Machine Learning for Forensic Printer Identification

Alexander Gokan, Zuoqian Xu

Purdue VIP Imaging And Printing

Vertically
Integrated
Projects

Introduction

Papers nowadays are used widely in many functions, such as currency, checks, contracts etc. But due to the improvement of digital image processing techniques and the advancement of printers, we need to know if these papers were printed by specific printers. For example, cash were printed by government authorized printer. But inkjet printer make it surprisingly easy to print fake cash.

Counterfeiting is a crime as old as money itself, with new and ever more sophisticated methods of forgery being developed every day. While they may copy the macro-scale features a bill (or document) what is often neglected is the intrinsic signature of the printer itself – details far too small for the naked human eye to detect.

The intrinsic features of the printer itself allow a document to be uniquely identified by our algorithm my identifying certain features, and comparing them to pre-collected data from other printers in order to match them

Our goal on this project is to investigate different patterns from scanned images that from different brands of printers. In the end, we will be able to recognize a image is from which printer.

The method we investigate includes image processing and machine learning techniques.

The three approach we are going to use are:

1. Dot Size and Standard Deviation
2. Dot compactness
3. NDSDF(Nearest Dot Sector Density Function)

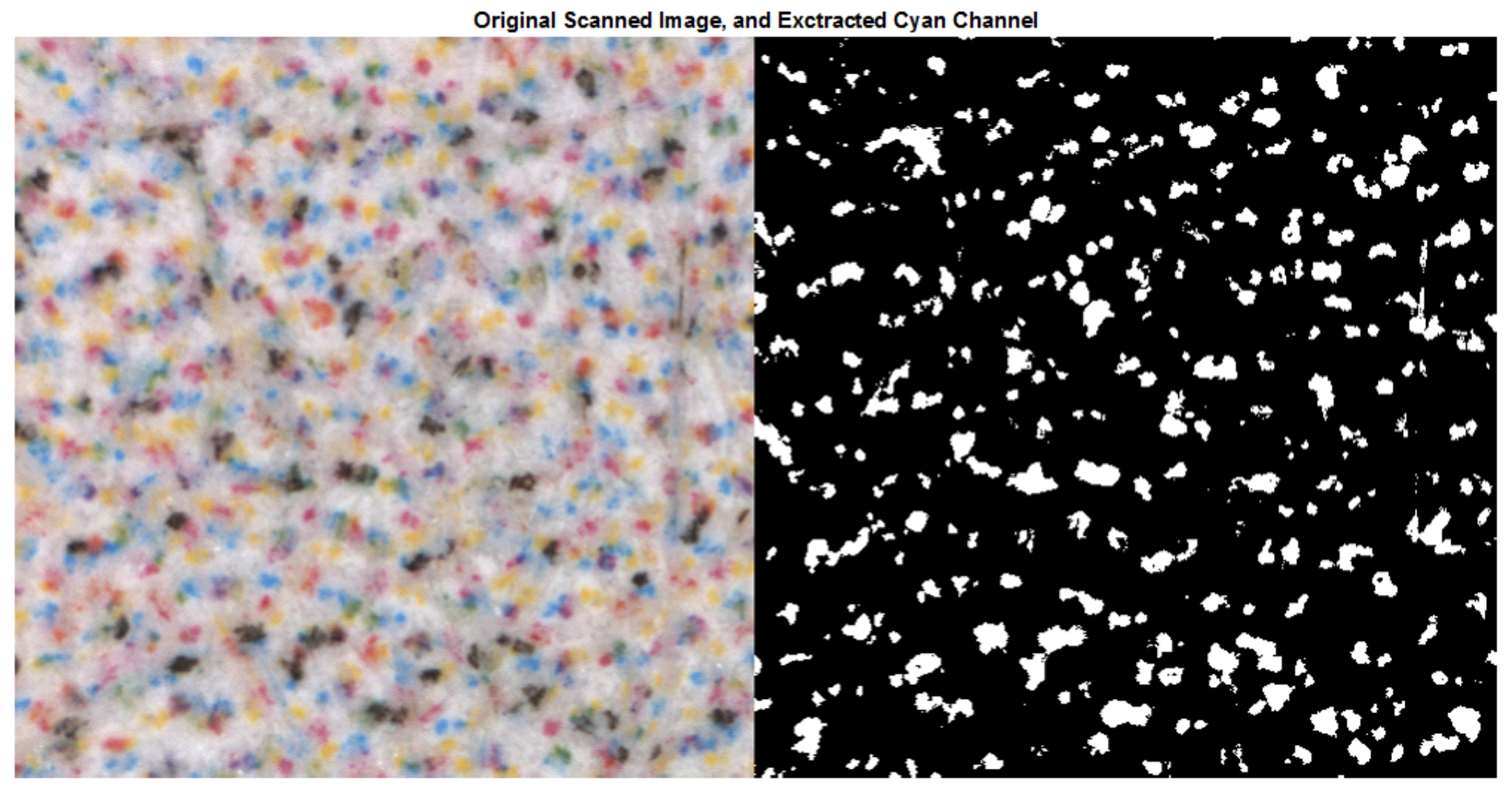
Ink Drop Isolation

In order to perform any sort of analysis on the ink drops, we first have to distinguish them from the backdrop.

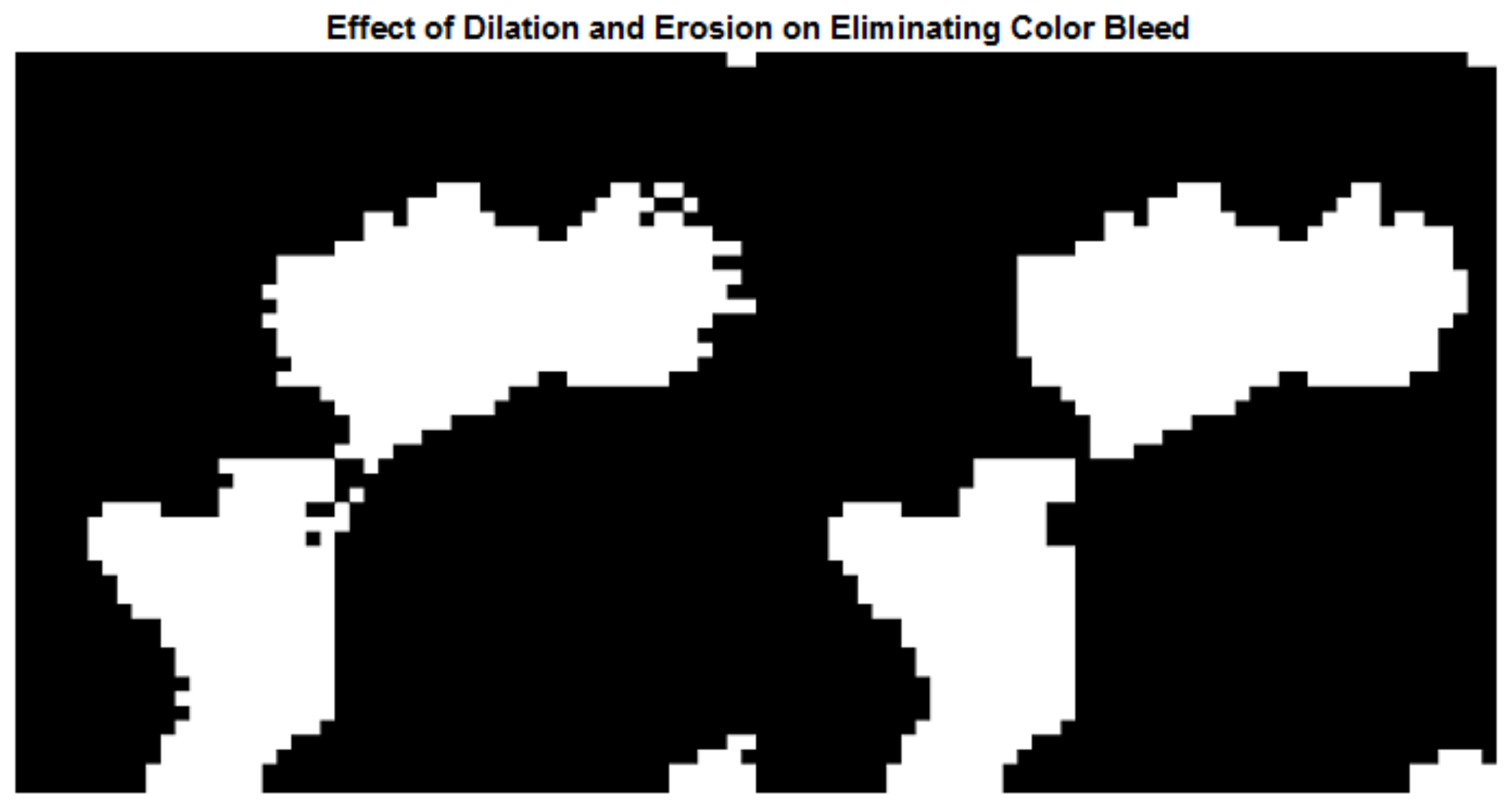
- First, the image is converted from RGB to LAB space
- Each pixel is assigned to an ink drop based on a KNN algorithm
- Determining the representative colors for the document is the only step requiring human input

The results of this method are shown below. They work as a rudimentary mask, but suffer from color bleed and high noise levels.

In order to help separate ink drops which have bled into each other, and reduce the noise inherent in the scan, a method of iterative erosion and dilation is applied, along with a simple removal of all ink drops with an area less than 35 pixels. From this final mask, we can identify each separate ink drop, and features about it such as its centroid, which is passed to other analysis functions



Original Scanned Image, and Exctracted Cyan Channel

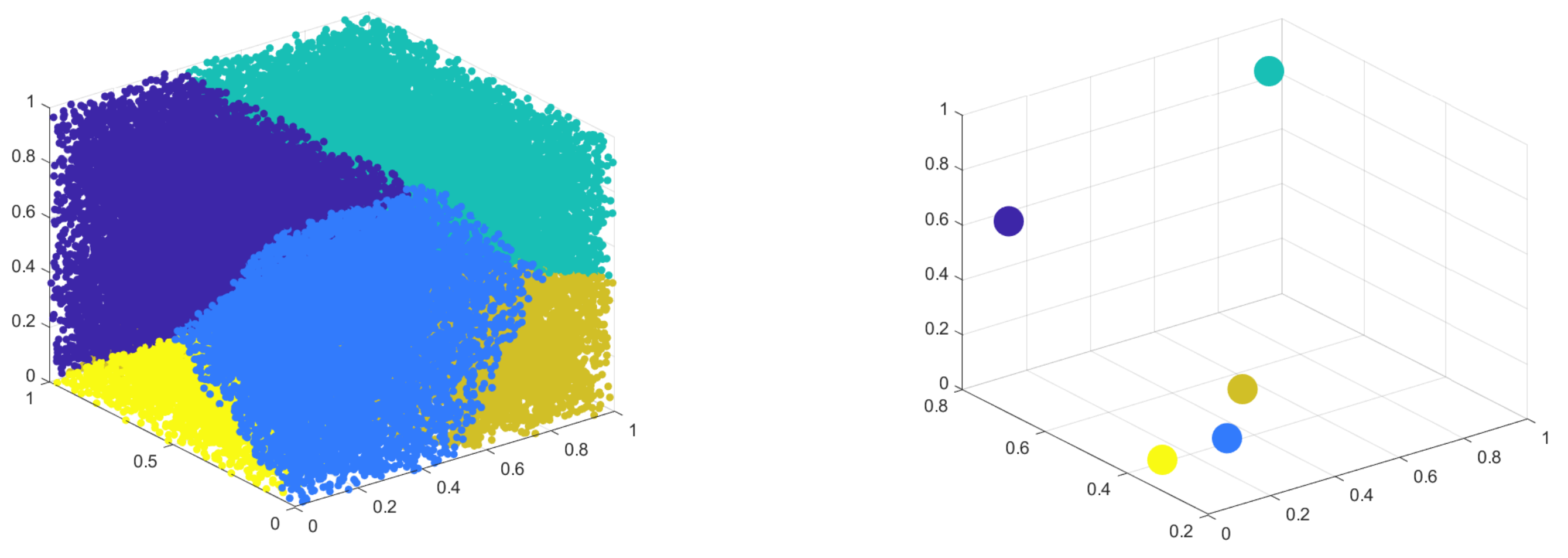


Effect of Dilation and Erosion on Eliminating Color Bleed

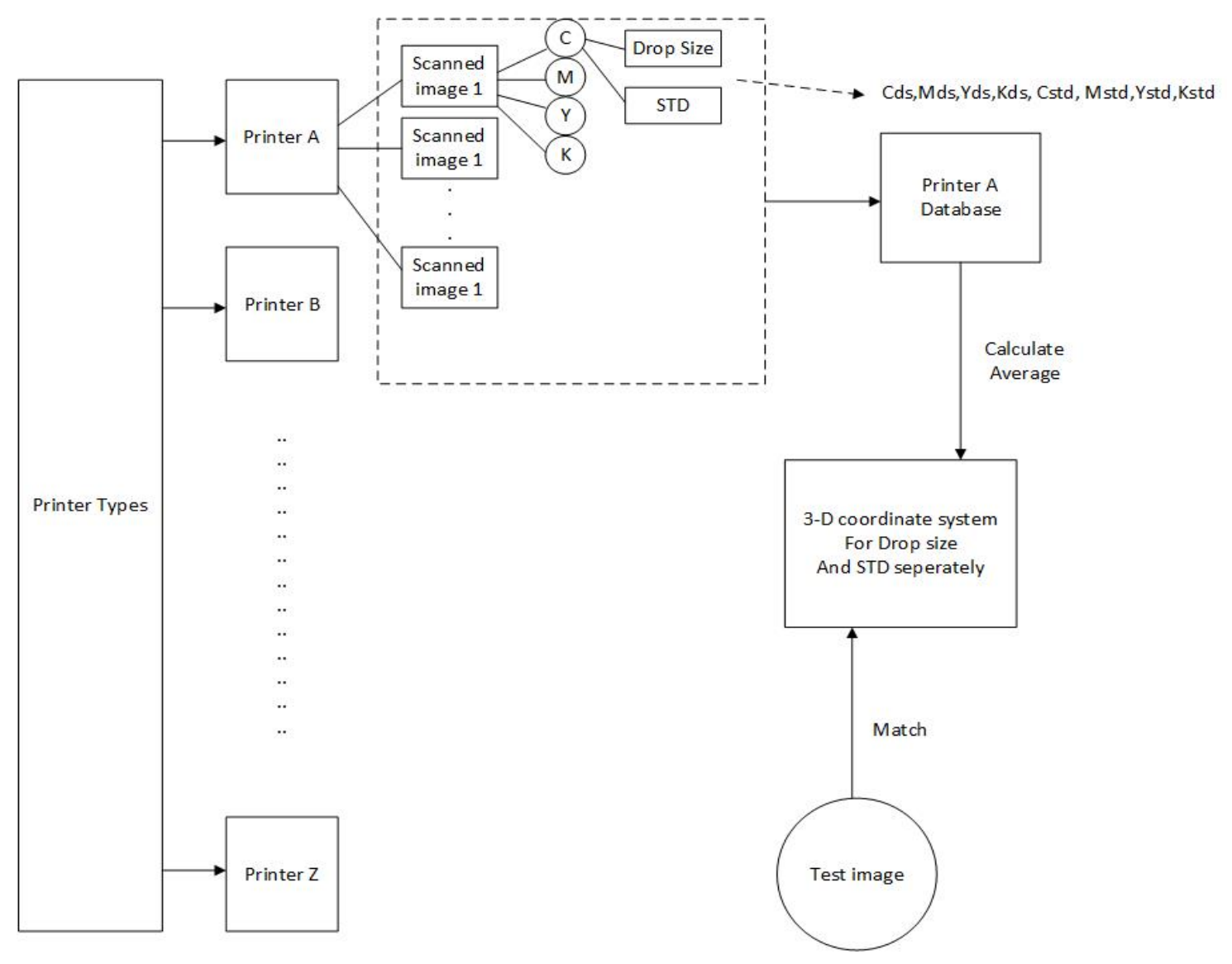
K-Nearest-Neighbors (KNN) Algorithm

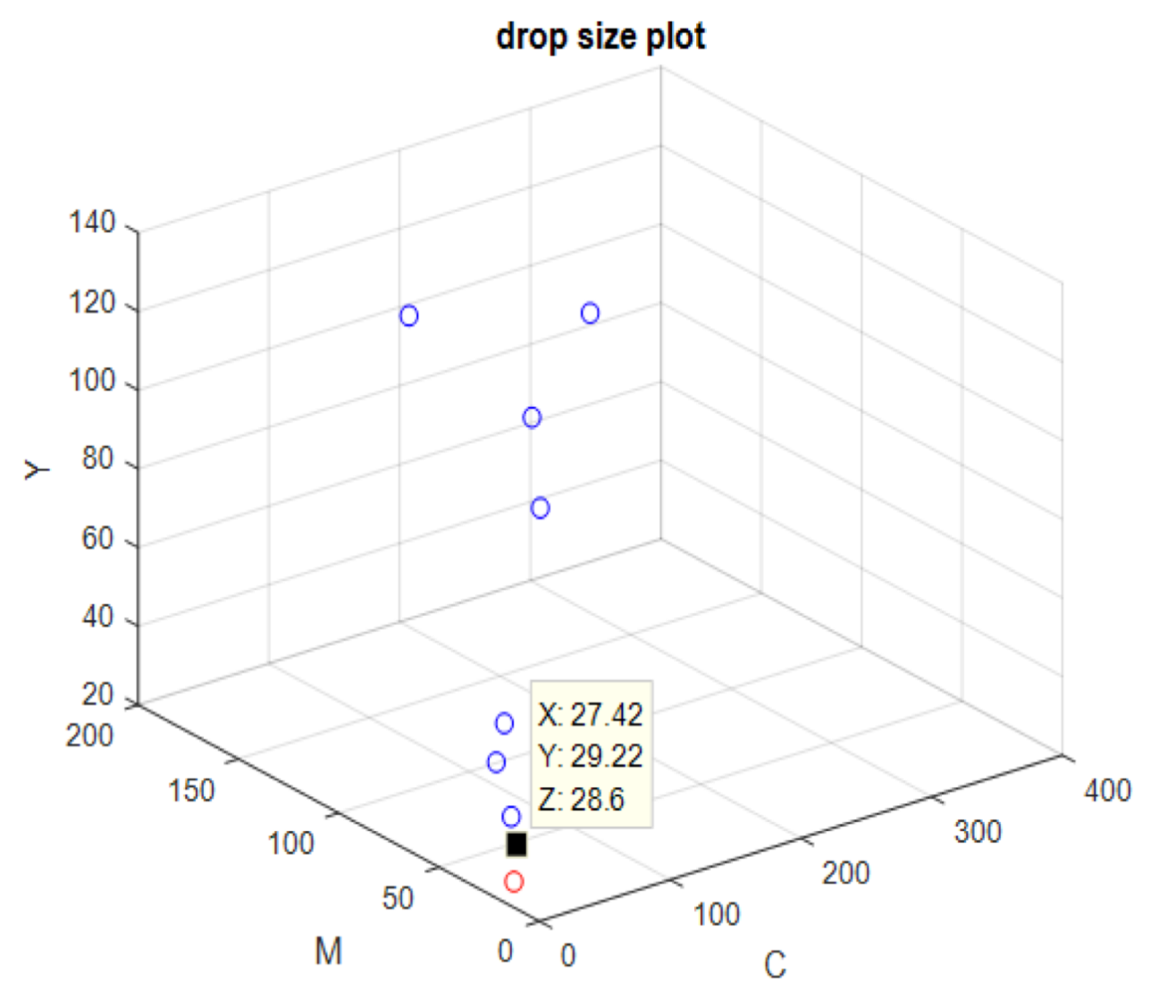
The K-Nearest-Neighbors algorithm attempts to find which pre-determined data point it most closely matches. It works by finding the Euclidian distance between the test vector $[t_1, t_2, ..t_n]$ and each of the pre-determined data vectors for each printer. The test point is then associated or "matched" with the data point that has the lowest distance (highest score in the score matrix below). This allows a "score" to how likely the algorithm thinks a document came from a certain printer

Below is an example showing how the KNN algorithm can be used to match data based on 3-dimensional location to a "printer". Each point on the left is matched to a "printer" on the right based on how far away it is. In a similar manner, each test document (a data point on the left) is matched to a printer (on the right) based on its distance in a higher-dimensional distance (in our case, 11 data dimensions)



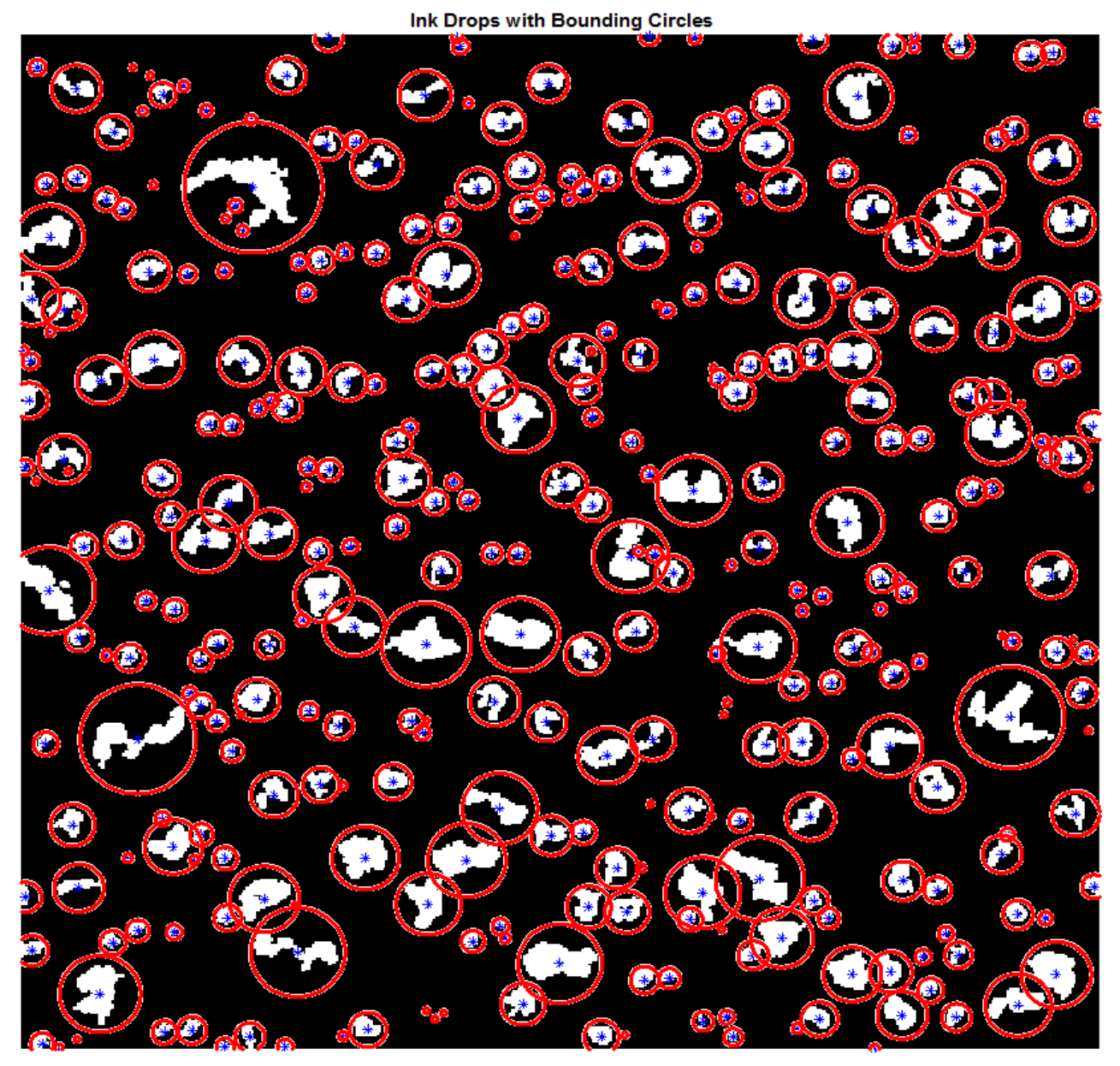
Drop Size and STD analysis





Ink Drop Compactness analysis

Compactness measures the shape of a droplet of ink, by comparing its area to the area of the smallest circle that encloses the entire drop. Drops that are very round have a high compactness, while drops with a stretched shape have a low compactness. Information about the compactness, such as its mean and standard deviation in each color channel can be used as data points in the KNN algorithm.



Ink Drops with Bounding Circles

Example of Low vs High Drop Compactness

Nearest-Dot-Sector Angle Sector Analysis

The nearest dot-sector density function finds the distribution of the angles in that the ink drops are lined up in. The distribution of angles can reveal information such as the direction of the print head. For example, if the print head is moving slightly downward as it moves to the right across the page, the sector histogram will be left skewed (for example Canon MG3620), and if the print head moves up as it moves to the right, the histogram will be right skewed

The angles are categorized in sectors to make the analysis of their angle histograms simpler. Matching the raw data, while more precise, would be computationally much more intensive

The results are analyzed by matching the histogram of a test image to the histogram of a known printer, using the KNN algorithm detailed in the "K-Nearest Neighbors" section

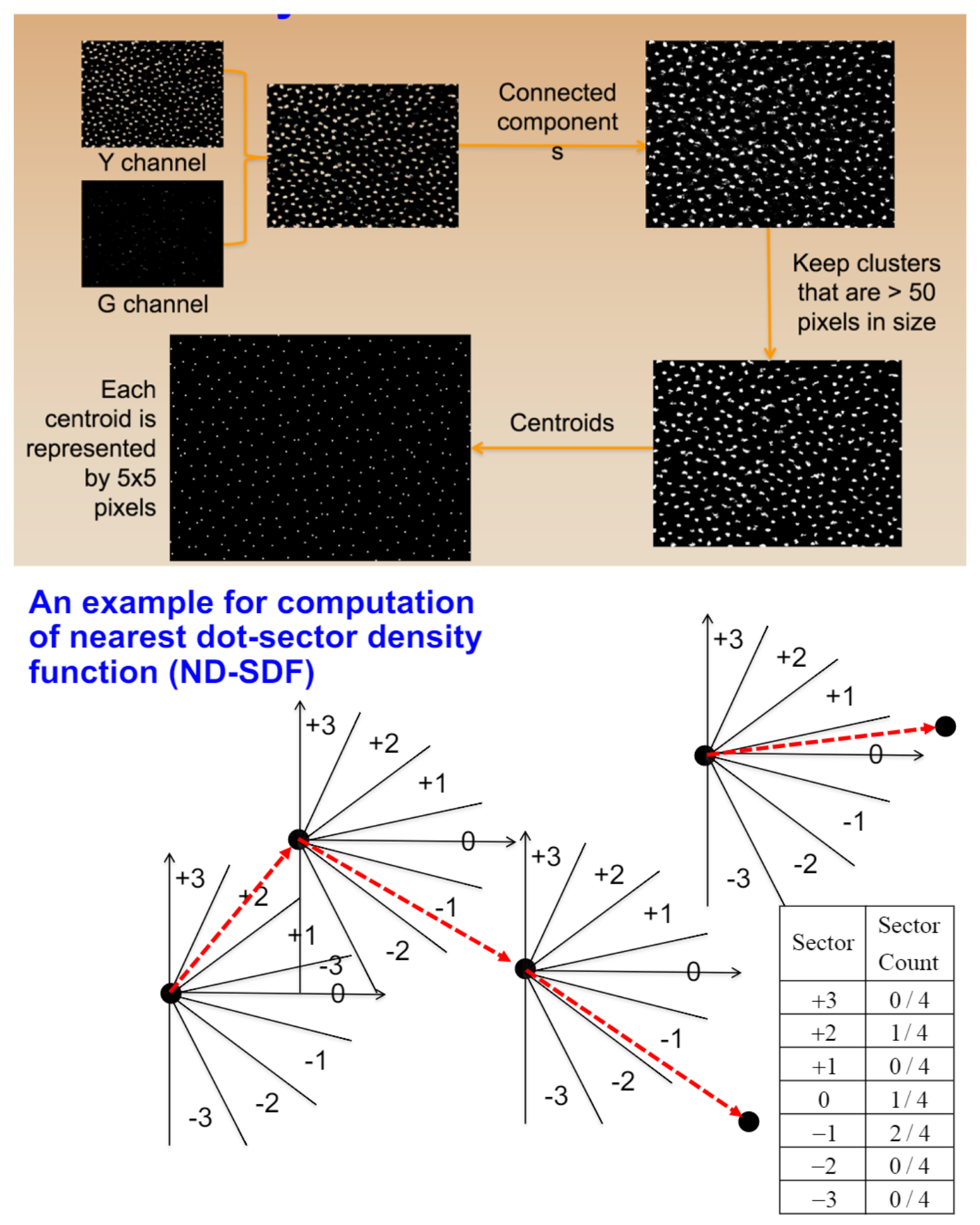
Procedure:

1) The centroids are passed in from the "Ink Drop Isolation" function

2) The distance between each drop centroid is found, and the angle is computed between each drop and its nearest neighbor

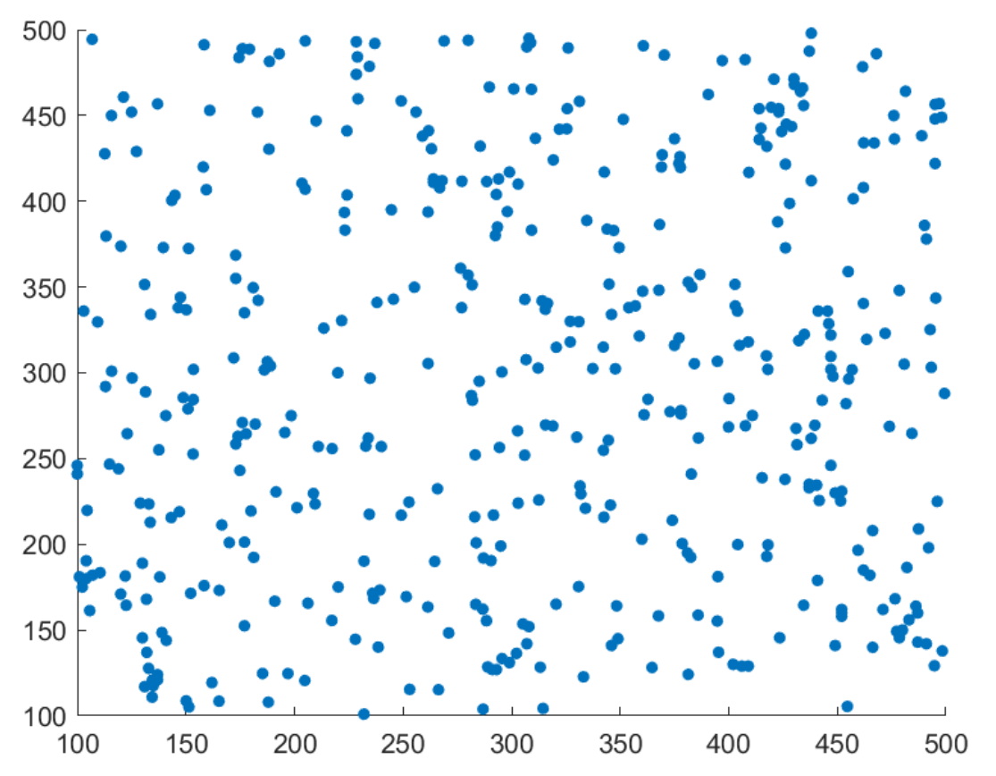
3) The angle is categorized into a sector (the meaning of each sector is explained in the example image), and is counted.

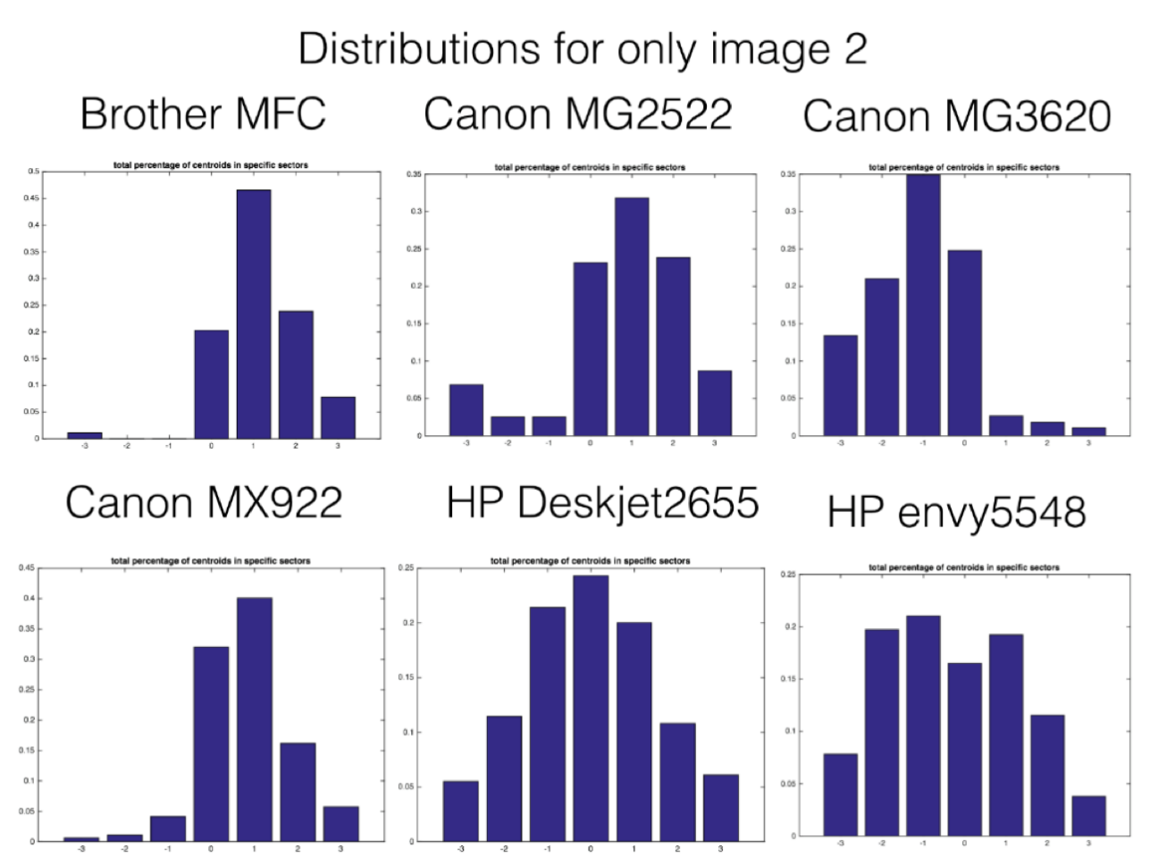
4) We can analyze these results by matching the histograms. For example, if the distribution we get is right or left skewed, the new can know that the centroids are arranged up or down a little bit with respect to the dots they form angle from. Results are shown below:


Y channel
G channel
Connected component s
Keep clusters that are > 50 pixels in size
Centroids
Each centroid is represented by 5x5 pixels
+3
+2
+1
0
-1
-2
-3

An example for computation of nearest dot-sector density function (ND-SDF)

Sector	Sector Count
+3	0 / 4
+2	1 / 4
+1	0 / 4
0	1 / 4
−1	2 / 4
−2	0 / 4
−3	0 / 4





Results/Score Matrix (Combined)

$$Score = -1 * log_{10}(norm(t-d))$$
$$t = test\ data\ of\ mystery\ printer$$
$$d = precalculated\ data\ for\ a\ given\ printer$$

The correct printer (along the diagonal) is always in the algorithm's top 2 choices, with varying degrees of confidence

		Brother	Canon	Canon	Canon	Epson	HP	HP
		MFC	MG3620	MG2522	MX922	XP340	Deskjet 2655	Envy 5548
Brother	MFC	-0.1502	-0.1597	-0.24134	-0.20532	-0.25493	-0.17855	-0.26991
Canon	MG3620	0.041594	0.076145	-0.03734	0.175084	-0.09251	-0.08538	-0.22002
Canon	MG2522	-0.07344	0.170696	0.515785	-0.08354	0.311777	0.022996	-0.15837
Canon	MX922	0.17222	0.751384	0.828677	0.795393	0.38411	0.086835	-0.15076
Epson	XP340	0.167586	0.302426	0.309125	0.308478	0.252107	0.09767	-0.13163
HP	Deskjet 2655	0.005728	-0.1343	-0.13289	-0.13538	-0.08036	-0.01453	-0.13969
HP	Envy 5548	1.316034	1.299167	1.360665	1.351071	1.206055	1.632878	2.212862

References

1) Jan Allebach, Electronic Imaging Systems Laboratory (EISL) Purdue University, "Intrinsic Signatures of Inkjet Devices"

2) Purdue University: ECE438 - Digital Signal Processing with Applications "Lab 10b"

3) Pei-Ju Chiang‡ , Nitin Khanna† , Aravind K. Mikkilineni‡ Maria V. Ortiz Segovia† , Sungjoo Suh† Jan P. Allebach† , George T. C. Chiu‡ , Edward J. Delp†, "Printer and Scanner Forensics"

4) Aravind K. Mikkilineni† , Osman Arslan† , Pei-Ju Chiang‡ , Roy M. Kumontoy† , Jan P. Allebach† , George T.-C. Chiu‡ , Edward J. Delp†, "Printer Forensics using SVM Techniques"

5) Li, Zhi," KMeans for Color Indexing" , https://wiki.itap.purdue.edu/display/wlxls5c201710/KMeans+for+Color+Indexing