IOP Conf. Series: Journal of Physics: Conf. Series **1487** (2020) 012031 doi:10.1088/1742-6596/1487/1/012031

Image Defect Detection and Root Cause Analysis

Tiancheng Liu, Haoran Yu, Xu Han and Haisong Gu

VisionX Global Foundation

Haisonggu@ieee.org

Abstract. Artificial Intelligence has played an increasingly important role in visual defect detection in recent years, while there are many challenges using deep learning for this application, such as the shortage of data, lack of knowledge of root cause of defects. In this paper, we combine deep learning with traditional AI methods, not only to solve unshaded defect detection but also find root causes of detected defects. First, we propose a taxonomy method called Dataonomy[SM] to extend a meta defect dataset with a small number of samples and a deep learning method to detect the image defects. For detected defect images, 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 a product line and later extended its use to watermark processing. Experimental results have shown great capability and versatility of our proposed methods.

1. Introduction

Visual inspection is a common task across the 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, Deep Learning (DL) based approach has shown great potential in solving complex problems and has proven to be successful in a variety of applications. In the field of visual inspection, several works [1][2][3] were proposed using DL based approach to classify and detect the defects. One of the biggest challenges for applying DL based approach to the industry is the lack of data samples.

In practice, a common approach [4] 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[SM], 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[SM] 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.

Another issue in the 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 [5], 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 rest of the paper is structured as follows. In Section 2, we provide a description of our Dataonomy[SM]. Section 3 gives a description of our deep learning-based framework for image defect detection, followed with the methodology for root cause analysis. The setup and results of the experiments will be presented in Section 4. Conclusions will be discussed in Section 5.

2. Dataonomy[SM]

The patent pending Dataonomy[SM] algorithm [1] 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.

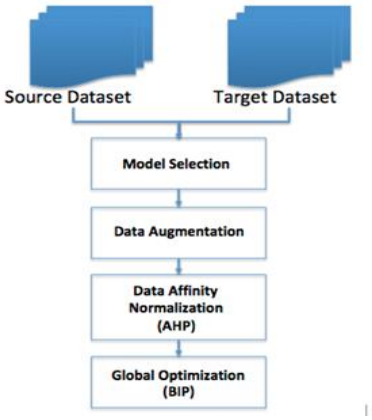


Figure 1. DataonomySM Pipeline

a) Make use of a pre-trained model for object classification, e.g. Inception V3 [6].
b) Find affinity matrix across dataset.
c) Get normalized data augmentation affinities using AHP (Analytic Hierarchy Process) [7].
d) Find global mapping taxonomy using BIP (Binary Integer Programming) [8].

The Dataonomy[SM] 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.

3. Framework for Image Defect Detection

(a) (b)

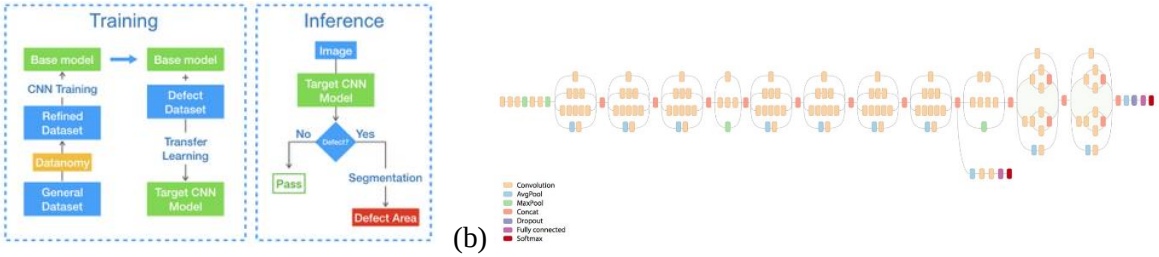


Figure 2. (a) Framework of Our Proposed Approach; (b) Inception V3

In this section, we present the framework of our proposed method for visual defect inspection. As shown in Figure 2 (a), the framework of deep learning based visual defect classification and detection

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.

3.1. Training of Base Model

In order to obtain the specific model for visual defect classification, the selection of 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^{SM} approach to prepare more useful and representative datasets related to our tasks. Then deep convolution neural network is applied with the state-of-the-art model architectures. Specifically, we introduced the InceptionV3 [6] 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 a number of inception modules which contains convolutions, pooling, concatenations, and fully connected layers. The original inception module was designed by stacking filters with multiple sizes in the same level of 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 Inception V3 network, the computational cost was further reduced by factorizing convolutional layers within inception module, where an NxN convolutional layer was decomposed into one 1xN convolutional layer followed by a 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^{SM} 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 weighs are stored as our base model and would be used in the next steps for transfer learning.

3.2. 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 dataset that is generated by Dataonomy^{SM}. 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 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^{SM}, which chooses sample images that have certain features that are more closely related to the classification task of defect inspection. This 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 initial learning rate are modified, and more details are presented in experiments.

3.3. Defect Area Detection

After the above steps, our model is capable of detecting the visual defects given an input image. Here, we further propose a segmentation approach so that the defect area can be located in the image. There are three components included in this stage, patch extraction, patch classification, heatmap generation, and localization. More specifically, we first segment the input image into patches, and then use the classifier obtained from Section 3.2 to classify each patch. The heatmap of the input image will be generated on the basis of the probabilities obtained from classification and then the defect area can be finally highlighted using the schema of binarization.

3.4. 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

$$p[m,n] = median\{g_k[m,n,k \in w]\} \quad (1)$$

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 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 at 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 3 is our defect analysis workflow.

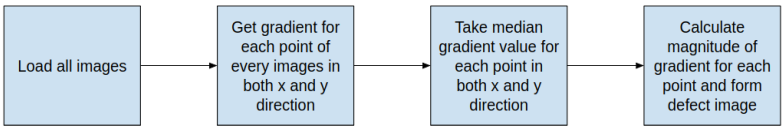


Figure 3. Defect Analysis Workflow

4. Experiments

4.1. Defect Detection

4.1.1. Datasets. We choose DAGM-2007 dataset [9] to evaluate the performance of our proposed framework. The dataset contains ten classes of different defect with different textured background, 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 is shown in Figure 4.

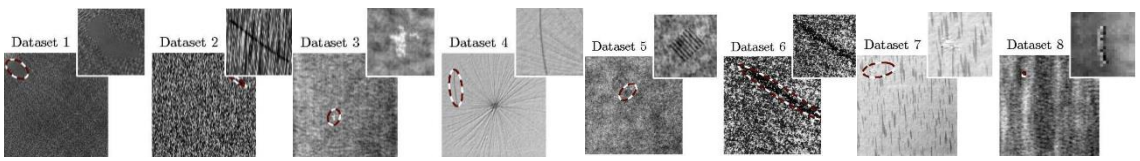


Figure 4. DAGM-2007 Dataset

4.1.2. 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[SM] from ImageNet to retrain the Inception V3 with the method in [1][2] we can prove the effectiveness of our method for data augmentation and thus showing the possibility of our method to solve the problem of limited dataset in deep learning based tasks.

Our experiment for retraining the Inception V3 using selected data from ImageNet was ran on computer with four GeForce GTX 1080 Ti graphics card. With use of transfer learning, the training of classifier for defect detection on the surface was ran on computer with two GeForce GTX 1080 Ti graphics card.

4.1.3. 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 use of transfer learning, we take the retrained Inception V3 on the selected 500 classes from ImageNet 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 provide the definition of TPR and TNR, respectively.

$$TPR = TP(TP + FN)^{-1} \tag{2}$$

$$\mathrm{TNR} = \mathrm{TN}(\mathrm{FP} + \mathrm{TN})^{-1} \tag{3}$$

Table 1 shows the performance of our framework compared to the state-of-art deep learning-based approach proposed in [10] 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 deep learning-based approach with limited data samples.

Table 1. Defect detection result (%).

No.	Weimer et al. [10]		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 on defect detection with the work in [2] and [11]. The accuracy of our method with 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 DAGM-2007 dataset is 99.88%. It can be seen that our framework using Dataonomy[SM] for data augmentation shows high performance on defect detection with 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. Part of the result for the texture data in this step is shown in Figure 5, showing a decent performance of our method.

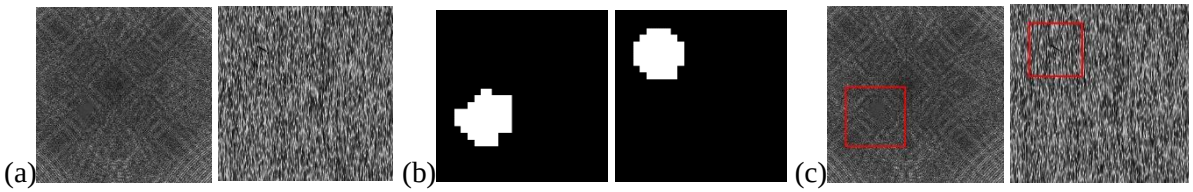


Figure 5. Defect Detection on DAGM-2007 (a) Original Image (b) Mask Image (c) Highlighted Defect

4.2. Defect Cause Analysis

4.2.1. Dataset for Root Cause Detection. In order to evaluate our method, we created a new dataset for our root cause detection, based on DAGM-2007 dataset [9]. 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 at the same position for each image. Figure 6 (a) and (b) are two examples with different systematic defects with the original defect from DAGM-2007.

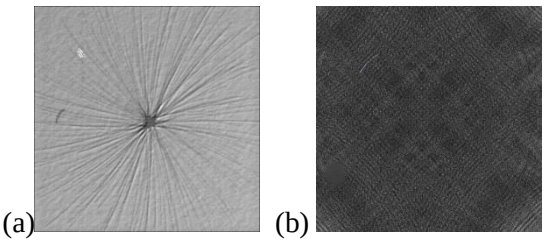


Figure 6. Example of 2 Types of Scratches.

4.2.2. Experimental Results for Root Cause Detection. Using the method in Sec. 3.3, we got common defect image for each type of scratch.

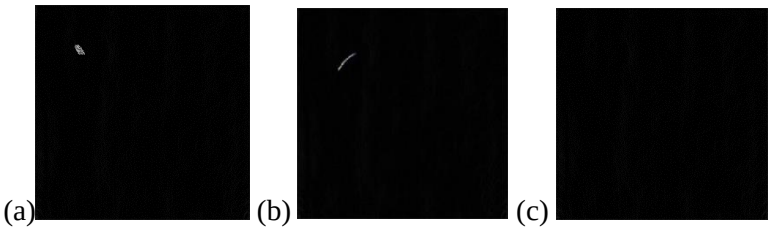
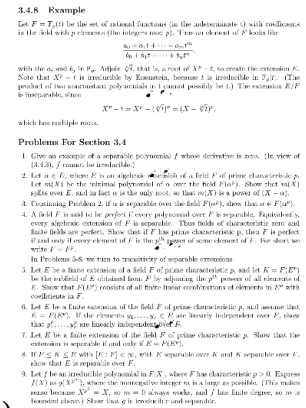


Figure 7. (a)(b) Results with Common Defect Detected; (c) Result with No Common Defect Detected.

From Figure 7 (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 7 (c) is the resultant image of the common defect detection for 1046 images with defect in DAGM-2007 dataset [9]. 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. We created a 500-large text defect dataset by adding the same ink defect at the same position of text images. Figure 8 (a) is an example from our dataset. Using our root cause detection method, the resulting image Figure 8 (b) does detect those 4 ink defects in the original dataset (we reversed colour for better notice). Therefore, in real life, we can know there is a problem in the printing system that causes the common defect using this method.



(a) (b)

Figure 8. (a) One of Original Text Defect Image; (b) Root Cause Detection Result.

5. Conclusion

In this paper, we provided a novel algorithm named Dataonomy[SM] 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. Detailed steps are provided regarding our approach for the tasks of defect image classification and defect area detection. In addition, a generalized multi-image matting algorithm was proposed to analyse defect cause and find defects associated to systematic errors and generated impressive results on our data. This method was also successfully applied in watermarker removal in our experiments. The well-designed and extensive experiments in this study clearly verified the effectiveness of the proposed framework for surface defect inspection tasks.

Reference

[1] 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.

[2] 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.

[3] 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.

[4] Ren R, Hung T, Tan KC. 2017 *A generic deep-learning-based approach for automated surface inspection*. IEEE Trans Cybern, **99**:1-12.

[5] Jia, Hongbin, 2005 *Surface defect detection, classification and root cause diagnosis in steel hot rolling process*

[6] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. 2016 *Rethinking the inception architecture for computer vision*. In CVPR.

[7] R. W. Saaty. 1987 *The analytic hierarchy process– what it is and how it is used*. Mathematical Modeling, **9**(3-5):161–176.

[8] MathWorks, https://www.mathworks.com/help/optim/examples/office-assignments-by-binary-integer-programming.html,

[9] https://hci.iwr.uni-heidelberg.de/node/3616. Accessed 10 Apr 2017.

[10] 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.

[11] Timm, F., Barth, E., 2011 *Non-parametric texture defect detection using Weibull features*. In Proc. SPIE 7877, Image Processing: Machine Vision Applications.