Global Journal of Science Frontier Research: I
Interdisciplinary
Volume 20 Issue 3 Version 1.0 Year 2020
Type: Double Blind Peer Reviewed International Research Journal
Publisher: Global Journals
Online ISSN: 2249-4626 & Print ISSN: 0975-5896

Surface Defect Detection and Root Cause Analysis

By Tianchen Liu, Fan Zhu, Haoran Yu & Haisong Gu

Abstract- Artificial Intelligence has played an increasingly important role in surface defect detection in recent years. At the same time, there are many challenges using deep learning for this area, such as the detection accuracy, shortage of data and, lack of knowledge of root cause of defects. To solve the problem of data shortage, we propose a taxonomy method called Dataonomy^{TM}to extend a meta defect datasets with a small number of samples for training defect classifiers. For the accuracy, we apply two latest deep neural network(DNN) architectures, Inception v3 and fully convolutional networks (FCN) so as not only to classify whether there are defects but also to make a pixel-wise prediction to inference the areas of defects. For those detected defects, we combine DNN with traditional AI methods to find root causes of detected defects. We use a generalized multi-image matting algorithm to extract common defects automatically. We apply this technology to identify defects that stem from systematic errors in the surface operation. Experimental results have shown great capability and versatility of our proposed methods.

Keywords: artificial intelligence, deep neural networks(DNN), Dataonomy^{TM}, defect detection, root cause finding.

GJSFR-I Classification: FOR Code: 080199

Surface Defect Detection and Root Cause Analysis

Tianchen Liu [α], Fan Zhu [σ], Haoran Yu [ρ] & Haisong Gu [ω]

Abstract- Artificial Intelligence has played an increasingly important role in surface defect detection in recent years. At the same time, there are many challenges using deep learning for this area, such as the detection accuracy, shortage of data and, lack of knowledge of root cause of defects. To solve the problem of data shortage, we propose a taxonomy method called Dataonomy™to extend a meta defect datasets with a small number of samples for training defect classifiers. For the accuracy, we apply two latest deep neural network(DNN) architectures, Inception v3 and fully convolutional networks (FCN) so as not only to classify whether there are defects but also to make a pixel-wise prediction to inference the areas of defects. For those detected defects, we combine DNN with traditional AI methods to find root causes of detected defects. We use a generalized multi-image matting algorithm to extract common defects automatically. We apply this technology to identify defects that stem from systematic errors in the surface operation. Experimental results have shown great capability and versatility of our proposed methods.

Keywords: artificial intelligence, deep neural networks(DNN), Dataonomy™, defect detection, root cause finding.

I. Introduction

Visual inspection is a common task not only across the industry, but also across the world. In industry, in order to improve the quality of products and reduce the cost, machine vision has been used for a long time. Visual inspection consists of three major tasks: defect detection, presence detection, and measurement. However, defect detection is still a challenging problem due to a large variety of shapes and patterns among products from different industries, and even different assembly lines of the same product. Recently, machine learning based approach has shown great potential in solving complex problems and has proven to be successful in a variety of applications. For example, using artificial intelligence method[1] for asphalt pavement pothole detection based on least squares support vector machine and neural network with steerable filter-based feature extraction. Another example[2]was to use GLCM to extract texture features for surface quality detection for steel sheet. Recently Deepneural network(DNN) has been applied to solve surface defect detection problem in various fields: automobile parts, car surface, etc. In the field of visual inspection, several works [3][4][5] were proposed using DNN or Deep Learning(DL) based approach to classify and detect the defects. One of the biggest challenges for applying DNN based approach to the industry is the lack of data samples. In practice, a common approach [6] to address this problem is to use transfer learning, in which a pre-trained model, such as VGG and Inception V3, is chosen and then retrained on the target dataset by keeping the model architecture and parameter weights of the lower layers constant and only updating the upper layers of the neural network. However, it is difficult to get a large number of training samples from a certain field or industry, for instance, images of defects on the surfaces of a specific type of ceramic product. Therefore, in this paper we propose a novel approach named Dataonomy™, which can be used to train the classifier for a specific task across the industry with relatively small data samples. Different from the method of adding a number of geometric transformations to the original image data to enlarge the number of samples in the training dataset, Dataonomy™ aims at quantifying the relationships between different datasets and extracting a structure out of them. The "structure" means a collection of relations specifying which dataset provides useful information to another, and by how much.

For the accuracy, we apply two latest deep neural network(DNN) architectures, Inception v3 and fully convolutional networks (FCN) so as not only to classify whether there are defects but also to make a pixel-wise prediction to inference the areas of defects. Both architectures have decent accuracies to find defects.

Another issue in the surface visual inspection field is that besides the basic defect detection tasks, few researches have dealt with root cause analysis for the detected image defects. In [7], the author proposed a knowledge-driven diagnosis approach when defect generation mechanism is known. Basically, there are two main kinds of root causes: systematic error and random error. Systematic error such as mechanic operation error will cause the same defect at the same position for each product. This kind of error does huge damage to the whole batch of products. In this paper, we will focus on finding out defects caused by systematic error.

The major contributions in this research consists of

1) Dataonomy™method to solve the sample datashortage
2) A novel AI approach to detect the surface detects with the combination of two latest DNNs

Author α σ ρ Ѡ: VisionX LLC, Cupertino, CA.
e-mails: andrew.liu@visionx.org, fanzhu@visionx.org, haoranyu@visionx.org, haisong.gu@visionx.org

3) A generalized multi-image matting algorithm is applied to a root-cause analysis from surface defects

The rest of the paper is structured as follows. In Section 2, we provide a description of our Dataonomy^{TM}. Section 3 gives a description of our deep learning-based framework for image defect detection. The setup and results of the experiments will be presented in Section 4. Conclusions will be discussed in Section 5.

II. Dataonomy^{TM}

The patent pending Dataonomy^{TM} algorithm [3][8] is a fully computational method for quantifying data class relationships and extracting a structure out of them. The following steps give the idea of the whole pipeline, and the framework of our approach is shown in Figure 1.

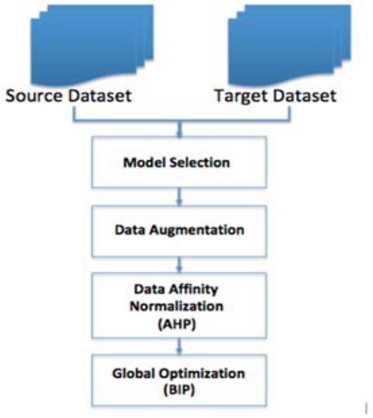


Figure 1: Dataonomy TM Pipeline

a) Make use of a pre-trained model for object classification, for example Inception V3 [9].
b) Find affinity matrix across the dataset.
c) Get normalized data augmentation affinities using AHP (Analytic Hierarchy Process) [10].
d) Find global mapping taxonomy using BIP (Binary Integer Programming) [11].

The Dataonomy^{TM} algorithm will pull the information from an ever-increasing pool of data to develop a highly specialized solution for new customers. Once the data of a company is added to the pool, the model can be fine-tuned to exceed 99.97% accuracy.

III. Framework for Image Defect Detection

In this section, we present the framework ofour proposed method for visual defect inspection. First is by using Inception V3, and second is by using FCN.

a) Inception V3

As shown in Figure 2 (a), the framework of deep learning based visual defect classification and detection by Inception V3 consists of three components. The first component is the base model training, the second component is transfer learning for visual defect classification, and the third component is defect segmentation.

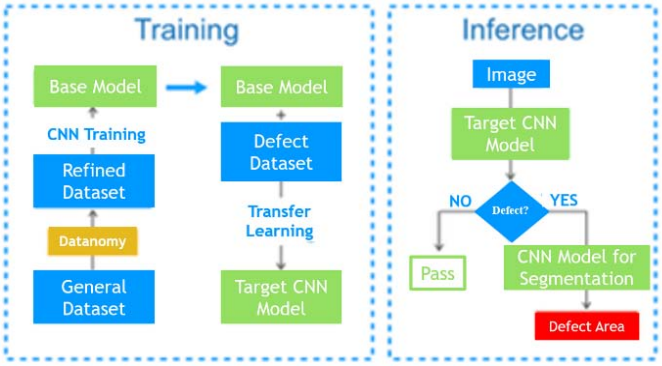


Figure 2: (a) Framework of Our Proposed Approach

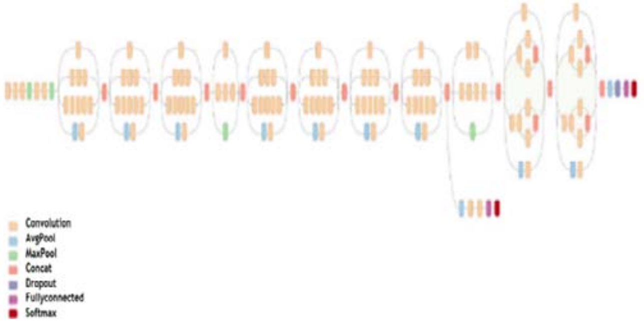


Figure 2: (b) Inception V3

i. *Training of Base Model*

In order to obtain the specific model for visual defect classification, the selection of the base model is important, and the way to train the base model is also crucial. These two factors would impact the overall performance of the base model and thereafter. During the base model training, we utilize the aforementioned Dataonomy^{TM} approach to prepare more useful and representative datasets related to our tasks. Then deep convolution neural network is applied with state-of-the-art model architectures. Specifically, we introduced the InceptionV3 [9] network, which has been widely used in image recognition and has shown promising performance on various datasets, as shown in Figure 2 (b). This network is made up of several inception modules which contain convolutions, pooling, concatenations, and fully connected layers. The original inception module was designed by stacking filters with multiple sizes in the same level of the network, which enables multiple receptive fields of each filter and, in turn, can extract features in multiple scales. In order to reduce the computational cost, within an inception module, 1x1 convolution layers were added to limit the number of input channels. In the Inception V3 network, the computational cost was further reduced by factorizing convolutional layers within the inception module, where an NxN convolutional layer was decomposed into one 1xN convolutional layer followed by an Nx1 convolutional layer. Lastly, batch normalization was added to auxiliary layers to improve the performance.

Given the InceptionV3 network structure, we modify the fully connected layers to fit the number of classes from the dataset generated by our proposed Dataonomy^{TM} approach. Then augmented data are collected into batches and feeding into the network for training. The Stochastic Gradient Descent (SGD) with momentum is applied for the training procedure. The whole training is set to stop when the network converged after a number of epochs. The trained weights are stored as our base model and would be used in the next steps for transfer learning.

ii. *Training of Defect Classifier*

The next step of our proposed framework is to train the visual defect model. The transfer learning scheme is applied in this step by utilizing our pertained models on the dataset that is generated by Dataonomy^{TM}. Particularly, the pre-trained Inception V3 model is used as the starting point for the model on the visual defect classification task. This transfer learning approach is considered to be effective since our base model is trained on a large corpus of photos with a large number of classes. It enables the model to efficiently learn to extract features from these images in order to perform well on a specific problem. Moreover, the model is pretrained on the dataset selected through Dataonomy^{TM}, which chooses sample images that have certain features that are more closely related to the classification task of defect inspection. This approach can further boost the capability of the base model to differentiate visual defects. During the training, we use the full model without freezing any layers, and only the last fully connected layer is modified to fit the two-class classification problem in defect inspection tasks. Hyperparameters such as the initial learning rate are modified, and more details are presented in experiments.

iii. *Defect Area Detection*

After the above steps, our model is capable of detecting the visual defects given an input image. Inspired by [5], we further propose a segmentation approach, fully convolutional networks, as shown in Figure 2 (c), for pixel-wise defect detection so that the defect area can be accurately located in the image. There are two components included in this stage, patch extraction, and model training. We crop patches in original images, and each patch as a training image. We label the patch whose defect area exceeds the threshold 0.6 as a defect, vice versa. The ratio of training defect patches and non-defect patches is 2:1. For the dataset of DAGM-2007, the size of the patch is 64*64 pixels with the stride of 64 pixels. When do testing, the whole image is the input image. In the FCN architecture, there are four convolutional layers as feature extractors followed by batch normalization and Relu, and two pooling layers. A deconvolutional layer is inserted before the score layer to maintain the resolution of the feature map for classification.

b) FCN

CNN has shown great quality and efficiency in different tasks. In order to take full advantage of CNN on surface defect inspection, we need to make predictions on every pixel. And that's the reason why we choose Fully Convolutional Networks (FCN) as our base model, which has been demonstrated to outperform other approaches in image segmentation. In this case, the networks can thus be trained end-to-end, pixel-to-pixel.

The other reason that FCN as our top choice is that FCN has the property to allow an arbitrary size of the image as the input of the networks. This property facilitates the processing of images in different sizes.

According to the paper "Fully Convolutional Networks for Surface Defect Inspection in Industrial Environment"[12], we use the method two stages method for base models.

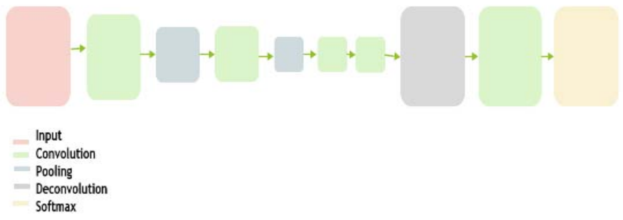


Figure 3: FCN

i. *Stage 1 -- Coarse Segmentation of Defect Area*

This stage is giving a quick and coarse inference of the defect area, which is also called the region of interest (ROI). The predicted ROIs would be the initialization of stage 2 in order to limit the search range of stage 2. The final goal is to improve inspection efficiency.

In the training phase, we cut the original image into several small patches. But in the test phase, we use the whole image as input. The receptive field should be a proper size, not too large or too small. If the receptive field is small, the network can focus on rich local spatial information rather than global object-level information. To interpret what influences the receptive field, we assume a network, the kernel size of the i-th layer (layer$i$) isK_i, s_i is the stride of layer i and S_i is the integral stride before layer i. We denote R_i as the receptive field of each neuron located on the i-th layer (noted as layer-i). Then the recurrence relation of R_i and S_i can be calculated as follows:

$$R_i = R_{i-1} + S_{i-1}(K_i - 1) \tag{1}$$

$$S_i = s_i \times S_{i-1} \tag{2}$$

It can be concluded from the recursion formula that the receptive field is influenced by K_i, s_i, and the depth of network layer-i. We pick Zeiler and Fergus's model trained for RPN in [13] as our basic architecture. We use only the first four layers as our feature extractor layer and append a scoring layer at the end of feature layers, and we change the strides of all convolutional layers from s_i to 1 (s_i is the original stride of layer i in ZF). Overlap pooling used in RPN controls model capacity and increases receptive field size, resulting in a coarse, highly-semantic feature representation. While effective and necessary for extracting object-level information, this general architecture results in low resolution feature that are invariant to pixel-level variations. This is beneficial for classification and identifying object instances but poses a challenge for pixel-labeling tasks. So, we change overlap pooling to non-overlap pooling as the former cause lager Ri in the following layers. To maintain the resolution of the feature map used for classification, we insert a deconvolutional layer [14] before the score layer. We use logistic regression as the loss function for segmentation. More details about the network structure are shown in Fig. 4.

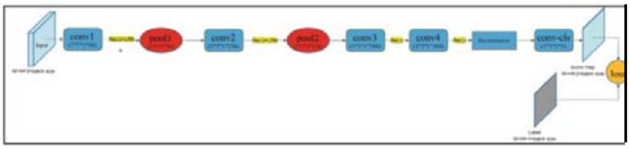


Figure 4: Structure of FCN on Stage 1

ii. *Stage 2 -- Segmentation Refinement with Instance-Sensitive Patches*

Stage 2 is to improve the result of segmentation from Stage 1 with a method of detection. The difference between stage 1 and stage 2 is that stage 1 focuses more on local information; stage 2 is a detection task to refine stage 1 with object-level information. In other words, stage 2 is detection instead of segmentation. We still use those cropped images as training data, but we do not use those manually annotated segmentation masks in the training process. We label the patches whose defect area covers over n% of the total area as the defect patches (n can be changed for different accuracy requirements, in our experiment in this paper, we design n = 40), and others as the non-defect ones. We also do not all samples of the whole image, we only do sample that around ROI from stage 1 for efficiency.

As shown in Figure 5, it's the fusion of stage 1 and stage 2. The result from stage 1, ROIs, is the initialization of stage 2. We crop the patches around ROI and do the classification, whether it is a defect or not. Then we keep the interaction of the two stages.

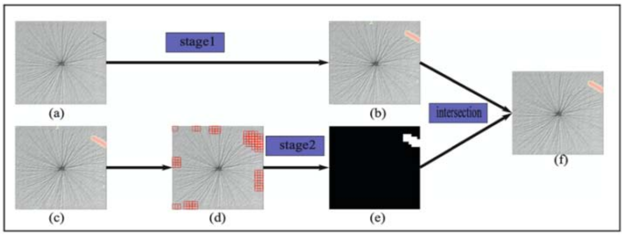


Figure 5: The Fusion of Stage 1 and Stage 2

We design a multi-loss-function in FCN to fuse information across layers to make a skip connection in order to increase the detection accuracy. All the loss function is logistic regression. As we still use FCN in a detection task, we label the patch with a label map that has the same resolution as the output layer, and its values are all the same—0 for defect patches and 1 for non-defect patches (shown in Fig. 6).

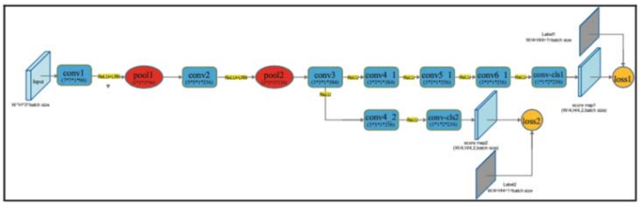


Figure 6: Multi-loss-function structure of FCN in stage 2

While inspection, we vote the score map to one single Soft Max layer and average the results from different Soft Max layers. This method can average the results of certain-size receptive fields under one patch and average the results of the different-size receptive fields under one patch, as shown in Figure 7.

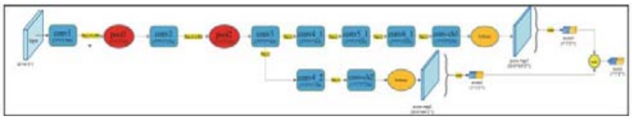


Figure 7: Illustration of inspection process

c) *Defect Cause Analysis*

There are many works dealing with defect detection. However, few of them can conduct the cause-finding automatically. We provide a way to find the root cause of common defects, which is also known as systematic error. Normally, if a systematic error exists, it will cause the same defect at the same location. The following workflow of a generalized multi-image matting algorithm shows our approach to extracting the common defect.

Assuming that a system error exists, our task is to determine if there is a common defect and what part belongs to a common defect in images. Basically, we first compute gradients at each pixel in both x and y directions for each image. Then we compute the median gradients, which are the medians of gradients obtained by a median filter, for x and y-direction independently.

Thus, we have two median gradient maps: one for x and one for y with all information from the dataset.

$$p[m,n] = median\{g_k[m,n,k \in w]\} \qquad (3)$$

Here, $p[m,n]$ is the median gradient value of a single pixel at position $[m,n]$ in either x or y direction for images within the filter window size of w. $g_k[m,n]$ is the gradient value of x or y direction in a single image at position $[m,n]$. After the experiment, we find out that window size around 30 will start to give us a good result. To further explain the filter window, imagine in a manufacturing line, every 30 or more consecutive products will be taken into analysis to get a median gradient map. We get the gradient values maps of all w images, and for each pixel, we find the median value of all *w* images at the same pixel as our median output. The reason why we use the median filter is to clear noise and speckles. As the number of images increases, the median gradient at the common defect area will be more consistent and significant than other points, because the systematic defect occurs in the same position for each image. Therefore, after computing the magnitude of the gradient for each point, we can get an output image that shows the common pattern, which normally gives the systematic error. Figure 8 is our defect analysis workflow.

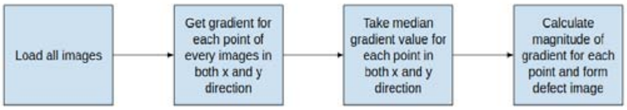

Load all images
Get gradient for each point of every images in both x and y direction
Take median gradient value for each point in both x and y direction
Calculate magnitude of gradient for each point and form defect image

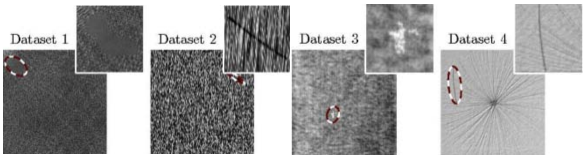
Figure 8: Defect Analysis Workflow

IV. Experiments

a) Defect Detection

i. *Datasets*

We choose the DAGM-2007 dataset [15] to evaluate the performance of our proposed framework. The dataset contains ten classes of the different defects with different textured backgrounds, even though the data is generated artificially, but similar to the real-world problems. The entire dataset consists of 8050 images for training, in which 1046 images contain defects; and 8050 images for testing, in which 1054 images contain defects. Each image in the dataset is saved in grayscale 8-bit PNG format of size 512x512. In our experiment, we split the training dataset into two parts, 80% for training and 20% for validation during the training stage. The example image for each class contained in this dataset shows in Figure 9.



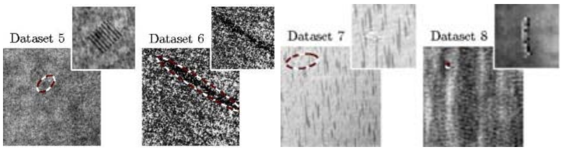


Figure 9: DAGM-2007 Dataset

ii. *Experimental Design*

In order to evaluate the effectiveness and performance of our proposed framework. We retrain the classifier for defect detection on the surface by using transfer learning. According to the accuracy on defect detection, we compare the relevant data extracted by Dataonomy^{TM} from Image Net to retrain the Inception V3 with the method in [3][4] we can prove the effectiveness of our method for data augmentation and thus showing the possibility of our method to solve the problem of the limited dataset in deep learning based tasks.

Our experiment for retraining the Inception V3 using selected data from Image Net was running on the computer with four Ge Force GTX 1080 Ti graphics cards. With the use of transfer learning, the training of classifiers for defect detection on the surface was running on the computer with two Ge Force GTX 1080 Ti graphics cards.

iii. *Experimental Results*

a. *Defect Image Detection for Texture Surface*

500 classes of data are selected from ImageNet to train the base model, and the total time for training takes around 252 hours.

With the use of transfer learning, we take the retrained Inception V3 on the selected 500 classes from Image Net as our base network. We evaluate the performance of our approach for surface defect detection in terms of the true positive rate (TPR) and true negative rate (TNR). Equation 2 and Equation 3 define TPR and TNR, respectively.

$$\mathrm{TPR} = \mathrm{TP}(\mathrm{TP} + \mathrm{FN})^{-1} \qquad (4)$$

$$\mathrm{TNR} = \mathrm{TN}(\mathrm{FP} + \mathrm{TN})^{-1} \qquad (5)$$

Table 1 shows the performance of our framework compared to the state-of-art deep learning-based approach proposed in [16] with DAGM-2007. From the table, we can see that our method outperforms the others, and therefore shows the effectiveness of our proposed framework for the deep learning-based approach with limited data samples.

Table 1: Defect detection result (%).

No.	Weimer et al. [16]		Inception V3(Ours)	
	TPR	TNR	TPR	TNR
1	100	100	100	100
2	100	97.3	100	100
3	95.5	100	98.8	100
4	100	98.7	100	100
5	98.8	100	98.8	100
6	100	99.5	97	100
7	NA	NA	100	100
8	NA	NA	96.7	100
9	NA	NA	100	100
10	NA	NA	99.3	100

In addition, we also compared the accuracy of our method of defect detection with the work in [4] and [17]. The accuracy of our method with the pre-trained base model on Wood Dataset is 99.12%, compared with the build-in Inception V3, which is 97.7%. And the average accuracy of our method on the DAGM-2007 dataset is 99.88%. It can be seen that our framework using Dataonomy^{TM} for data augmentation shows high performance on defect detection with a limited dataset compared to the state-of-the-art method.

b. *Defect Area Detection on Texture Surface*

The next step of our proposed framework is to highlight the defect area on the surface. Besides the dataset of DAGM-2007, we use the samples of the phone screen with scratches to validate our methods. Part of the result for the texture data in this step by using Inception V3 and FCN is shown in Figure 10 and Figure 11, showing a decent performance of our two methods.

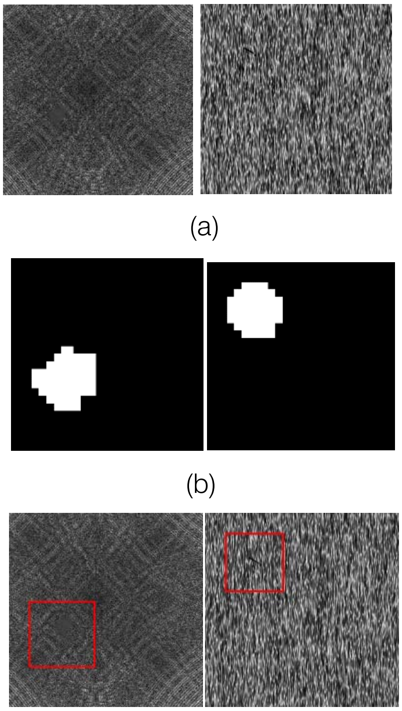


Figure 10: Inception V3: (a) DAGM-2007 Original Image (b) Mask Image (c) Highlighted Defect

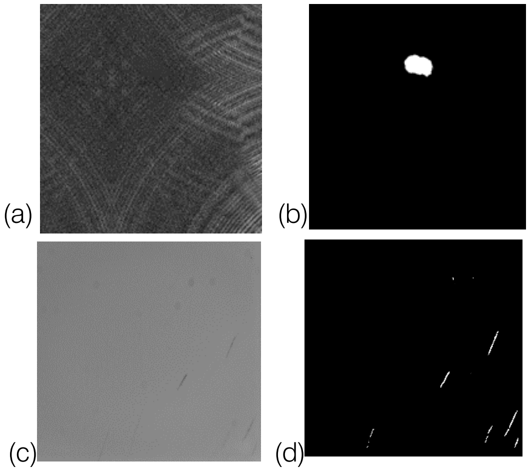


Figure 11: FCN: (a) DAGM-2007 Original Image (b)Result for DAGM-2007; (c) Phone Screen Original Image (d)Result for Phone Screen

b) *Defect Cause Analysis*

i. *Dataset for Root Cause Detection*

In order to evaluate our method, we created a new dataset for our root cause detection based on the DAGM-2007 dataset [15]. We chose all 1046 images which contain a common defect with the existing types of scratch defect independently on chosen images. In this case, we have 1046 training images for each scratch type and around 10,000 images in total. In order to simulate the systematic error, the added scratch is the same size and in the same position for each image. Figure 12 (a) and (b) are two examples with different systematic defects with the original defect from DAGM-2007.

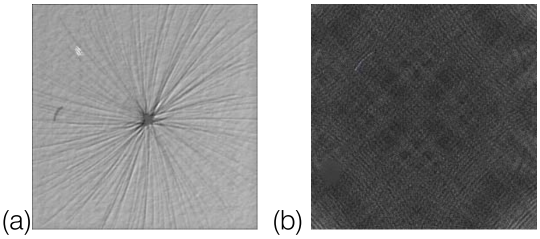


Figure 12: Example of 2 Types of Scratches

ii. *Experimental Results for Root Cause Detection*

Using the method in Sec. 3.1.3, we got common defect image for each type of scratch.

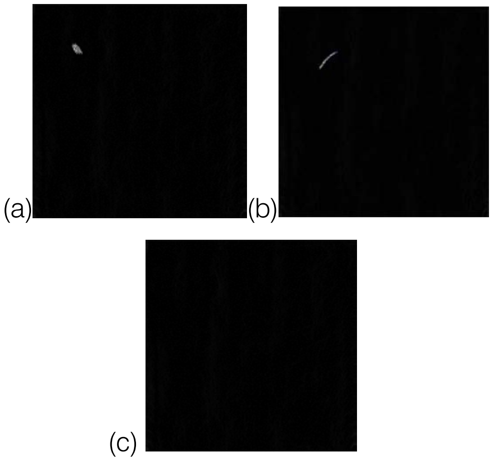


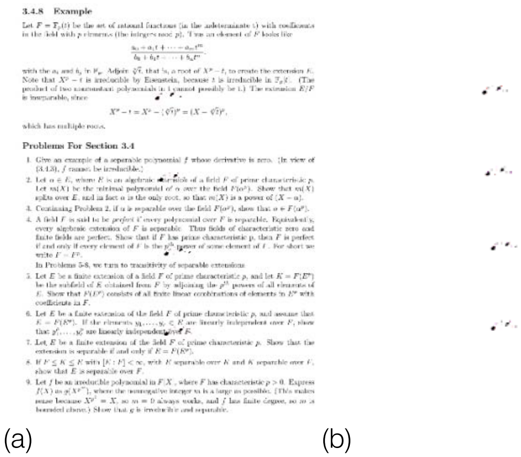
Figure 13: (a)(b) Results with Common Defect Detected; (c) Result with No Common Defect Detected

From Figure 13 (a) and (b), we can find out where the common defect is. This defect may come from the mechanical error in the product assembly line, which can cause a huge loss in production if not detected automatically.

Figure 13 (c) is the resultant image of the common defect detection for 1046 images with defects in the DAGM-2007 dataset [15]. Because there are random defects (scratches), the resultant image obtained by the generalized multi-image matting algorithm is a blank image.

This technique can also be used in other areas, such as troubleshooting in printing systems(Figure 14).We created a 500-large text defect dataset by adding the same ink defect at the same position of text images. image.

Using our root cause detection method, the resulting image Figure 14 (b) does detect those four ink defects in the original dataset (we reversed the colour for better notice). Therefore, in real life, we can know there is a problem in the printing system that causes a common defect using this method.



3.4.8 Example

Let $F = \mathbb{F}_p(t)$ be the set of rational functions (in the indeterminate t) with coefficients in the field with p elements (the integers mod p). Thus an element of F looks like

$$\frac{a_0 + a_1 t + \cdots + a_m t^m}{b_0 + b_1 t + \cdots + b_n t^n}$$

with the a_i and b_j in $\mathbb{F}_p$. Adjoin $\sqrt[p]{t}$, that is, a root of $X^p - t$, to create the extension E. Note that $X^p - t$ is irreducible by Eisenstein, because t is irreducible in $\mathbb{F}_p[t]$. (The product of two nonconstant polynomials in t cannot possibly be t.) The extension E/F is inseparable, since

$$X^p - t = X^p - (\sqrt[p]{t})^p = (X - \sqrt[p]{t})^p,$$

which has multiple roots.

Problems For Section 3.4

1. Give an example of a separable polynomial f whose derivative is zero. (In view of (3.4.3), f cannot be irreducible.)
2. Let $\alpha \in E$, where E is an algebraic extension of a field F of prime characteristic p. Let $m(X)$ be the minimal polynomial of α over the field $F(\alpha^p)$. Show that $m(X)$ splits over E, and in fact α is the only root, so that $m(X)$ is a power of $(X - \alpha)$.
3. Continuing Problem 2, if α is separable over the field $F(\alpha^p)$, show that $\alpha \in F(\alpha^p)$.
4. A field F is said to be *perfect* if every polynomial over F is separable. Equivalently, every algebraic extension of F is separable. Thus fields of characteristic zero and finite fields are perfect. Show that if F has prime characteristic p, then F is perfect if and only if every element of F is the p^{th} power of some element of F. For short we write $F = F^p$.

In Problems 5-8, we turn to transitivity of separable extensions.

5. Let E be a finite extension of a field F of prime characteristic p, and let $K = F(E^p)$ be the subfield of E obtained from F by adjoining the p^{th} powers of all elements of E. Show that $F(E^p)$ consists of all finite linear combinations of elements in E^p with coefficients in F.
6. Let E be a finite extension of the field F of prime characteristic p, and assume that $E = F(E^p)$. If the elements $y_1, \ldots, y_r \in E$ are linearly independent over F, show that $y_1^p, \ldots, y_r^p$ are linearly independent over F.
7. Let E be a finite extension of the field F of prime characteristic p. Show that the extension is separable if and only if $E = F(E^p)$.
8. If $F \le K \le E$ with $[E : F] < \infty$, with E separable over K and K separable over F, show that E is separable over F.
9. Let f be an irreducible polynomial in $F[X]$, where F has characteristic $p > 0$. Express $f(X)$ as $g(X^{p^m})$, where the nonnegative integer m is as large as possible. (This makes sense because $X^{p^0} = X$, so $m = 0$ always works, and f has finite degree, so m is bounded above.) Show that g is irreducible and separable.

Figure 14: (a) One of Original Text Defect Image; (b) Root Cause Detection Result

V. Conclusion

In this paper, we provided a novel algorithm named Dataonomy^{TM} to improve the performance of the deep learning-based approach to detect product defects with limited data samples for training, which proved to be successful in our experiments. Also, the fully convolutional networks have been proved as effective end-to-end tools for defect segmentation. Detailed steps are provided regarding our approach for the tasks of defect image classification and defect detection. Besides that, a generalized multi-image matting algorithm was proposed to analyse defect cause and find defects associated with systematic errors and generated impressive results on our data. The well-designed and extensive experiments in this study verified the effectiveness of the proposed framework for surface defect inspection tasks.

References Références Referencias

1. Nhat-Duc Hoang, *An Artificial Intelligence Method for Asphalt Pavement Pothole Detection Using Least Squares Support Vector Machine and Neural Network with Steerable Filter-Based Feature Extraction*, Hindawi, Advances in Civil Engineering, Volume 2018, Article ID 7419058
2. Ssurabh Ghatnekar, *Use Machine Learning to Detect Defects on the Steel Surface*, https://software.intel.com/content/www/us/en/develop/articles/use-machine-learning-to-detect-defects-on-the-steel-surface.html
3. R. Zamir, A. Sax, W. Shen, L. Guibas, J. Malik, and S. Savarese 2018 Taskonomy: *Disentangling task transfer learning.* In Computer Vision and Pattern Recognition, CVPR.
4. K. Muto, T. Matsubara and H. Koshimizu 2018 *Proposal of local feature vector focusing on the differences among neighboring ROI's*. International Workshop on Advanced Image Technology (IWAIT), Chiang Mai, pp. 1-3.
5. Yu, Z., Wu, X., Gu, X, 2017 *Fully Convolutional networks for surface defect inspection in industrial environment*. In: Liu, M., Chen, H., Vincze, M. (eds.) ICVS 2017. LNCS, vol. 10528, pp. 417–426. Springer, Cham.
6. Ren R, Hung T, Tan KC. 2017 A generic deep-learning-based approach for automated surface inspection. IEEE Trans Cybern, 99:1-12.
7. Jia, Hongbin, 2005 Surface defect detection, classification and root cause diagnosis in steel hot rolling process
8. W. Xu,Y. Zhu, K. Sun,D. Wang, H. GU,2018 *Visual Defect Inspection Across Industry* . In Proc. ViEW, IS2-C11
9. C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. 2016 *Rethinking the inception architecture for computer vision*. In CVPR.

10. R. W. Saaty. 1987 The analytic hierarchy process–what it is and how it is used. Mathematical Modeling, 9(3-5):161–176.
11. MathWorks, https://www.mathworks.com/help/optim/examples/office-assignments-by-binary-integer-programming.html
12. Zhiyang Yu, Xiaojun Wu, and Xiaodong Gu, *Fully Convolutional Networks for Surface Defect Inspection in Industrial Environment*
13. Ren, S., He, K., Girshick, R., Sun, J.: *Faster R-CNN: towards real-time object detection with region proposal networks. In: Advances in Neural Information Processing Systems*, pp. 91– 99 (2015)
14. Zeiler, M.D., Krishnan, D., Taylor, G.W., Fergus, R.: *Deconvolutional networks*. In: 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2528–2535. IEEE (2010)
15. https://hci.iwr.uni-heidelberg.de/node/3616. Accessed 10 Apr 2017.
16. D. Weimer, B. Scholz-Reiter, and M. Shpitalni. 2016 Design of deep convolutional neural network architectures for automated feature extraction in industrial inspection. CIRP Annals-Manufacturing Technology.
17. Timm, F., Barth, E., 2011 *Non-parametric texture defect detection using Weibull features*. In Proc. SPIE 7877, Image Processing: Machine Vision Applications.