using additional data augmentation techniques during network training, including scaling and translation of images, the invariance of the network can be increased. The second approach involves modifying the original CNN architecture [2, 3, 5]. The authors of reference [3] suggest introducing anti-aliasing layers into the network, such as the BlurPool operator. This operator smooths the feature maps before downsampling to reduce high-frequency components, thereby enhancing spatial smoothness. In this way, the network can achieve greater consistency, reducing the impact of small positional changes in the input on the output, and thus improving translational invariance. In reference [5], CNNs are modified by introducing the Adaptive Polyphase Sampling (APS) scheme. APS is a simple subsampling method designed to address the issue of displacement invariance caused by downsampling (stride) layers in CNNs. In reference [6], to improve classifier accuracy under translational conditions, the authors employed robust optimization methods and trained against worst-case adversaries. This approach successfully enhanced the model's robustness to translations. Compared to random transformation-based data augmentation methods, the use of robust optimization and worst-case adversarial training resulted in significant improvements in the model's spatial robustness.

To enhance the robustness of the trained model, it is necessary to generate a set of adversarial samples based on the original dataset. Techniques for generating adversarial samples include spatial transformation techniques [7] and pixel manipulation techniques [8-11]. In reference [12], the authors introduced up to 15 types of image corruptions, such as shot noise, impulse noise, glass blur, defocus blur, and motion blur, to the MNIST dataset [12]. In fact, generating adversarial samples partially simulates the complexity of real-world application scenarios, where the degradation of ideal images may result from the superposition of multiple factors.

To enhance the accuracy of object recognition within application contexts, scholars and researchers have engaged in various productive explorations. Karim and Muhammad proposed a technique that integrates YOLO with Deep Sport, focusing on improving the reliability and accuracy of object recognition while also addressing data processing speed [13]. Reference [14] reviews different recognition and localization methods and their applications in agricultural robotics vision, aiming to improve the efficiency and precision of fruit picking. Reference [15] investigates the use of RGB images and object shape features to address object pose issues, combining 3D surface normal features with image plane highlights to detect fruits. This approach segments point cloud data into categories such as apples, branches, and leaves, thereby providing a more comprehensive perceptual capability and demonstrating notable improvements in recognition rates, increasing the average precision from 0.52 to 0.55. Reference [16] introduces the Rep-ViG-Apple algorithm, an enhanced version of the YOLOv8 model. The Rep-ViG-Apple algorithm features a sophisticated structure designed to enhance apple detection performance under challenging conditions. This approach improves the extraction of apple features, leading to a 1.5% increase in model accuracy. Reference [17] notes that to achieve small object detection in complex environments, they first created a real dataset based on complex application contexts, followed by image augmentation. On the network side, they employed YOLOV5 and made adaptive improvements in three areas—attention mechanisms, head networks, and backbone networks—resulting in performance surpassing the baseline model. Reference [18] designed a YOLOv5-based algorithm incorporating frame difference fusion for ship target detection, utilizing an enhanced DeepSORT tracking algorithm as the ship tracking module, and combined both to achieve automatic detection and tracking of inland vessels.

2.2 Segmentation and Recognition of Handwritten Digits

In the context of handwritten digit segmentation and recognition, extensive research has been conducted on standard datasets such as CENPARMI, CEDAR, and MNIST, achieving very high accuracy rates. However, in practical scenarios, the accuracy significantly drops, with performance falling below 90% in cases of touching digits [19], and dropping to below 50% in the presence of overlapping digits. This decline is due to the variability in digit shape and size in real-world settings, as well as the inconsistency in writing standards [20]. Reference [20] proposes the use of vertical and horizontal projection methods for segmentation, convex hull algorithms for feature extraction, and support vector machines for digit recognition.

Traditional segmentation-based methods have played a significant role in the field of digit recognition. These methods primarily involve various techniques to separate digits from the background in an image, facilitating subsequent recognition processes. Specific methods include: thresholding, edge detection, contour extraction, projection methods [20], region-based approaches, dynamic programming, and over-segmentation. The over-segmentation method divides the image into multiple small regions, followed by a merging step to determine the final segmentation. While this approach has the potential to improve segmentation accuracy, it incurs higher computational costs. Overall, these traditional segmentation-based methods have demonstrated relatively high recognition rates in the situation of handling clear and well-structured digits. When the digits are clear in appearance and possess a well-defined structure, these methods can effectively perform recognition tasks, thus achieving a satisfactory level of recognition rates. Nevertheless, these traditional methods do encounter certain limitations. Specifically, when they are faced with digits that are overlapping, touching each other, or are in a blurred state, problems start to emerge. Overlapping digits mean that parts of two or more digits are superimposed on one another, while touching digits are adjacent to each other in such a way that it becomes difficult to distinguish their boundaries. Blurred digits lack clear outlines and details, often due to factors such as low-quality imaging or interference during the digitization process. When dealing with such types of digits, these traditional segmentation-based methods may not function as effectively as they do with clear and well-structured digits, and this can have a negative impact on the overall recognition performance.

Reference [19] introduces a segmentation-free approach, which inputs the entire digit string as a whole into the model, rather than first segmenting it into individual digits. By incorporating contextual information, such as the length of the digit string, the model can better comprehend the input data. However, digits may be irregularly

distributed within the image and may not necessarily form a contiguous string. Even within a single string, the widths of the individual digits may vary, and the spacing between different digits can differ, making the length of the digit string less informative. Consequently, the method proposed in Reference [19, 21, 22] is only applicable to specific scenarios.

3 Our Strategies and Methods

The research carried out by the above-mentioned experts and scholars has furnished us with highly valuable insights. In the current era, object detection and recognition are experiencing a period of vigorous development. With the continuous progress in this field, various models have emerged, each having its own unique features and capabilities. In such a context, the selection of an appropriate model becomes crucial. Different models may be more suitable for different types of data, application scenarios, or performance requirements. Moreover, adaptively integrating the selected model with other relevant technologies can further enhance its effectiveness. This integration can leverage the strengths of different technologies, such as combining deep learning algorithms with traditional image processing techniques. By doing so, it will be conducive to providing effective solutions for the real-world problems that are often complex and diverse in nature. These real-world problems may involve object recognition in various environmental conditions, handling of different object scales, or dealing with occluded objects, etc.

3.1 Adopting a Segmentation-first Approach Followed by Recognition

For effective segmentation-free digit string recognition, it typically requires hundreds of thousands to millions of samples to achieve optimal performance. Given that segmentation-free methods generally demand such a large volume of training data, we will not pursue this direction further at this time. For handwritten digit recognition, we will continue to follow the approach of segmenting digits individually before recognition.

We adopt the method of segmentation first and then recognition. The overall recognition rate is the product of the recognition rates of each stage. Specifically, if we define the recognition rate of the segmentation stage (the ratio of correct segmentation) as P_s, and the ratio of correct recognition as P_r, then the formula for the overall recognition rate P, which we denote as Formula (1), is the product of Ps and Pr. Here, both Ps and Pr are less than or equal to one. Obviously, in order to maximize P, Ps and Pr need to reach their maximum values. The larger Ps and Pr are, the better.

$$P = P_s * P_r \tag{1}$$

Traditional segmentation methods, for instance, those contour-based approaches, have demonstrated satisfactory performance when dealing with clear and well-structured digits. In such cases, the characteristics of the digits are relatively easy to identify and distinguish, which enables these traditional methods to effectively segment the digits based on the pre-defined rules or algorithms related to contour extraction and analysis. However, when faced with digits that are overlapping, touching each other, or are in a blurred state, the performance of traditional segmentation methods deteriorates significantly. In order to maintain effective segmentation performance under these challenging conditions, it becomes necessary to resort to deep learning methods for training classifiers.

We need to utilize a substantial amount of labeled handwritten digit images for training the classifier. These labeled images serve as the basis for the classifier to learn. Through the training process, the classifier is able to capture and learn the diverse features of digits, including not only the general shape and structure but also the subtle characteristics that are unique to different digits. Moreover, it can also learn about the appropriate segmentation points, which are crucial for accurately dividing the digits from the complex background or from other adjacent digits in case of overlapping or touching situations.

Once the classifier has been trained, it can be employed to segment new images. More specifically, we take advantage of deep-learning-based models, such as CNNs. As one of the core technologies in deep learning, CNN has continued to demonstrate strong development momentum and innovative potential in recent years [23]. These models are well-known for their ability to automatically learn image features. Through a series of convolutional layers, pooling layers, and fully-connected layers in the CNN architecture, the model can effectively extract hierarchical features from the input images. This automatic feature learning process is highly beneficial for digit

segmentation as it can adapt to various types of digit appearances and complex situations.

Although this method typically demands a large volume of data for training to ensure the generalization ability of the model and the accuracy of the learned features, it has the advantage of providing relatively good segmentation results. Additionally, compared to some complex and heuristic-based traditional methods, it is relatively straightforward to implement, which makes it a promising approach for digit segmentation in various practical applications.

3.2 Use Compact Segmentation to Ensure no Translation

In training recognition models, we continue to use the widely adopted MNIST dataset, a portion of which is illustrated in the Fig. 1 below. As seen in the figure, there is a spacing between the effective digit part of each small image and the image's bounding box, with the spacing varying within a certain range. This effectively allows the digit part to translate within a certain amplitude range. As previously discussed, convolutional neural networks (CNNs) do not possess complete translation invariance.

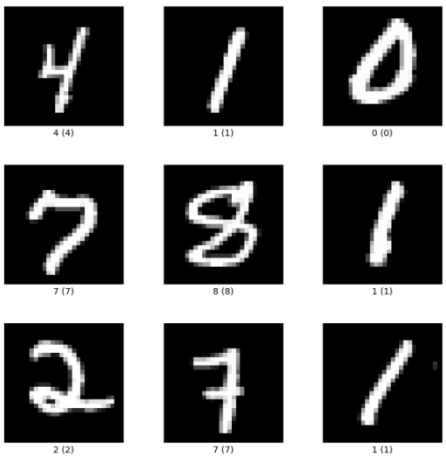

Fig. 1. Partial samples from the MNIST dataset

To mitigate the impact of translation uncertainty, we have boldly decided to perform a surgical operation on the dataset by retaining only the effective digit portions. This methodology comprises the following steps:

1. Load the raw MNIST dataset, which consists of grayscale images, each with dimensions of 28 pixels by 28 pixels. The method of manual annotation is adopted to remove the obvious noise points in the data set. This step is necessary because, if not carried out, the recognition rate may decrease by more than 0.2% according to our method of processing.

2. In the MNIST dataset, we binarize each image. Let I(x,y) denote the grayscale value at the coordinate (x,y) of an image I in the dataset. We set a threshold of 0. According to formula (2), we can calculate the transformation results, where I_b(x,y) represents the binarized image. That is, for each pixel in the image, if its grayscale value I(x,y) is greater than 0, it is converted to 255, and if its grayscale value is less than or equal to 0, it remains unchanged. As a result, we obtain binary images.

$$\mathrm{I_b}(x,y)=\begin{cases}255, & \mathrm{I}(x,y)>0\\ \mathrm{I}(x,y), & \mathrm{I}(x,y)\le 0\end{cases} \tag{2}$$

3. In the context of a binary image, it is essential to accurately identify the bounding rectangle of the foreground. The foreground in a binary image consists of pixels with a particular value (usually 255, assuming a binary convention where 0 represents the background). To determine the bounding rectangle, we need to precisely find the top, bottom, left, and right boundaries. Let's assume that the binary image is represented as a

two-dimensional matrix I with dimensions m by n, where m is the number of rows and n is the number of columns. We can define a function f(x,y) to represent the value of the pixel at location (x,y) in the image, such that $f(x,y) \in \{0, 255\}$. The left boundary x_{min} can be found by minimizing the x-coordinate over all foreground pixels. Mathematically, we can express it as formula (3). The right boundary x_{max} is found by maximizing the x-coordinate over all foreground pixels, formulated as formula (4). For the top boundary y_{min}, we minimize the y-coordinate among the foreground pixels as formula (5). And the bottom boundary y_{max} is obtained by maximizing the y-coordinate for the foreground pixels as formula (6). The bounding rectangle of the foreground is then defined by the coordinates (x_{min}, y_{min}), (x_{max}, y_{min}), (x_{max}, y_{max}) and (x_{min}, y_{max}).

$$x_{\min} = \min\{x \mid \exists y, f(x,y) = 255\} \tag{3}$$

$$x_{\max} = \max\{x \mid \exists y, f(x,y) = 255\} \tag{4}$$

$$y_{\min} = \min\{y \mid \exists x, f(x,y) = 255\} \tag{5}$$

$$y_{\max} = \max\{y \mid \exists x, f(x,y) = 255\} \tag{6}$$

4. Crop the grayscale image based on the obtained foreground bounding rectangle to create a compact grayscale image, ensuring that the margins on the top, bottom, left, and right are all zero. The reason why the margin is set to 0 here is as follows. In practical situations, when we segment targets from complex backgrounds, in most cases, the margin is 0. Based on this practical situation, we also adjust the data set to a state where the margin is equal to 0. Consequently, after implementing the segmentation according to the actual situation, we only need to perform a scaling operation, and there is no need for translation. In this way, the influence caused by translation invariance can be avoided.

5. Scale and center the compact grayscale image to obtain the final preprocessed grayscale image. When the margins of top, bottom, left and right are all zero, the object we are training is naturally centered.

The pseudo-codes of step 4 and step 5 are presented as follows. These codes can be easily debugged and passed in the Python-OpenCV environment.

```
  def cutEdge(X_train):
      X_trainp = []
      for i in range(len(X_train)):
          img = X_train[i]
          thresh = cv2.threshold(img, 0, 255, cv2.THRESH_BINARY)[1]
          cnts = cv2.findContours(thresh, cv2.RETR_EXTERNAL, cv2.CHAIN_AP-
PROX_SIMPLE)[0]
          cnts = sorted(cnts, key=cv2.contourArea, reverse=True)  # Sort
by area, descending
          cnt = cnts[0]
          x, y, w, h = cv2.boundingRect(cnt)
          timg = img[y:y + h, x:x + w]
          timg = cv2.resize(timg, (28, 28))
          X_trainp.append(timg)
      X_trainp = np.array(X_trainp)
      print(X_trainp.shape)
      return X_trainp
```

Therefore, we obtained a variant of the MNIST dataset, provisionally named the MNIST-J dataset, where "J" stands for "compact." The primary rationale for this modification is that the digit image patches extracted during segmentation possess a compact characteristic.

Based on past experience, classifiers trained on the MNIST dataset typically achieve accuracy rates exceeding 99.5% when tested on data within the MNIST dataset. Currently, using the MNIST-J dataset and employing the same neural network architecture for training, we have observed a slight decrease in terms of accuracy, which struggles to reach the 99.3% threshold. However, it is important to note that this situation is not the primary concern, as our research goal is to enhance the final accuracy of digit segmentation and recognition in complex environments. We refer to the model trained during this phase as Model-1.

3.3 Adopting a Hybrid Segmentation Strategy

In real-world complex scenarios, segmenting handwritten digit blocks presents the most challenging task. While there are methods and strategies that do not involve segmentation, these approaches are based on several assumptions, the realization of which is inherently difficult. Therefore, we continue to employ a segmentation strategy. Our chosen approach is a hybrid strategy: for the overall image, we use a segmentation strategy based on traditional methods; for specific local regions, such as excessively wide or high areas, we employ a deep learning-based segmentation strategy.

The rationale for adopting a hybrid segmentation strategy is as follows: Traditional segmentation methods and strategies treat large and small features equally, performing well for both large and small features, but they are not optimal in highly complex real-world scenarios. Thus, we use traditional segmentation methods for the overall image. For local features, we use deep learning-based segmentation strategies because models trained with deep learning often exhibit lower recall rates for smaller features. However, for local regions, we can apply an enlargement strategy to increase the size of small features, thereby improving recall rates without significantly compromising performance. In this scenario, recall rates for deep learning models can be maintained at a very high level.

In our adopted hybrid strategy, the deep learning-based segmentation approach is applied solely to local features. Consequently, the focus is primarily on the accuracy of the deep learning model, while speed is relatively secondary. After a comprehensive comparison of Mask R-CNN [24], YOLO V5 and subsequent versions [25, 26], RetinaNet [27], and EfficientNet [28], we ultimately chose Mask R-CNN for training the deep learning segmentation model. The main reason for this choice is that Mask R-CNN, as a two-stage object detection model, although somewhat slower in detection speed compared to single-stage object detection models (such as YOLO, SSD, etc.), excels in recall rate and detection accuracy. Particularly for objects with complex shapes and unclear boundaries, Mask R-CNN is capable of achieving superior detection and segmentation, thereby attaining a higher recall rate. We refer to the deep learning segmentation model trained at this stage as Model-2. The following Fig. 2 illustrates our overall technical roadmap.

RoIAlign is a key component in Mask R-CNN for accurate instance segmentation [24]. Given a region-of-interest (RoI) proposal with coordinates on the feature map, RoIAlign performs bilinear interpolation to extract a fixed-size feature map for each RoI. However, there is still room for improvement in its performance. In this study, we propose to optimize it through the combination of two techniques.

1. Cubic Interpolation: It is used to estimate the values within a given interval. Given a set of discrete data points (xi, yi) for i = 0, 1, 2, ..., n. The cubic interpolation polynomial I(x) in the interval [xi, xi+1] is of the form (7):

$$\mathrm{I}(x) = a_i(x - x_i)^3 + b_i(x - x_i)^2 + c_i(x - x_i) + d_i \tag{7}$$

where the coefficients ai, bi, ci, di are determined by the values of the function y = f(x) and its derivatives at the endpoints xi and xi+1. For a one-dimensional case, assume we have four data points y0, y1, y2, y3 at positions x0, x1, x2, x3. The cubic interpolation formula for estimating the value at x ∈ [x1, x2] is:

$$y(x) = y_1 \frac{(x - x_2)(x - x_3)}{(x_1 - x_2)(x_1 - x_3)} + y_2 \frac{(x - x_2)(x - x_3)}{(x_2 - x_1)(x_2 - x_3)} + y_3 \frac{(x - x_2)(x - x_1)}{(x_3 - x_2)(x_3 - x_1)} \tag{8}$$

In the context of ROI Align, when dealing with the feature maps, cubic interpolation can be used to more accurately calculate the values of the features at non-integer positions within the ROI.

2. Adaptive Sampling: It aims to adjust the sampling points according to the characteristics of the ROI. Let S be the set of sampling points in the original ROI. The number of sampling points N = |S|. We define a function f(R) based on the geometric and semantic information of the ROI as formula (9).

$$\mathrm{f}(R) = \omega_1 \frac{s - s_{\min}}{s_{\max} - s_{\min}} + \omega_2 \frac{a - a_{\min}}{a_{\max} - a_{\min}} + \omega_3 p \tag{9}$$

where R is a ROI, s represents the size of the ROI (for example, the area, which can be obtained by calculating the number of pixels covered by the ROI), a represents the aspect ratio of the ROI (a = widthROI/heightROI), and p represents the class probability of the ROI (this may be the probability value in a certain category obtained by a pre-classifier). smin and smax are the pre-defined minimum and maximum values of the ROI size respectively, amin and amax are the pre-defined minimum and maximum values of the aspect ratio respectively, and ω_1, ω_2 and ω_3 are weighting coefficients, which can be adjusted according to specific requirements. The new set of sampling points S' is obtained by formula (10).

$$\left|\mathrm{S}'\right| = g(S, f) = \mathrm{N} * (1 + k * f(x_0)) = |S| * (1 + k * f(x_0)) \tag{10}$$

where g is an adjustment function. For example, for larger ROIs or ROIs with higher uncertainty, we may increase the number of sampling points. k is a positive constant, and x0 is a specific element in the set S (if it has a special meaning). Here it is assumed that the function f can take a value at a specific element x0, and this value is related to the adjustment of the number of sampling points. For example, if f represents an uncertainty function regarding the elements of the set, the larger the value of f(x0), the higher the uncertainty, and thus more sampling points will be added (here the number of sampling points is related to the value of g).

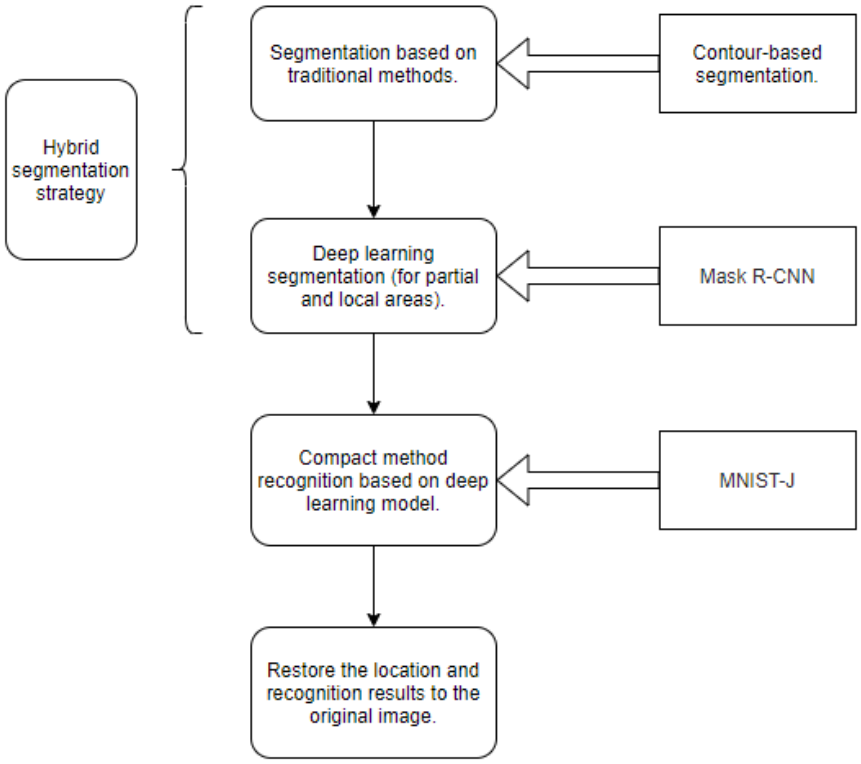

Fig. 2. Overall technical roadmap

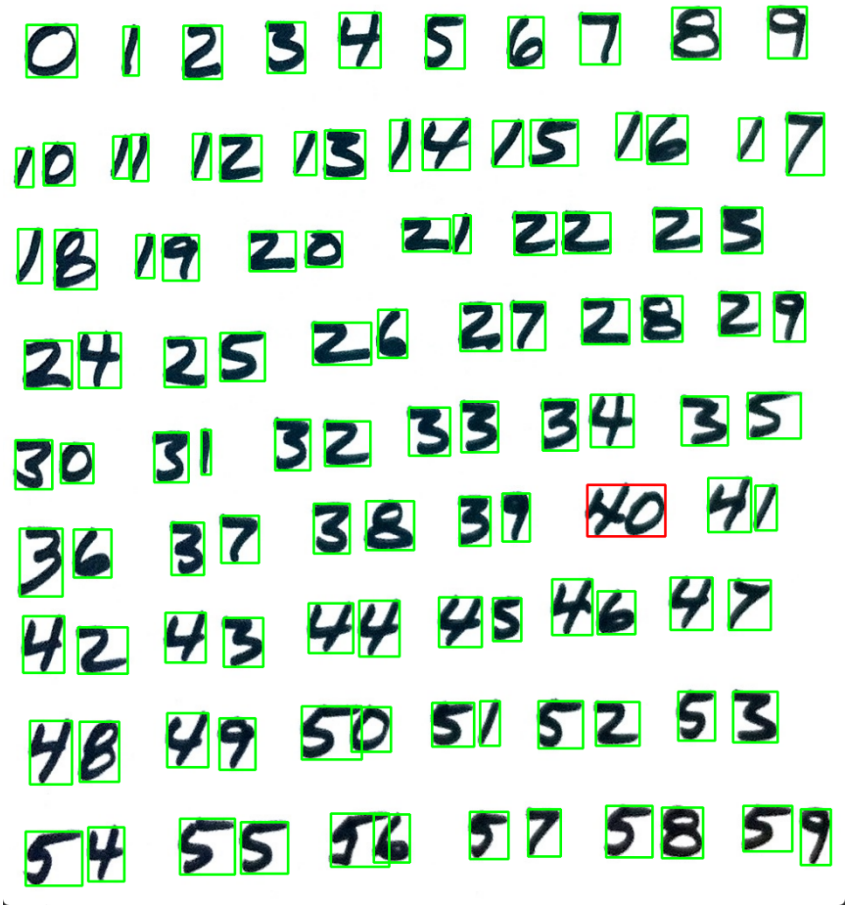

Fig. 3. The digits 40 are mutually adhered, not segmented when using traditional methods

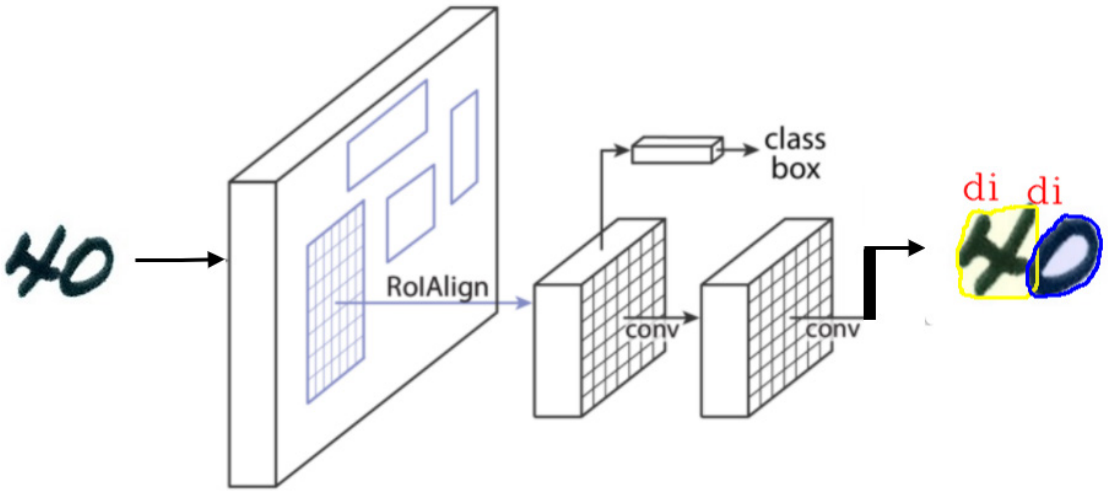

Fig. 4. Local anomalous image patches are re-segmented using Model-2

3. Combination of Cubic Interpolation and Adaptive Sampling in ROI Align: When performing ROI Align, first, we use the adaptive sampling method to determine the appropriate sampling points within the ROI. Then, for each sampling point, if it is at a non-integer position on the feature map, we use cubic interpolation to cal-

culate the feature value. Let I be the input feature map, and R be the ROI. After adaptive sampling, we get a set of sampling points S' = $\{p_i\}$ within R. For each p_i, if its coordinates on the feature map are not integers, we use cubic interpolation on I to get the feature value at p_i. The output of the optimized ROI Align is a set of feature values corresponding to the sampling points, which can be further used for subsequent operations such as classification, bounding box regression, and mask generation in Mask R-CNN.

Traditional segmentation methods, while capable of achieving satisfactory results, fail to effectively segment certain adhered digits, as illustrated in Fig. 3. For such adhered or overlapping regions, secondary segmentation is required, wherein a deep learning model, namely Model-2, is employed to perform the segmentation. As shown in Fig. 4, the two adhered digits have been successfully separated.

4 Experiment

To train a deep learning model for handwritten digit object detection based on Mask R-CNN, constructing an appropriate dataset is crucial. This study adopts a dataset construction method based on MNIST to conserve effort. In the first part of the constructed dataset, each image contains 5 to 10 handwritten digits. The gaps between these digits are random, and the dataset is balanced, ensuring that different types of digits are approximately equally distributed among these images. A total of 4,250 such images were created. Although this dataset could be designed to be larger, due to annotation limitations, it was restricted to 4,250 images. However, this size is generally sufficient for our purposes. The purpose of this paper is merely to verify the feasibility of the research approach. The problem we aim to address is the handwritten digit adhesion issue, which is most likely to occur horizontally. To enhance the robustness of deep-learning models in handling adhered handwritten digits, we need to design more samples. In these samples, digits appear in pairs. These paired digits may be mildly adhered or separated from each other. We plan to create 2,000 such sample images. As shown in Fig. 5, these are partial examples of the constructed dataset. The reason we only design pairs of digits is that consecutive digit adhesions can be regarded as combinations of pairwise adhesions. Now, the issue at hand is the problem of annotation. There are three annotation methods. The first is rectangular annotation, the second is polygon annotation, and the third is a more refined and special polygon annotation, namely mask annotation. After repeated deliberation, we finally chose a hybrid annotation method. That is, for separated digits, rectangular annotation is used. For two mutually adhered digits, polygon annotation is employed to mark each digit clearly, as shown in Fig. 6. Under no circumstances will we use mask annotation, because mask annotation is extremely time-consuming and labor-intensive.

Before the actual training, it is necessary to unify the annotation format. Since a rectangle is a special type of polygon, we can use a program to convert rectangular box annotations into polygon annotations with four vertices. In this way, all our annotation formats will be polygon annotations. Consequently, the training is carried out based on polygon annotations. Through extensive training on sample segmentation of adhered digits, our deep-learning model can effectively segment the adhered digits. For the segmented parts, we convert them into rectangles first (Note that the pixels outside the inner contour but within the rectangle should be filled with background pixels uniformly, instead of cropping the pixels from the original image at these positions. This approach can improve the accuracy of subsequent recognition.), and then scale them to a standard square size. After that, these standardized samples are fed into the model for recognition, which enables excellent segmentation and recognition results.

In the MNIST dataset, the object size of the images is relatively small, measuring 28×28, which may lead to a relatively low recall rate when using Mask R-CNN for object detection. Therefore, during the construction of the aforementioned dataset, the images were enlarged to [84,112]×[84,112] (square images). This enlargement makes the details of each handwritten digit more pronounced, aiding the model in better capturing features. Bilinear interpolation was employed for the image enlargement, a method that preserves the edges and details of the digits to the greatest extent, as used in the method described by Kaiming He in his paper [24]. The enlarged images are more suitable for data augmentation operations during training, such as rotation, further scaling, and translation, thereby further enriching the dataset.

We developed the training program for the Mask R-CNN model based on the open-source code accessible online (https://github.com/matterport/Mask_RCNN). In the process of model development, we made certain improvements to the RoI Align operation. Our improvements were aimed at enhancing the precision of this alignment process. Furthermore, in order to boost the performance of the model, we incorporated anti-overfitting operations into the program. Overfitting is a common issue in machine learning models, where the model becomes

too specialized in the training data and fails to generalize well on new, unseen data. By adding anti-overfitting operations, such as regularization techniques (e.g., L1 or L2 regularization), dropout, or early stopping, we were able to mitigate this problem and improve the model's generalization ability, which ultimately led to enhanced performance on both the training and test datasets.

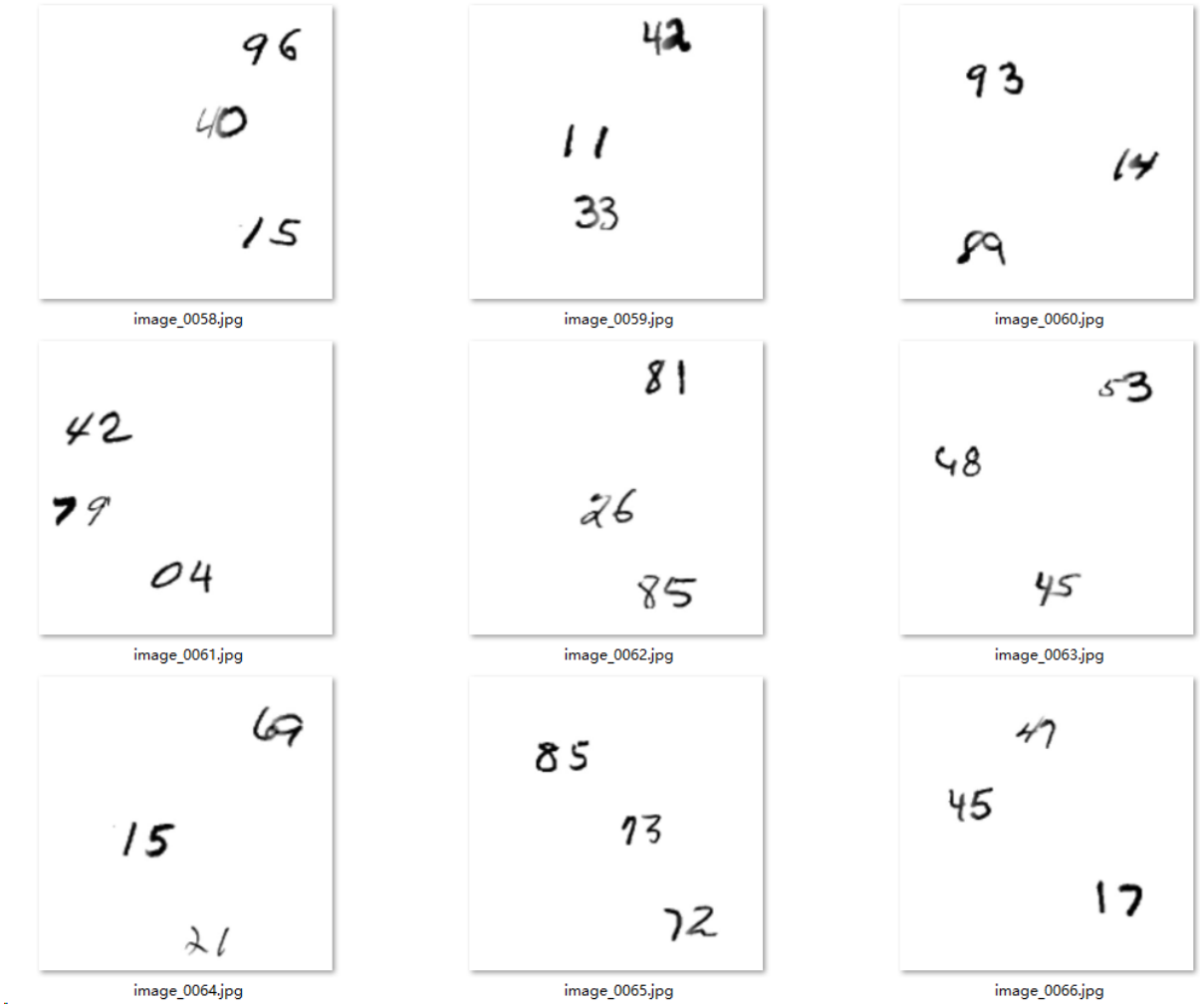

Fig. 5. Partial examples of the constructed dataset

Fig. 6. The hybrid annotation applied to the samples involves both rectangular annotation and polygon annotation

The previously created dataset was used for training in handwritten digit object detection. The training was conducted using an NVIDIA GeForce GTX 1660 SUPER graphics card. After 20 epochs of training, the model's validation loss reached approximately 0.11, at which point the training was halted, yielding Model-2, as shown in Fig. 7.

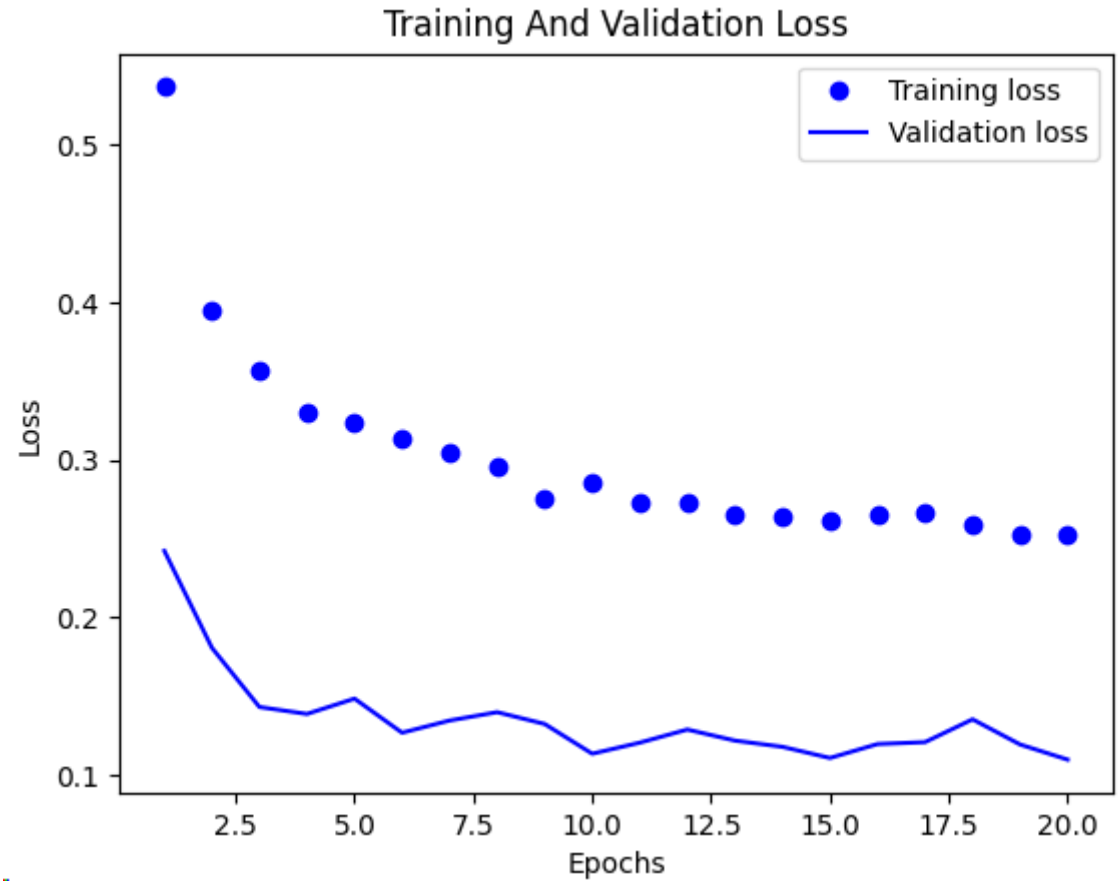

Fig. 7. The training process of Model-2

Fig. 8. Model-2 performed flawlessly on the test set

From the perspective of the test set, Model 2 performs well in dealing with adhered digits. It can effectively segment digits that are close to each other and adhered (as shown in Fig. 8). The ideal application scenario for Model-2 is to process localized abnormal image blocks, such as excessively wide image blocks, as well as image blocks with low confidence scores during recognition. These image blocks can be input into Model-2 for further

segmentation (as shown in Fig. 4), followed by recognition using Model-1. Upon verification, this approach has proven to be feasible.

By employing a hybrid segmentation approach that combines traditional segmentation with segmentation using Model-2, the accuracy of handwritten digit recognition—specifically, the accuracy of Model-1—exceeds 98-100% if the segmentation can be completed. For the majority of handwritten digit images in practical scenarios, a recognition accuracy of over 98% can be achieved.

In fact, through a series of limited comparative tests between this method and the Optical Character Recognition (OCR) system of a well-known instant messaging tool WeChat, a comprehensive evaluation was carried out. These tests were carefully designed with specific test datasets and controlled experimental conditions. The high accuracy of this method, which reaches over 99%, can be attributed to the effective hybrid segmentation method and the self-defined compact dataset. The hybrid segmentation method combines different segmentation algorithms in an innovative way, enabling more precise separation of characters. The self-defined compact dataset is specifically tailored for this OCR task, which contains carefully selected and pre-processed samples. These samples are representative and can effectively train the model to recognize various types of characters accurately. On the other hand, when examining the OCR accuracy of the well-known instant messaging tool WeChat, it was found to be less than 97%, as shown in Fig. 9. The fact that our method outperforms WeChat OCR in this case indicates, to some extent, the practicality of our approach.

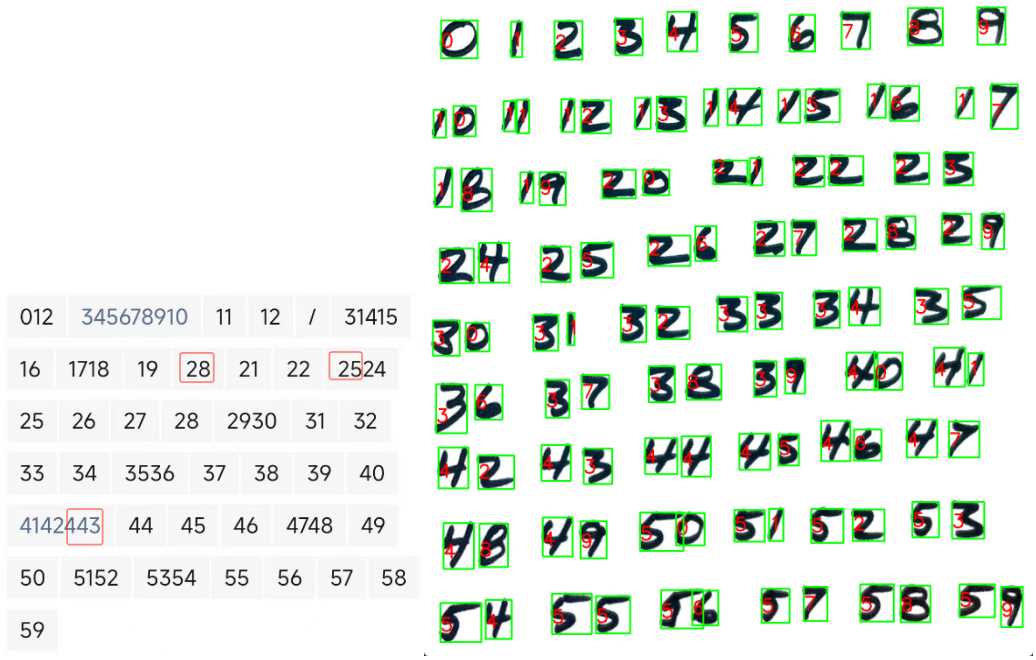

Fig. 9. (left) A recognition output from a renowned OCR system (with 3 errors detected) (right) Our results in segmentation and recognition (0 errors detected)

This study focuses on enhancing object recognition accuracy within specific contextual application scenarios. Although the research is exemplified using handwritten digits, the methodologies employed are broadly applicable and can be extended to other domains, such as agricultural fruit harvesting.

The underlying principles of our method are based on fundamental characteristics that are not specific to handwritten digits but are more general in nature. Our method focuses on extracting key features and patterns from the data. In the case of handwritten digits, it is able to identify the unique shapes and strokes that define each digit. This feature-extraction mechanism is not limited to the domain of digits. When considering the application in agricultural fruit harvesting, especially in the task of identifying fruits in images captured by robotic vision systems, there are several parallels. In the agricultural context, fruits like apples can be in complex situations within the images. They may be adhered to branches and leaves or even occluded by them. However, our method's strength lies in its ability to analyze the overall structure and local details within an image. Just as it can

distinguish the different parts of a handwritten digit, it can also differentiate between the fruit and the surrounding elements in an agricultural image.

The algorithms used in our method are designed to handle variations and uncertainties. Handwritten digits can have a wide range of variations in terms of writing styles, sizes, and orientations. Similarly, fruits in an agricultural setting can have different shapes, colors, and levels of maturity. Our method has proven to be robust against these variations in the digit-recognition domain, and this robustness can be transferred to the agricultural domain. Moreover, the pre-processing steps in our method, such as noise reduction and normalization, are also applicable in other domains. In agricultural images, there may be noise due to lighting conditions, wind-induced movement of branches, or the natural texture of the foliage. The normalization techniques used for handwritten digits can be adapted to standardize the agricultural images, making them more suitable for further analysis.

5 Conclusion

After comprehensive analysis and in-depth demonstration, this paper puts forward an effective method for handwritten digit processing. Initially, a hybrid segmentation approach is adopted, followed by the recognition process. In this method, two models are trained specifically. Model-1 adopts a compactness approach, which plays a crucial role in ensuring translational invariance. Model-2, on the other hand, utilizes a slightly improved Mask R-CNN. This model is trained with a custom-made dataset for performing local image segmentation on handwritten digits that are interconnected or partially overlapped. Through the application of this method, a high level of accuracy can be attained when dealing with complex real-world handwritten digit images. Moreover, this approach shows excellent adaptability not only in this field but also in other areas such as agricultural product production and growth.

6 Acknowledgements

This research has been supported by National Natural Science Foundation of China (No. 62272320), Department of Science and Technology of Guangdong Province project “Precision Monitoring System and Promotion Application for Agricultural Product Production Processes” (KTP20200222), Department of Education of Guangdong Province (No. 2022ZDZX4102), Shenzhen Science and Technology Innovation Commission (No. 20220812222043002), Shenzhen Polytechnic University Research Fund(No. 6025310045K, 602531008PQ, 40130050401).

References

[1] Y. LeCun, Y. Bengio, G. Hinton. Deep learning, Nature 521(7553)(2015) 436-444.
[2] A. Azulay, Y. Weiss. Why do deep convolutional networks generalize so poorly to small image transformations? Journal of Machine Learning Research 20(184)(2019) 1-25.
[3] R. Zhang, Making convolutional networks shift-invariant again, in: Proc. 2019 Proceedings of Machine Learning Research (PMLR), 2019.
[4] Md A. Islam, S. Jia, N.D.B. Bruce, How much position information do convolutional neural networks encode? in: Proc. 2020 International Conference on Learning Representations, 2020.
[5] A. Chaman, I. Dokmanić, Truly shift-invariant convolutional neural networks. <https://arxiv.org/abs/2011.14214>, 2021 (accessed 03.06.2024).
[6] L. Engstrom, B. Tran, D. Tsipras, L. Schmidt, A. Madry, Exploring the Landscape of Spatial Robustness. <https://arxiv.org/abs/1712.02779>, 2019 (accessed 03.06.2024).
[7] C.-W. Xiao, J.-Y. Zhu, B. Li, W. He, M.-Y. Liu, D. Song, Spatially transformed adversarial examples, in: Proc. 2018 International Conference on Learning Representations (ICLR), 2018.
[8] I.J. Goodfellow, J. Shlens, C. Szegedy, Explaining and harnessing adversarial examples, in: Proc. 2015 International Conference on Learning Representations (ICLR), 2015.
[9] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, R. Fergus, Intriguing properties of neural networks, in: Proc. 2014 International Conference on Learning Representations (ICLR), 2014.
[10] A. Kurakin, I. Goodfellow, S. Bengio, Adversarial examples in the physical world. <https://arxiv.org/abs/1607.02533>, 2017 (accessed 04.06.2024).

[11] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z.B. Celik, A. Swami, The limitations of deep learning in adversarial settings, in: Proc. 2016 IEEE European Symposium on Security and Privacy, 2016.
[12] N. Mu, J. Gilmer, MNIST-C: A Robustness Benchmark for Computer Vision. <https://arxiv.org/abs/1906.02337>, 2019 (accessed 04.06.2024).
[13] A. Karim, I. Muhammad, Target Identification and Tracking in Complex Environment. KazUTB 2(23)(2024) 28-37.
[14] Y.-C. Tang, M.-Y. Chen, C.-L. Wang, L.-F. Luo, J.-H. Li, G.-P. Lian, X.-J. Zou, Recognition and localization methods for vision-based fruit picking robots: A review, Frontiers in Plant Science 11(2020) 510.
[15] Y.-T. Tao, J. Zhou, Automatic apple recognition based on the fusion of color and 3D feature for robotic fruit picking, Computers and Electronics in Agriculture 142(2017) 388-396.
[16] B. Han, Z. Lu, J.-J. Zhang, R. Almodfer, Z.-T. Wang, W. Sun, L. Dong, Rep-ViG-Apple: A CNN-GCN Hybrid Model for Apple Detection in Complex Orchard Environments, Agronomy 14(8)(2024) 1733.
[17] J.-K. Zhong, Q. Cheng, X.-C. Hu, Z. Liu, YOLO Adaptive Developments in Complex Natural Environments for Tiny Object Detection, Electronics 13(13)(2024) 2525.
[18] B.-H. Qi, P. Zhang, W.-B. Huang, Research on Improved YOLO and DeepSORT Ship Detection and Tracking Algorithms, in: Proc. 2024 IEEE 4th International Conference on Electronic Technology, Communication and Information (ICETCI), 2024.
[19] A.G. Hochuli, L.E.S. Oliveira, A.S. Britto Jr, R. Sabourin, Segmentation-Free Approaches for Handwritten Numeral String Recognition. <https://arxiv.org/abs/1804.09279>, 2018 (accessed 05.06.2024).
[20] R. Sethi, I. Kaushik, Hand Written Digit Recognition using Machine Learning, in: Proc. 2020 IEEE 9th International Conference on Communication Systems and Network Technologies (CSNT), 2020.
[21] F.C. Ribas, L.S. Oliveira, A.S. Britto Jr, R. Sabourin, Handwritten Digit Segmentation: A Comparative Study, International Journal of Document Analysis and Recognition 16(2)(2013) 127-137.
[22] A. Hochuli, L.S. Oliveira, A.S. Britto Jr, R. Sabourin. Handwritten Digits Segmentation: Is it still necessary? Pattern Recognition 78(2018) 1-11.
[23] C.S. Jemimah, S.A.K. Arakkal, L. Mathew, Prediction of Coronary Artery Disease using Artificial Intelligence – A Systematic Literature Review, International Journal of Advanced Research in Engineering and Technology 14(1)(2023) 104-115.
[24] K.-M. He, G. Gkioxari, P. Dollár, R. Girshick, Mask R-CNN, in: Proc. 2017 IEEE International Conference on Computer Vision (ICCV), 2017.
[25] A. Nazir, M.A. Wani, You Only Look Once - Object Detection Models: A Review, in: Proc. 2023 10th International Conference on Computing for Sustainable Global Development (INDIACom), 2023.
[26] A. Elaoua, M. Nadour, L. Cherroun, A. Elasri, Real-Time People Counting System using YOLOv8 Object Detection, in: Proc. 2023 2nd International Conference on Electronics, Energy and Measurement (IC2EM), 2023.
[27] T.-Y. Lin, P. Goyal, R.B. Girshick, K.-M. He, P. Dollár, Focal Loss for Dense Object Detection, in: Proc. 2017 IEEE International Conference on Computer Vision (ICCV), 2017.
[28] S. Kar, M. El-Sharkawv, Object Detection Using Vision Transformed EfficientDet, in: Proc. 2023 IEEE National Aerospace and Electronics Conference (NAECON), 2023.