

# Local Defect Detection and Print Quality Assessment

Jianguo Wang<sup>†</sup>, Terry Nelson<sup>‡</sup>, Renee Jessome<sup>‡</sup>, Steve Astling<sup>‡</sup>, Eric Maggard<sup>‡</sup>, Mark Shaw<sup>‡</sup>, Jan P. Allebach<sup>†</sup>

<sup>†</sup>School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907, USA

<sup>‡</sup>HP Inc., Boise, ID 83714, USA

## Abstract

During the development of a new printer product, units of the product undergo extensive testing, during which thousands of print quality test pages are printed. Compared with visual evaluation of these pages by print quality experts, it is more efficient and cost-effective to scan the pages, and then perform image analysis to assess their quality. In this project, we develop an algorithm for the detection of a specific type of print quality artifact: local defects, and the prediction of the overall print quality that would be assigned by an expert observer to prints that exhibit such defects. Local defects are print defects in the form of spots and patches. In the detection procedure, the local standard deviation of the scanned pages is firstly computed to find the candidate regions where local defects are likely to occur. Subsequently, an automatic thresholding algorithm, namely valley-emphasis thresholding, is applied to find the local defects. Multiple features of each local defect are further calculated. With a database of print samples for which grades have been assigned by a print quality expert, a print quality predictor is trained by the support vector machine (SVM) method. For a new print sample, the detection procedure firstly finds the local defects and their features, and then the print quality is determined by the trained SVM model.

## Introduction

Early in the product development cycle at major printer manufacturers, a batch of printers goes into extensive testing during which humidity, temperature, media type and job type are varied. During the extensive testing that these units undergo, an abundant number of test pages are printed. Assessing the print quality (PQ) of the test images can reveal multiple printer design deficiencies. Traditionally, the print quality of these test pages is assessed by human experts through visual evaluation. Thus, the general PQ rank and more detailed print defect information are both obtained manually. Given the large number of test pages, the visual evaluation procedure is very costly and time-consuming. Furthermore, human observers are usually inconsistent in determining the print quality. Therefore, an algorithm that can assess a scanned page, find print defects in the page, and give appropriate rank is highly desirable.

In this project, we develop such an algorithm to detect and characterize one specific class of print defect: local defects. A print quality predictor is further trained by the support vector machine (SVM) algorithm to give the PQ rank of test pages with the obtained defect information. Local defects are print defects in the form of a spot or patch of which the size and shape can vary dramatically. An example of large patch local defects is shown in Fig. 1. The sample is printed on letter-size paper; and the local

defects are easily visible. An example of small spot local defects is shown in Fig. 2, where the local defects can only be seen after zooming in.

Local defects could be caused by numerous printer malfunctions, such as foreign contamination on the intermediate transfer belt and photoconductor drum defects. Therefore, they play an important role in the print quality assessment.

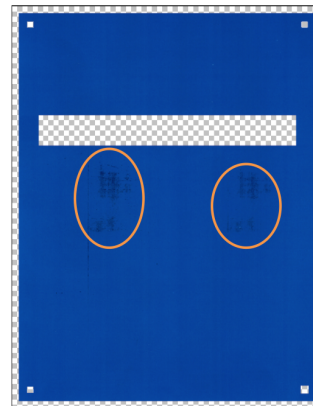


Figure 1. Example of large patch local defects. The orange circles are not part of the print sample.

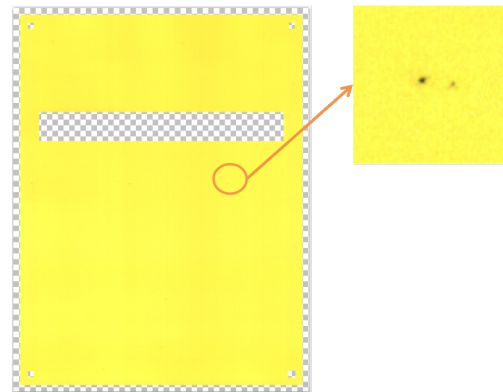


Figure 2. Example of small spot local defects.

Some studies have been conducted in the area of print defect detection. Jing et al. [1] uses image quality assessment methods such as the structural similarity index (SSIM) [2] to assess a wide range of print defects. Zhang et al. [3][4] studied banding detection with a histogram-based method. Nguyen et al. [5][6] described a method based on local and global variance to detect graininess and mottle. Yan et al. [7] and Ju et al. [8] gave novel

Research supported by HP Inc., Boise, ID 83714, USA.

algorithms for predicting the visibility of fading defects. However, even though the local defect is a very common type of print defect, studies of local defect detection are relatively rare.

Our project pipeline can be separated into three stages. In the first stage, local defects are detected and located. Then multiple features of each defect are calculated. Finally, a print quality predictor is trained based on the local defect features. The following sections describe the three stages in detail.

## Local Defect Detection

In the stage of detecting the presence of local defects, we first decide the candidate regions where the local defects are likely to appear, then use automatic thresholding to separate the defect pixels from properly printed pixels inside the candidate regions. A diagram of the whole process is shown in Fig. 3.

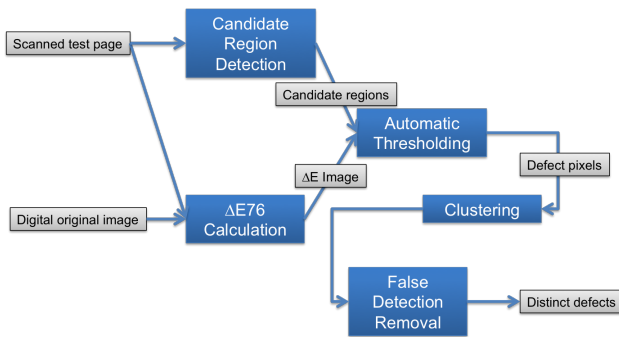


Figure 3. Block diagram for local defect detection process.

## Deciding Candidate Region

Candidate regions are the locations in the page where local defects are likely to occur. All our future processing, such as automatic thresholding, will only be conducted in the candidate regions instead of the whole page.

The method we are using consists of two steps. Firstly a page is separated into grid blocks, and the standard deviation of the  $\Delta E$  difference values between original pixel values and the block average is calculated. This standard deviation is referred as the local standard deviation  $\sigma_l$ . Afterwards an autonomous method is applied to choose the blocks that have relatively higher local standard deviation, which are the candidate regions.

## Local Standard Deviation Calculation

Currently, the test pages that we use are all letter-sized, with uniform color, and have 4 fiducial marks at the four corners of the pages. Test information such as the printer model is put at approximately 1/3 height of the page from the top. The printed test page is scanned in 600 dpi; and a Gaussian filter is applied to the scanned image to remove the halftone pattern.

The scanned page is firstly transformed from *sRGB* into the *CIE L\*a\*b\** color space. Subsequently, the page is separated into a block grid. The size of each block is 150 pixels  $\times$  150 pixels, which is equal to 0.25 inch  $\times$  0.25 inch at 600-dpi resolution. Within each block, the average values of the  $L^*$ ,  $a^*$  and  $b^*$  chan-

nels are computed as

$$L_{avg}^* = \frac{1}{N} \sum_{i=1}^N L_i^*, \quad (1)$$

$$a_{avg}^* = \frac{1}{N} \sum_{i=1}^N a_i^*, \quad (2)$$

$$b_{avg}^* = \frac{1}{N} \sum_{i=1}^N b_i^*, \quad (3)$$

where the subscript  $i$  is the pixel index, and  $N = 150^2$  is the number of pixels in the block. Then, the  $\Delta E$  difference between each pixel and the block average is calculated as

$$\Delta E_i = \sqrt{(L_i^* - L_{avg}^*)^2 + (a_i^* - a_{avg}^*)^2 + (b_i^* - b_{avg}^*)^2}. \quad (4)$$

With all pixels'  $\Delta E$  values, the local standard deviation  $\sigma_l$  is determined as

$$\sigma_l = \sqrt{\frac{1}{N} \sum_{i=1}^N (\Delta E_i - \Delta E_{avg})^2}, \quad (5)$$

where

$$\Delta E_{avg} = \frac{1}{N} \sum_{i=1}^N \Delta E_i. \quad (6)$$

If local defects exist in a block, the  $\Delta E$  values of defect pixels and those of background pixels will be very different, which leads to a relatively large standard deviation. On the other hand, if all pixels inside a block appear uniform, those  $\Delta E$  values should be similar (all close to 0). Therefore, we are able to distinguish the candidate regions from other blocks by comparing their  $\sigma_l$ 's. An example of the calculated  $\sigma_l$ 's of a page is shown in Fig. 4. We use a color coding that uses darker color to indicate a block for which  $\sigma_l$  is larger. It can be easily observed that the  $\sigma_l$  values of blocks where local defects exist are much greater than those of their neighboring blocks.

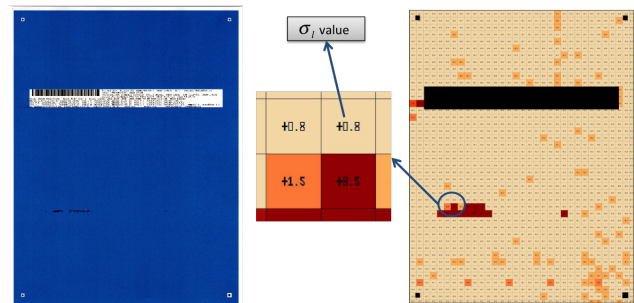


Figure 4. Local standard deviation. The left image is the original scanned image; and the right image is the result of the local standard deviation calculation.

## Choosing candidate blocks

After obtaining the local standard deviation  $\sigma_l$ 's, we apply an autonomous method to choose the blocks with relatively large  $\sigma_l$  values as the candidate regions.

We first calculate a histogram of all blocks based on their  $\sigma_l$  values. The bin edges of this histogram are  $[0, 0.5, 1, 1.5, 2, 2.5, 3, 5, inf]$ ; and the histogram is thus an 8-bin histogram. The peak of this histogram is further located. Since this peak represents the most “common” print quality in the page, the blocks falling into this bin are unlikely to have local defects. Here we assume most blocks do not contain local defects. In addition, all blocks within bins that are on the left side of the peak have better quality than peak blocks, and thus are also not regarded as candidate regions, either. For the bins on the right side of the peak, we set a fixed threshold to decide if the blocks within them are candidate regions. In our experiment, a threshold that is set to be 5% of the total number of blocks in the test page gave a satisfactory result. Therefore, if a histogram bin located at the right side of the peak contains fewer than 5% of the total number of blocks, blocks within this bin are classified as candidate regions. Finally, we combine the neighboring candidate region blocks to one single candidate region.

### Automatic Thresholding

In this section, we describe the automatic thresholding method applied to the pixels inside candidate regions.

### Difference Image Calculation

Our automatic thresholding algorithm will be applied to a  $\Delta E$  difference image. However, this  $\Delta E$  difference image is not the same as the previous within-block  $\Delta E$ . Here, the  $\Delta E$  is between the scanned test page and the digital master page, which is a defect-free version of the test page. The computation can be expressed as

$$\Delta E(i, j) = \left( (L_t^*(i, j) - L_m^*(i, j))^2 + (a_t^*(i, j) - a_m^*(i, j))^2 + (b_t^*(i, j) - b_m^*(i, j))^2 \right)^{1/2}, \quad (7)$$

where  $(i, j)$  is the pixel index in the page, and subscripts  $t$  and  $m$  indicate test and master images, respectively. Before the computation of  $\Delta E$ , color calibration is needed to eliminate the color difference between the test page and master page. An example of the difference image calculation is shown in Fig. 5. It can be seen that the defect pixels have greater  $\Delta E$  values than properly printed pixels.

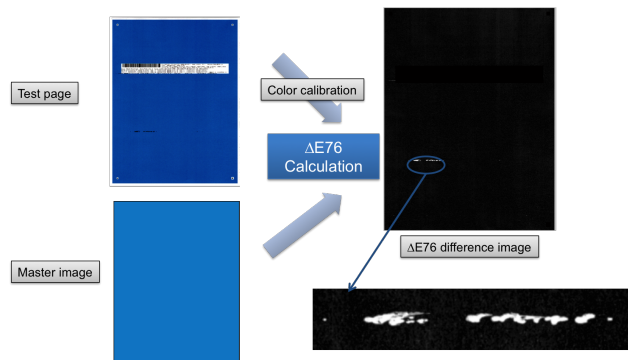


Figure 5. Calculation of difference image.

### Valley-emphasis Thresholding Algorithm

After getting this  $\Delta E$  difference image, we simply map the candidate regions we obtained from the previous steps to this difference image at the same position. An example can be seen from Fig. 6. As expected, the defect pixels are located inside candidate regions.

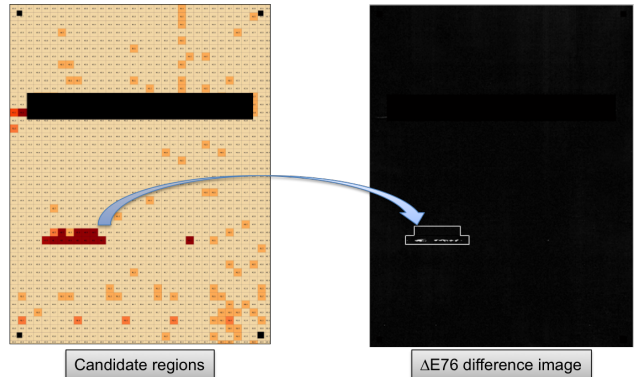


Figure 6. Mapping candidate regions to the difference image.

The next step is to apply an automatic thresholding algorithm inside each candidate region. Among numerous automatic thresholding strategies, Otsu’s algorithm is one of the most widely used [9]. Based on the deduction of Liao et al. [10], the preferred threshold  $t^*$  in the Otsu’s algorithm of a gray level image is chosen automatically by maximizing the between class variance, which can be written as

$$t^* = \operatorname{argmax}_{0 \leq t < L} \{ \omega_1(t) \mu_1(t)^2 + \omega_2(t) \mu_2(t)^2 \}, \quad (8)$$

where  $t$  is a gray level value and  $L$  is the number of distinct gray levels;  $\omega_1$  and  $\omega_2$  are the percentages of pixels falling into the background and foreground, respectively; and  $\mu_1$  and  $\mu_2$  are the average gray level of background and foreground pixels.

Despite the wide acceptance of the Otsu’s algorithm, Ng et al. [11] proposed to add a new term in the maximization to emphasize the “valley” in the image histogram. This modification is based on the intuitive assumption that the correct threshold should be located at a histogram “valley”; so the foreground and background can be properly separated. The refined expression is

$$t^* = \operatorname{argmax}_{0 \leq t < L} \{ (1 - p(t)) (\omega_1(t) \mu_1(t)^2 + \omega_2(t) \mu_2(t)^2) \}, \quad (9)$$

where  $p(t)$  is the percentage of pixels at gray level  $t$ . As shown in the equation, a gray level that corresponds to a smaller number of image pixels, i.e. a valley in the histogram, is favored in the argmax calculation.

To apply this to our defect detection pipeline, we feed pixels inside each candidate region into the valley-emphasis algorithm. After getting the optimized threshold, any pixels with gray level value greater than the threshold are set to be 1; and other pixels are set to be 0. All pixels outside the candidate regions are set to be 0. By doing this, we obtain a binary output mask where white pixels indicate defects pixels and black pixels indicate properly printed pixels. In our experiment, valley-emphasis thresholding achieves robust performance, as an example shows in Fig. 7.

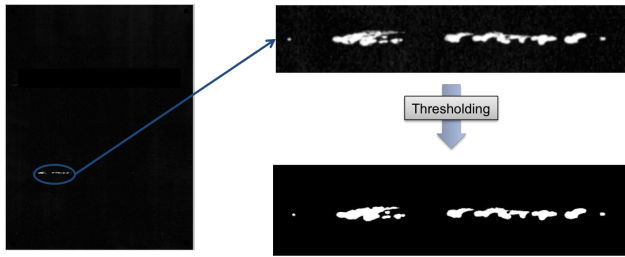


Figure 7. Result of valley-emphasis thresholding.

### Defect Pixel Clustering and False Detection Removal

In the thresholding step, we separate the defect pixels from the properly printed pixels. To make the output easier to interpret, we cluster those pixels into distinct defects.

The clustering strategy we use is a generalized connected component method. In a traditional 8-neighborhood connected component procedure, a  $3 \times 3$  neighborhood of the current foreground pixel is searched to find whether there exist any other foreground pixels. In our generalized connected component method, we instead use a  $161 \times 161$  neighborhood. All defect pixels within this neighborhood will be labeled as belonging to the same distinct defect. We use this much bigger window size because defect pixels produced by a single printer deficiency are usually close to, but not necessarily connected to each other. But it makes more sense to label them as one single defect. As shown in Fig. 8, each distinct defect is given an ID number. If we used a traditional connected component algorithm, the defect with ID 4 would be marked as four distinct defects.

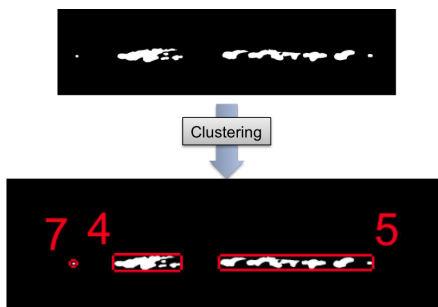


Figure 8. Example of defect pixel clustering.

After getting distinct defects, we calculate two features of them, which are the area size in pixels and the contrast to the background. While the defect area size is straightforwardly defined, we define the defect contrast as the ratio between the average pixel value of the defect and the average pixel value of the background in the  $\Delta E$  difference image. We only use background pixels inside a certain sized neighborhood around the defect to make the contrast a localized value, which is more consistent with the human observation. This neighborhood is defined as a rectangle concentric with the bounding box of the defect but 40% bigger in both dimensions. Furthermore, we define an area size threshold of 25 pixels and a contrast threshold of 2. Any detections with area size smaller than 25 pixels or contrast lower than 2 will be removed. In our experiment, those detections are hard to observe by human viewers.

### Three Categories of Local Defects

After removing the false alarms, we further divide the local defects into three categories: small-area local defect, large-area local defect, and repeating defects. These categories are useful to engineers for determining the printer malfunction. We first separate small-area local defects and large-area local defects with a threshold of the area size of the defect bounding box. We use the area size of the defect bounding box rather than the defect itself, because if a defect consists of “sparse” spots dispersed in a large area, it appears to be large even if its own area size is limited. We set the threshold on the defect bounding box area to be 25000 pixels. Any defect with bounding box area greater than 25000 pixels is categorized as large-area local defect. Among other defects that have smaller area sizes, if more than three of them align along either vertical or horizontal directions, they are categorized as repeating defects. We further use a method similar to the one described in [3] to calculate the repeating intervals of the repeating defects. Defects which are not categorized as large-area or repeating defects are small-area local defects. A final result of local defect detection is shown in Fig. 9. Defects marked with ID 8 are a set of repeating defects. For each test page, an RGB output image and a binary output image are produced, where the defects, defects bounding boxes, and ID’s are marked.

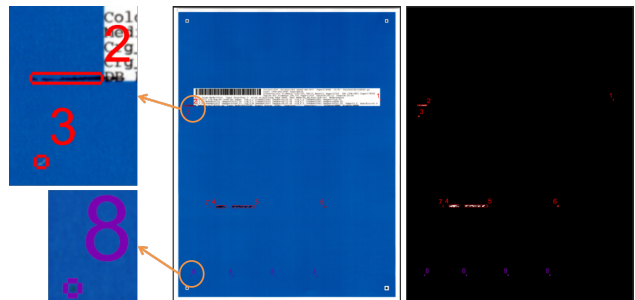


Figure 9. Final result of local defect detection.

### Local Defect Feature Calculation

With the distinct defects, we calculate multiple features for each defect. These features are helpful to printer development engineers for analyzing those defects and further deciding the cause of those defects. In addition, some of them will also be used in training the print quality predictor. The calculated features include area size, centroid position, contrast to the background (in the  $\Delta E$  difference image), average RGB value, bounding box information (position and dimension), edge strength, darker or lighter than the background, perimeter, and repeating interval (only for repeating defects). Most features are straightforward to calculate. For edge strength, since the edge pixels can be extracted from the defect detection stage, we simply compute the Difference of Gaussian (DoG) filter response of these edge pixels and use the absolute summation as the defect’s edge strength. These features are listed in a comma-separated values (CSV) spreadsheet, and can give useful feedback to the printer design engineers.

### Print Quality Predictor

Traditionally, human experts assess and give print quality ranks to test pages printed by the test units. In order to au-

tonomously evaluate the print quality of test pages, we train a print quality predictor by the support vector machine (SVM) method [12]. With a trained SVM model, the PQ rank of a new test page can be autonomously given after going through the local defect detection and feature calculation stages.

### Training Database

We use 80 scanned pages as the training data. They have been assessed by human print quality experts in the Hewlett-Packard Company. In our project, each test page can be assigned to one of the four ranks: A (good, PQ-Pass), B (fairly good, PQ-Pass), C (bad, PQ-Fail), and D (very bad, PQ-Fail). The numbers of pages with different ranks are: 13 rank A, 19 rank B, 23 rank C, and 25 rank D, respectively. Examples of rank B, C and D pages can be seen in Figs. 10, 11, and 1. We are not showing rank A examples because these pages are usually just defect-free.

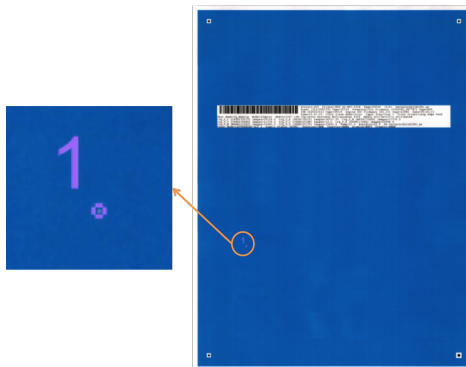


Figure 10. Example of rank B page.

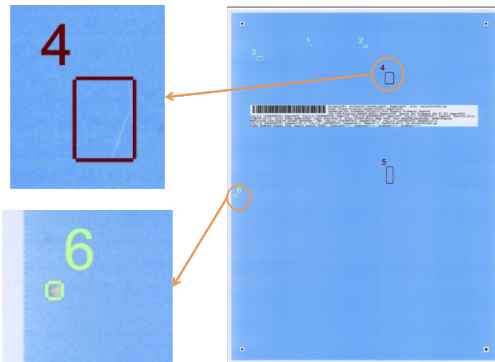


Figure 11. Example of rank C page.

### Training Features

The first step of the machine learning pipeline is to produce a feature vector for each test page. Unfortunately, we cannot simply use the features we calculated in the defect feature calculation stage. This is due to the fact that different pages have different numbers of defects, and thus using the defect features directly will lead to a feature vector whose length depends on the number of defects on the page. SVM cannot accept variable length feature vector as the prediction input, so we need to modify the defect feature representation.

For this purpose, we first define a new feature for a single defect, namely defect volume  $V_d$ :

$$V_d = A_d \cdot C_d, \quad (10)$$

where  $A_d$  is the area size of the defect in  $mm^2$  and  $C_d$  is the contrast of the defect. They are both calculated in the previous feature calculation step. We name this feature “volume” because if the defect area is regarded as the base area of a cylinder and contrast as the height, this feature is comparable to the cylinder’s volume. We then define two features for each test page: the maximum defect volume and the sum of all defects’ volumes. Consequently, with these two features we take into account both the most noticeable single defect and the overall appearance of all defects on the page, which makes this feature vector representative.

### Training Strategy

Support vector classification is a two-class classification method. Our problem is a four-class classification problem, so we need a strategy to apply SVM in our project.

Different from general multi-class classification problems, our four classes can be divided into two super-classes, i.e. PQ-pass pages (ranks A and B) and PQ-fail pages (ranks C and D). Therefore, we use a binary decision tree as shown in Fig. 12 to deal with this problem. The first level of the decision tree separates the test page into PQ-pass pages (ranks A and B) and PQ-fail pages (ranks C and D) with an SVM classifier. Afterwards, if the page is PQ-pass, we use a second SVM classifier to decide whether it belongs to rank A or rank B. Otherwise, a third SVM classifier is applied to decide whether it is rank C or rank D. In practice, the first level (PQ-pass or PQ-fail decision) is the most important classifier, since it decides whether the printer’s performance is still acceptable. This classifier is trained with all 80 of our training images (32 PQ-pass images and 48 PQ-fail images). The rank A/B classifier is trained with 13 rank A and 19 rank B images. The rank C/D classifier is trained with 23 rank C and 25 rank D images.

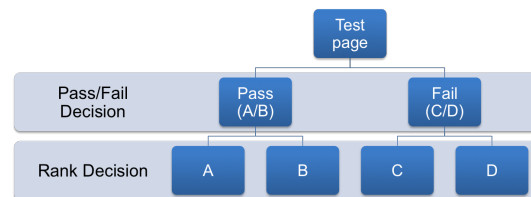
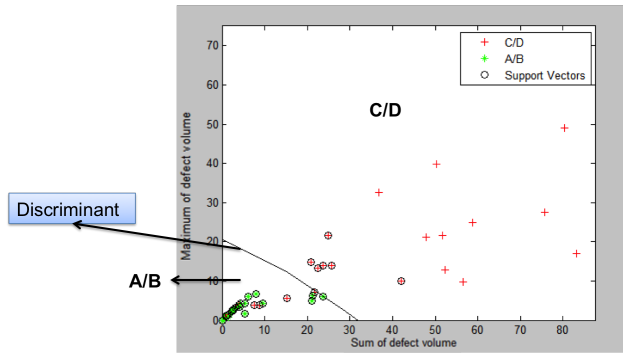


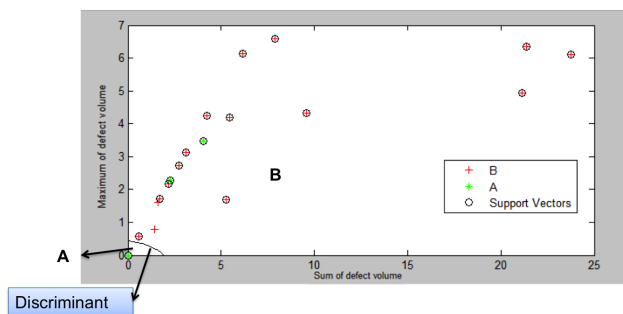
Figure 12. Binary decision tree for multi-class classification.

### Accuracy Analysis

We adopt 10-fold cross-validation to measure the accuracy of each SVM classifier [13]. In addition, to make the result more representative, the cross-validation is repeated five times with different random partitions. The first classifier is the pass/fail decision as shown in the first decision tree level in Fig.12. The mean 10-fold cross-validation accuracies obtained from the 5 repetitions are 86.25%, 85.00%, 85.00%, 86.25%, and 83.75%, respectively. Each mean accuracy is the average of 10 accuracies that are calculated from the 10 folds of one repetition. The mean accuracies for



**Figure 13.** Discriminant of PQ-pass or PQ-fail decision. Note that this figure is not showing all training data since many rank C and rank D data are out of this figure's range.



**Figure 14.** Discriminant of rank A/B decision. Note many rank A data with no defects overlap at the coordinate origin.

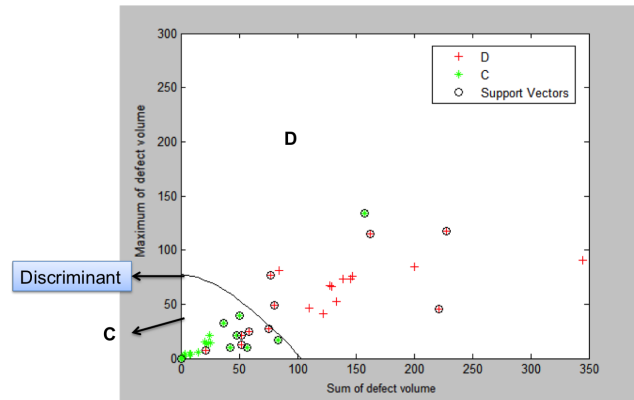
the 5 repetitions (50 accuracies in total) is 85.25% and the standard deviation of the 50 accuracies is 12.55%. The discriminant trained with all training data (instead of part of them as in cross validation) is shown in Fig. 13.

For the rank A/B decision, the same cross-validation strategy is adopted. The mean accuracies of the 10-fold cross-validations are 85.83%, 84.17%, 83.33%, 85.00%, and 85.00%, respectively. The mean accuracies for the 5 repetitions (50 accuracies in total) is 84.67% and the standard deviation is 18.55%. The discriminant trained with all training data is shown in Fig. 14.

Similarly, the same cross-validation strategy is adopted for the rank C/D decision. The mean accuracies of the 10-fold cross-validations are 84.00%, 85.50%, 84.50%, 85.50%, and 83.00%, respectively. The mean accuracies for the 5 repetitions (50 accuracies in total) is 84.50% and the standard deviation of the 50 accuracies is 14.58%. The discriminant trained with all training data is shown in Fig. 15.

## Conclusion

In this project, we developed a method to automatically detect the local defects on a scanned print sample. After finding the defects, we are able to cluster and label them and then divide them into 3 categories (small-area local defect, large-area local defect and repeating defects). Then multiple features are calculated for each defect. Finally, a print quality predictor is trained with the support vector machine method. Satisfactory accuracy is achieved for this predictor.



**Figure 15.** Discriminant of rank C/D decision.

## References

- [1] Xiaochen Jing, Steve Astling, Renee Jessome, Eric Maggard, Terry Nelson, Mark Shaw, and Jan P Allebach, "A general approach for assessment of print quality," in *IS&T/SPIE Electronic Imaging*. International Society for Optics and Photonics, 2013, pp. 86530L–86530L.
- [2] Zhou Wang, Alan Conrad Bovik, Hamid Rahim Sheikh, and Eero P Simoncelli, "Image quality assessment: from error visibility to structural similarity," *Image Processing, IEEE Transactions on*, vol. 13, no. 4, pp. 600–612, 2004.
- [3] Jia Zhang, Stephen Astling, Renee Jessome, Eric Maggard, Terry Nelson, Mark Shaw, and Jan P Allebach, "Assessment of presence of isolated periodic and aperiodic bands in laser electrophotographic printer output," in *IS&T/SPIE Electronic Imaging*. International Society for Optics and Photonics, 2013, pp. 86530N–86530N.
- [4] Jia Zhang and Jan P Allebach, "Estimation of repetitive interval of periodic bands in laser electrophotographic printer output," in *IS&T/SPIE Electronic Imaging*. International Society for Optics and Photonics, 2015, pp. 93960J–93960J.
- [5] Minh Q Nguyen, Renee Jessome, Steve Astling, Eric Maggard, Terry Nelson, Mark Shaw, and Jan P Allebach, "Perceptual metrics and visualization tools for evaluation of page uniformity," in *IS&T/SPIE Electronic Imaging*. International Society for Optics and Photonics, 2014.
- [6] Minh Q Nguyen and Jan P Allebach, "Controlling misses and false alarms in a machine learning framework for predicting uniformity of printed pages," in *IS&T/SPIE Electronic Imaging*. International Society for Optics and Photonics, 2015, pp. 93960I–93960I.
- [7] Ni Yan, Eric Maggard, Roberta Fothergill, Renee J Jessome, and Jan P Allebach, "Autonomous detection of iso fade point with color laser printers," in *IS&T/SPIE Electronic Imaging*. International Society for Optics and Photonics, 2015, pp. 93960F–93960F.
- [8] Yanling Ju, Eric Maggard, Renee Jessome, and Jan Allebach, "Autonomous detection of text fade point with color laser printers," in *IS&T/SPIE Electronic Imaging*. International Society for Optics and Photonics, 2015, pp. 93960G–93960G.
- [9] Nobuyuki Otsu, "A threshold selection method from gray-level histograms," *Automatica*, vol. 11, no. 285–296, pp. 23–27, 1975.
- [10] Ping-Sung Liao, Tse-Sheng Chen, and Pau-Choo Chung, "A fast algorithm for multilevel thresholding," *J. Inf. Sci. Eng.*, vol. 17, no. 5, pp. 713–727, 2001.
- [11] Hui-Fuang Ng, "Automatic thresholding for defect detection," *Pat-*

*tern recognition letters*, vol. 27, no. 14, pp. 1644–1649, 2006.

- [12] Christopher M Bishop, *Pattern recognition and machine learning*, vol. 1, Springer New York, 2006.
- [13] Ron Kohavi et al., “A study of cross-validation and bootstrap for accuracy estimation and model selection,” in *IJCAI*, 1995, vol. 14, pp. 1137–1145.

## Author Biographies

*Jianyu Wang is a Ph.D. student at School of Electrical and Computer Engineering, Purdue University, supervised by Prof. Jan P. Allebach. He received his B.E. from Tianjin University (2011). His research has mainly focused on image quality, photo aesthetic quality assessment, 2.5D printing, and machine learning.*

*Terry Nelson has been a Development Engineer with Hewlett Packard for over 35 years. He started sparking with a Tesla Coil, radios, and HP calculators. After reviving an A.A.S. in electronics from Spokane Community College in 1979, he joined Hewlett-Packard Company working disk storage technology. Moving to printing, he developed systems for ink jet, wet/dry electro-photography, and machine control. Several years were spent doing system modeling, image processing, color science, system testing, and way too many programming languages. Along the way he picked up eight patents in ink jet, image processing, data compression, color science, and printer calibration.*

*Renee Jessome received a B.S. in Mechanical Engineering from the Rochester Institute of Technology, Rochester, NY, in 1992. She has worked on a variety of printing devices over the past 14 years from the smallest LaserJet printers to Indigo Printing presses. She currently leads a Product Development team focused on high end LaserJet Printers and Multi-Function Printers at HP Inc. in Boise, Idaho. During her early career at HP she was developing storage devices such as disk drives, tape drives and disk arrays.*

*Stephen Astling is a systems engineer who worked at Hewlett Packard on the development of commercial and enterprise color laser printers and scanners. Stephen focused on print quality, including defect troubleshooting, color calibration, and psychometric testing to verify print quality relative to competitive benchmarks.*

*Eric Maggard received his B.S. degree in Physics from Northwest Nazarene University, Nampa, Idaho in 1991 and the M.S. degree in Computer Science specializing in image analysis and processing from Walden University in 2006. He is an Expert Imaging Scientist in the LaserJet Hardware Division and has developed programs and image quality algorithms for the last 15 years that are used in the testing of LaserJet print and scan image quality. His interests include machine vision, object recognition, machine learning, robot control and navigation.*

*Mark Q. Shaw received the B.Sc. degree in print technology from the University of Hertfordshire, London, U.K., in 1997, the M.Sc. degree in color science from the Rochester Institute of Technology, Munsell Color Science Laboratory, Rochester, NY, in 1999, and the Ph.D. degree in electrical and computer engineering from Purdue University, West Lafayette, IN. He is a Color and Imaging Architect with the Laserjet Hardware Division, Hewlett-Packard Company, Boise, Idaho. He has over 15 years experience in the Color and Imaging Industry having previously worked for other major corporations including 3M and Xerox. His interests include video processing, multispectral color reproduction, color modeling, HDR image rendering, color management, and image understanding.*

*Jan P. Allebach is Hewlett-Packard Distinguished Professor of Electrical and Computer Engineering at Purdue University. Allebach is a Fellow of the IEEE, the National Academy of Inventors, the Society for Imaging Science and Technology (IS&T), and SPIE. He was named Elec-*

*tronic Imaging Scientist of the Year by IS&T and SPIE, and was named Honorary Member of IS&T, the highest award that IS&T bestows. He has received the IEEE Daniel E. Noble Award, and is a member of the National Academy of Engineering. He currently serves as an IEEE Signal Processing Society Distinguished Lecturer (2016-2017).*