EXPLOITING SPARSITY FOR IMAGE-BASED OBJECT SURFACE ANOMALY DETECTION

Woon Huei Chai, Shen-Shyang Ho^{*}, and Chi-Keong Goh[†]

Rolls-Royce @ Nanyang Technological University Corporate Lab, Singapore Energy Research Institute, Interdisciplinary Graduate School, Nanyang Technological University, Singapore * School of Computer Engineering, Nanyang Technological University, Singapore † Rolls-Royce Singapore Pte Ltd, Singapore

ABSTRACT

The anomaly detection task plays an important role in quality control in many industrial or manufacturing processes. However, in many such processes, anomaly detection is done visually by human experts who have in-depth knowledge and vast experience on a product in order to perform well in the detection task. In this paper, we present an approach that (i) identifies anomalies in an image based on the sparse residuals (or errors) obtained during image reconstruction using sparse representation and (ii) learns the threshold to classify an image pixel based on its residual value. The intuitions for our proposed sparse approximation driven approach are, namely: (i) anomalies are infrequent and (ii) anomalies are unwanted portions of an image reconstruction. Empirical results on a real-world image dataset for an industrial surface defect detection task are used to demonstrate the feasibility of our proposed approach.

Index Terms— Parameter optimization, sparse reconstruction, anomaly detection, defect detection, quality control and automated inspection

1. INTRODUCTION

Anomaly detection is the task of finding abnormality in data that do not conform to the expected patterns. Its real-world applications include intrusion detection [1, 2], fraud detection [3], abnormal event detection from surveillance video [4, 5] and many others [6]. Traditionally, anomaly detection for product quality control is done by human experts who are trained and have experience to identify anomalies. However, anomaly detection by human is not efficient and the results are very subjective. Hence, product inspection by human should be replaced or complemented by objective automated inspection [7, 8, 9, 10].

One main characteristic of image-based inspection process is that images are taken from multiple fixed viewpoints of a three-dimensional object of interest. These images are then used for anomaly detection. This is to ensure that the object surface is exhaustively analyzed. One challenging characteristic of surface anomalies is that they come in different forms, shapes, and sizes. Figure 1 shows image patches of different anomalies that have significant different characteristics on a metallic surface. Besides different viewpoints, there could also be variation in the illumination (or light intensity) in the images.



Fig. 1: Examples of anomalies on metallic surfaces. (Left to Right): Melt, Melt, Shadowing, Plus Metal.

Sparsity enforcement for approximation (or coding) has been widely used in different fields and problems in estimating sparse high dimensional vectors. The earliest work on sparse approximation reconstructs signals with a preconstructed overcomplete dictionary (or basis set) [11]. Previous work that exploited sparsity property for anomaly detection utilized the reconstruction cost [12] and localized sparsity [13]. These approaches are applied to abnormal event detection in videos. To improve the accuracy of sparse approximation and to reduce redundancy in a dictionary, the dictionary could be learned from a training dataset [14]. Two recent works closely related to this paper utilize sparse approximation for background subtraction in images [15] and for face recognition when occlusion or corruption occurs [16].

In this paper, we present an approach that (i) identifies anomalies in an image based on the sparse residuals (or errors) obtained from a reconstruction using sparse representation and (ii) learns the threshold to classify an image pixel from its residual value. Residuals are the deviation of the observed feature values (e.g., grayscale values of pixels in an image) from the reconstructed (or estimated) values. Figure 2 shows the flow diagram of our proposed approach.

The rest of the paper is organized as follows. In Section 2, we describe and discuss our proposed anomaly detection approach and parameter optimization in detail. Section 3 presents detailed experimental result on an industrial problem related to anomaly detection for quality control to show the



Fig. 2: Flow diagram of anomaly detection using sparse image reconstruction. In the results, white and black pixels represent the estimated anomalous and estimated normal pixels.

feasibility and performance of our proposed method.

2. METHODOLOGY

2.1. Sparse Reconstruction

The objective of sparse reconstruction is to utilize a few atoms (or words) in a dictionary to reconstruct an input data (e.g., an image) with high similarity (See Figure 2). The main motivation for using sparse approximation to detect anomalies is as follows: An image that does not contain anomalies is ideally reconstructed by sparse approximation with zero residual when the dictionary consists of atoms that describe normal situations. On the other hand, an image that contains anomalies is represented by a sparse representation together with sparse residual.

A *p*-dimensional data vector $x = (x_1, x_2, ..., x_p)$ can be decomposed into the form $x = D \cdot \alpha$ where *D* is a $m \times p$ underdetermined matrix (m >> p) and $\alpha = (\alpha_1, \alpha_2, ..., \alpha_m)$ is a *m*-dimensional vector. *D* is called a dictionary (or design matrix) with *m* atoms (or words) of size *p*. If α is sparse, the decomposition is called sparse approximation (or decomposition) and α can be estimated by solving the following minimization problem:

$$\min_{\alpha, e} \|\alpha\|_1 + \|e\|_1$$
(1)
s.t. $x = D\alpha + \lambda' e$

The two l_1 norm functions in the minimization problem are used to learn sparse α and e vectors, e is a p-dimensional vector which represents the errors and λ' controls the tradeoff between the sparsities of α and the errors¹.

2.2. Anomaly Detection

After identifying the likely regions R_j , j = 1, ..., l that contain anomalies, we locate the anomalies using a simple pixel-

based threshold classifier

$$C(x_i) = \begin{cases} 1, |e_i| - T > 0\\ 0, |e_i| - T \le 0 \end{cases}$$
(2)

where pixel $x_i \in R_j$ and $C(x_i)$ returns 1 when the pixel is predicted to be an anomaly based on the absolute residual value $|e_i|$ obtained from (1) and a predefined threshold T. $C(x_i)$ returns 0 when the pixel is predicted to be normal. The main challenge is to select the most discriminative threshold, T.

Although the anomaly detection task is formulated as a binary classification task (See Figure 2), it is similar to the task of one-class classification with outliers. Hence, we learn the pixel-based threshold classifier (i.e., T) using a simple variation of the one class support vector machine [17] as follows.

$$\min_{T} T + \frac{1}{\eta} \sum_{i,k} \xi_{i,k}$$
(3)
s.t. $T - |e_{i,k}| \ge -\xi_{i,k}, i = 1...m, k = 1...p$
 $\xi_{i,k} \ge 0, T \ge 0$

where m is the total number of training images, $\xi_{i,k}$ are slack variables corresponding to the residual $e_{i,k}$ for pixel k in image i. η controls the tradeoff between the threshold and the total sum of slack variables.

2.3. Finding Best λ' , η and T

To achieve a competitive detection performance, one needs to find the best λ' , η , and T using the training images whose pixels are marked either as abnormal or normal. Algorithm 1 shows the procedure to obtain these parameters. Given a dictionary D, the inputs to the algorithm consist of the set of labeled training images, I_t , the stopping criterion S for the procedure, and the search granularity N that affects the computational time of the algorithm.

Step 1 initializes λ' and η and the variables required to compute the stopping criterion, i.e., the difference between the two best accuracies in one iteration. p is the total number of pixels for each image in I_t . The accuracy computed in Algorithm 1 for the stopping criterion is the balanced accuracy described in Section 3.2. When the stopping criterion is not satisfied, the algorithm repeats steps 3 to 27, the search procedure to find the best parameters.

For steps 3, maxAccp, λ'_p and η_p store the best accuracy and the λ' and η obtained during the previous iteration of the outer while loop. Minimum and maximum of λ' domain are also initiated for the while loop from steps 4 to 12. From steps 5 to 11, we narrow down the possible λ' domain that gives the best λ' by looking for the best accuracy (based on I_t) at a fixed η based on a search in the λ' domain depending on the granularity N. Step 7 finds the likely anomalous regions based on the sparse residual in the sparse approximation (1) for a given $\lambda'[i]$. Step 8 finds threshold T[i] using (3) using

¹Note that in this paper, *errors* has similar definition as *residuals* defined in Section 1.

the fixed η . Step 9 performs classification on the pixels of the images in I_t based on the pixel-based threshold classifier (2) using T[i]. Then, the balanced accuracy acc1[i] (see (4)) is computed (similarly for acc2[i] in step 19).

For steps 11, the indices of the λ' vector and their corresponding highest and second highest accuracies are found and stored. The best λ' to be used for steps 15 to 21 and the smaller λ' domain to be searched in the next iteration of the inner while loop (steps 4 to 12) when the stopping criterion in step 4 is not satisfied are stored too.

Steps 15 to 21 follow similar procedure as steps 5 to 11 except that we now search for the best η . Step 23 stores the best accuracy maxAcc achieved in the iteration of the outer while loop. It is clear the highest accuracy is obtained in the procedure from steps 14 to 22 since this procedure also uses the best parameters obtained from steps 4 to 11 for anomaly detection. It also stores the threshold used to construct the best accuracy maxAcc obtained from this iteration is not better than the previous iteration, revert λ' and η to the values in the previous iteration and the optimization ends.

Algorithm 1: Finding Best λ' , η and T				
Input : <i>I_t</i> , set of labeled training images; stopping				
criterion. S: search granularity. N				
Output: λ'_{1} , n_{2} and T_{2}				
1: $\eta_0 = \frac{p}{2}; \lambda'_0 = 0; maxAccp = 1; maxAcc = 0; acc1a =$				
1; acc1b = 0; acc2a = 1; acc2b = 0				
2: while $ maxAcc - maxAccp > S$ do				
3: $maxAccp = maxAcc; \lambda'_p = \lambda'_o; \eta_p = \eta_o; \lambda'_{min} = 0.001;$				
$\lambda'_{max} = 1000$				
4: while $\ acc1a - acc1b\ > S$ do				
5: for $i = 0$ to N do				
6: $\lambda'[i] = \lambda'_{min} + \frac{i}{N} (\lambda'_{max} - \lambda'_{min})$				
7: Find likely anomaly regions using (1) for $\lambda'[i]$				
8: Find $T[i]$ using (3) given η_0				
9: Apply (2) to obtain $acc1[i]$ using $T[i]$ on I_t				
10: end for				
11: $k1 = \arg \max_i(acc1); k2 = \arg \max_i(acc1 \setminus acc1[k1]);$				
$acc1a = acc1[k1]; acc1b = acc1[k2]; \lambda'_o = \lambda'[k1];$				
$\lambda'_{min} = \min(\lambda'[k1], \lambda'[k2]); \lambda'_{max} = \max(\lambda'[k1], \lambda'[k2])$				
12: end while				
13: $\eta_{min} = 1; \eta_{max} = p$				
14: while $ acc2a - acc2b > S$ do				
15: for $j = 0$ to N do				
16: Find likely anomaly regions using (1) for λ'_o				
17: $\eta[j] = \eta_{min} + \frac{j}{N}(\eta_{max} - \eta_{min})$				
18: Find $T[j]$ using (3) for $\eta[j]$				
19: Apply (2) to obtain $acc2[j]$ using $T[j]$ on I_t				
20: end for				
21: $k3 = \arg\max_j(acc2); k4 = \arg\max_j(acc2 \setminus acc2[k3]);$				
$acc2a = acc2[k3]; acc2b = acc2[k4]; \eta_o = \eta[k3];$				
$\eta_{min} = \min(\eta[k3], \eta[k4]); \eta_{max} = \max(\eta[k3], \eta[k4])$				
22: end while				
23: $maxAcc = acc2[k3]; T_o = T[k3]$				
24: if $maxAccp > maxAcc$ then				
25: $maxAcc = maxAccp; \lambda'_o = \lambda'_p; \eta_o = \eta_p$				
26: break				
27: end if				
28: end while				

Note that the best λ' and T for the anomaly detection approach is highly dependent on the dictionary. Hence, Algorithm 1 needs to be rerun after the dictionary is changed. However, note that the focus of this paper is not on the learning of dictionary. In Section 3, we demonstrate and compare the performance of our approach on different fixed and learned dictionaries.

Algorithm 1 can be considered as a two-step greedy search first on λ' and then on η . The training accuracy, in general, is concave with respect to λ' and η . However, it is non-smooth locally. Hence, we utilize a greedy approach to find a near-optimal λ' and η that achieve good accuracy.

3. EXPERIMENTAL RESULTS

3.1. Metallic Object Dataset

The dataset used in our experiment is a subset of large number of images collected from an automated visual inspection process for quality control in metallic object manufacturing². The anomalies that we are interested in identifying are defects on the metallic surface. The dataset consists of 641 images with a resolution of 2448×2050 pixels. 368 images do not contain any defect and 273 images contain one of the three defects: "melt", "plus metal", and "shadowing". Each image is taken from one of the six viewpoints of the object of interest. Some defects have complicated illumination dependent characteristics. For example, "plus metal" can only be visible in dark condition. These defects are labeled as the anomalies in the images for our experiment. Each image with anomaly has a ground truth image for performance evaluation purposes. In the ground truth image, a pixel can be either a defect (abnormal) pixel or a defect-free (normal) pixel.

3.2. Experimental Design

Due to high computational cost of high resolutions, images are downsampled using bicubic interpolation. The pixel values which are larger than zero in the downsampled ground truth images are labeled as anomalous pixels. All 368 images containing defect-free objects are used to construct the dictionaries. For comparison purposes, three dictionaries are used in our experiment, namely: (i) normalized images, (ii) dictionary learned using the online robust dictionary learning (ORDL) [18], and (iii) dictionary learning using scale adaptive dictionary learning (SADL) [19].

For performance evaluation, the balanced accuracy [20]

$$Acc_b = \frac{1}{2} \left(\frac{TP}{P} + \frac{TN}{N} \right) \tag{4}$$

is used. TP and TN are the number of true positive pixels and true negative pixels, respectively. P and N are the num-

²Due to confidentiality, the description is limited.

bers of positive and negative test pixels, respectively. Acc_b is the average between the sensitivity and specificity measures.

3.3. Empirical Results and Discussions

3.3.1. Performance of different dictionaries at fixed λ'

Figure 3 shows receiver operating characteristic (ROC) curves for three dictionaries stated in subsection 3.2 and three λ' values: 0.001, 1, and 10. The image resolution is 30×30 after downsampling. To learn a dictionary using ORDL, the number of images in a batch and the number of atoms are set to 200 and 368 (the number of images without defect), respectively. To learn a dictionary using SADL, the maximum number of atoms in the dictionary is set to 500. Note that the dictionary learned using SADL has only 14 atoms. The pixels of the 273 images with anomaly are used to construct the ROC curves. The true and false positive rates in the ROC curves are computed by varying the thresholds from the lowest to the highest absolute residual value.



Fig. 3: ROC curves of three dictionaries at different λ' : (top left) normal dictionary; (top right) ORDL dictionary; (bottom left) SADL dictionary; (bottom right) comparison of ROC curves for all 3 dictionaries.

Figure 3 (bottom right) shows a comparison of the best anomaly detection performance using the three dictionaries for sparse approximation. One interesting observation is that the best performance using the learned dictionary of only 14 atoms from SADL is comparable to using a dictionary with 368 atoms. Online approach of learning dictionary may result in lost in critical information. As a result, it does not perform as well as the other two dictionaries.

3.3.2. Performance of different dictionaries using proposed parameter learning method

The images containing anomalies are split into five sets to perform the five-fold cross-validation for the proposed parameter learning method (Algorithm 1). Table 1 shows the average balanced accuracy, Acc_b when the near-optimal λ' and learned T from Algorithm 1 are used to identify the anomalies in the test images with image resolution of 15×15 . It shows that Algorithm 1 performs the best when all the images containing defect-free objects are used in the dictionary.

Dictionary	Normalized images	ORDL	SADL
Average Acc_b	0.69	0.63	0.63

Table 1: Comparison of (balanced) accuracies of different dictionaries with parameter optimization at 15×15 pixel resolution.

3.3.3. Effect of downsampled resolution on performance.

Table 2 shows the average Acc_b when parameter learning (Algorithm 1) with dictionary containing images of defect-free object at different image resolutions are used. It should not be surprising to observe that higher resolution images result in better Acc_b . However, higher resolution images mean higher computational cost since some iterative steps in Algorithm 1 requires handling the images in the dictionary.

Resolution	5×5	10×10	15×15	20×20
(pixels)				
Average Acc_b	0.56	0.65	0.69	0.70

 Table 2:
 Comparison of (balanced) accuracies of dictionary containing images of all images of defect-free object with parameter optimization at various image resolutions.

4. CONCLUSIONS

We present an approach that (i) identifies anomalies in an image based on the sparse residual (or error) obtained from an image reconstruction using sparse representation and (ii) learn the threshold to classify an image pixel from the sparse residual. The intuitions for our proposed sparse approximation driven approach are, namely: (i) *anomalies are infrequent* and (ii) *anomalies are unwanted portions of an image reconstruction*. Empirical results on an image dataset for surface defect detection obtained from an industrial setting demonstrate the feasibility of our proposed approach.

5. ACKNOWLEDGMENTS

This work was conducted within the Rolls-Royce@NTU Corporate Lab with support from the National Research Foundation (NRF) Singapore under the Corp Lab@University Scheme and Energy Research Institute@NTU under Interdisciplinary Graduate School in Nanyang Technological University.

6. REFERENCES

- Aleksandar Lazarevic, Levent Ertöz, Vipin Kumar, Aysel Ozgur, and Jaideep Srivastava, "A comparative study of anomaly detection schemes in network intrusion detection.," in SDM. SIAM, 2003, pp. 25–36.
- [2] Jingwei Huang, Z. Kalbarczyk, and D.M. Nicol, "Knowledge discovery from big data for intrusion detection using Ida," in *Big Data (BigData Congress)*, 2014 IEEE International Congress on, June 2014, pp. 760–761.
- [3] Richard J Bolton and David J Hand, "Statistical fraud detection: A review," *Statistical science*, pp. 235–249, 2002.
- [4] Tao Xiang and Shaogang Gong, "Video behavior profiling for anomaly detection," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 30, no. 5, pp. 893–908, 2008.
- [5] Vijay Mahadevan, Weixin Li, Viral Bhalodia, and Nuno Vasconcelos, "Anomaly detection in crowded scenes," in *Computer Vision and Pattern Recognition (CVPR)*, 2010 IEEE Conference on. IEEE, 2010, pp. 1975–1981.
- [6] Varun Chandola, Arindam Banerjee, and Vipin Kumar, "Anomaly detection: A survey," ACM Computing Surveys (CSUR), vol. 41, no. 3, pp. 15, 2009.
- [7] Roland T Chin and Charles A Harlow, "Automated visual inspection: A survey," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, , no. 6, pp. 557–573, 1982.
- [8] Roland T Chin, "Automated visual inspection: 1981 to 1987," *Computer Vision, Graphics, and Image Processing*, vol. 41, no. 3, pp. 346–381, 1988.
- [9] Timothy S Newman and Anil K Jain, "A survey of automated visual inspection," *Computer vision and image understanding*, vol. 61, no. 2, pp. 231–262, 1995.
- [10] S Ravikumar, KI Ramachandran, and V Sugumaran, "Machine learning approach for automated visual inspection of machine components," *Expert Systems with Applications*, vol. 38, no. 4, pp. 3260–3266, 2011.
- [11] Bruno A Olshausen and David J Field, "Sparse coding with an overcomplete basis set: A strategy employed by v1?," *Vision research*, vol. 37, no. 23, pp. 3311–3325, 1997.
- [12] Yang Cong, Junsong Yuan, and Ji Liu, "Sparse reconstruction cost for abnormal event detection," in *Computer Vision and Pattern Recognition (CVPR)*, 2011 IEEE Conference on. IEEE, 2011, pp. 3449–3456.

- [13] Huamin Ren and Thomas B Moeslund, "Abnormal event detection using local sparse representation," in Advanced Video and Signal Based Surveillance (AVSS), 2014 11th IEEE International Conference on. IEEE, 2014, pp. 125–130.
- [14] Michael Elad and Michal Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *Image Processing, IEEE Transactions on*, vol. 15, no. 12, pp. 3736–3745, 2006.
- [15] Zhao Cong, Wang Xiaogang, and Cham Wai-Kuen, "Background subtraction via robust dictionary learning," *EURASIP Journal on Image and Video Processing*, vol. 2011, 2011.
- [16] John Wright, Allen Y Yang, Arvind Ganesh, Shankar S Sastry, and Yi Ma, "Robust face recognition via sparse representation," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 31, no. 2, pp. 210– 227, 2009.
- [17] Bernhard Schölkopf, Robert C Williamson, Alex J Smola, John Shawe-Taylor, and John C Platt, "Support vector method for novelty detection.," in *NIPS*, 1999, vol. 12, pp. 582–588.
- [18] Cewu Lu, Jiaping Shi, and Jiaya Jia, "Online robust dictionary learning," in *Computer Vision and Pattern Recognition (CVPR)*, 2013 IEEE Conference on. IEEE, 2013, pp. 415–422.
- [19] Cewu Lu, Jianping Shi, and Jiaya Jia, "Scale adaptive dictionary learning," *Image Processing, IEEE Transactions on*, vol. 23, no. 2, pp. 837–847, 2014.
- [20] Kay Henning Brodersen, Cheng Soon Ong, Klaas Enno Stephan, and Joachim M Buhmann, "The balanced accuracy and its posterior distribution," in *Pattern Recognition (ICPR), 2010 20th International Conference on*. IEEE, 2010, pp. 3121–3124.