Paper:

An arithmetic sequence method for counting ordered pillboxes

Maofan,Cheng [1], Hongbin,Ma [*2] and Ying,Jin [*3]

*2 Beijing Institute of Technology, Beijing 100081, China
E-mail: mathmhb@bit.edu.cn
*3 Beijing Institute of Technology, Beijing 100081, China
E-mail: jinyinghappy@bit.edu.cn
[Received 2022/06/15; accepted 2022/07/22]

Abstract. It is often necessary to count pillboxes in the medical and health care industry. However, there are many challenges in the counting process, such as complex patterns on the surface of the pillboxes, surface reflection, etc. It is difficult to achieve satisfactory results using existing methods, such as object detection and image segmentation. In this paper, a simple and novel method for arithmetic sequence algorithm based on deep learning and traditional image processing technology fusion is proposed, which can effectively and efficiently count ordered pillboxes. Firstly, we use one deep-learning-based real-time instance segmentation model to segment the pillbox image, and secondly use the arithmetic sequence algorithm we proposed to count the pillboxes. The method requires little computation overhead to obtain an accurate result. Our experiment shows that the algorithm we proposed is effective, efficient, with counting accuracy up to 99.89 % in our dataset.

Keywords: instance segmentation; objects counting; arithmetic algorithm;

1. Introduction

Recently, with the development of artificial intelligence technology, the intelligent development of different industries has been promoted. The work of counting has always been very important in many different industries, and it is an important part of many different task requirements. The work of counting is mainly done manually in many existing industries, this work mode is not only time-consuming and labor-intensive, but also prone to errors. It may cause fatal errors in other work links in severe cases. Therefore, it is necessary to design an algorithm that can automatically count objects, which can not only save costs, but also reduce the error rate of counting and counting, and assist workers to improve the quality of work.

It is often necessary to count the number of pillboxes filled with medicines in the smart medical and health care industries, and determine whether the medicines need to be supplemented based on the statistical results. Using computers to automatically count objects has a certain research history, which mainly includes traditional-image-processing-based methods and some deep-learning-based methods. Mehmet BAYGIN [2] et al. proposed an object counting method based on image processing to count bottles. Researchers have proposed many CNN-based object counting methods[4] to count crowds in the field of crowd counting research. But the pillboxes to be counted are closely arranged in our counting scenario, and have different surface colors, surface patterns, and different widths. Pillboxes with different surface colors, surface patterns, and different widths are shown in Fig.1 (a), (b), (c) respectively. The closely-arranged pillboxes have high similarity, and the use of object detection methods is prone to misidentification. The light-colored pillbox has a problem of reflection during image acquisition, and the reflection will cause the information loss of the surface image of the pillbox. Some existing deep-learning-based object detection and counting methods cant complete the work well in such a scene, these methods generally require high computation overhead. There are often very different patterns on the image surface of different types of pillboxes. Some patterns are relatively simple, while others are more complex. It is difficult to overcome the influence of complex surface patterns only by the image-processing-based object counting method. Different types of pillboxes have different widths, which will also have a certain impact on counting. An intuitive idea to solve the problem of different widths is to introduce different prior width reference information for different pillboxes to assist counting, but it is not easy to collect and annotate prior information, and it will also affect the robustness of the model.

To address these issues, a simple and novel method for arithmetic sequence algorithm based on deep learning and traditional image processing technology fusion is proposed, which can effectively and efficiently count ordered pillboxes. We first segment the ordered image regions of the pillboxes using a real-time instance segmentation model, which is YOLACT [3], and then count the pillboxes in the segmented image using the arithmetic sequence algorithm proposed by us. In this paper, we focus on counting pillboxes. The arithmetic sequence method proposed by us does not rely on the prior width information of the pillbox, and at the same time, since we allow jump counting when counting, it can also alleviate the in-

fluence of reflection in the middle part of the image of the pillbox to a certain extent. The method proposed by us detects the straight lines in the segmented image by using the straight lines detection algorithm, and then uses the detected straight lines information to count the pillboxes, which alleviates the influence of complex patterns on the image surface on the counting results to a certain extent. The method has several practical advantages, firstly, it is simple, because it doesn't rely on prior information and only need some simple line detection methods. Secondly, the counting procedure is fast enough, it only need a tiny amount of computation overhead to get the final results. Finally, it can easily be extended to other similar applications.

The processing flow of our entire program is shown in Fig.2. The image segmentation model is used to segment the image, and then the line detection and the algorithm proposed by us are used to count the pillboxes in the segmented image.

The remainder of the paper is organized as follows: we highlight the existing works about instance segmentation and line detection in Section 2. In Section 3, we illustrate the proposed method in detail. The experiments and results are discussed in Section 4. Finally, we conclude the paper in Section 5.



(a) (b) (c)

Fig. 1. Surface reflection, surface complex patterns and different widths of pillboxes

2. Related Work

2.1. Instance Segmentation

Instance segmentation is one of the important and challenging areas in computer vision research, aimed at predicting the object class and pixel-specific object instance-mask [10]. A lot of research effort has been made for image instance segmentation. Some of them made a lot of effort push instance segmentation accuracy, such as Mask-RCNN [5], FCIS [8], while some of which focus on the instance segmentation speed. YOLACT is the first real-time instance segmentation algorithm with competitive performance in contrast to some state-of-the-art instance segmentation algorithm.

The YOLACT model consists of two parallel branches, the first branch is protonet, which is responsible for generating a series of prototype masks, and the second branch is a weight prediction branch [3], which predicts one weight coefficient for each prototype mask, and then the masks are linearly combined according to the weight to obtain the segmentation result. Since it is difficult to obtain a good result by directly using the object detection method to count the pillboxes, our idea of counting pillboxes is to segment the ordered pillboxes from the image, and then count the pillboxes.

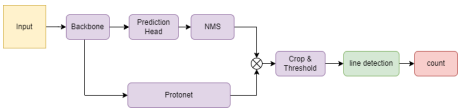


Fig. 2. The pipeline of models with the algorithm we proposed.

The program we designed consists of two stages, the first stage is to segment the object we need from the image captured by the camera using the instance segmentation methods, and the second stage is to count the objects using the arithmetic sequence algorithm proposed by us. Since speed is an indicator that we are concerned about in our scenario, so we choose the real-time instance segmentation model YOLACT to complete the image segmentation work, which prepares for the subsequent counting work. While our application scenario is a little different from the application scenario in the original YOLACT. The original YOLACT adopted FPN [9], FPN was proposed by Tsung-Yi Lin et al. in 2017. This work is dedicated to solving the problem of detecting objects of different scales in images. There are no objects with large scale differences in our scenario, so we can consider removing FPN for detecting the objects with different scales, and this change will also speed up the model a little bit.

2.2. Lines Detection

Line detection is an important problem in image processing and computer vision. There has been a lot of research effort in the area of image line detection, and many well-known algorithms have been proposed, such as the classic Hough line detection, LSD [12] algorithm, FLD [7] algorithm, EDlines [1] algorithm, LSWMS [11] algorithm, etc. The EDlines algorithm proposed by Cuneyt Akinlar [1] et al can process any image regardless of its origin, orientation or size, and produce accurate and robust line detection. Furthermore, it needs no parameter tuning at all, and the lines in image can be segmented in a very short amount of time and even in real-time. There is a gap between every two of the ordered pillboxes. This gap can be easily detected by the line detection algorithm. Similarly, the gaps between all the pillboxes can be detected by the line detection algorithm. A specific algorithm can be used to calculate the number of pillboxes after obtaining these straight lines. Therefore, the line detection on the segmented images is the first step of the counting stage. We choose the EDlines algorithm to segment the lines considering our requirements for real-time.

3. Proposed Methodology

In this section , we'll discuss the pre-process to the image captured by the camera and the method we pro-



(a) (b) (c)

Fig. 3. Real image data captured by the camera in the real application scenarios.



Fig. 4. Segmented pillbox image.

posed for counting the drug box in an image. The method mainly includes two steps: (1) extraction of boundary lines and (2) using arithmetic sequence algorithm to compute the number of pillboxes.

3.1. Pre-process

YOLACT-based Image Segmentation There are two parallel parts in YOLACT model: (1) protonet branch and (2) mask coefficients predicting branch. The protonet generates a set of prototypes masks using the feature map from the backbone and the mask coefficients predicting branch predicts the mask coefficients for per-instance. Then the instance masks are produced by linearly combining the prototype masks with the mask coefficients. We adopt ResNet50 or ResNet101 [6] as the backbone network of the model to extract features. There are some images captured by the camera in Fig.3. We view all the ordered pillboxes in a row as an object, there are two, one, and zero object respectively in Fig.3. We can obtain the location of the object in the image from the mask produced by the YOLACT model. Objects can be easily segmented once we know their location in the image. The pillbox image segmented by the YOLACT model is shown in Fig.4.

Line Image Pre-process We can obtain a lot of lines in an image after Lines detection, as is shown in Fig.5, but many of them are not what we need, we need to filter out some of them. What is beneficial to count the pillboxes is the vertical lines instead of the horizontal ones, a natural idea is to preserve the vertical ones and remove the others. We can obtain the coordinates of the two endpoints of each line from line detection, denote the coordinates by (x_{istart},y_{istart}), (x_{iend},y_{iend}).Define

$$k = (x_{istart}\text{-}x_{iend})/(y_{istart}\text{-}y_{iend}) \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad (1)$$

k of the vertical lines is very small so that we can choose an appropriate threshold **T** to filter out the non-vertical lines in the image. There are many short vertical lines

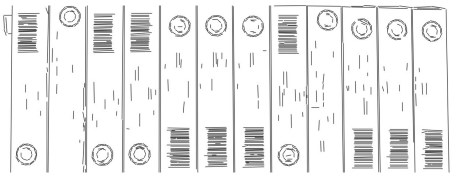


Fig. 5. EDlines for the segmented pillbox image.

after filtering out the non-vertical lines, as is shown in Fig.5, which are obviously not the edge lines of the pillbox. They are not helpful for us to count. We can get rid of them by their length, whose length is less than **l** will be filtered out. There are many adjacent lines in images, which are not the target lines we need. So we use the straight line in the middle of these adjacent straight lines to represent these adjacent straight lines, map these adjacent straight lines to a straight line, and then we proceed to the next step. The filtered image which we called line image is depicted in Fig.6.

3.2. Arithmetic Sequence Algorithm

The main idea of Arithmetic Sequence Algorithm comes from that the width of the same pillbox is a constant. Obviously, the pixel width of pillbox in an image is still a constant, as the pillbox is ordered. We can obtain a series of straight lines from the segmented pillbox image after preprocessing and the coordinates of the straight lines in the segmented image at the same time. Since the pillboxes are ordered in the image, the straight lines obtained after image preprocessing are almost equally spaced. The difference of abscissa value between the lines can be seen as the pixel width of a single pillbox. In the same pillbox image, each type of pillbox is the same, and these pillboxes have the same pixel width. When these pillboxes are neatly arranged, the abscissa value of the straight line at the edge of the pillbox can be approximately regarded as an arithmetic sequence. If we can find this arithmetic sequence by algorithm, then at the same time we can get the number of pillboxes in the image.

Suppose that there is a sequence a_1, a_2, ..., a_n, if a_{i+1} - a_i = δ, i=1,2...., n-1, it is called arithmetic sequence. Consider another sequence b_1, b_2, ..., b_n:

$$b_j - b_i \in [\delta\text{-}\epsilon,\delta+\epsilon], \ i = j\text{-}1, \ j = 2,3,....,n \quad . \quad . \quad (2)$$

This sequence is not a true arithmetic sequence, but it is similar to an arithmetic sequence. If the elements in the sequence satisfy the relation mentioned in Eq.2, we can approximate it as an arithmetic sequence. Then, we can make a rule to find the elements in the sequence which satisfy the relation in Eq.2. When all the elements in the sequence satisfy the relation in Eq.2 were found, denote the new elements sequence by c_1, c_2, ..., c_m, $m < n$.

$$c_{i+1} - c_i = \delta_i, \ \mathrm{i} = 1, 2, ..., m - 1 \quad . \quad . \quad . \quad . \quad . \quad . \quad (3)$$

Sequence element c_i can be regarded as the line between two adjacent pillboxes. δ_i represents the width of the i-th pillbox. There is an sequence after the pre-process

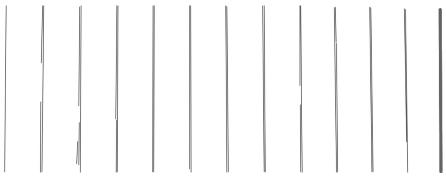


Fig. 6. The lineimage.

of the segmented image, an appropriate arithmetic sequence will be found when using the arithmetic sequence algorithm on it. Since the pillboxes have the same size, although δ_i is not identical, it is within a certain range. We can approximate the width of the pillbox by choosing δ_j with close values to calculate their mean, where $j = 1, 2, ..., J, J \leq m$.

$$\bar{\delta} = (\sum_{j=1}^{J} \delta_j)/J \quad . . . . . (4)$$

The number of pillboxes *num*:

$$num = \sum_{i=1}^{m-1} \delta_i / \bar{\delta} \quad . . . . . (5)$$

We select the m-th (m = 1, 2, ... , M, M < n) line at the right end of the straight line image as the initial line, and use the k-th (k = m+1, m+2,...K, K < n) line to make the difference of x-coordinate value between the k-th line and the initial line to get δ_1, take the k-th straight line as the benchmark, use the abscissa of the straight line after the k-th straight line and the abscissa of the k-th straight line to make the difference. If the difference and δ_1 are within a certain error range, the straight line can be considered as a straight line at the edge of the pillbox, and then use the straight line as a new benchmark, and repeat the previous algorithm until no straight line that meets the conditions is found. We obtain an arithmetic sequence under this δ_1 now. For different initial straight lines, the algorithm will find many different arithmetic sequences. There is a longest arithmetic sequence among these arithmetic sequences. This longest arithmetic sequence corresponds to the pillbox in the image. Using this arithmetic sequence makes it easy to get the number of pillboxes in the image.

4. Experiments And Results

We made a dataset of pillbox images collected from real scenarios, and used 1000 of them to test our designed program. The whole program is broken into two steps. (1) The deep learning-based real-time instance segmentation model YOLACT is used to segment the collected images in the first step of the program and obtain the images of the pillbox area that we are interested in, and (2) then the arithmetic sequence algorithm proposed by us is used to count the pillboxes after the segmented image is pre-processed.

Table 1. Instance segmentation accuracy and speed, objects refer to all objects presented in the test images, error objects are the ones presented in the image but not detected, all images with one object or zero object are correctly detected.

images	objects	error	acc	speed(s)
280	453	6	98.67 %	2.0
350	631	9	98.57 %	2.0
370	697	11	98.42 %	2.0

4.1. Instance Segmentation

The YOLACT model was trained using the 1500 images in the image dataset we made as training data in the image segmentation stage. Since the ordered pillboxes are regarded as an overall object in our application scenario, these objects have a certain similarity. There is no object with very different scales and categories, so a model with high segmentation accuracy can be trained with a small amount of data. The test dataset is input into our trained YOLACT model for testing when the model training is complete. There may be two overall pillbox objects, one object or no object in a single image captured in the application scenario. There are a total of 1781 objects in the 1000 test images. 1755 objects were successfully segmented by our trained model, and the segmentation accuracy is 98.5%. The segmentation accuracy is shown in Table 1.

However, there were a small number of inaccurate objects in the segmented images. One possible reason for the inaccurate segmentation is the similarity between pillbox object and background. So the segmentation model misidentifies part of the background as part of the object. There are also some objects that cannot be successfully segmented. As is shown in Fig.7, the brightness difference presented by the two objects under the same lighting conditions is large in such a scene image, which makes the model unable to segment correctly. The resolution of the image is 2448×2048, and the average time taken to segment a single image is 2s. There is still a certain gap between the real-time performance and the reasoning speed can be further improved.



(a) (b) (c)

Fig. 7. Some images with relatively large brightness difference between two objects.

4.2. Counting

We used the segmented image dataset to test Yolov4-based, prior-knowledge-based and the arithmetic se-



Fig. 8. The error result of Yolov4-based method.



Fig. 9. The error result of prior-knowledge-based method.



Fig. 10. The error result of our method.

Table 2. The result of counting using the Arithmetic Sequence algorithm we proposed.

method	samples	error	acc
Yolov4-based	500	154	69.20%
prior-knowledge-based	1755	106	93.96%
ours	1755	2	99.89 %

quence method proposed by us respectively. The counting results obtained based on these three methods are shown in Table 2, where Yolov4-based method was tested using 500 images and errors occurred in 154 images. It is difficult to obtain a satisfactory result using the object detection method. One possible reason is that the pillboxes are closely arranged in the image and there are some straight lines on the surface of some pillboxes, so that the object detection model appears to recognize one pillbox as multiple ones or the identified objects overlap incorrectly, which results in wrong counting results, as shown in Fig.8.

The prior-knowledge-based method has the possibility of obtaining higher counting accuracy in theory. It must manually measure the pixel width of the pillbox in advance using the prior-knowledge-based method and then the prior knowledge is input into the counting program as reference information. However, errors are unavoidable in the measurement process, the measurement process is very time-consuming and labor-intensive due to the different sizes of the pillboxes and errors will be introduced. The error result is shown in Fig.9. The prior-knowledge-based counting method is also interfered by the straight line on the surface of the pillbox, which results in counting errors. The prior-knowledge-based counting result is shown in Table 2, and the final counting accuracy is 93.96 %.

The image obtained from the segmentation in the previous stage is firstly pre-processed in the stage of pillbox counting, and the pre-processing includes image filtering, line detection, line filtering and so on. The abscissa of the straight line in the image can be obtained after preprocessing the segmented pillbox image. Then the arithmetic sequence algorithm we proposed is used to count the segmented object images, these coordinates are input into the arithmetic sequence algorithm program we designed to get the counting results, and we make statistics on the counting results. There are a total of 1755 objects, as shown in Table 2, in our pillbox test dataset, there were only 5 samples with incorrect counting results in the test result, and the error between the result of this wrong counting and the real number of pillboxes in the image is no more than 2, which shows the effectiveness of counting algorithm. A typical error sample is shown in Fig.10. The reason for the wrong sample is that there is a pillbox at the left side of this sample that is not closely arranged with other pillboxes. There is a distance between this pillbox and other pillboxes, which will affect the accuracy of the algorithm. There is also some image data collected with low quality sometimes in real scenarios, and there is still room for further improvement in the processing capability of the algorithm for data with low quality. The object processed by the arithmetic sequence algorithm is an array and the length of this array is relatively short, so the running speed of the arithmetic sequence algorithm is very fast. The average running time of the arithmetic sequence part of the program is 5ms according to the statistics of the experimental results.

5. Conclusion

An arithmetic sequence method that can count the pillboxes with high accuracy was proposed by us in this paper. Firstly, the YOLACT model is used to automatically complete the segmentation of the image area of the pillbox for the input image. Secondly£ the EDlines method is used to automatically perform line detection, and finally, the number of pillboxes in the image area of the pillbox is automatically identified by the arithmetic sequence algorithm proposed by us. 1000 image data were used to test the method proposed by us. 1755 pillbox images were successfully segmented from these image data, and the image segmentation accuracy was 98.5 %. Then using the images obtained from the segmentation tested the arithmetic sequence method. Result shows that only 2 samples were counted incorrectly in the 1755 pillbox ob-

jects, the error range is within 2 compared with the real number of pillboxes in the image. The counting accuracy reached 99.89%. The arithmetic sequence counting part of the whole program runs with an average running time of 5ms, but the running time of the image segmentation part is still far from real-time, which is also a direction that needs to be improved in future research.

Acknowledgements

This work was partially funded by the National Key Research and Development Plan of China (No. 2018AAA0101000) and the National Natural Science Foundation of China under grant 62076028.

References:

[1] C. Akinlar and C. Topal, "EDLines: A real-time line segment detector with a false detection control", Pattern Recognition Letters, 32(13):1633–1642, 2011.

[2] M. Baygin, M. Karakose, A. Sarimaden, and E. Akin, "An image processing based object counting approach for machine vision application", arXiv preprint arXiv:1802.05911, 2018.

[3] D. Bolya, C. Zhou, F. Xiao, and Y. J. Lee, "Yolact: Real-time instance segmentation", In Proceedings of the IEEE/CVF international conference on computer vision, pp. 9157–9166, 2019.

[4] G. Gao, J. Gao, Q. Liu, Q. Wang, and Y. Wang, "Cnn-based density estimation and crowd counting: A survey", arXiv preprint arXiv:2003.12783, 2020.

[5] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn", In Proceedings of the IEEE international conference on computer vision, pp. 2961–2969, 2017.

[6] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition", In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770–778, 2016.

[7] J. H. Lee, S. Lee, G. Zhang, J. Lim, W. K. Chung, and I. H. Suh, "Outdoor place recognition in urban environments using straight lines", In 2014 IEEE International Conference on Robotics and Automation (ICRA), pp. 5550–5557. IEEE, 2014.

[8] Y. Li, H. Qi, J. Dai, X. Ji, and Y. Wei, "Fully convolutional instance-aware semantic segmentation", In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 2359–2367, 2017.

[9] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection", In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 2117–2125, 2017.

[10] A. Mueed Hafiz and G. Mohiuddin Bhat, "A Survey on Instance Segmentation: State of the art", arXiv e-prints, pp. arXiv–2007, 2020.

[11] M. Nieto, C. Cuevas, L. Salgado, and N. García, "Line segment detection using weighted mean shift procedures on a 2D slice sampling strategy", Pattern Analysis and Applications, 14(2):149–163, 2011.

[12] R. G. Von Gioi, J. Jakubowicz, J.-M. Morel, and G. Randall, "LSD: A line segment detector", Image Processing On Line, 2:35–55, 2012.