10.2352/EI.2022.34.15.COLOR-259

Measuring Colorant Fading Within Raster Regions of Printed Scanned Customer Content Using a Novel Unsupervised Clustering Method*

Runzhe Zhang [a], *Yousun Bang* [c], *Minki Cho* [c], *Mark Shaw* [b], *Jan Allebach* [a]
[a] *School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47906, U.S.A.*
[b] *HP Inc., Boise, ID 83714, U.S.A.*
[c] *HP Inc., Suwon City, KOREA.*

Abstract

Colorant fading defects in raster region of interest (ROI) [1] are among the most common printing issues in electrophotographic printers. Colorant fading manifests as faint print or customer content and is usually caused by low-level ink/cartridge. This paper presents an accurate method to detect the colorant fading defects and classify the defects based on their severity. There are two modules for this method. The first module uses the SLIC super-pixel method to separate the raster ROI and extract the smooth super-pixels. We design a novel unsupervised clustering algorithm to automatically extract the three or four main colors from the smooth super-pixels. Unsupervised data clustering is an important problem that arises in many image processing applications. We propose a new approach to unsupervised data clustering called Data Inter-Distance MEdiated Clustering (DIDMEC). Our new method is based on analyzing the matrix of Euclidean distances between each pair of points in the data set. Based on three simple properties, we devise an approach that effectively yields the same accuracy as the K-Means algorithm but at a much lower computational cost for a moderate number of sample points. The second module extracts feature vectors for each main color clustered by DIDMEC to classify the colorant fading defects based on their severity.

1. Introduction

There are many different solutions to detect and evaluate raster ROI colorant fading defects in electrophotographic printers. The most common method to detect raster ROI colorant fading defects is by calculating the color average difference between the printed and the original electronic images [2]. This method is inaccurate because the average of raster ROI colors may not be the real color present in the original raster ROI.

This paper presents an accurate method for detecting colorant fading defects and classifying them based on their severity. There are two modules for this method. The first module uses the SLIC super-pixel method [3] to separate the raster ROI and extract the smooth super-pixels. We design a novel unsupervised clustering algorithm to automatically extract the three or four main colors from the smooth super-pixels. We propose the Data Inter-Distance MEdiated Clustering (DIDMEC) algorithm to reduce the amount of calculation. It can classify the data points by only calculating the Euclidean distance between every pair of sample points while building a distance matrix. In the DIDMEC method, we first calculate the best in-cluster threshold for the distance matrix. Then, we use three properties to extract the clusters one by one. The DIDMEC clustering method effectively yields the same accuracy as the K-Means algorithm [4,5] but at a much lower computational cost for a moderate number of sample points. The second module extracts feature vectors for each main color clustered by DIDMEC to rank the colorant fading defects based on their severity. In Figure 1, we show some sample images for raster ROI color fading defect: (a) is the digital master image raster ROI saved in the computer; (b) is the rank A of raster ROI color fading defect, and there is no defect with good print quality; (c) is the rank B of raster ROI color fading defect with a little cyan fading; (d) is the rank C of the raster ROI color fading defect with some magenta fading; (e) is the rank D of the raster ROI color fading defect with severe magenta fading.

Figure 1: Raster ROI color fading defects sample images.

2. Raster ROI color fading detection and classification procedure

This section introduces the procedure of raster ROI colorant fading defects detection and classification. Figure 2 shows the overall pipeline of the proposed method. This procedure includes the raster ROI preprocessing part and the color fading classification part.

In the first raster ROI preprocessing part, we need to align the test raster ROI based on the master raster ROI

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

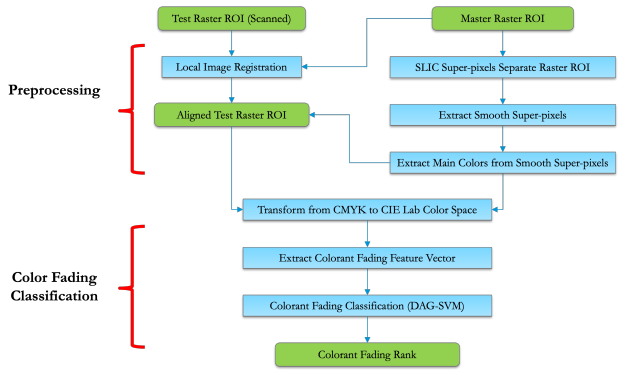


Figure 2: The overall pipeline of the raster ROI colorant fading detection and classification.

a. Master and test raster ROI blending before image registration

b. Master and test raster ROI blending after image registration

Figure 3: The master raster ROI and test raster ROI blended images before and after image registration process.

to get the aligned test raster ROI. We use the SLIC super-pixels method to separate the master raster ROI into several super-pixels based on the pixels' color and position. Then, we extract the uniform super-pixels to detect the color fading defects because the high variance super-pixels are not good for extracting the color difference between the master raster ROI and test raster ROI. We use the DIDMEC method to extract the main colors for all the smooth super-pixels.

The second part extracts the color fading feature vector for each main color by comparing the master and test raster ROI smooth super-pixels in CIE $L^*a^*b^*$ color space. Then, we still use the Directed Acyclic Graph-Support Vector Machine (DAG-SVM) [6] to classify the rank of the color fading defect in the test raster ROI.

We will now introduce the details of raster ROI colorant fading defects detection and the classification procedure.

3. Raster ROI preprocessing

This section introduces the raster ROI preprocessing for colorant fading detection and classification. It includes four parts: raster ROI image registration, SLIC super-pixels separating the master raster ROI, extracting smooth super-pixels, and main colors extraction using DIDMEC. The image registration is to remove the misalignment between the master and test raster ROI. The SLIC super-pixels algorithm separates the master raster ROI based on the color and position of the pixels. After we have the SLIC super-pixels result, we prefer to remove the super-pixel with high variance and keep the smooth super-pixels. We propose the DIDMEC clustering algorithm to automatically extract the main colors from the smooth super-pixels at the end of the preprocessing.

Because the raster ROI image registration part is the same as the previous image registration process [7,8,9], we do not introduce the detail of image registration here and only show the raster ROI image registration result in Figure 3.

3.1 SLIC super-pixels for master raster ROI

After we have the aligned test raster ROI with the master raster ROI, we want to separate the master raster ROI into several small parts because each small part of the raster ROI may have a uniform color, and it is easy to extract the difference between two uniform color areas in the master and test raster ROIs. Before we continue to introduce the SLIC super-pixels method, we want to explain a detail about the master image. In Figure 1, we can find that the color of the master raster ROI (a) is much different from the test raster ROI's good quality image (b). That means there is no color fading defect in the raster ROI, and there is a significant color difference between the digital master image and the scanned test image. So, if we want to use the master image as a standard to detect the color fading defect, we must do the color calibration before we use the master image. Instead of using a color-calibrated master image, we chose an excellent printed quality test image as our master image in this project, such as Figure 1 (b). Because we use this good quality printed image as the color-calibrated master image, we will call this good quality printed test image the master image in the following paragraphs.

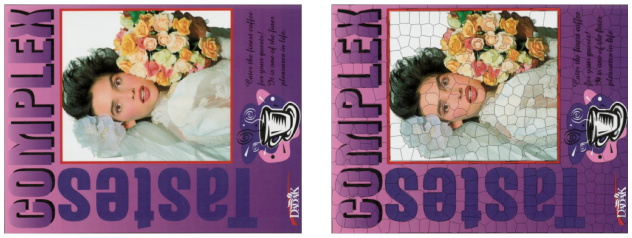
The SLIC algorithm is easy to process. The only input is k, which is the number of approximately equally-sized super-pixels. We first transform the master raster ROI to the CIE $L^*a^*b^*$ color space and initialize the k cluster centers $C_i = [l_i, a_i, b_i, x_i, y_i]^\top$ sampled on a regular grid spaced S pixels apart, where $S = \sqrt{N/k}$ and N is the number of pixels in the raster ROI. To avoid having the centers of the super-pixels fall on the edges of the ROI, we calculate the gradient of the 3×3 neighbor pixels of the center and move it to the lowest gradient position. After we have k initialized cluster centers, we assign each pixel an index i. We calculate the distance measure D for each pixel in the $2S \times 2S$ area for each super-pixel center. We will introduce this distance measure of D later. Because each super-pixel size is approximately $S \times S$, each pixel i will have several distance measures D for neighboring super-pixel centers, and the smallest distance measure D determines the nearest super-pixel center for each pixel i.

After every pixel in the raster ROI has been assigned a nearest super-pixel center, we update the center of every new super-pixel by calculating the mean $[L^*, a^*, b^*, x, y]^\top$ vector of all the pixels belonging to the same super-pixel. Then, we repeat to calculate the distance measure D for each pixel and refind the center of new super-pixels until the centers of super-pixels are almost fixed or we reach a fixed iteration number.

$$
\begin{aligned}
d_c &= \sqrt{(L_j - L_i)^2 + (a_j - a_i)^2 + (b_j - b_i)^2} \\
d_s &= \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2} \\
D &= \sqrt{\left(\frac{d_c}{N_c}\right)^2 + \left(\frac{d_s}{N_s}\right)^2} = \sqrt{\left(\frac{d_c}{m}\right)^2 + \left(\frac{d_s}{S}\right)^2}
\end{aligned} \tag{1}
$$

The distance measure of D is defined in Equation 1. d_c is the color L_2 norm distance between the pixel i and the super-pixel center j, and d_s is the space L_2 norm distance between the pixel i and the super-pixels center j. It is necessary to normalize color proximity and spatial proximity by their respective maximum distance within a cluster if we want to combine the two distances into a single measure. N_s is the maximum spatial distance in a given cluster and $N_s = S = \sqrt{N/k}$. N_c is the maximum color distance in a given cluster, and it is also a constant for each given super-pixel, and we use m to represent this constant.

Figure 4 shows the SLIC super-pixels result of the master raster ROI when k = 1000. In Figure 4 (b), we use the black lines to show each super-pixel's edge.



a. Master Raster ROI b. "SLIC" result of master raster ROI

Figure 4: The SLIC super-pixel result of master raster ROI with $k = 1000$.

3.2 Extract smooth super-pixels

We want to detect the color fading defect using the SLIC super-pixel result, but some super-pixels include multiple colors. This kind of super-pixel is not good for detecting the color fading defect. We prefer to compare the uniform color area between master and test raster ROIs to extract the color differences. So, the next step is to extract the smooth super-pixels from the master raster ROI. We transform the master raster ROI to the CIE $L^*a^*b^*$ color space and use Equation 2 to calculate the variance of the super-pixel. In Equation 2, M is the number of pixels in the corresponding super-pixel, and i indicates each pixel.

$$
\begin{aligned}
&L_{ave} = \frac{\sum_{i=1}^{M} L_i}{M},\ a_{ave} = \frac{\sum_{i=1}^{M} a_i}{M},\ b_{ave} = \frac{\sum_{i=1}^{M} b_i}{M}, \\
&\Delta E_i = \sqrt{(L_i - L_{ave})^2 + (a_i - a_{ave})^2 + (b_i - b_{ave})^2}, \\
&\Delta E_{ave} = \frac{\sum_{i=1}^{M} \Delta E_i}{M}, \\
&\Delta E_{variance} = \frac{\sum_{i=1}^{M} (\Delta E_i - \Delta E_{ave})^2}{M-1},
\end{aligned} \tag{2}
$$

After we calculate the $\Delta E_{variance}$ of each super-pixel, we set a threshold to remove the colorful super-pixels, for which $\Delta E_{variance}$ is higher than the threshold. This threshold is an experimental parameter. We test it with several master raster ROIs and from 3 to 15. Figure 5 shows three examples of using different thresholds to extract the smooth super-pixels. Based on the experimental results, we set the threshold to be 5, as shown in Figure 5 (a).

a. Threshold is 5 for smooth super-pixels extraction b. Threshold is 10 for smooth super-pixels extraction c. Threshold is 10 for smooth super-pixels extraction

Figure 5: Different threshold to extract the smooth super-pixels.

3.3 Main colors extraction with DIDMEC

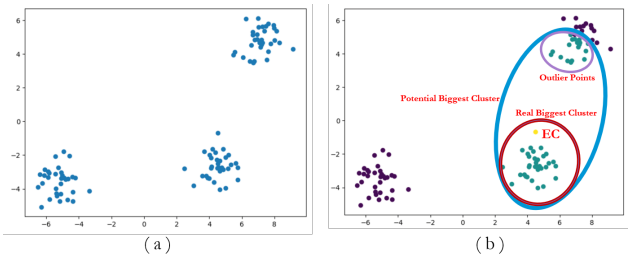
In Section 3.2, we have extracted the smooth super-pixels from the raster ROI. Usually, there are three or four main colors in each raster ROI. We prefer to extract the super-pixels belonging to the same main color in the raster ROI and use these same color super-pixels to detect the colorant fading defects. So the first step is to extract the main colors of the raster ROI.

There are many unsupervised cluster classification methods, such as K-means and mean shift algorithms [10,11]. We have used these algorithms to extract the main colors from raster ROI, but the result is not good enough. There are two reasons: One reason is that these cluster classification algorithms classify all the smooth super-pixels in the raster ROI, but we only want to extract the most similar main cluster color super-pixels and do not need to classify all the smooth super-pixel. Another reason is all these algorithms need many calculations. In this project, we propose a new approach to unsupervised data clustering called Data Inter-Distance MEdiated Clustering (DIDMEC), which can extract the most similar main color super-pixels only and very efficiently. We will introduce this algorithm first.

To explain our new cluster classification algorithm, we use 100 sample points, which belong to three different Gaussian distributions. Here, we assume from the outset that there are three clusters. We will explain later how we choose the number of clusters for the data in a more general setting, where this number is not known *a priori*. We use the position (x, y) coordinates of the sample points to classify them. Figure 6 (a) shows 100 sample points; we can see that these sample points belong to three clusters.

The first step is to calculate the Euclidean Distance between every two sample points and get the distance matrix for the sample points. $D_{i,j}$ denotes the distance from Point i to Point j. This distance matrix shows the distance between every two points, so the diagonal elements of this matrix always are zero; this distance matrix must be symmetric because the distance from Point A to Point B must be the same as the distance from Point B to Point A.

Intuitively, the assignment of points to clusters can be based on the distance between two different points. We want to find the best threshold to separate the in-cluster and out-



(a) (b)

Figure 6: 100 sample point distribution used to illustrate the new DIDMEC clustering method. (a) Original point set. (b) Initial clustering result after applying threshold based on mean plus standard deviation of data in leftmost peak of histogram shown in Figure 7.

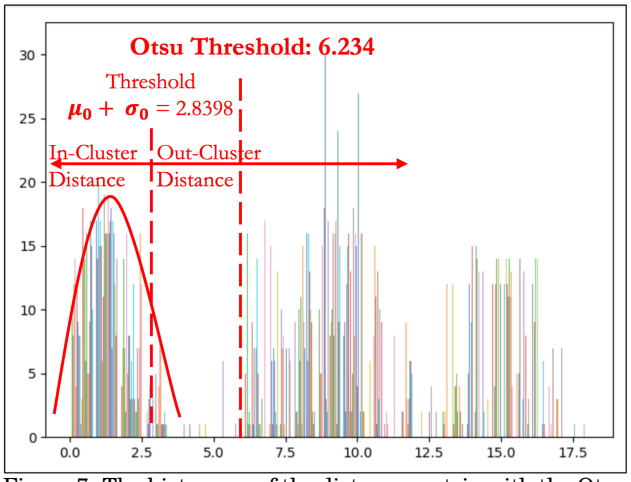


Figure 7: The histogram of the distance matrix with the Otsu thresholding result.

cluster distances. We plot the histogram distance matrix to find this best threshold, as shown in Figure 7. We can see that the apparent valley of this histogram is at a distance of around 5, and there are peak values before and after this valley. We define the distance values to the left of this valley as the in-cluster distances and the distance values to this valley's right as the out-cluster distances. First, we use Otsu's method [12] to calculate the valley of this histogram because Otsu's method has demonstrated success in separating background and foreground pixels by using the gray value histogram [13]. However, the result from Otsu's method is not the best threshold. Instead, we use the average plus the standard deviation of the distances as the best threshold, which in this case is less than the Otsu threshold. We will describe below how we calculate the mean and standard deviation. Here, we assume that the distribution of the in-cluster distances and the out-cluster distances are Gaussian distributed and draw on hypothesis test theory to minimize the probability of a Type II error [14]. However, we still need to use Otsu's method to separate the leftmost peak in the histogram from the one to its right. We have to calculate the mean and standard deviation of the data below the threshold provided by Otsu's method.

After we get the best threshold to classify the in-cluster and out-cluster distances, we use it to process the entire distance matrix. We label the distance values less than the threshold as True and the distance values larger than the threshold as False, as shown in Figure 8.

	Point 1	Point 2	Point 3 (EC)	Point 4	…	Point 99	Point 100	Sum_{True}
Point 1	True	False	False	False	…	False	True	24
Point 2	False	True	True	True	…	False	True	30
Point 3 (EC)	False	True	True	True	…	True	False	36
Point 4	False	True	True	True	…	False	False	…
…	…	…	…	…	…	…	…	…
Point 99	False	False	True	False	…	True	False	30
Point 100	True	True	False	False	…	False	True	28

Figure 8: The True/False matrix for the 100 sample points shown in Figure 6.

In the True/False matrix, "True" means the corresponding points belong to the same cluster, and "False" means the corresponding points belong to different clusters. We want to extract the biggest cluster from this dataset using the True/False matrix. We calculate the number of True values for each Point by summing all the True's in each row. The Point (row) with the maximum number of True values is the estimated center (EC) point connected to the most other points within the in-cluster distance. From Figure 8, Point 3 has the maximum number (36) of True connections to other points. So Point 3 is the estimated center of the biggest cluster. In Figure 6 (b), we color Point 3 in Figure 6 (a) yellow, and all the points within the in-cluster distance to Point 3 with green. All other points are colored purple. Now, all the green points may belong to the potential biggest cluster of the dataset.

However, some of these green points may not belong to the same cluster. They are only labeled green because they are near the estimated center of the in-cluster distance points. As shown in Figure 6 (b), some green points do not belong to the largest cluster. We call these points *outlier points*.

	Point 2	Point 3	Point 4	…	Point 99
Point 2	True	True	True	…	False
Point 3	True	True	True	…	True
Point 4	True	True	True	…	False
…	…	…	…	…	…
Point 99	False	True	False	…	True

36

Figure 9: The potential biggest cluster True/False matrix.

To remove the outlier points from the potential biggest cluster, we first extract all the points belonging to the potential biggest cluster from the True/False matrix to build the potential biggest cluster True/False matrix, shown in Figure 9. For our example dataset, this matrix has 36 rows. This matrix will have entries labeled False corresponding to points farther apart than the in-cluster distance threshold. However, each such entry corresponds to a pair of points – one of which may be in the biggest cluster and the other one not in the biggest cluster. Once we remove the outlier points from the potential biggest cluster, the matrix in Figure 9 will have no entries labeled False; and all the remaining points will be no farther apart than the in-cluster distance threshold.

The following three properties can help us to remove the outlier points from the potential biggest cluster.

Property 1. The number of False entries in the potential biggest cluster True/False matrix is twice the real num-

ber of False connections, because each False connection appears twice in the matrix.

Property 2. Each False entry indicates a connection between one point that is an inlier and another point that is an outlier. The number of the False connections of all inlier points is the same as the number of the False connections of all outlier points.

Property 3. Each outlier point will have more False connections than the inlier points because the outlier points do not belong to the real biggest cluster.

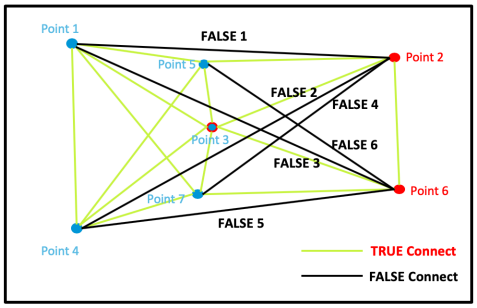


Figure 10: Seven-point example illustrating the removal of outlier points.

We use the simple example shown in Figure 10 to explain how to remove the outlier points from the potential biggest cluster. In this example, we have seven sample points; and all of these points are connected to the estimated center Point of 3 within the in-cluster distance threshold. Point 1, Point 3, Point 4, Point 5, and Point 7 belong to the real biggest cluster, whereas Point 2 and Point 6 are outlier points. Each connection of an outlier point to an inlier point will be labeled False, such as FALSE 1 in Figure 10. We also can observe this condition in the True/False matrix, as shown in Figure 11. The outlier points cause all the False connections, and if we remove these outlier points, the True/False matrix will be filled with all True values without any entries labeled False. The number of False connections of all inlier points is four in this example, and the number of False connections of all outlier points also is four. This confirms Property 2. There are only four False connections; there are eight False values in the True/False matrix. This confirms Property 1. The number of False connections to the outlier points is three and three. They are all larger than the number of False connections to any given inlier point.

	1	~~2~~	3	4	5	~~6~~	7	# False
1	True	False1	True	True	True	False3	True	2
~~2~~	False1	True	True	False2	True	True	False4	3
3	True	True	True	True	True	True	True	0
4	True	False2	True	True	True	False5	True	2
5	True	True	True	True	True	False6	True	1
~~6~~	False3	True	True	False5	False6	True	True	3
7	True	False4	True	True	True	True	True	1

Number of False Connections of Inlier Pointers 2+2+0+1+1 = 6

Number of False Connections of Outlier Pointers 3+3 = 6

Figure 11: Seven-point True/False matrix for the data set shown in Figure 10.

We apply these properties to the potential biggest cluster True/False matrix. We first calculate the number of False connections for each point, as shown in Figure 11. Then, we sort the number of False connections of each point from maximum to minimum. For the next step, we add the sorted numbers from the maximum value to the minimum value, one by one, until the partial sum is equal to half the sum of the whole list. The outlier points are the number of False connections included in this partial sum. In this example, the outlier points are Point 2 and Point 6, because $3+3=6=12/2$.

After using this method to remove the outlier points from the potential biggest cluster among the 100 date points, shown in Figure 6, we can extract the real biggest cluster from these 100 points, as shown by the yellow points in Figure 12 (a). All the values of this biggest cluster True/False matrix are True.

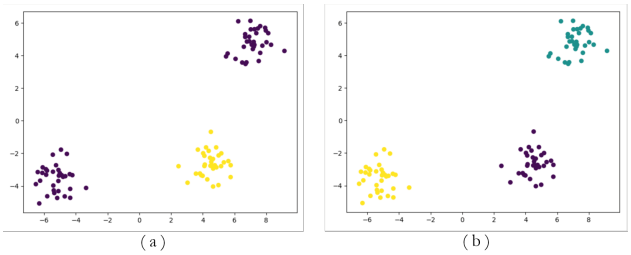


Figure 12: Illustration of clustering results for the data set shown in Figure 6. (a) The biggest cluster extracted from 100 sample points with points colored yellow. The rest of the points are colored purple, and have not yet been assigned to any cluster. (b) Final clustering result.

After we extract the biggest cluster sample points, we modify the original 100 sample points True/False matrix. All the rows and columns corresponding to the biggest cluster sample points are set to be False. Next, we repeat the process described above to extract the second biggest cluster using this True/False matrix. Then, we repeat the process once more to obtain our final three clusters. The result is shown in Figure 12 (b). We refer to this as Stage 1 of the DIDMEC clustering algorithm.

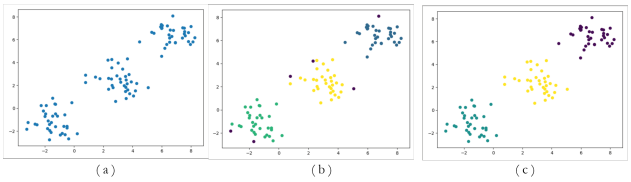
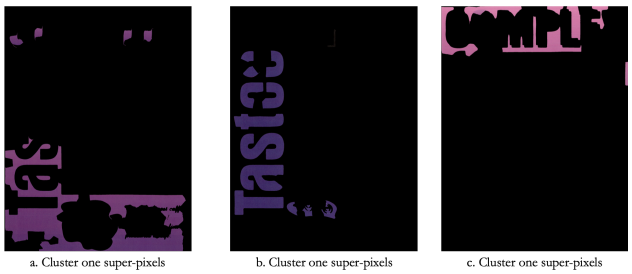


Figure 13: Illustration of a 100-point data set generated from three Gaussian distributions for which some unclustered points remain after Stage 1. (a) Initial set of 100 unclustered data points. (b) Clustering result after Stage 1. The points assigned to the three clusters are colored green, yellow, and blue. The unclustered points are colored purple. (c) Final clustering result after application of Stage 2.

As mentioned earlier, we assumed here at the outset that there were three clusters. We will later discuss below how we decide how many clusters there should be. However, given a fixed number of clusters, there may remain a few unclustered points at the conclusion of Stage 1. These are shown as purple points in Figure 8 (b), which is a different set of data points than that (Figure 6) used to illustrate Stage 1. We

classify these points to their nearest cluster center to complete the clustering of the data set. This is Stage 2 of our algorithm. As is the case with Stage 1, in Stage 2, we assume the best number of clusters is known. The final result for this data set is shown in Figure 8 (c).

Now, we address the question of how to choose the number of clusters. In some cases, this number is fixed by the nature of the application. If it is not, we use the Gap Statistics method [15] to determine the best number of clusters.



a. Cluster one super-pixels b. Cluster one super-pixels c. Cluster one super-pixels

Figure 14: The three main cluster color super-pixels result.

Here, we apply this method to extract main cluster colors from the smooth super-pixels in the raster ROI. The result is shown in Figure 14. We will use these three main clusters of color results to predict the depleted cartridge.

4. Raster ROI color fading detection and classification

In this section, we extract the color fading feature vector by comparing the difference between the master and test raster ROI and use this feature vector to detect and classify the test raster ROI color fading defect.

4.1 Color fading defect feature vector extraction

The first step is to transform the master and test raster ROI to the CIE $L^*a^*b^*$ color space because the most common way to assess the difference between different colors is to calculate the ΔE value, which is the Euclidean distance between two colors in CIE $L^*a^*b^*$ color space.

In the color fading feature vector, we calculate the average L^*, a^*, b^* for the master and test raster ROIs. These are the first six features in the color fading feature vector. We then calculate the average ΔE between the master and test raster ROIs for smooth super-pixels. We calculate the ΔE value for each pair of corresponding smooth super-pixels in the master and test raster ROI first by calculating the Euclidean distance between the average L^*, a^*, b^* of corresponding master and test raster ROI super-pixels. We calculate the average of the whole set of raster ROI super-pixel ΔE values and the smooth raster ROI super-pixel ΔE values separately for the color fading features. These two ΔE values and the average of the L^*, a^*, b^* values of the master and test raster ROIs are the features to classify the color fading defect.

4.2 Color fading defect multi-classification

After we extract the feature vector of the raster ROI color fading defect, we use this feature vector to classify the raster ROI color fading defect into four different ranks. Rank A means no color fading defects in the raster ROI; Rank B means there are color fading defects in the raster ROI, but the defects do not affect the regular use of the printed page; Rank C means there is an observable color fading defect and people can find it easily; Rank D means there are a lot of color fading defects and the defects influence the regular use of the printed page. These ranks have been assigned by HP Inc. Figure 1 shows an example of the four different rank samples of raster ROI color fading defects.

We have 52 raster ROI streak defects images, and they are 15 images of Rank A, 5 images of Rank B, 15 images of Rank C, and 17 images of Rank D. The multi-classification model is Directed Acyclic Graph-Support Vector Machine (DAG-SVM). To evaluate this multi-classification result, we apply cross-validation for this model [16]. We separate 52 raster ROIs into 5 folds, such as (10/10/10/11/11) raster ROI color fading defects ground truth for each fold. Each fold is used as the test set one time, and the other four folds are used as training sets. The final confusion matrix for this five-folds cross-validation is shown in Table 1. The accuracy of this multi-classification model is 82.7%.

Table 1: Raster ROI color fading defect DAG-SVM classification confusion matrix

	Predict A	Predict B	Predict C	Predict D
Real A	13	2	0	0
Real B	0	4	1	0
Real C	0	1	12	2
Real D	0	0	3	14

5. Conclusion

In this paper, we proposed a colorant fading defect detection and classification method. The input is the master raster ROI and the corresponding test raster ROI, and the output is the colorant fading defect feature vector, and rank classification.

To achieve this result, we design two parts in our procedure: raster ROI preprocessing, and color fading classification. In the first part, we use an image registration algorithm to align the test raster ROI to the master raster ROI. For the master raster ROI, we use the SLIC super-pixels algorithm to separate pixels based on their color and position. Then, we choose smooth super-pixels. We proposed a new method called Data Inter-Distance Mediated Clustering (DIDMEC) for unsupervised clustering of a set of data points. Unlike previously proposed methods, DIDMEC only computes the distances between pairs of points in the data set one time. Our experimental results show that DIDMEC can achieve essentially the same accuracy as K-Means, but at a significantly lower computational cost for a moderately sized data set. In the second part, we transform the color space to CIE $L^*a^*b^*$ and extract the ΔE value between the master and test raster ROIs. We use these features to classify the rank of the color fading defect in test raster ROI.

References

[1] R. Zhang, E. Maggard, Y. Bang, M. Cho, and J. Allebach, "Region of interest extraction for image quality assessment," *Electronic Imaging, Image Quality and System Performance XVII,* San Francisco, CA, 26 January 2020.

[2] Z. Xiao, M. Nguyen, E. Maggard, M. Shaw, J. Allebach, and A. Reibman, "Real-time print quality diagnostics," *Electronic Imaging, Image Quality and System Performance XIV,* Burlingame, CA, January 2017.

[3] R. Achanta, A. Shaji, K Smith, A. Lucchi, P. Fua and S.Süsstrunk, "SLIC superpixels compared to state-of-the-art superpixel methods," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 34(11), pp.2274-2282, 2012

[4] J. MacQueen, "Some methods for classification and analysis of multivariate observations," *In Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability,* Vol. 1, No. 14, pp. 281-297, June, 1967.

[5] A. Likas, N. Vlassis, and J.J Verbeek, "The global k-means clustering algorithm," *Pattern Recognition,* 36(2), pp.451-461, 2003.

[6] P. Chen, and S. Liu, "An improved DAG-SVM for multi-class classification," *IEEE Fifth International Conference on Natural Computation,* (Vol. 1, pp. 460-462), August, 2009.

[7] R. Zhang, E. Maggard, Y. Bang, M. Cho, and J. Allebach, "A comprehensive system for analyzing the presence of print quality defects," *Electronic Imaging, Image Quality and System Performance XVII,* Burlingame, CA, January 2020.

[8] R. Zhang, Y. Bang, M. Cho, M. Shaw, and J. Allebach, "Streak Detection In Color Printed Customer Content," *Electronic Imaging, Color Imaging XXVI: Displaying, Processing, Hardcopy, and Applications,* Burlingame, CA, January 2021.

[9] R. Zhang, Y. Bang, M. Cho, M. Shaw, and J. Allebach, "Color Text Fading Detection," *Electronic Imaging, Color Imaging XXVI: Displaying, Processing, Hardcopy, and Applications,* Burlingame, CA, January 2021.

[10] D. Comaniciu, and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Transactions on Pattern Analysis and Machine iIntelligence,* 24(5), pp.603-619, 2002.

[11] D. Comaniciu, and P. Meer, "Mean shift analysis and applications," *In Proceedings of the Seventh IEEE International Conference on Computer Vision,* (Vol. 2, pp. 1197-1203), September, 1999.

[12] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Transactions on Systems, Man, and Cybernetics,* 9(1), pp.62-66, 1979.

[13] C.A. Glasbey, "An analysis of histogram-based thresholding algorithms," *CVGIP: Graphical Models and Image Processing, 55(6),* pp.532-537, 1993.

[14] J.P. Shaffer, "Multiple hypothesis testing," *Annual Review of Psychology,* 46(1), pp.561-584, 1995.

[15] R. Tibshirani, G. Walther, and T. Hastie, "Estimating the number of clusters in a data set via the gap statistic," *Journal of the Royal Statistical Society: Series B (Statistical Methodology),* 63(2), pp.411-423, 2001.

[16] P. Refaeilzadeh, L. Tang, and H. Liu, "Cross-validation," *Encyclopedia of Database Systems,* 5, pp.532-538, 2009.

Author Biography

Runzhe Zhang *received his B.S. degree in Mechanical Engineering from Qingdao University of Technology (2013), Shandong, China and received the M.S. degree in Mechanical Engineering from Boston University, MA, USA. Currently, he is pursuing the Ph.D. in Electrical and Computer Engineering at Purdue University. His research areas include digital image processing, computer vision, and machine learning.*

Yousun Bang *is a manager of Image Quality Part in the Imaging Lab at HP Printing Korea Co. Ltd. She received her BS and MS in mathematics from Ewha Womans University, Seoul, Korea in 1994 and 1996, and her Ph.D. in the School of Electrical and Computer Engineering, Purdue University, West Lafayette, Indiana in 2005. She worked for Samsung Advanced Institute of Technology and Samsung Electronics Company from 2004 to 2017. Her current research interests include image quality diagnosis and metrics and ML/DL based prediction for smart device services.*

Minki Cho *is an engineer with HP Printing Korea. He received B.S.(1997) and M.S.(1999) in electrical engineering from the Inha University, Korea. From 2003 to 2017, he worked for Samsung Electronics and Samsung Advanced Institute of Technology. His research areas are print image processing, print image quality diagnosis and calibration. Recently, he is researching the above interests using ML and DL.*

Mark Shaw *received his B.S. in Printing and Publishing Technology from University of Herfordshire in 1997 and his M.S. in Color Science from Rochester Institute of Technology in 1999. Then he went to Boise State University to take Electrical Engineering classes to meet the pre-requisites to study under Prof. Jan Allebach at Purdue. Finally he received his Ph.D. in Electrical and Computer Engineering from Purdue University in 2015. He is now a principal color and imaging Distinguished Technologist at HP Inc.*

Jan P. Allebach *is Hewlett-Packard Distinguished Professor of Electrical and Computer Engineering at Purdue University. Jan P. 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 Electronic 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, the IS&T/OSA Edwin Land Medal, the IS&T Johann Gutenberg Prize, and is a member of the National Academy of Engineering.*