Proceedings of the 2017 IEEE
International Conference on Robotics and Biomimetics
December 5-8, 2017, Macau SAR, China

Learning to Detect Small Impurities with Superpixel Proposals

Yue Guo, Yijia He
School of Computer and Control Engineering,
University of Chinese Academy of Sciences
Institute of Automation,
Chinese Academy of Sciences
Beijing, China
Email: {guoyue2013, heyijia2013}@ia.ac.cn

Haitao Song and Kui Yuan
Institute of Automation,
Chinese Academy of Sciences
Beijing, China
Email:{haitao.song, kui.yuan}@ia.ac.cn

***Abstract*—In this paper, we introduce a simplified end-to-end framework for impurity detection in opaque glass bottles with liquor that learns to directly distinguish between small impurities and backgrounds. Despite promising results using convolutional neural networks in various vision tasks, few works have provided specific solutions under inadequate exposures and large background fluctuations. Two contributions are made for this problem. Firstly, we have built a feasible detection system with a cascade hardware structure, and each FPGA provides a host computer with 12 images which are most confident for containing potential impurities respectively. Secondly, most previous convolutional network architectures generally work in large-scale notable object detection benchmarks, however, such networks cannot transfer well when detecting small objects in gray images. Therefore, we propose a superpixel proposal generation method for image augmentation and a fast convolutional network with an overlapped grid structure to detect small impurities, and experiments show that our binary detection results are comparable with human checkers.**

I. INTRODUCTION

Liquid exists everywhere in our lives, ranging from common drinks to medical drugs. Qualities of liquid products are closely related to people's lives, and many types of equipment on market can merely detect impurities in transparent containers, but greatly increasing variations of drinks leave consumers too much choice in the real world, and generally opaque bottles with beautiful patterns have higher liquid quality than those with poor packages. Therefore, opaque containers seem more commonly used than transparent bottles for special purposes. In this paper, we focus on impurity detection for liquor in opaque glass bottles.

A. Impurity Detection System

An impurity detection system contains several different components. To improve the system reliability, a two-stage workflow with cascade hardware architecture is proposed, as shown in Fig. 1. Real-time frames preprocessed by FPGAs are sent with Ethernet to the host computer which then receives three types of information. Specifically, each frame typically includes a gently filtered gray image, a processed binary image and a shared text file containing the results of proposal features among 12 images.

To avoid interference from stains on the outer bottom bottle surface, an actuator is designed to stop abruptly after spinning the bottle with liquor for a while. In this case, impurities still continue moving with inertia, and we can discuss this problem into three situations: as for a transparent bottle with a patternless wall or bottom, impurities can be observed directly through its wall or bottom; as for a transparent bottle with printings on its wall or bottom, impurities may still be captured through its wall or bottom by outside cameras with an intense light; as for an opaque glass bottle with carved and coated patterns, impurities can only be viewed from an inside camera because an intense illuminant outside the bottle cannot light up the inner part of such a bottle. After choosing several light locations, we found that the light located under the bottle bottom produces relatively better imaging performance. Intuitively, the final occasion is the generalization of previous two, so the impurity detection method for the third task should be transferred easily to those for other tasks.

B. Impurity Detection Problem

Over the years, some traditional computer vision researchers have made great contributions on designing general feature descriptors such as SIFT [1], HOG [2], and Bag of Features [3], and they have utilized them to discriminate common objects such as faces, people, and cars in large numbers of scenes. Large-scale image benchmarks such as PASCAL-VOC, Imagenet [4], and COCO [5] provide incredible platforms for computer vision researchers to evaluate their solutions and variants, developments of graphical cards and deep learning [6] [7] [8] [9] also accelerate computer vision applications, so impurity detection problems may also be solved using current research works.

Impurity detection performances in an opaque glass bottle depend on several influence factors including nonuniform light exposures, bottom fragment motions of the carved pattern, impurity-background discriminative feature selection and impurity detection. Specifically, nonuniform exposures result from various bottom thicknesses and colors of the bottle; fragment motions of the bottom carved pattern are the local shifts of partial designs caused by tilts and fluctuations of

978-1-5386-3742-5/17/$31.00 © 2017 IEEE

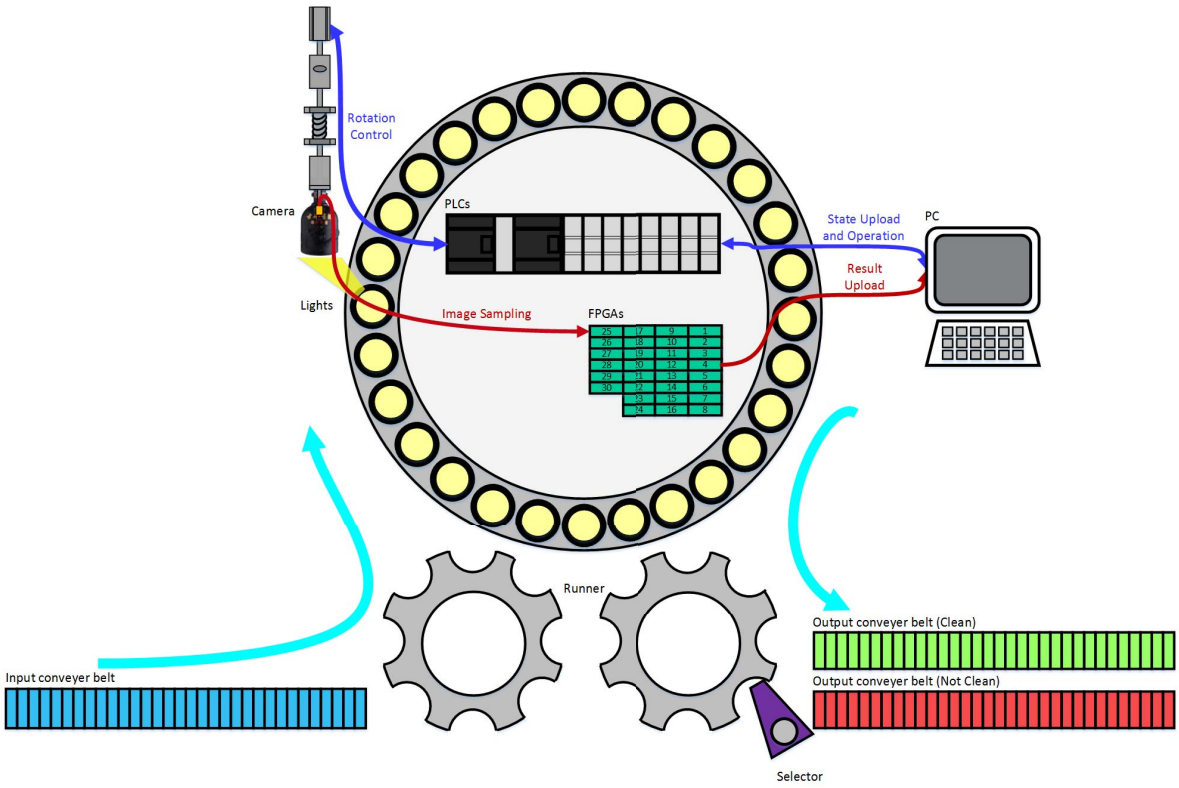

Fig. 1. System architecture for the impurity detection: hardware workflow can be depicted using the sequential movement of a liquor product. Unchecked products move one by one on the input conveyer belt, then the runner pushes them into a big revolving stage which includes around 30 area light sources whose areas are the same as that of a bottle bottom. Once a liquor product moves onto the stage, an actuator will put its head down to catch, rotate and abruptly stop rotating the bottle. After running a round, the bottle moves down from the stage, and the other runner pushes it into the output conveyer belt. If impurities are detected, then a selector pushes the corresponding bottle onto the "not clean" output conveyer belt, otherwise, the selector pushes it onto the other "clean" belt. Software workflow can be described with the cascade transmission of each frame. Once an actuator catches the bottle tightly and stops swing it, a camera on the actuator head begins to sample subsequent frames and sends them to a FPGA processor, then the processor detects impurities with high recall rate, and offers the most confident detection results and corresponding images to a host computer to make a final decision.

liquor level under intense light conditions; performances of discriminative feature selections and detections mainly depend on imaging qualities and detection frameworks.

Conventional region features are not suitable for two intuitive reasons. Firstly, impurities like tiny grains are negligible compared with others such as insects, so even moderate blob erosions should be inhibited after image binarization. Meanwhile, ranges of gray pixels inside impurity regions remain uncertain, making it hard to differentiate with spots caused by water waves. Secondly, labeling regions with binary values inside ground truth bounding boxes is substantially difficult even for human experts, because after binarization, many object blobs are broken up into different local regions including large cognizable parts and small boundaries with few valuable features. Human checkers must decide whether impurity fragments in various sizes are positive or not, mainly according to impurity locations and relationships among nearby parts instead of their individual shape features. Specific experiments using binary region shapes are detailed in Appendix A.

The rest of this paper is organized as follows: firstly, proposal labeling and generation based on superpixels are briefly introduced in Section II; secondly, a convolutional neural network for bucket segmentation is introduced in Section III, and drawbacks using the whole image are qualitatively analyzed, then a grid detection framework using a convolutional network and a superpixel image augmentation method for small impurity detection are demonstrated; experimental results are evaluated in Section IV; finally, conclusions and future works are given in Section V.

II. Superpixel Proposal Labeling and Generation

Gray image labeling is decomposed into general and task-oriented branches. In other words, we follow the procedure of making typical computer vision benchmarks using bounding box annotations and label each output image proposals suitable for our cascade hardware structure.

To sample large data from small-scale datasets and bridge the gap between tiny objects and large ones, we use a feasible method especially useful for tiny object augmentation with SLIC (Simple Linear Iterative Clustering) [13], which is initially used to generate superpixels, then we extract the contours and their corresponding minimum circumscribed rectangles of these output superpixels, as illustrated in Fig. 3.

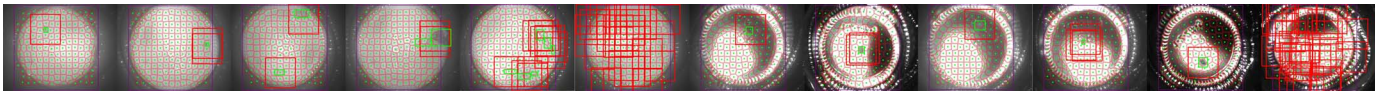

Fig. 2. Image proposal generation: Green rectangles represent ground truth bounding boxes, and red ones are extracted proposals. Specifically, background proposals are randomly sampled in bottles the sixth and twelveth images, and image proposals suitable for small objects are extracted with superpixels in the remaining images.

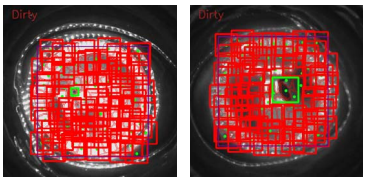

Fig. 3. Superpixel proposal generation: if we observe contours of superpixels (red line, and green dots represent their centers), when impurities are cropped with the minimum circumscribed rectangles, the small point remains complete, and the butterfly is not cut into pieces. Besides, impurities are not located in image patches of backgrounds (red rectangles) when a proper IoU between a minimum circumscribed rectangle and the ground truth bounding box is set.

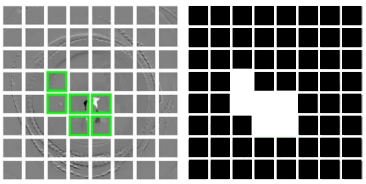

Fig. 4. A mask from a differential image: when a bucket intersects with a region labeled as impurity (denoted by a green bounding box), its mask is equal to 255, otherwise it becomes 0; then the binary mask image is resized until its size is the same as that of the original image. Then, when the number of buckets is small, combining pixels with bilinear interpolation is preferable than nearest-neighbor one, because the nearest-neighbor interpolating method may cause some mask pixels corresponding to impurity targets disappear. Finally, the mask is renormalized into $[0, 1]$.

Positive bounding boxes and negative ones can be generated with these superpixel rectangles. Specifically, positive bounding boxes can be obtained with proposals of superpixels near the ground truth bounding boxes, and when IoU between a minimum circumscribed rectangle and any ground truth bounding box in the same image is larger than 0.7, then it represents a nearby superpixel bounding box. Randomly collected rectangles in each image without impurities are used as negative bounding boxes. Besides, patches are sampled in much larger size for data augmentation, as shown in Fig. 2.

III. Applications

A. Bucket Segmentation

Experiment reports from Deep Sense suggest transferring a bounding box regression problem (see Appendix B) into a bucket classification problem. Moreover, to better understand effects of object sizes, a segmentation experiment is taken instead of a classification one. Since a single gray patch inside a fix-size grid is called a bucket, grid patches in a whole gray image are divided and labeled, as shown in Fig. 4.

U-net [11] is modified as our network which inputs a 60×60 gray image and outputs a grid heatmap. U-net mainly contains a convolutional encoder and a decoder. When the level of an encoder layer equals that of a decoder one, pixel-wise addition is used between the feature map of the decoder layer and that of the encoder one.

Dice Coefficient is also used as the network training criterion. Qualitative experiment results demonstrate that large impurities are easier to be detected, and few fluctuations are observed, and impurity locations tend to approach their ground truth ones, as shown in Fig. 5. High-resolution image inputs can solve this well, but its high computational and memory cost may limit our application.

B. Data Augmentation

Before image proposal augmentation, we randomly split the whole sample set into parts for training and test, and augment image proposals for each part.

During the sampling period, we are more likely to sample more images when impurities in liquor cannot be clearly observed, and it makes categories of the new dataset extremely unbalanced, so the numbers needed for balance is shown in Table. I, where operations include translation, rotation, scaling, brightness changing, and contrast variation. Specifically, "Sample" attribute lists the total number of samples a per class. Given the annotation $\times m[\times n]$, the sample numbers will be increased to $a \times (m + 1)$ and $a \times (n + 1)$ for balancing multiple classes and binary categories respectively.

C. Detection in Overlapped Grids

Instead of typical R-CNN frameworks [14] [15] or SLIC to connect classification and detection tasks, an overlapped grid structure is a relatively faster alternative for online detections (18 fps with a GTX 1080). Because the overlap ratio between two neighboring proposals in a grid structure is constantly 0.5, and under the prior assumption that small objects cannot be easily extracted (see Fig. 5), selecting small objects in a relatively large bounding box is much more computationally effective, comparing with computing forward a region proposal network in the R-CNN framework or other heuristic selection methods [16] [17]. However, SLIC is still necessary for training, because when augmenting image patches with the grid structure, tiny impurities may sometimes be cut by the edges of fixed nonoverlapping grid bounding boxes, and large impurities may be cut into different parts (see Fig. 4), which brings more noises in positive training samples, and a

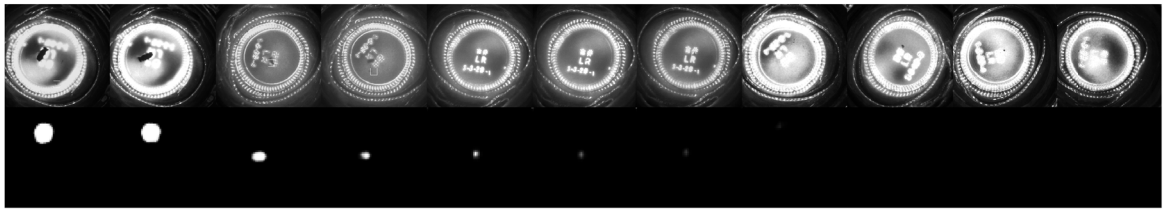

Fig. 5. U-net segmention results given a complete gray images containing various size of impurities: tiny impurites cannot be extracted, because the loss function considering the whole image range changes little even when predicted masks for tiny impurities are all zeros.

little higher computation costs are also acceptable for offline training.

IV. Experimental Results

Impurities in different bottles are independently sampled, and humans cannot label all kinds of impurities under poor imaging conditions, so impurity types may be different in various bottle colors.

A network with six convolutional layers is constructed in our task, and its input is a 60×60 gray image patch, and the classifier predicts one-hot outputs using softmax regression.

Multiple impurity classifications in the same bottle color are trained and evaluated independently, because bottles in different colors can easily be distinguished before detection, and in practice, customers usually need products of the same bottle color at the same time. Impurities and backgrounds in various bottle colors also have different data distributions. Emperically, backgrounds in red bottles should not be detected in those with other colors such as gray and black. As a result, fibers and splashes are easily identified impurities, while white patches are the most difficult ones to be detected in red bottles; performances of white points and white patches are acceptable in gray bottles, but results of backgrounds and plugs are quite poor; points in black bottles are comparable with humans in Fig. 7.

Binary classification can achieve acceptable performances, but simultaneously predicting the type of an impurity and its corresponding bottle color using one single model might be difficult when gray images are used, as shown in Fig. 7. Since our only goal is to distinguish impurities from background fluctuations in a fixed bottle color, so impurity categories may not be necessarily required. Performances of some samples using binary classification models in fixed bottle colors are displayed in Fig. 8.

V. Conclusion

In this paper, we apply convolutional networks to the impurity detection in opaque glass bottles with liquor. A feasible data augmentation method based on superpixels is introduced to mitigate the difficulties using deep learning in small-scale datasets, especially for small object detection. Also, instead of using conventional deep learning detection frameworks in public benchmarks, a customized convolutional network is constructed and evaluated in our practical tasks, and a small object detection framework with the overlapped grid structure significantly reduces memory and computational costs comparing with general purpose end-to-end convolutional networks for object detection, but it may still need more model compression for parallel detection. Finally, performance evaluations in different impurity types remind us of further improving proposal sampling methods and imaging conditions in our future work, and to improve the speed of transmission and that of detection, sampled images at reasonable time steps may further be replaced with corresponding image proposals filtered in FPGAs.

VI. Acknowledgement

This work is supported by National Natural Science Foundation (NNSF) of China under Grant 61421004.

Appendix

A. Region Feature Classification

Classical features are considered at first, and we use these features as the inputs of support vector machines for binary classification (impurity or background).

Locations and region features of image proposals are obtained to detect dark impurities. Specifically, gray images are binarized with an empirical threshold set as 110 to remove relatively lighter pixels at first, and we get a binary image mask, then differential images are obtained from two adjacent frames and multiplied with the binary mask to get masked differential images. Secondly, renormalize and calculate the median pixel values from previous output images. Thirdly, another threshold set as the medium divided by 1.5 to exclude light pixels (slowly moving blobs).

Areas, central locations, lengths of major and minor axes are extracted as region features and are normalized with the normal distribution to eliminate the range differences among features, and then they are split into sets for training and test $(0.7 : 0.3)$. A support vector machine with polynomial kernels inputs region features and classifies impurities or backgrounds. As a result, the detection performance using conventional features has many false negative ground truth labels and predictions, as shown in Fig. 9.

B. Direct Bounding Box Regression

Changes of gray pixel values at the current time step relative to the last time step are obtained after frames differencing. Although the original image size is 480×480, we resize it into a 240×240 image. Then a convolutional network is

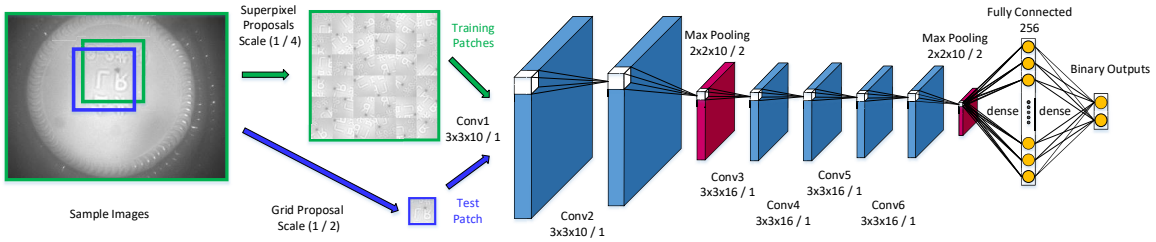

Fig. 6. Convolutional neural network architectures: generally larger models with larger-scale datasets tend to perform better than others, especially when some model regularization technologies such as Dropout [12] are used, which enable convolutional network models to be considered as ensembles of different subnetworks and generalize better in several tasks. Then classification experiments in three bottle colors are taken, and transformation from detection tasks to classification ones is achieved with the overlapped grid structure.

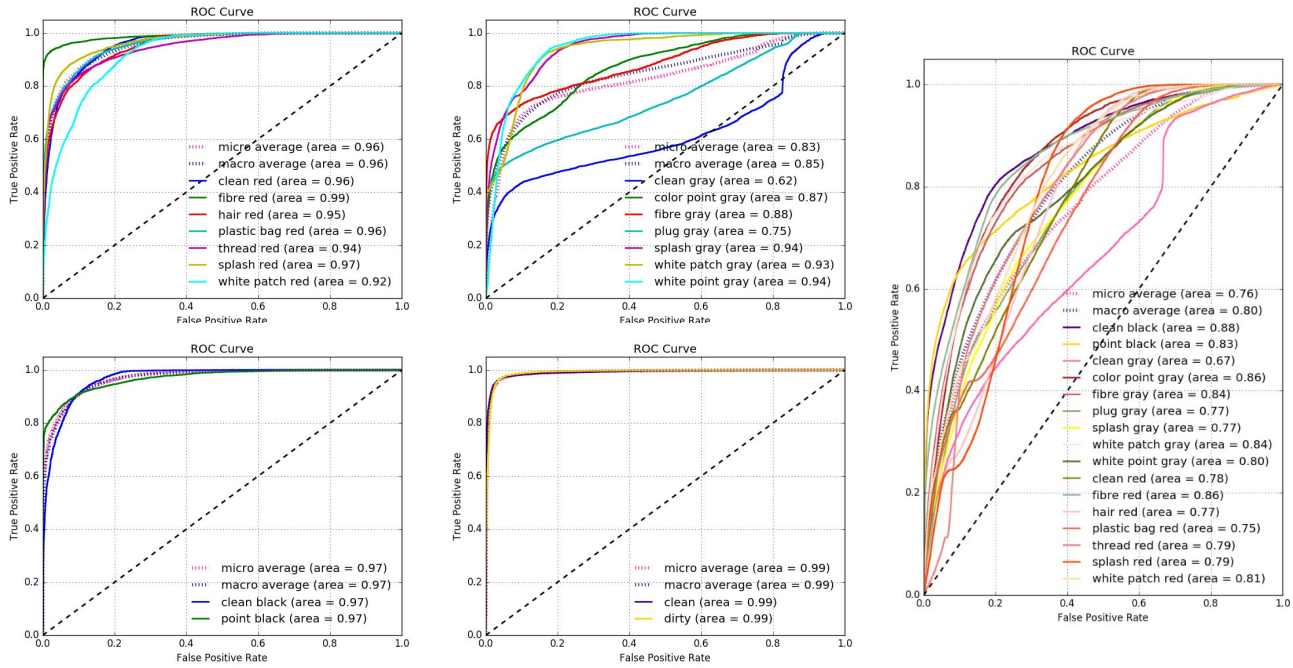

Fig. 7. Receiver Operating Characteristic curves in three types of classification rules (from top to bottom, from left to right): multi-class performances in red bottles, black bottles, and gray bottles, binary classifications in all bottles, multi-class performances in all bottles.

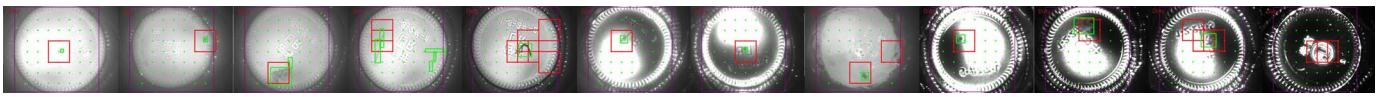

Fig. 8. Detection results in all bottles (red bounding boxes represent the predicted impurity proposals, while green ones contain ground truth impurities): the first five images belong to red bottles, but detectors sometimes misclassify backgrounds as fibres; the next six images are in gray bottles; the last image corresponds to a black bottle, which is difficult for both labeling and detection. Performances in some impurity classes are quite well though not all ROC curves in gray bottles are acceptable.

designed to predict the bounding box features including central location (x, y), width w, and height h. Unfortunately, both root mean square errors for the training set and the test set are unacceptable (35.57 and 40.08 respectively).

Bounding boxes are abstract concepts including both locations and classes. Specifically, low-level features are only sensitive to locations, while higher-level ones are good at learning semantic contents. However, the brute force regression network merely uses top-level features to localize targets.

References

[1] Lowe D G, Object recognition from local scale-invariant features, IEEE International Conference on Computer Vision, 1999, pp. 1150-1157.

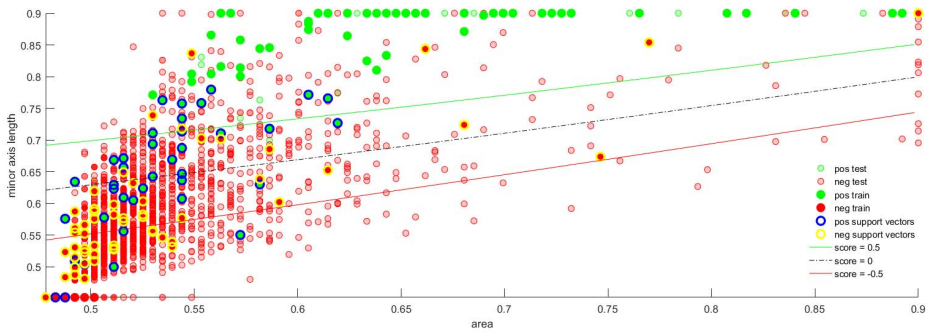

Fig. 9. Scatterplot of two features (area and minor axis length): decision boundaries of the support vector machine cannot discriminate impurity features and background ones with high confidence.

TABLE I
DATASET BALANCE AND AUGMENTATION

Impurity	Bottle	Sample	Augment Ratio	Scale	Angle	Translated Width	Translated Height	Bright	Contrast
clean	black	4260	$\times 3[\times 3]$	$\times 1[\times 1]$	$\times 3[\times 3]$	$\times 1[\times 1]$	$\times 1[\times 1]$	$\times 1[\times 1]$	$\times 1[\times 1]$
clean	gray	14490	$\times 1[\times 1]$	$\times 1[\times 1]$	$\times 1[\times 1]$	$\times 1[\times 1]$	$\times 1[\times 1]$	$\times 1[\times 1]$	$\times 1[\times 1]$
clean	red	2370	$\times 6[\times 6]$	$\times 1[\times 1]$	$\times 3[\times 3]$	$\times 1[\times 1]$	$\times 2[\times 2]$	$\times 1[\times 1]$	$\times 1[\times 1]$
color point	gray	224	$\times 64[\times 16]$	$\times 2[\times 2]$	$\times 8[\times 2]$	$\times 2[\times 2]$	$\times 2[\times 2]$	$\times 1[\times 1]$	$\times 1[\times 1]$
fibre	gray	266	$\times 45[\times 16]$	$\times 1[\times 2]$	$\times 5[\times 2]$	$\times 3[\times 2]$	$\times 3[\times 2]$	$\times 1[\times 1]$	$\times 1[\times 1]$
fibre	red	133	$\times 128[\times 32]$	$\times 2[\times 2]$	$\times 8[\times 2]$	$\times 2[\times 2]$	$\times 2[\times 2]$	$\times 1[\times 1]$	$\times 2[\times 2]$
hair	black	0	$\times 0[\times 0]$	$\times 0[\times 0]$	$\times 0[\times 0]$	$\times 0[\times 0]$	$\times 0[\times 0]$	$\times 0[\times 0]$	$\times 0[\times 0]$
hair	gray	0	$\times 0[\times 0]$	$\times 0[\times 0]$	$\times 0[\times 0]$	$\times 0[\times 0]$	$\times 0[\times 0]$	$\times 0[\times 0]$	$\times 0[\times 0]$
hair	red	418	$\times 36[\times 12]$	$\times 1[\times 1]$	$\times 9[\times 3]$	$\times 2[\times 2]$	$\times 2[\times 2]$	$\times 1[\times 1]$	$\times 1[\times 1]$
plastic bag	red	527	$\times 32[\times 9]$	$\times 1[\times 3]$	$\times 4[\times 3]$	$\times 2[\times 1]$	$\times 2[\times 1]$	$\times 1[\times 1]$	$\times 2[\times 1]$
plug	black	0	$\times 0[\times 0]$	$\times 0[\times 0]$	$\times 0[\times 0]$	$\times 0[\times 0]$	$\times 0[\times 0]$	$\times 0[\times 0]$	$\times 0[\times 0]$
plug	gray	297	$\times 64[\times 16]$	$\times 2[\times 2]$	$\times 8[\times 2]$	$\times 2[\times 2]$	$\times 2[\times 2]$	$\times 1[\times 1]$	$\times 1[\times 1]$
point	black	95	$\times 160[\times 36]$	$\times 2[\times 3]$	$\times 10[\times 3]$	$\times 2[\times 2]$	$\times 2[\times 2]$	$\times 1[\times 1]$	$\times 2[\times 1]$
splash	gray	275	$\times 64[\times 16]$	$\times 1[\times 2]$	$\times 9[\times 2]$	$\times 2[\times 2]$	$\times 2[\times 2]$	$\times 1[\times 1]$	$\times 1[\times 1]$
splash	red	341	$\times 36[\times 16]$	$\times 1[\times 2]$	$\times 8[\times 2]$	$\times 2[\times 2]$	$\times 2[\times 2]$	$\times 1[\times 1]$	$\times 2[\times 1]$
thread	black	0	$\times 0[\times 0]$	$\times 0[\times 0]$	$\times 0[\times 0]$	$\times 0[\times 0]$	$\times 0[\times 0]$	$\times 0[\times 0]$	$\times 0[\times 0]$
thread	red	520	$\times 32[\times 8]$	$\times 1[\times 2]$	$\times 8[\times 1]$	$\times 2[\times 2]$	$\times 2[\times 2]$	$\times 1[\times 1]$	$\times 1[\times 1]$
white patch	black	0	$\times 0[\times 0]$	$\times 0[\times 0]$	$\times 0[\times 0]$	$\times 0[\times 0]$	$\times 0[\times 0]$	$\times 0[\times 0]$	$\times 0[\times 0]$
white patch	gray	92	$\times 128[\times 48]$	$\times 2[\times 2]$	$\times 8[\times 3]$	$\times 2[\times 2]$	$\times 2[\times 2]$	$\times 1[\times 1]$	$\times 2[\times 2]$
white patch	red	209	$\times 72[\times 16]$	$\times 1[\times 2]$	$\times 9[\times 2]$	$\times 2[\times 2]$	$\times 2[\times 2]$	$\times 1[\times 1]$	$\times 2[\times 1]$
white point	gray	57	$\times 288[\times 72]$	$\times 2[\times 3]$	$\times 8[\times 3]$	$\times 3[\times 2]$	$\times 3[\times 2]$	$\times 1[\times 1]$	$\times 2[\times 2]$

[2] Dalal N, Triggs B, Histograms of oriented gradients for human detection, IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005, pp. 886-893.

[3] Lazebnik, Svetlana, Schmid, Cordelia, Ponce, Jean, Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories, IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2006, pp. 2169-2178.

[4] Deng J, Dong W, Socher R, ImageNet: A large-scale hierarchical image database, IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 248-255.

[5] Lin T Y, Maire M, Belongie S, Microsoft COCO: Common Objects in Context, 2014, 8693, pp. 740-755.

[6] Lecun Y, Bengio Y, Hinton G, Deep learning, Nature, 2015, 521(7553), pp. 436-444.

[7] Krizhevsky A, Sutskever I, Hinton G E, Imagenet classification with deep convolutional neural networks, Advances in Neural Information Processing Systems, 2012, pp. 1097-1105.

[8] Szegedy C, Liu W, Jia Y, Going deeper with convolutions, IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 1-9.

[9] Simonyan K, Zisserman A, Very deep convolutional networks for large-scale image recognition, arXiv preprint arXiv:1409.1556, 2014.

[10] Krizhevsky A, Sutskever I, Hinton G E, ImageNet classification with deep convolutional neural networks, International Conference on Neural Information Processing Systems. Curran Associates Inc, 2012, pp. 1097-1105.

[11] Ronneberger O, Fischer P, Brox T, U-Net: Convolutional Networks for Biomedical Image Segmentation, Medical Image Computing and Computer-Assisted Intervention, Springer International Publishing, 2015, pp. 234-241.

[12] Srivastava N, Hinton G, Krizhevsky A, Dropout: a simple way to prevent neural networks from overfitting, Journal of Machine Learning Research, 2014, 15(1), pp. 1929-1958.

[13] Achanta R, Shaji A, Smith K, SLIC Superpixels Compared to State-of-the-Art Superpixel Methods, IEEE Transactions on Pattern Analysis and Machine Intelligence, 2012, 34(11), pp. 2274.

[14] Girshick R, Donahue J, Darrell T, Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation, International Conference on Computer Vision and Pattern Recognition, 2013, pp. 580-587.

[15] Girshick R, Fast R-CNN, IEEE International Conference on Computer Vision, 2015, pp. 1440-1448.

[16] Najibi M, Rastegari M, Davis L S, G-CNN: An Iterative Grid Based Object Detector, Computer Science, 2015.

[17] Uijlings J R, Sande K E, Gevers T, Selective Search for Object Recognition, International Journal of Computer Vision, 2013, 104(2), pp. 154-171.