A novel stock counting system for detecting lot numbers using Tesseract OCR

Parkpoom Lertsawatwicha1 · Phumidon Phathong1 · Napatsorn Tantasanee1 · Kotchakorn Sarawutthinun1 · Thitirat Siriborvornratanakul

Received: 9 April 2022 / Accepted: 15 September 2022 / Published online: 6 October 2022
© The Author(s), under exclusive licence to Bharati Vidyapeeth’s Institute of Computer Applications and Management 2022

Abstract Counting stock is one of the warehouse’s methods for preventing insatiable stock. Moreover, it could help the company forecast how many products they need to store and predict the replenished goods for customers. However, stock count in the medical business, which sells specialized medical equipment, needs more focus on, because it uses to treat the patient. So that lack of inventory should not happen. In a normal situation, stock count at some hospitals is quite hard for salespeople, especially hospitals in upcountry that far away. During the COVID-19 situation, many limits need to be strict. At this point, it causes a shortage of goods in many hospitals. In this paper, we represent how computer vision can help this process. When the hospital’s officer sends images of stock to our system. The system will recognize the quantity and lot number of goods that remain in the hospital. Therefore, salespeople can decrease the times to visit hospitals. The result showed that for text detection and text recognition in a specific use case. Our prototype system achieves 84.17% in accuracy.

Keywords Computer vision · Object detection · Optical Character Recognition · Tesseract

1 Introduction

Replenishing stock with precise number is a key in medical business. To keep customers using and reordering the goods, on one hand, a salesperson needs to visit the hospital and count stock more often. On the other hand, traveling to hospitals in some areas is inconvenient. Even there is a central system that hospitals can fill in the usage to inform the company, sometimes they don’t report it after use immediately. In the COVID-19 situation, there are lots of new limitations as many rules need to be strict and followed by the hospital’s requests. According to information from a medical supply company, the lead time to replenish goods during COVID-19 is longer than a normal situation. This problem has many effects on the company such as losing the chance to sell goods to the hospital again or making them feel unpleasant. Solving this problem requires a system that can count quantities and detect lot numbers from images (taken by the hospital’s officer or salespeople) of the remaining stock, decreasing time to visit the hospital and allowing the sale company to acknowledge the remaining stock fast.

In this paper, our contribution is to use high accuracy text detection and recognition to precisely recognize the quantity and lot number of goods that remain in the hospital, helping salespeople to visit hospitals less often but more efficiently. We first process an input image to detect the word “LOT” enclosed by a rectangular box. The regions containing lot numbers will be relatively inferred from the positions of

1 Graduate School of Applied Statistics, National Institute of Development Administration, 148 Serithai Road, Bangkok 10240, Thailand
the detected “LOT” rectangles. After these regions with lot numbers are properly cropped out from the input image, Optical Character Recognition (OCR) is applied to read the lot numbers and output them as text sequence. Speaking of OCR for reading text from an image of textual document, previous works involve using OCR for historical documents [1] and automatic detection of books [2]. However, these mentioned works have different purposes from ours because our research purpose is to recognize the quantity and lot number of goods in the hospital context so that users may use our results to compare with the sale company’s database.

2 Related work

Apart from demand forecasting [3], stock or inventory counting is another important problem in Supply Chain Management (SCM). Because of each object’s uniqueness, previous solutions span widely from using a multi-robot system [4], using an automatic camera to record visual inventory for being counted manually later by human [5], or using vision-based template matching to locate and count target objects [6, 7]. Our work is different from these previous works as the counting is done on the stock of small medical products stored in a hospital’s stock room, requiring neither automatic robots nor cameras as in the large-scale inventory. Also, our goal is to detect and recognize lot numbers printed on each box package to count different lots of medical products accurately. The static pattern of how the lot number is printed on the box (as shown in Fig. 1) implies that it may not be necessary to apply general-purpose visual template matching techniques like [6, 7].

According to the survey of [8, 9], the solutions for obtaining text from natural images could be categorized into two ways. The first way is a step-wise method which consists of a series of processing steps including detection, segmentation, and recognition; the other way is an integrated method that combines several steps into a unified framework. In this paper, the first way of step-wise method is applied as we can design our processing steps and assemble several existing methods to suit requirements.

In fact, there are several previous works trying to recognize objects using computer vision, such as [10] hand motion recognition, [11] human motion analysis for recognition from 3D gait signatures, and [12] color recognition using a Bayesian classifier. To recognize lot numbers printed on medical product packages, logo or object detection is one possible solution. For a complicated image with lot of visual variants such as perspective distortion, multi-colored text, artistic font, uneven light or too much shadow etc., machine learning techniques and deep neural networks (a.k.a., deep learning) are said to be more efficient than handcrafted or rule-based techniques. Yufeng and Bo [13] proposed a solution to detect logos from bicycles using Haar Classifier and AdaBoost. Despite its high recognition rate, the precision is low compared to the local binary pattern algorithm. Since 2012, deep learning methods have gained a lot of attention according to [14]. Many works like [15, 16] used vision-based deep learning to obtain logo or text from an image with very promising results. Also, there is the work of [17] that compared the performance of several deep learning-based object detector’s architecture. This work concludes that Swin Transformer achieves the highest average precision (AP) with 57.70% followed by Detecting Objects with Recursive Feature Pyramid and Switchable Atrous Convolution (DetectoRS) (53.30%), EfficientDet-D2 (43.00%), and YOLOv4 (43.00%).

However, as a deep learning method often needs a huge dataset, it is not suitable for our medical stock situations where images are scarce. Unlike deep learning, traditional methods don’t require a lot of datasets and provide a good result in some situations. Kuznetsov and Savchenko [18] compared a custom deep learning algorithm based on DEep Local Features (DELF) and traditional methods (neither machine learning nor deep learning) such as scale-invariant feature transformation (SIFT), accelerated KAZE (AKAZE) and binary robust invariant scalable keypoints (BRISK) for a logo detection task. The results reveal that SIFT gets the most precision (0.89), BRISK gets the most recall (0.68), and AKAZE gets a good overall result (0.62 for precision and 0.64 for recall). In conclusion, our research work will experiment on rule-based techniques to detect lot numbers from images and input them to OCR.
3 Proposed method

In this work, extracting lot numbers from an input image is done by Algorithm 1. First, RotateImage is the process that rotates an input image to horizon orientation by using an orientation detected by Tesseract [19], one of the popular open-source OCR tools for converting an image into textual information. Then, in the rotated image, all squares are detected in FindSquares as shown in Fig. 2; this process includes converting the rotated image to grayscale, thresholding with Otsu, applying Morphology operation, doing Canny Edge, and finding contours. All detected squares are further classified into red (squares without the word “LOT”) or green (squares with the word “LOT”) squares in FilterCandidate as shown in Fig. 3. To get lot numbers written next to the detected green squares, each green square is extended (stretched) to cover its corresponding lot number as shown in Fig. 4 (FindTextRegionWithLotNumber), then the extended square is cropped out from the image and straightened up using image warping. Finally, Tesseract OCR is applied to each resultant region and the output lot numbers are read as shown in Fig. 5.
Algorithm 1 Extracting lot numbers from an input image

1: `img ←` an RGB input image
2: Text localization:
3: `img_rotate ← RotateImage(img, TesseractTextOrientation(img))`
4: `all_squares ← FindSquares(img_rotate)`
5: Text verification:
6: `lot_squares, nolot_squares ← FilterCandidate(all_squares)`
7: Text segmentation:
8: `lotnumber_squares ← FindTextRegionWithLotNumber(lot_squares)`
9: `ocr_inputs ← Straighten( Crop(lotnumber_squares) )`
10: Text recognition:
11: `Text_sequence ← Tesseract(ocr_inputs)`

The proposed system was evaluated with 43 images containing 240 lot numbers. The overall accuracy was 84.17%. In addition, we divided the evaluation into two parts. (1) Detecting word “LOT”: We calculated accuracy in this part by dividing the number of detected words “LOT” by the total number of all words “LOT.” (2) Detecting lot numbers: In this part, we measured accuracy on a word level. For example, if there was one character in lot number which was wrongly recognized, we defined that lot number as a false prediction. We decided to use this approach because the

4 Experimental results and discussion

To evaluate our prototype system in real production, we host our system on the Amazon Web Services (AWS) cloud so that users can use the system by accessing our website. In the website, users can import an input image. After finishing all processes, the result will be displayed, allowing users to export the results as a CSV (Comma-Separated Value) file for their further use in the stock counting database. Example results are shown in Fig. 6.

![Fig. 6](image_url)  
All processes will be demonstrated on our website hosted in the AWS cloud. The results of lot numbers will show in summarized text and can be exported to a CSV file.

Fig. 6 All processes will be demonstrated on our website hosted in the AWS cloud. The results of lot numbers will show in summarized text and can be exported to a CSV file.

Springer
correct lot numbers should be found to be able to match lot numbers in the database. Accuracy results of our proposed methods are 91.67% for detecting word “LOT,” 91.82% for detecting lot numbers, and 84.17% for the overall.

After evaluation, we could categorized the errors into four types. First, the inability to detect some contours that contain the “LOT” text. Because we currently use the RETR_EXTERNAL find-contour mode in OpenCV (Open Source Computer Vision), the program will ignore any inner contours, causing the program to miss detecting some ROIs with the “LOT” text. We tried experimenting with other find-contour modes like RETR_LIST or RETR_CCOMP to include all contours but it resulted in too many contours detected; this made the program 9-10x slower than the RETR_EXTERNAL mode despite slight improvement in accuracy. To further investigate this error, we have to reduce the number of ROIs to reduce the number of times that the program needs to check whether each ROI contains the text “LOT.” For example, using the area threshold to filter out regions that are too big or too small, trying a binarization method designed specially for document image [20], or applying image denoising for better character recognition.

The second type of error is when OCR cannot read some “LOT” text. This error occurs after we get a list of ROIs, warp them, and feed the warped ROIs to Tesseract. It came out that Tesseract can’t read some “LOT” text properly. The third type of error is when Tesseract OCR cannot detect the proper text orientation regarding some images. This error occurs when an image contains few textual information so that Tesseract cannot detect a proper textual orientation. The fourth type of error is when Tesseract OCR cannot read the lot number accurately. This is caused by the low quality of input images such as low resolution, noise, and poor lighting conditions. To resolve this, we may have to improve our preprocessing methods to enhance input images.

For the last three types of errors (types 2, 3, and 4), trying alternative OCR engines may help resolve or ease the errors. According to [22], the top performing OCR engines are Google Cloud Vision and AWS Textract. However, this study uses Tesseract because it is a common and public tool. For the error of detecting the word “LOT,” we found that they were caused by an inability to detect some contours that contain text (70%) and an inability to read text “LOT” by OCR (30%). Hence, improving the contour detection method with a more robust solution, like deep learning-based object detection or text-specific contour detection method [23], may significantly ease this error. As for the error in detecting the lot numbers, we could not discover any specific error pattern; for example, \( W \rightarrow V?? \) (1 occurrence), \( W \rightarrow VWW \) (1 occurrence), \( W \rightarrow \ \backslash A \) (1 occurrence) and \( W \rightarrow \ \backslash AJ \) (2 occurrences). Trying alternative OCR engines may help clarify this.

5 Conclusion and future work

This paper presents an approach for text detection and text recognition in a specific use case of medical stock counting. Our prototype system achieves 84.17% in the overall accuracy. However, some lot numbers still cannot be detected accurately due to the low quality of images. Our future work would focus on improving algorithms to achieve higher accuracy and faster computation. One interesting alternative is to detect unstructured text from an input image using deep learning-based techniques for information extraction like Named-Entity Recognition (NER). Another future direction is to develop a GUI (Graphic User Interface) application and design a complete workflow for actual production deployment.

References

1. Martínek J, Lenc L, Král P (2020) Building an efficient OCR system for historical documents with little training data. Neural Comput Appl 32:17209–17227
2. Fatema K, Ahmed MR, Arefin MS (2021) Developing a system for automatic detection of books. In: International Conference on image processing and capsule networks (ICIPCN), Lecture Notes in Networks and Systems (LNNS), vol 300, pp 309–321. https://doi.org/10.1007/978-3-030-84760-9_27
3. Zohdi M, Rafiee M, Kayvanfar V, Salamirad A (2022) Demand forecasting based machine learning algorithms on customer information: an applied approach. Int J Inf Technol 14:1937–1947
4. Casamayor-Pujol V, Morenza-Cinos M, Gastón B, Pous R (2020) Autonomous stock counting based on a stigmatic algorithm for multi-robot systems. Comput Ind 122:103259
5. Cidal GM, Cimbek YA, Karahan G, Boler OE, Ozkardesler O, Uvet H (2019) A study on the development of semi automated warehouse stock counting system. In: International Conference on electrical and electronics engineering (ICEEE), 16–17 April 2019, Istanbul, Turkey. https://doi.org/10.1109/ICEEE2019.2019.000069
6. Kejriwal N, Garg S, Kumar S (2015) Product counting using images with application to robot-based retail stock assessment. In: IEEE International Conference on technologies for practical robot applications (TePRA), 11–12 May 2015, Woburn, MA, USA. https://doi.org/10.1109/TePRA.2015.7219676
7. Sharma T, Jain A, Verma NK, Vaskarla S (2019) Object counting using KAZE features under different lighting conditions for inventory management. In: IEEE Applied Imagery Pattern Recognition Workshop, 15–17 October 2019, Washington, DC, USA. https://doi.org/10.1109/AIPR47015.2019.9174578
8. Chen X, Jin L, Zhu Y, Luo C, Wang T (2021) Text recognition in the wild: a survey. ACM Comput Surv 54(2):1–35
9. Ye Q, Doermann D (2015) Text detection and recognition in imagery: a survey. IEEE Trans Pattern Anal Mach Intell 37(7):1480–1500
10. Kerdvibulvech C (2014) Hand motion recognition using an extended particle filter. Lect Notes Comput Sci 8563:71–80
11. Kerdvibulvech C, Yamauchi K (2014) 3d human motion analysis for reconstruction and recognition. Lect Notes Comput Sci 8563:118–127
12. Kerdvibulvech C (2010) Real-time augmented reality application using color analysis. In: IEEE Southwest Symposium on image analysis and interpretation (SSIAI), pp 29–32, 23–25 May 2010, Austin, TX, USA. https://doi.org/10.1109/SSIAI.2010.5483927

13. Yufeng D, Bo Z (2018) Intelligent identification method of bicycle logo based on Haar classifier. In: International Conference on systems and informatics (ICSAI), pp 973–977, 10–12 November 2018, Nanjing, China. https://doi.org/10.1109/ICSAI.2018.8599499

14. Long S, He X, Yao C (2021) Scene text detection and recognition: the deep learning era. Int J Comput Vis 129(1):161–184

15. Xue W, Li Q, Xue Q (2020) Text detection and recognition for images of medical laboratory reports with a deep learning approach. IEEE Access 8:407–416

16. Sharma N, Mandal R, Sharma R, Pal U, Blumenstein M (2018) Signature and logo detection using deep cnn for document image retrieval. In: International Conference on frontiers in handwriting recognition (ICFHR), pp 416–422, 05–08 August 2018, Niagara Falls, NY, USA. https://doi.org/10.1109/ICFHR-2018.2018.00079

17. Zaidi SSA, Ansari MS, Aslam A, Kanwal N, Asghar M, Lee B (2021) A survey of modern deep learning based object detection models. arXiv arXiv:2104.11892 [cs.CV]

18. Kuznetsov A, Savchenko A (2019) Sport teams logo detection based on deep local features. In: International Multi-Conference on engineering, computer and information sciences (SIBIRCON), pp 0548–0552, 21–27 October 2019, Novosibirsk, Russia. https://doi.org/10.1109/SIBIRCON48586.2019.8958301

19. Zacharias E, Teuchler M, Bernier B (2020) Image processing based scene-text detection and recognition with tesseract. arXiv arXiv:2004.08079 [cs.CV]

20. Rani U, Kaur A, Josan G (2019) A new binarization method for degraded document images. Int J Inf Technol 2019:1–19. https://doi.org/10.1007/s41870-019-00361-3

21. Hussain J, Vanlalruata (2022) Image denoising to enhance character recognition using deep learning. Int J Inf Technol 2022:1–13. https://doi.org/10.1007/s41870-022-00931-y

22. Best OCR by text extraction accuracy in 2022. https://research.aimultiple.com/ocr-accuracy/. Accessed 20 Jan 2022

23. Shekar BH, Raveeshwara S (2022) Contour feature learning for locating text in natural scene images. Int J Inf Technol 14:1719–1724

Springer Nature or its licensor holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.