TSB-funded Project 'TADD' - Trainable vision-based anomaly detection and diagnosis Technical Report for August 2013

Ran Song and Tom Duckett

Agri-Food Technology Research Group, Lincoln Centre for Autonomous Systems Research, School of Computer Science, University of Lincoln, UK

Abstract

In this project report, we propose a method which can partially address the problem that we raised in the last report: semantic labelling based on image segmentation. Note that due to the lack of ground truth, we have not implemented any experimental evaluation or quantitative comparison for the proposed method. However, preliminary visual results demonstrate that our method is potentially an effective solution, or at least a promising one that we can work with towards a prototype, to the 'Trainable Vision-based Anomaly Detection and Diagnosis' (TADD) project.

We also summarise the findings revealed in the development of the reported technique, which forms a solid base for the next stage of the project and gives helpful hints for further improvement of the current method.

1. Review

We first review the targets of the 'Trainable Vision-based Anomaly Detection and Diagnosis' (TADD) project. The key features of TADD are shown in Fig. 1

In the last report, we provided feasible methods to calculate tray size and the orientation of tray. In this report, we shall show how to achieve other features listed in Fig. 1 (such as seal region auto detection and recognition, price/date tick box auto detection and recognition) in an intelligent and autonomous manner. And we demonstrate our method by implementing it on the real images offered by our project partner Ishida Europe Ltd. Basically,

Software Phase 1	
Key feature	Development feature
Position of tray	What is the orientation of the tray on arrival to the machine.
Tray size	Auto detection of the tray to auto set speeds, timings of sensing and reject.
Detecting top film on a tray.	Printed, clear.
Detecting seal width on a tray.	Measures min/max width. Adjustable by simple +/- setting.
Detecting film position on a tray.	Overhang of film with tolerance. Bowed tray with shrink. Printed film position.
Detecting label on a tray.	
Detecting label position on a tray.	
Barcode detection	
Price detection	Auto character recognition with tick box for fixed or variable data.
Weight detection	Auto character recognition with tick box for fixed or variable data.
Date detection	Auto character recognition with tick box for fixed or variable data.
Detecting other text	
Product in seal detection	Adjustable by example setting. Pure visual, laser scattered light and stress pattern analysis.
Product ID for diverger/converger	

Figure 1: University of Lincoln tasks in Software Phase 1, as taken from the Second Level Project Plan

the method still follows the workflow that we proposed in our last report as shown in Fig. 2.

2. Methodology

As we mentioned in our last report, at the first stage, we selected the graph-based method [2] for image segmentation, a process of partitioning a digital image into multiple components. We also revised the original method to accelerate it and demonstrated that it worked well for our specific test images through visualised results (Fig.4 in the last report). However, our further research done in August showed that it is not capable of segment-

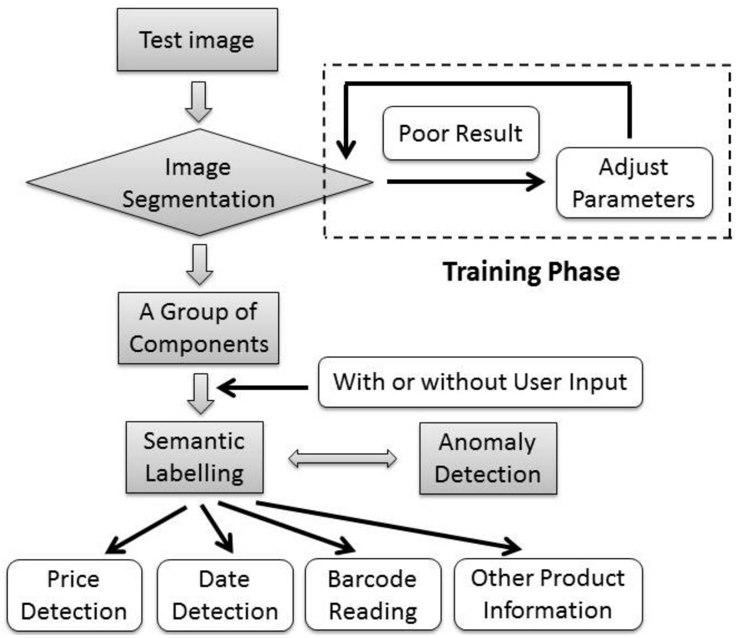

Figure 2: The workflow of the proposed system.

ing the seal region on a tray, which means that we cannot further measure minimum/maximum width of seal as requested in Fig. 1. To overcome this shortcoming, we develop a method by combing a graph-based background segmentation and a superpixel-based over-segmentation. The graph-based background segmentation is actually done at a very coarse scale where image details in and around the seal region are not significantly considered. In this way, it is implemented rapidly. The superpixel-based over-segmentation cares more about local image details so that the seal region can be segmented more accurately.

In the following, we describe the proposed method to detect seal region step by step.

1. First, we coarsely partition an input image into foreground and background using the graph-based segmentation method [2] where the input image is downscaled by a factor of 0.1 before running the graph-based segmentation and rescaled to a larger size S after the segmentation. This process is illustrated in Fig. 3. Please note that in all image resizing/rescaling operations involved in this work, we use nearest-neighbour interpolation.

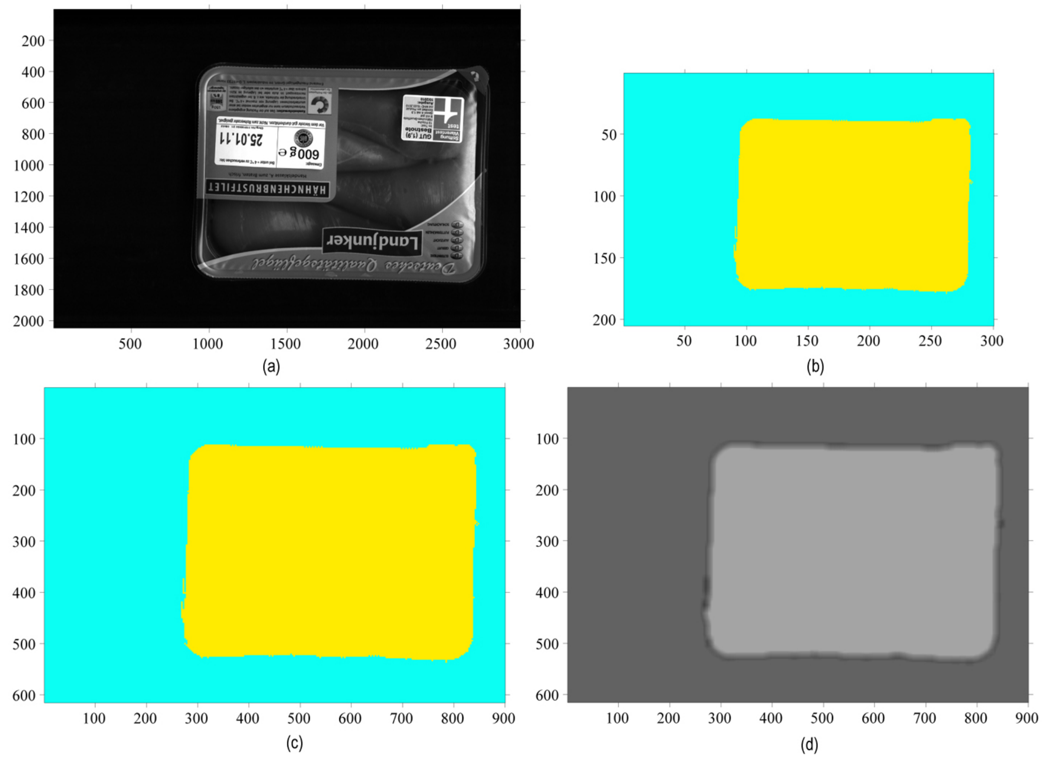

Figure 3: (a) The original input image (b) The binary result of background segmentation (c) The result of rescaling the background segmentation (d) Average filtering result of the rescaled background segmentation

2. We then introduce a 2D average filter. The result of applying such a filter to the binary background segmentation is an image with a blurred boundary as shown in Fig. 3 (d).
3. We rescale the original input image to size S and implement the SLIC algorithm [1] for producing superpixels on the rescaled image. The starting size of the superpixels should be set at roughly the same magnitude of the seal width. The result of superpixel over-segmentation is shown in Fig. 4 (a).
4. We detect the seal region by classifying the superpixels across the blurred boundary.
5. We finally rescale the seal segmentation to the size of the original input image as shown in Fig. 4 (b).

As we demonstrated in the last report, semantically meaningful tick boxes

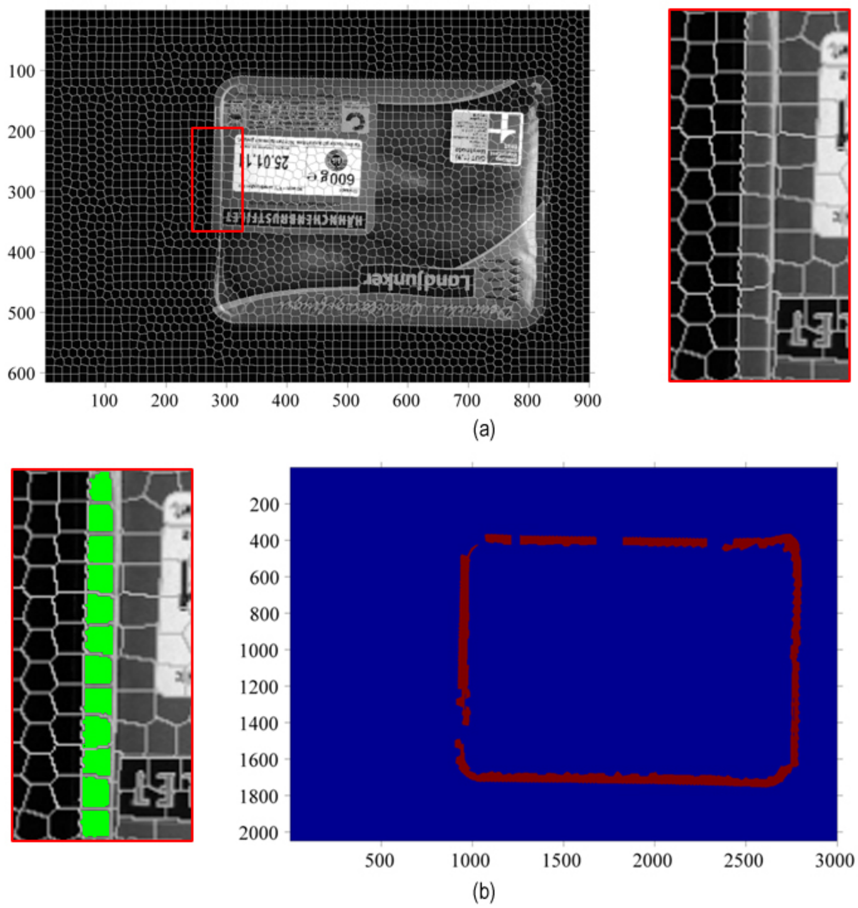

Figure 4: Seal region detection via superpixel classification. (a) Superpixel over-segmentation (b) Seal detection

can be directly detected by graph-based method. In the work reported here, we take a further step towards a complete system for semantic segmentation and recognition. In order to detect and recognise the date/price tick box, we analyse the intensity information of each component for an automated and intelligent understanding of what it is. For example, the mean intensity value of a tick box should be much higher than that of the background region. Thus finally, we integrate segmentation with recognition. And the result of our method is shown in Fig. 5.

We have implemented the proposed method on 25 images offered by Ishida Europe Ltd. On the average, the time for processing one image is roughly

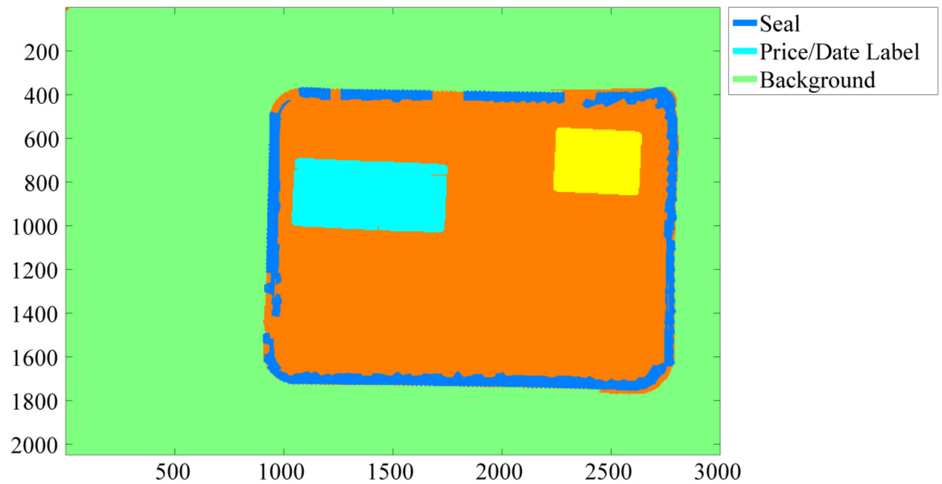

Figure 5: The final output of the proposed method where the input is the image shown in Fig. 3 (a). The entire processing is fully automatic.

11 seconds.

3. Future work

Here we raise several issues for future work.

Firstly, future development can potentially focus on the acceleration of the method. Apart from software (i.e., converting the current mixture of MATLAB and C++ codes to C++ codes only) or hardware ways (i.e., using GPU and CUDA-based programming), more interestingly, one algorithmic way is to reduce the image regions that we input to the SLIC algorithm. On one hand, in our experiments we found that SLIC cannot be implemented rapidly even through C++ on a CPU when the input image is of high resolution and it typically costs more than half of the total execution time. Note that images of high resolution are always desired from a quality control point of view and sometimes even necessary due to the requirement of character recognition (e.g., barcode detection, price detection, date detection, etc). On the other hand, we notice that in the tray images, the superpixel over-segmenation of some regions is actually not necessary. For instance, the over-segmentation in the background region is always pointless in any case. Also, if we just need SLIC for segmenting seal region, obviously a large image region inside the tray can be ignored. In short, we may first define a (potentially as small as possible) region of interest (ROI) for the SLIC algorithm and then merely implement it in the ROI. Nonetheless, the detection of ROI need to be automatic and ideally it should consist of a collection of rectangular image regions because the initialisation of SLIC requires a regular grid.

Secondly, to more accurately detect and recognise each component in an image, training is definitely needed. Training could be very important for the fine segmentation of a high-resolution image. Once the user selects a specific collection of pixels (e.g., superpixels) and labels them, the system should be able to vectorise the intrinsic features of the selected superpixels and improve current segmentation/classification based on them. For example, the user can select a collection of pixels in the seal region and compute their features such as intensity, colour, gradient, etc. If some pixels automatically labelled as 'seal' by the system have very inconsistent features, the segmentation system (essentially a classifier) can be trained to relabel such pixels. Besides such kind of online interactive training, offline training is also very useful for parameter settings. This will significantly ease the manual burden for fine tuning the controlling parameters of the algorithm and reliably produce good segmentation results.

Thirdly, the ground truth data are always required for both scientific evaluation and the aforementioned offline training. Therefore, in the future, we also plan to construct some ground truth data by manually labelling the tray images.

Finally, we have tested the seal detection method on colour images produced by a low-cost webcam and observed improved results compared with using corresponding grey images as the input. Hence, future work will contain the exploration of the utility of colour images.

References

[1] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, S. Susstrunk, Slic superpixels compared to state-of-the-art superpixel methods, IEEE Transactions on Pattern Analysis and Machine Intelligence 34 (2012) 2274–2281.

[2] P.F. Felzenszwalb, D.P. Huttenlocher, Efficient graph-based image segmentation, International Journal of Computer Vision 59 (2004) 167–181.