Czech technical university in Prague
Faculty of electrical engineering

department of radio engineering

MASTER'S THESIS

Metrics for determination of patterns similarity

Author: Bc. Tomáš Krč

Supervisor: Ing. Elena Anisimova

Prague, 2014

Název práce: Metriky pro zjištění shody vzorů

Autor: Tomáš Krč

Katedra (ústav): Katedra radioelektroniky

Vedoucí diplomové práce: Ing. Elena Anisimova

Abstrakt Tato diplomová práce se zabývá vytvořením metody pro klasifikaci tiskárenského papíru založené na textuře daného papíru. Popisuje, jakým způsobem získat detailní fotografie textury a jak je zpracovat pomocí singulárního rozkladu matice. Práce pak navrhuje jedenáct různých způsobů pro klasifikaci papíru založených na různých definicích vzdálenosti dvou matic, zkoumá jejich vlastnosti, a následně porovnává dosažené výsledky s dalšími univerzitními pracovišti.

Klíčová slova: textura, klasifikace, tiskárenský papír, singulární rozklad matice, vzdálenost dvou matic, hlasovací algoritmus

---

Title: Metrics for determination of patterns similarity

Author: Tomáš Krč

Department: Department of Radio Engineering

Supervisor: Ing. Elena Anisimova

Abstract A creation of method for classifying ink jet paper based on surface texture will be discussed in this thesis. It describes a way to obtain detailed photographs of the texture and how to process them using singular value decomposition. Eleven different ways of ink jet paper classification based on various distances and similarities between two matrices are proposed, their attributes tested, and the results are compared to those obtained by other university teams.

Keywords: texture, classification, ink jet paper, Singular value decomposition, matrix distance measure, voting algorithm

Acknowledgments

I would like to thank the supervisor of this thesis, Ing. Elena Anisimova for her tolerance and unlimited patience. I would also like to thank University of Wisconsin in Madison and especially William Sethares for inviting me in and making this thesis possible. His advises were always invaluable and insightful. The last acknowledgment goes to my parents for their continuing support for the entire time of my studies.

Declaration

I hereby declare that I have completed this thesis independently and that I have listed all the literature and publications used according to Methodological guideline of ethical principals connected with thesis writing.

I have no objections against lending, publishing and other usage of this work as long as Department of Radio Engineering will allow it.

In Prague on September 4, 2014 ....

student's signature

Contents

List of abbreviations

2D-DWT – 2D - Discrete wavelet transform
EROS – Extended Frobenius Norm
HWT – Hyperbolic wavelet transform
PCA – Principal Component Analysis
SVD – Singular value decomposition
M – number of pictures in the data set
$X_{i,j}$ – patch drawn from image sized $p \times p$ pixels
N – number of patches drawn from image
A_j – matrix consisting of reordered patches $X_{i,j}$
U_j – right eigenvector matrix
Σ – rectangular diagonal matrix filled with eigenvalues
V_j – left eigenvector matrix
n_u – number of columns taken into account from U_j
$d_{TYPE}(A, B)$ – distance of defined $TYPE$ between two elements
w – weight vector
W – matrix with elements of w on the main diagonal
Q – number of image patches drawn inside voting algorithm
p_j – probability of making the right classification
g – quality assessment measure
P – number of patches drawn in iteration of optimized voting algorithm
t – number of classes thrown away during each iteration of optimized voting algorithm
ϕ– maximum number of iterations in the optimized voting algorithm
α – factor used to decrease P in each iteration of the optimized voting algorithm

List of Figures

List of Tables

Introduction

Texture is a one of the most important attributes of photographic paper. In the 1920s, manufacturers started manipulating texture of the photographic paper in order to somehow differentiate their products and to satisfy the aesthetic and functional requirements of professional photographers. Prior to Second World War, when black and white silver gelatin paper was the dominant photographic medium [1], dozens of manufacturers worldwide produced a wide array of surfaces. Velour Black, produced by the Defender Photo Supply Company of Rochester, New York, and a favorite of Edward Weston in the 1930s, was available in 22 different surface finishes including Buff Platinum Matt and Velvet grain White Luster. One of the greatest achievements of this period of great diversity was Gevaluxe Velours, produced by the Geveart Company of Antwerp, Belgium starting in 1933 [2]. Promoted by the company as the "most beautiful paper ever made," Gevaluxe Velours had a unique texture that produced an extraordinarily matte surface with intensely deep black shadows unlike any photographic paper before or since.

From this period a book of specimen prints by the Belgian company Gevaert lists twenty five different surfaces made up of combinations of texture, reflectance, color and paper thickness [3]. Around the same time, a sample book from the Defender Company of Rochester New York lists twenty seven surfaces [4], Mimosa twenty six [5] and Kodak twenty two [6]. Each listed surface was proprietary to the different manufacturers and each was used across their multiple brands of paper with changes, additions, and deletions occurring over a span of many years.

Texture of the paper surface affects the visibility of fine detail and thus provides insight into the intent of the photographer and the envisioned purpose of his print. For example, prints made for reproduction or documentary functions are usually better suited to smooth-surface papers that render details with sharpness and clarity. More impressionistic or expressive subjects, especially those depicting large un-modulated masses of shadows or highlights, are best suited for papers with rough, broadly open textures [7]. A result of a careful and deliberate manufacturing process, texture applied to photographic paper is designed to be distinct and distinguishable through processing and post-processing procedures. Likewise manufacturer-applied texture endures despite localized defects such as abrasions and deterioration caused by poor handling, storage environment and enclosures [8].

A collector paid $1m for a photograph for the first time in 1999, a 1932 print of "Glass Tears" made by Man Ray [9]. More and more collectors wanted to cash in the fine-art photography market. Therefore the authenticity scandal attributed to Lewis Hine worried the collecting community. Hine was sociologist and photographer born in Oskhosh, Wisconsin. His most famous images were of the aerial construction workers, who built the Empire State Building. In the late 1930s, Hine met art historian and curator Robert Rosenblum. After Hine's death in 1940, Rosenblum became Hine's conservator and gave nearly 4000 negatives and 6000 prints to America's first photographic museum, the George Eastman House in Rochester. Rosenblum's vintage Hine photographs were almost greyish in tone when compared to other Hine's vintage work. Paul Messier, a photographic conservator from Boston was asked to test the authenticity of some of the Hine's prints. By finding optical brightness agents that glow in ultraviolet light he discovered that they were in fact made between 1958 and 1975, because optical brightness agents were used from the 1950s to make the paper appear whiter.

Collectors and curators are still willing to pay a huge amount of money for prints, however a demand for sophisticated scientific proof of the origin of the print began to arise. Given the attributes of the paper surface texture, an encyclopedic collection of surface textures could reveal vital clues about an unknown photographic print. Likewise a method for classifying textures could provide a means to link prints to specific photographers or to other prints of known origin. The same method might be used not only to classify photographic paper, but any other kind of paper, such as ink jet papers, canvases, bills, etc..

The goal of this thesis is to describe a possible way of classifying ink jet paper images, examine it's sensitivity to various conditions and objectively asses the results.

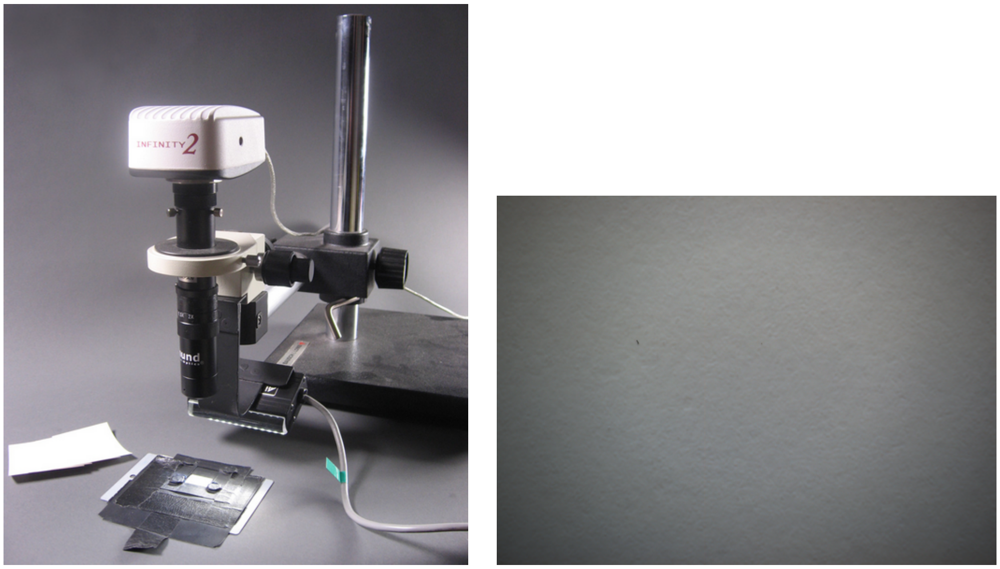
1 Texture image preparation

In order to work with surface texture of ink jet paper, we need to find a non-destructive way of capturing the texture. The easiest way is to capture the texture into images. A microscope system for acquiring the images was therefore assembled by Paul Messier [8]. The system used an Infinity 2-3 imager made by the Lumenera Corporation fitted with an Edmund Optics VZM 200i lens, as shown in Fig. 1.1.

The imager includes an Interline Sony ICX262 3.3 megapixel color progressive scan CCD sensor, that produces 2080 × 1536 pixels large images. Picture from each sample was taken from an 100 × 135 mm sample, so one pixel covers an area of $3.45 \mu m^2$.

A fixed point illumination source using a 3 inch LED line light manufactured by Advanced Illumination placed at a 25° raking angle to the surface of the paper then allows to take the raking light photomicrographics. Each raking light photo-micrograph was then saved in a 16-bit TIFF image file, approximately 18.2 MB large. No further post processing was then used on the images (i.e. no filters, sharpening, etc.) Detailed schematic of the system can be found in Appendix A.

This technique for capturing images is non-contact, non-destructive and therefore can be applied on prints or documents of high value.



(a) (b)

Figure 1.1: Raking-light photo-micrograph acquisition imager (a) and an example of such photo-micrograph (b). [8]

2 Collaborative challenge

As a part of a materials-based characterization project of modernist silver gelatin photographs, raking light photomicrographics described in the section above were made from each print from the Thomas Walter Collection at the Museum of Modern Art (MoMA) in New York [10]. The goal was to develop an automated way to classify alike prints based on their surface texture. Several university teams with signal processing experience were then asked to participate in a collaborative competition with a goal to develop algorithms for classifying those photomicrographics. Four university teams ended up joining this project:

- **Worcester Polytechnic Institute**: Andrew G. Klein, Christopher Brown, Anh Hoang Do, and Philip Klausmeyer
- **Ecole Normale Superieure de Lyon**: Patrice Abry, Stéphane Jaffard, Herwig Wendt Stéphane Roux, and Nelly Pustelnik
- **Tillburg University**: Nanne van Noord, Laurens van der Maaten and Eric Postma
- **University of Wisconsin**: William A. Sethares

Each team tried a different approach to develop the two standard parts of an automatic classifier: feature vector extraction and degree of similarity quantification. These approaches to texture image classification are described in subsection 2.2.

Each team then created a prototype algorithm using a training set of 50 silver gelatin samples from the Thomas Walter Collection with some known texture matches. This early work suggested that the placement of the raking light didn't have any significant impact on the results of classification, even though the primary orientation of fibers in the paper was changing. However, this doesn't mean that silver gelatin surfaces don't possess any other form of anisotropy. Still, the initial work provided a basis for relatively effective classification of silver gelatin prints using their surface texture.

2.1 Ink jet data set creation

Some ink jet paper surfaces do seem to exhibit anisotropy based on primary orientation of fibers. Since ink jet papers were not included in the early work, a natural expansion to that work would be to test and modify the methods on ink jet paper as well as other canvases.

In order to do that, another data set containing raking light photomicrographics of desired papers and canvases needed to be created [11]. This data set consisted of 120 images of ink jet papers with known meta-data, such as manufacturer, brand, gloss and date of issue. Papers in the data set were chosen to offer a varying degrees of self-similarity (the Appendix B lists all samples used in this data set).

In general, the data set consists of nine groups of ten somehow related paper samples, and thirty samples picked to span the large range of textures associated with ink jet papers. In the first nine groups, three similarity subsets can be found:

- images made from the same sheet of paper,
- images made from sheets taken from the same manufacturer package of paper,
- images from papers made to satisfy the same manufacturer specifications over a period of time.

Common sense suggests that raking light photomicrographics taken from the same sheet of paper should appear nearly identical. Likewise, photomicrographics from different sheets of paper taken from the same manufacturer package should also be largely similar. Raking light images obtained from the third subset, i.e. images from papers manufactured to the same specifications but made at different times should show strong similarity, but to a somewhat lesser degree. The remaining thirty samples demonstrating diversity might appear similar to the group of ninety images, or they might appear to be unique.

The university teams are then facing a challenge to not only classify one texture to the texture closest in the data set, but also to be able to discover and describe the similarity groupings. In other words, saying that sample x is closest to sample y is not enough, greater context needs to be addressed.

2.2 Technical approaches

As stated before, each of the four team took a different approach to this task. In general, those approaches to feature detection can be divided into two groups [12]:

- non-semantic (Wisconsin and Tilburg),
- multi-scale (Lyon and WPI).

The difference is that non-semantic features are extracted directly from the image data, while multi-scale features are based on a structural model that is supposed to be relevant to the data. Below is a basic summary of the approaches of other teams. I was privileged enough to work with professor Sethares from University of Wisconsin on trying to solve this problem. The method developed by Wisconsin team is the main subject of this thesis, and will be explained in detail in section 3.

2.2.1 Anisotropic wavelet multi-scale analysis (Ecole Normale Superieure de Lyon)

Team from Lyon based their algorithm on Hyperbolic Wavelet Transform (HWT) [13, 14] which is a variation of the 2D-Discrete Wavelet Transform (2D-DWT) [15]. HWT is defined from the use of two independent dilation factors $a_1 = 2^{j_1}$ and $a_2 = 2^{j_2}$ (along directions x and y respectively), instead of relying on single dilation factor a used along both directions of the image as in 2D-DWT. This allows it to capture anisotropy in texture images, that might or might not be associated to scaling properties.

The Hyperbolic Wavelet coefficients of image i (denoted as $T_i\left(\left(a_1,\, a_2\right),\, \left(k_1,\, k_2\right)\right)$) are theoretically defined as:

$$T_i\left(\left(a_1,\, a_2\right),\, \left(k_1,\, k_2\right)\right) =< i\left(x,\, y\right),\, \frac{1}{\sqrt{a_1 a_2}}\psi\left(\frac{x-k_1}{a_1},\, \frac{y-k_2}{a_2}\right) > . \tag{2.1}$$

From these HWT coefficients, structure functions (consisting of space averages at given scales a_1 and a_2) are defined as:

$$S_i\left(\left(a_1,\, a_2\right),\, q\right) = \frac{1}{n_a}\sum_k \left|T_i\left(\left(a_1,\, a_2\right),\, \left(k_1,\, k_2\right)\right)\right|^q, \tag{2.2}$$

where n_a is the number of $T_i\left(\left(a_1,\, a_2\right),\, \left(k_1,\, k_2\right)\right)$ really computed and not degraded by image border effects.

The similarity between two images i and j is then computed using a cepstral distance between their structure functions $S_i\left(\left(a_1,\, a_2\right),\, q\right)$ and $S_j\left(\left(a_1,\, a_2\right),\, q\right)$. It consists of a classical L^p norm computed on log-transformed normalized structure functions:

$$d(i,j) = \left(\sum_a \left|\hat{S}_i(a,q) = \hat{S}_j(a,q)\right|^p\right)^{\frac{1}{p}}, \tag{2.3}$$

where $\hat{S}_i(a,q) = \ln\frac{S_i(a,q)}{\sum_{a'} S_i(a',q)}$.

2.2.2 Pseudo-area-scale analysis (Worcester Polytechnic Institute)

In surface meteorology, area-scale analysis is a technique which has been applied to various problems. Both the measured length of coastline and the measured area of surface depend on the scale of observation and therefore the resolvability of small features. Fractal analysis is used in the area-scale approach [16] to decompose a surface into a patchwork of triangles of a given size. Smaller surface features become less resolvable as the size of triangles is increased. This applies to the "relative area" of the surface as well. The topological similarity of two surfaces is then computed by comparing relative areas at various scales. This technique has traditionally been employed on topographic data sets containing height information over a certain surface. Though lacking a direct measure, area-scale analysis can be applied to the photomicrographics by using light intensity as a proxy for height.

The proposed approach proceeds in three steps:

- preprocessing,
- feature extraction,
- classification.

In the preprocessing step, a square $N \times N$ region from the center of the image (where N was chosen to be 1024) is extracted, and intensity of the resulting extracted image is normalized. The $N \times N$

grid of equally spaced points (representing pixel locations) is then decomposed into a patchwork of $2\left(\frac{N-1}{s}\right)^2$ isosceles right triangles, where s is a scale parameter representing the length of two legs of each triangle. The pixel values at each of the triangle vertices are then taken as the "pseudo-height" of each of the vertices. The area of each triangle in 3-D space is then computed and the areas of all triangular regions are summed, resulting in the total relative area A_s at the chosen scale s.

In the feature extraction, the relative area for an image is computed over a range of scale values (8 scale values were used ranging from 1 pixel to 34 pixels, which correspond to lengths of 6.51 μm to 0.221 mm, respectively).

Finally, a χ^2 distance measure $d(i,\, j)$ is used to classify and compare the similarity of two images i and j:

$$d(i,\, j) = \sum_{s \in S} \frac{\left(A_s^{(i)} - A_s^{(j)}\right)^2}{A_s^{(i)} + A_s^{(j)}}\,, \tag{2.4}$$

where $A_s^{(i)}$ is the relative area of image i at scale s and S is the set of chosen scale values. Small values of $d(i,\, j)$ indicate high similarity between images i and j, while large values indicate low similarity.

2.2.3 Random-feature texton method (Tillburg University)

This method is a combination of random features and textons, i.e. the random-feature texton method. It was developed by Liu and Fieguth [17] and is an adaptation of the texton approach [18] using random features. Textons are a prototypical exemplar image patches that capture the "essence" of the texture in an image. Random-features (RF) are random projections of these image patches with $N \times N$ pixels to vectors with D elements ($N = 9$, $D = 20$, $D < N \times N$). More specifically, a random feature is defined as a $D \times N^2$ matrix, the elements of which are sampled from the standard multivariate normal distribution $\mathcal{N}(0,\, 1)$.

The random-feature texton method is used in the photomicrographics classification problem in the following way: a set of X sub-images sized $M \times M$ pixels is selected for each gray-value texture image in the 120 sample data set ($M = 512$). The sub-images are defined to be the central regions of $M \times M$ pixels of which the intensity distributions are normalized to zero mean and unit variance. A sample of 45 000 randomly selected $N \times N$ ($N \ll M$) patches (represented as vectors of length N^2) of the sub-images are contrast-normalized and then multiplied with RFs, yielding RF vectors of length D. A texton dictionary is created by applying k-means clustering to all RF vectors of the X sub-images of each texture image of the data set consisting of 120 samples.

Each image of the data set is transformed into a texture histogram by comparing all of its patches (represented as RF vectors) to the entries in the texton dictionary. Finally, the histograms are classified using a k-nearest neighbor algorithm using the χ^2 similarity measure.

3 Eigentexture approach (University of Wisconsin)

In the Eigentexture approach developed by the Wisconsin team, a number N of small $p \times p$ pixels patches $X_{j,i} \in \mathbb{R}^{p \times p}$ is randomly selected from the image j. Each element of such patch contains three values in a vector, each value representing one of the RGB color channels. These $X_{i,j}$ patches are then reordered lexigraphically into column vectors $a_{j,i} \in \mathbb{R}^{3p^2}$. Now that we have N such vectors, we can create matrix $A_j = [a_{j,1}, a_{j,1}, \dots, a_{j,N},]$ for each image j. We will use Singular Value Decomposition to simplify and order matrices A_j. The choice of values for parameters p and N will be discussed in subsection 3.5.2.

3.1 Singular Value Decomposition

Singular value decomposition (SVD) is widely used as a meaningful simplification of large amounts of data [19]. Formally, the singular value decomposition of an $m \times n$ matrix A_j is a factorization of the form:

$$A_j = U_j \Sigma V_j^T, \tag{3.1}$$

where U_j is a $m \times n$ orthogonal unitary matrix, Σ is an $m \times n$ rectangular diagonal matrix with non negative real numbers on the diagonal, and V_j^T is an $n \times n$ orthogonal unitary matrix again. The diagonal entries Σ_i of Σ are known as the eigenvalues of A_j.

The SVD can be performed on matrices $A_j \in \mathbb{R}^{m \times n}$, where $m \neq n$. In case that $m = n$ there will be only non-zero positive diagonal elements in Σ. The SVD can be performed such that the diagonal values of Σ are descending.

SVD can certainly be used in our case to simplify the A_j matrices. For example, if $p = 25$ and $N = 3000$, vector a_j will have size $3 \times 25^2 = 1875$ values. Matrix A_j is then sized 1875×3000. After applying SVD to A_j and extracting certain number of columns n_U corresponding to the n largest singular values, we will end up with only $1875 \times n_U$ sized normalized matrices U_j containing only the most important information. For example for $n_U = 30$, that is a noticeable difference. The reason for choosing n_U to be 30 is again described in subsection 3.5.2. Matrix U_j is used as a feature to describe class j, it can be thought of as the most relevant direction.

3.2 Distance metrics

Given a data set of j classes, we can create a dictionary by obtaining the SVD matrices $U_j \Sigma V_j$ for $j = 1, \dots, 120$. All that is left to do is determine how to compare them (i.e. the classification) - how different are U_j from each other? Sadly, there is no unified definition of distance or similarity between matrices, there are many ways to compute the distance. First of all, what can be considered a valid distance measure?

A method of determining distance $d(A, B)$ between elements A and B can be called a metric, if it fulfills the following requirements:

- $d(A, B) \geq 0$,
- $d(A, B) = 0 \iff A = B$,

- $d(A, B) = d(B, A)$,
- $d(A, B) + d(B, C) \geq d(A, C)$ (triangle inequality).

If the triangle inequality is the only property that is not fulfilled, metric can still be used for calculation. Such method can then be called similarity between elements A and B. Distance metrics measure how different two instances are (distance metric is large when instances are very different and is small when they are close). Similarity on the other hand measures how close to each other two instances are (the closer the two instances are to each other, the larger is the similarity value).

3.3 Commonly used distance metrics

Let the elements A and B be matrices of the same size (in our case $A = U_i$ and $B = U_j$, both sized 1875×3000). Distance between A and B isn't strictly defined. There are may ways of computing such distance, and each of them may be useful for different situations and scenarios. Some of the most used metrics for calculating distance between matrices are listed below. Following notation is used through the chapter: U_i and U_B are left-singular matrices obtained by applying SVD to A and B, respectively, $u_{i_{kl}}$ and $u_{j_{kl}}$ are single elements of U_i and U_j.

3.3.1 Minkowski distance

Minkowski distance for 2 matrices U_i and U_j of the same size $m \times n$ is defined as follows [20]:

$$d_{MINKOWSKI}(U_i, U_j) = \left(\sum_{k=1}^{m} \sum_{l=1}^{n} |u_{i_{kl}} - u_{j_{kl}}|^r \right)^{\frac{1}{r}}, \tag{3.2}$$

where r is an integer, $r > 0$, and $||$ represents absolute value. For $r = 1$ this expression defines Manhattan distance, a distance based on strictly vertical or horizontal paths (lines). If $r = 2$, the equation defines Euclidean distance, an "ordinary" distance given by the Pythagorean formula, sometimes also called Frobenius measure [21]. When $r \to \infty$, the expression describes Chebyshev distance, also known as chessboard distance, since in the game of chess the distance that a chess piece must travel between two squares is equal to Chebyshev distance between centers of those squares. When $d_{MINKOWSKI}(U_i, U_j) = 0$, U_i and U_j are the same.

3.3.2 Assembled matrix distance

Assembled matrix distance is defined as the following expression [21]:

$$d_{ASM}(U_i, U_j) = \left(\sum_{k=1}^{n} \left(\sum_{l=1}^{m} (u_{i_{kl}} - u_{j_{kl}})^2 \right)^{\frac{1}{2p}} \right)^{\frac{1}{p}}, \tag{3.3}$$

where $p > 0$ is a real number. According to [21], best result are obtained when $p = 0.125$. The expression can also be rewritten in following form, that is easier to implement as an algorithm:

$$d_{ASM}(U_i, U_j) = \left(\sum_{l=1}^{n} < (u_{i_{kl}} - u_{j_{kl}}), (u_{i_{kl}} - u_{j_{kl}})^T >^{\frac{1}{2p}} \right)^{\frac{1}{p}}, \tag{3.4}$$

where $< (u_{i_{kl}} - u_{j_{kl}}), (u_{i_{kl}} - u_{j_{kl}})^T >$ is the inner product of $(u_{i_{kl}} - u_{j_{kl}})$ and $(u_{i_{kl}} - u_{j_{kl}})^T$. When $d_{ASM}(U_i, U_j) = 0$, U_i and U_j are the same.

3.3.3 Yang distance

Yang distance is a distance measure based on Euclidean distance (3.2) proposed in [22]:

$$d_{YANG}(U_i, U_j) = \sum_{l=1}^{n} \left(\sum_{k=1}^{m} (u_{i_{kl}} - u_{j_{kl}})^2 \right)^{\frac{1}{2}}. \tag{3.5}$$

Again, this formula can be rewritten using inner product between $(u_{i_{kl}} - u_{j_{kl}})$ and $(u_{i_{kl}} - u_{j_{kl}})^T$ in the following way:

$$d_{YANG}(U_i, U_j) = \sum_{l=1}^{n} < (u_{i_{kl}} - u_{j_{kl}}), (u_{i_{kl}} - u_{j_{kl}})^T >^{\frac{1}{2}}. \tag{3.6}$$

When $d_{YANG}(U_i, U_j) = 0$, U_i and U_j are again the same.

3.4 Distance similarities

As stated before, distance can be measured with metrics satisfying all four requirements described in subsection 3.2 as well as with similarities where triangle inequality doesn't hold. In fact, distance measures listed above are not sufficient for determining distance between U_i and U_j in our case on their own. Moreover distance similarities may provide additional robustness against various properties of the ink jet paper images, as well as additional context.

3.4.1 Weight vector

Different attributes of A and B may be of different importance for the purposes of determining distance. That is true for our case, because the columns of the right eigenvector matrices U_i and U_j are already listed in order of importance. Therefore it seems a good idea to somehow quantify the importance as a weight of each column. A weight vector might be used for this purpose [23].

In remaining methods of determining distance in this subsection, a weight vector w of length n_U is required. Second matrix Σ resulting from SVD with input parameter n_U (number of columns taken into account) contains exactly n_U eigenvalues s_i on a diagonal. Let s be a vector of length

n_U filled with eigenvalues s_i. Firstly, all the eigenvalues s_i in the vector are normalized:

$$s_i \leftarrow \frac{s_i}{\sum_{j=1}^{n} s_{ij}}. \tag{3.7}$$

Assuming there are j matrices in our data-set, there are j vectors with eigenvalues. After normalizing each one, the vectors are arranged in one $m \times j$ matrix S. Each number in column S_j is replaced by a mean value of the whole column S_j (S is made of j columns, each of them now has a same value in every position). Finally, each row in S is replaced by a single value determined again as a norm of all the values in the row. This results in the desired weight vector w of length n_U computed over the whole data set.

3.4.2 Extended Frobenius Norm

*E*xtended F*ro*beniu*s* Norm (EROS) similarity measure is proposed in [23]. Let $U_i = [a_1, \dots, a_n]$ and $U_j = [b_1, \dots, b_n]$, where a_i and b_i are column orthonormal vectors of size m. The EROS distance between U_i and U_j is then defined as:

$$d_{EROS}(U_i, U_j) = \sum_{i=1}^{m} w_i \left| < a_i, b_i > \right| = \sum_{i=1}^{m} w_i \left| \cos\theta_i \right|, \tag{3.8}$$

where $< a_i, b_i >$ is the inner product of a_i and b_i, and w_i is the i^{th}element of a weight vector (see subsection 3.4.1) based on eigenvalues of A and B, $\sum_{i=1}^{m} w_i = 1$ and $\cos\theta_i$ is the angle between a_i and b_i. Range of $d_{EROS}(U_i, U_j)$ is between 0 and 1, 1 being the most similar - exactly the same.

3.4.3 Principal Component Analysis

Principal component analysis (PCA) similarity measure is widely used technique in multivariate analysis. It allows to convert a set of matrices for possibly correlated variables into a set of linearly uncorrelated variables called principal components [24]. PCA (3.9) is defined between two matrices of the same number of columns, but not necessarily the same number of rows. Let W be a matrix with the elements of weight vector w on the leading diagonal. PCA is then defined as:

$$d_{PCA}(U_i, U_j) = \text{trace}\left(U_i U_j^T W U_j U_i^T\right) = \sum_{j=1}^{n} \sum_{i=1}^{m} \cos^2 \theta_{ij}, \tag{3.9}$$

where trace represents a summation along the main diagonal. Range of PCA is between 0 and 1. Since PCA is an similarity measure, $d_{PCA}(U_i, U_j) = 1$ again means that A and B are the same.

3.4.4 Cosine Similarity

Cosine similarity is another used method for determining distance (similarity) between two vectors, that measures the angle between them [25]. Let u and v be two vectors of the same length x. The cosine similarity measure can be written as:

$$d_{COS}(u,v) = \frac{< u, v >}{< \|u\|, \|v\| >} = \frac{\sum_{l=1}^{x} u_l v_l}{\sqrt{\sum_{l=1}^{x} (u_l)^2}\sqrt{\sum_{l=1}^{x} (v_l)^2}}, \tag{3.10}$$

where $\|\ldots\|$ represents Euclidean norm. This formula can be adjusted for our problem, i.e. computing the distance between two are right eigenvector matrices U_i and U_j using inner product between them. Since both U_i and U_j are the outcomes of SVD, they are already normalized. A weight vector w (resp. matrix W with the elements of w on the leading diagonal) can be used as well:

$$d_{COS}(U_i, U_j) = \sum_{k=1}^{m} W \max |< U_{i_k}, U_j >| . \tag{3.11}$$

Range of $d_{COS}(U_i, U_j)$ is between 0 and 1, 1 being the most alike.

3.5 Voting algorithm

The above distance and similarity measures can be used to classify the ink jet images directly. The results obtained by using those and their comparison can be found in section 4. However, the method that was used in the original silver gelatin photographs classification problem works a bit differently.

The first step is the same as previously described, i.e. compute dictionary consisting of right eigenvector matrices U_j sized $3p^2 \times n_U$ from all the classes j. Instead of classifying one unknown paper image by computing it's right eigenvector matrix and comparing it to the dictionary, a different approach was chosen.

First, we randomly select Q $p \times p$ pixels patches Q_i from the unknown image i. These patches are again reordered lexigraphically into column vectors $q_i \in \mathbb{R}^{3p^2}$. Then we calculate the Euclidean distance (same as Minkowski norm with $r = 2$ in equation (3.2)) from the i^{th} patch to the j^{th} class from the dictionary:

$$d_{VOTE}(q_i, U_j) = \left\| q_i - U_j \left(U_j^T q_i \right) \right\|_2 , \tag{3.12}$$

where $\|\ldots\|_2$ denotes the Euclidean distance. Each patch Q_i is closest to one of the classes j. The best match for patch Q_i is then found using $f_{Q_i} = \arg\min_j d_{vote}(q_i, U_j)$. If we tally the set of all such f_{Q_i}, $i = 1, 2, \ldots, Q$, the most common entry among the f_{Q_i} is the most likely class for this image, the second most common entry is the second most likely class, etc.

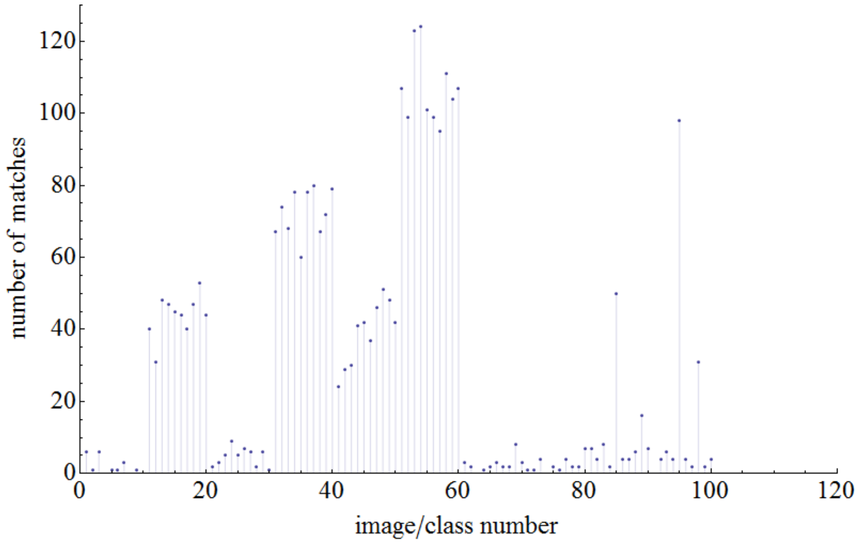


Figure 3.1: Example of classifying image number 54 with $Q = 3000$.

The example of classifying image number 54 can be found in Fig. 3.1. In this figure, we draw $Q = 3000$ patches from image $i = 54$ and compute the Euclidean distance from each of these patches to the matrices U_j, $j = 1, \dots, 120$ calculated for the 120 images in the data set using (3.12). For each patch we find the smallest distance and record the class corresponding to this smallest distance. The highest point in Fig. 3.1 occurs at image class $j = 54$ with a value of 124. So from 3000 patches drawn from image 54, 124 were closest to the U_{54} matrix. This might be a poor performance if the goal was just classification. However, recall the structure of our data set described in subsection 2.1. Images numbered 51-60 are all taken from the same kind of paper. Thus all the values between $j = 51, \dots, 60$ are very likely to be classified as coming from other images in this group. From 3000 patches, 1070 were in fact closest to matrices U_j, $j = 51, \dots, 60$. We can also see that groups of images 11-20 and 31-40 appear to be similar to class 54, though less so. There is also isolated peak at class 95, suggesting that paper 95 has some similarities to number 54. So this method provides the basic classification as well as the context to all other classes in the dictionary.

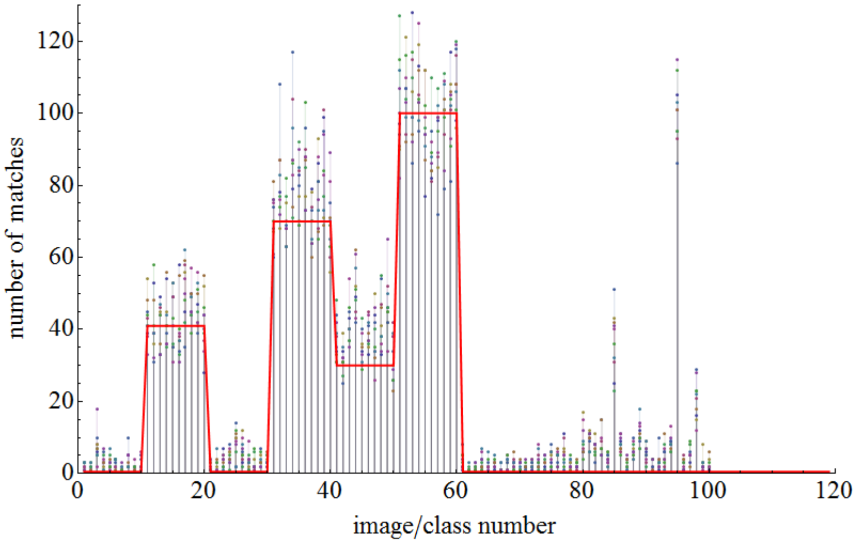


Figure 3.2: Superimposed classification of images 51-60 with $Q = 3000$, including an empirically derived distribution (red line).

We can use data from Fig. 3.1 as a histogram, normalize that histogram by the number of patches Q and view it as a probability density function. Then Fig. 3.1 shows the probability that a patch from image 54 is matched to some other image number. This can be used to estimate the choice of parameter Q (number of patches to extract) as well as measure the reliability of the whole procedure. Fig. 3.2 shows the counts from images 51-60 superimposed (red line). Assume that the probability of a patch matching an image is given by some (empirically derived) distribution, i.e the flat red line. It should then be possible to compare the achieved density function (the histogram) and the assumed density and create an asymptotic analysis showing how the distribution converges as $Q \to \infty$ (i.e. show how far away from the assumed similarity values any given trial with finite Q is). The asymptotic analysis is described in more detail in subsection 3.5.1. This test should also be able show if the peak at class 95 is worth any further investigation, or if it is a statistical outlier that should be expected given the randomness of the algorithm.

The histogram may also be used to give a measure of assurance that the classification is correct. Meaning if 90 % of the patches in an image are closest to one class, classification can be made with confidence. Similarly if 10 % are closest to one class, 9 % to other and so on, the classification will not be very reliable. However, these values are available to the algorithm when it is running, and therefore the algorithm can provide the classification as well as an assessment of the confidence of that classification.

3.5.1 Asymptotic analysis

As stated before, the histogram created by classifying image i against the dictionary can be interpreted as underlying distribution with $M = 120$ classes and a certain "win probability" p_i for each class $1 \leq i \leq M$, such that $\sum_{i=1}^{M} p_i = 1$. Given the specific design of this experiment, the classes can be divided into groups of 10 so that the sequences of probabilities satisfy $p_1 = p_2 = \ldots = p_{10}$, $p_{11} = p_{12} = \ldots = p_{20}$ etc. This can be represented using the underlying probability function, i.e.

the red line in Fig. 3.2. Even though this function in not known in applications, for the purpose of this analysis, consider the ideal case that all the probabilities p_i for all classes are known.

We can define a sequence of independent, identically distributed Bernoulli random variables for $1 \leq n \leq Q$. Let $X_n^{(i)} = 1$ if class i "wins" the n^{th} competition and 0 otherwise (i.e., class i is the smallest of the $d(q_i, U_j)$). We can define the frequency of wins for class i after Q competitions as follows: $\overline{X_Q^{(i)}} = \frac{1}{Q}\sum_{n=1}^{Q} X_n^{(i)}$. The law pf large numbers show that $\overline{X_Q^{(i)}} \xrightarrow{P} p_i$ as $Q \to \infty$.

To decide how many experiments Q must be held to gain a desired level of accuracy, rate of convergence can be used. Chebyshev's inequality [26] states that:

$$P\left(\left|\overline{X_Q^{(i)}} - p_i\right| < \varepsilon\right) \geq 1 - \delta, \tag{3.13}$$

where $\delta = \frac{Var\left(X_n^{(i)}\right)}{Q\varepsilon^2} = \frac{p_i(1-p_i)}{Q\varepsilon^2}$ and ε is the desired accuracy. We can use this to determine the number of experiments Q to achieve a certain level of probabilistic accuracy ε given probability p_i.

For example, suppose $p_i = \frac{1}{12}$ and we desire an accuracy of at least 10^{-2} with 90% probability:

$$1 - \delta \geq 0.90, \tag{3.14}$$

$$\frac{p_i\left(1-p_i\right)}{Q\left(10^{-2}\right)^2} \leq 0.01, \tag{3.15}$$

$$Q \geq 7639, \tag{3.16}$$

meaning that approximately 8000 experiments will be enough. This calculation can be repeated for each class j, and the largest Q may be used in the final algorithm. The spikes in the histogram may be processed in similar way: first calculate the probability p_i of observing such spike, and if it exceeds a certain threshold, we can increase Q to examine it further.

However, extracting 8000 patches and calculating the Euclidean distance for each of those 120 times takes a lot of computation time. Therefore, we chose Q to be 3000, which theoretically corresponds to an accuracy of 10^{-2} with probability 75 %, or an accuracy of $10^{-1.8}$ with probability 90 %. We can see that the accuracy does not change much, but the saving of computation time is substantial - it is more than 2.5 times faster.

3.5.2 Refining the parameters

As stated before, the data from Fig. 3.1 can also be used to determine the effectiveness of parameters within the algorithm. No matter what the data set consists of, we always know that an individual patch should be classified with reasonable frequency with the image from which is drawn. In our example, patches from image 54 should tend to be close (in the Euclidean distance sense) to the U_{54} derived from class 54. We can use this prior knowledge to measure the quality of the parameters.

This might be accomplished in many ways. Procedure used in this algorithm is to order the counts of matches in descending order, sorting the classes from most-frequent to least-frequent. Values are then assigned as:

$$g_j = \begin{cases} 1 & \text{if the } j\text{th image is first} \\ \frac{1}{2} & \text{if the } j\text{th image is second} \\ \vdots & \\ \frac{1}{n} & \text{if the } j\text{th image is } n\text{th} \end{cases} . \tag{3.17}$$

For the specific case of $j = 54$, the sorted counts are $\{\{54, 124\}, \{53, 123\}, \{58, 111\}, \{60, 107\}, \ldots\}$, meaning that the patches extracted from image 54 were closest to the U_{54} most often (124 times), closest to U_{53} slightly less often (123 times) and so on.

Overall assessment can be made over all images, providing a quality measure:

$$g = \sum_{j=1}^{M} g_j. \tag{3.18}$$

Quality measure g is used as the final distance measure. However, this time $0 \leq g < 2$, not like distance similarities, where the measure would be between 0 and 1. And if we try to classify an image against itself, the result will not be 1, because we draw different patches from the image inside voting algorithm.

We can also use g to compare various sets of parameters such as:

- the number of patches N used to compute U_j for each class
- the size $p \times p$ of the patches
- the size n_U for defining the size of each U_j

The values used in the simulations of this algorithm ($N = 3000$, $n_U = 30$ and $p = 25$) were a result of trying different combinations in an organized fashion. For example, with N and p fixed, we tried $n_U = 15, 20, 25, 30, 35$. Since $m = 30$ gave the highest g, this was chosen. This was repeated iterating through the three parameters until no improvements occurred.

A similar procedure can be used to asses the possible effectiveness of several algorithmic alternatives. For example, after the patches from image are extracted, the pixel values are transformed into vectors in a column-wise fashion. An obvious variation would be to to order the vectors q_i into row-wise fashion. Another option would be to sometimes use rows and sometimes columns. After averaging the quality assessment g of these alternatives over a number of runs (25) showed that there was no significant difference in the behavior of the three variations for the ink jet data set. This variation might be more appropriate for a data set in which the images had significant differences in the orientation (if the image looks different when photographed under horizontal raking light than under vertical raking light).

We could also choose pixels in the patches from different scales. Large patch could be selected and re-sampled to size $p \times p$ before using it as an input into the algorithm. Several values were tested as the initial size of the patch ($2p$, $3p$, $4p$, $8p$), but again, no significant improvement in the quality assessment g was found.

In conclusion, one of strengths of this approach is the ability to compare different variations of the algorithm using only information available when the method is applied. Meaning no labeled training data is required to refine the parameters.

3.6 Optimized voting algorithm

It seems rather ineffective to compute Q tests for each patch against the whole dictionary, since we know that many of the classes will not get any (or a very small amount) matches. We don't want to waste time testing patches with matrices U_j which we know are surely not close to the right class.

In order to speed the algorithm up, we created an iterative version of the algorithm. The basic idea is to classify a number of patches P ($P \ll Q$) for the first time on all classes. Then throw away classes that do not have many matches, and repeat with a new set of patches only against classes that remained after the previous pass. The new set of patches does not necessarily be the same size P. To further speed up the algorithm, it would be only logical do decrease P with each iteration, because we already obtained the classes that are most likely to be the right answer in the previous round (and therefore we do not need to test them so thoroughly as at the beginning). We can either continue the iterations until there is only one class left, or set a limited number of iterations. The summary of the algorithm then looks like:

1. Start with M classes (in our case, $M = 120$).
2. Test P different patches and form a ranked list for the classes.
3. Throw out t of the classes from the bottom of the list.
4. Update $M \leftarrow M - t$. Decrease P.
5. Repeat till maximum number of iterations ϕ is reached or only one class is left.

Different classes will take part in different number of competitions, depending on whether or not they get thrown out at some point. As a result, we not only keep track of the sorted counts as in the original algorithm, but of the number of competitions each class entered as well. For example, say that we started with $P = 300$ in the first iteration, and decrease P by factor of $\alpha = 0.8$ during the next iteration. If a particular class gets discarded in third iteration, it entered $\sum_{i=1}^{3} 0.8^{(i-1)}P = P + 0.8P + 0.8^2P = 732$ competitions. The number of entered competitions is preserved in the output of classifying one image, which might look like $\{\{54,124,1106\}, \{53,123,1006\}, \{58,111,1008\}, \{60,107,886\},\ldots\}$, meaning that the patches extracted from image were closest to the U_{54} most often (124 times) and entered 1106 competitions in 5 iterations, etc.

The g_j values from (3.17) cannot be used this time to obtain the final distance measure, because each of the classes made it through different number of iterations. Classes that remained to the end will have very high number of matches, since they entered the most competitions. So the g_i would be significantly biased towards these classes. However, we know the precise number of competitions each class entered, so we can use the following ratio as the final distance measure:

$$r_i = \frac{\text{number of competitions won}}{\text{number of competitions entered}}. \tag{3.19}$$

3.6.1 Analyzing parameters of the optimized algorithm

Once again, to determine the right parameters of this optimization, we need to analyze it [27]. Let us simplify things and consider a 1-D case, where instead of the usual patch, we extract just one value from the image. For further simplification, say we only have $M = 4$ classes (i.e. 4 other values u_j, $1 \leq j \leq 4$ we are trying to match). Suppose we draw a patch b from class j. We know that $b = u_j$ for some j. However, due to noise caused by rounding, after several tries we actually see a random measurement denoted by little Gaussian peak around each "true" value u_j. This situation is pictured in Fig. 3.3.

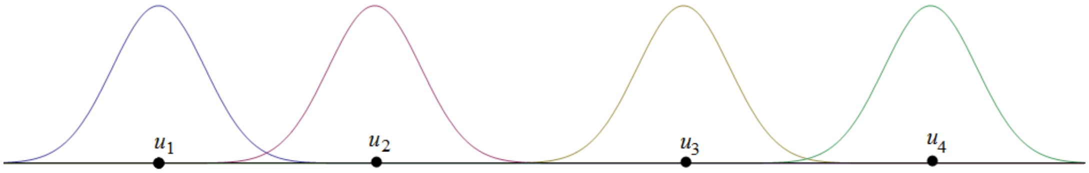


Figure 3.3: An example of simple 1-D case with $M = 4$ classes.

In 1-D, this situation is trivial, we just need to pick the nearest point. The probability of making the right classification p_j can be written as:

$$p_j = \int_{(u_{j-1}+u_j)/2}^{(u_j+u_{j+})/2} f(x; u_j, \sigma^2), \tag{3.20}$$

for $1 \leq j \leq 4$, where $f\left(x; u_j, \sigma^2\right)$ is the Gaussian density function with mean $\mu$and variance σ^2. Note that we are assuming we know the true values of u_j and the noise variance σ^2.

An error is made every time we announce any other class then j as a result of the classification. Given that we tested P independently drawn patches, we want to know what is the probability that the right class j is discarded in the 4^{th} step of the algorithm because it is not near the top of the list. The exact answer is difficult to calculate, since it requires use of order statistics for a multinomial distribution. Instead, bounding can be used:

$$\begin{aligned} Pr(j \text{ is not discarded}) \geq & Pr(j \text{ is ranked first}), && (3.21)\\ \geq & Pr(j \text{ gets} \geq P/2 \text{ votes}), && (3.22)\\ = & \sum_{k=P/2}^{P} \binom{P}{k} p_j^k \left(1-p_j\right)^{P-k}, && (3.23)\\ \geq & 1 - \exp\left(-\frac{1}{2p_j}\frac{\left(Pp_j - P/2\right)^2}{P}\right). && (3.24) \end{aligned}$$

The last inequality (3.24) follows from a Chernoff bound for sum of identically distributed Bernoulli random variables which works provided $p_j > \frac{1}{2}$.

By selecting $Pr(j$ is not discarded) to be for example 0.95, we now have a way to determine the optimal number of patches P we want to extract (i.e. number of tests we need to run). The problem is that P is dependent on the probability of making the right classification p_j, which is unknown in our problem. However, we might be able to estimate p_j using the original algorithm. If we run it with different number of classes M we can then empirically calculate p_j for each M. For example, say $M = 20$ (that is, first 20 classes from the dictionary). Suppose we draw Q patches from the first image and classify them against the first 20 classes. Given the structure of the data set, we know that correct answer is any number between 1 and 10, since the first 10 classes are created from single sheet of paper. The answer remains to be correct for any of the first ten images we try to classify. For images 11-20, the correct answer is between 11 and 20, etc. This can be applied to all images up to 90, since they are grouped in sets of 10.

Back to the example: say we try gradually classify all the images from 1 to 20. For each, we obtain a result being the class closest to the image. We can now analyze, how many of those result are correct, since we know the right answer. Probability of right classification will then be:

$$p_j = \frac{\text{number of correct classifications}}{M}. \tag{3.25}$$

Suppose that we obtain the following results: {10, 4, 3, 9, 5, 1, 2, 4, 1, 1, 18, 14, 18, 14, 8, 7, 1, 14, 14, 18} for $M = 20$. We can see, that only 18 of the 20 classifications gave right answer, so $p_j = \frac{18}{20} = 0.9$.

This can be repeated for a different number of random classes in dictionary M, yielding different pairs $\{M, p_j\}$. One pair is already known: if $M = 1$, then p_j must be 1 (or 100 %) as well, since there is only one possible result. Several other such pairs can be found in Tab. (3.1. Those pairs were obtained in the following way: first, let's randomly select M classes. From those M classes randomly select one that will be used for testing. This class should be from the interval $\langle 1, 90 \rangle$, since those are the classes grouped in sets of 10. Then the original voting algorithm is used to classify selected class from dictionary M, and p_j can be calculated using (3.25).

Size of dictionary M [-]	1	20	30	40	50	60
p_j [%]	100.00	85.44	75.67	79.78	75.78	81.33

Size of dictionary M [-]	70	80	90	100	110	120
p_j [%]	86.56	72.56	75.33	73.22	70.00	78.78

Table 3.1: Experimentally obtained values of p_j for different sizes of dictionary M.

We can see that the value of p_j tends to decrease with size of M, even though it is slightly irregular. The lowest obtained value is $p_j = 0.7$ (or 70 %). Using (3.24) and selecting $Pr(j \text{ is not discarded})$ to be 0.95, we get the initial value of $P \geq 110$, so about 200 should be enough.

There are still few questions unanswered. How to decrease number of drawn patches P in each iteration? Since the probability of making the right classification p_j tends to increase with smaller M, P should be decreasing. An easiest way is to simply decrease the P by a factor of α each iteration, simulating the expected change in p_j. We chose α to be 0.8, since it seemed to be the best compromise between computation time and accuracy of the result.

Another unknown parameter is t, i.e. how many classes to throw away after each iteration. Easiest choice is to throw away those classes that did not get any matches or an amount of matches below some small threshold. We chose the former in the implementation. It is nearly impossible to get to the stage where only one class is left, because the smaller number of classes remains, the more likely they will be matched and therefore never discarded. Therefore a maximum number of iterations ϕ was defined, so that the algorithm would not get stuck in infinite loop. We chose the value of $\phi = 10$, since it was very rare for any classes to drop at this point.

3.7 Combining the voting algorithm and distance similarities

Both the voting algorithm and the optimized voting algorithm are using Euclidean distance to classify the selected class to the dictionary. Other similarity measures can be used for this task as well. Suppose we want to combine the PCA distance with our voting algorithm. Instead of using (3.12), i.e. $d_{VOTE}(q_i, U_j) = \left\| q_i - U_j \left(U_j^T q_i \right) \right\|_2$ for computing the distance, we would need to use (3.9), i.e. $d_{PCA}(U_i, U_j) = \text{trace}\left(U_i U_j^T W U_j U_i^T \right)$. These two equations have different input parameters, so one cannot be replaced with the other. However, recall the way voting algorithm works: First we draw Q $p \times p$ pixels patches Q_i from the unknown image i, and reorder them lexigraphically into column vectors q_i, in order to compute Q classifications against the selected U_j from the dictionary. What can be done instead is drawing certain number of patches R in each test Q_i, again reorder them into vectors q_i. This time, all the vectors q_i are composed into one matrix, and SVD of that matrix is computed. The right eigenvector matrix U_i can now be used to compute $d_{PCA}(U_i, U_j)$.

This method has an obvious disadvantage: instead of just drawing Q patches and reordering them into vectors, we now have to draw R patches, reorder them into one matrix and compute it's SVD, all Q times. Even if $R < Q$, this will take significantly more computation time. On the other hand, we are using larger amount of data from the original image and processing them in a way that should provide additional robustness against various properties of the ink jet paper images.

All three similarities EROS, PCA and Cosine distance can be combined with both the original and optimized voting algorithm. Together with the original EROS, PCA and Cosine distance similarities, this gives us 11 different methods of classifying ink jet paper images. How to determine, which one is the best for our cause?

3.8 Visualizing the results

Since all papers in our data set need to be compared to all remaining, there will be $119^2 = 14161$ different distances for each of the 11 different classification methods. In order to represent them in a sensible way, we can plot the 119×119 similarity matrix into one graph using *Mathematica's* [28] function "ArrayPlot", and compare them visually, as well as by elements. Each row and column in the similarity matrix is a number that represents a measure of the "distance" between image i and image j. Example of such matrix can be seen at Fig. 3.4. White color is representing values close to zero, and black being one (i.e. the most similar).

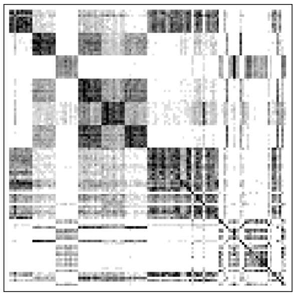


Figure 3.4: Example of similarity matrix.

Maximum value of each similarity measure is always one, since each SVD matrix is compared to itself at one point (black diagonal stripe). However, the "ArrayPlot" function adjusts each figure, so that the minimum in the picture is always zero, maximum is always one. Therefore some of the results might appear different then they actually are. Nevertheless, this visualization helps understanding the results of given metric. For example the clusters along the main diagonal in Fig. 3.4 clearly correspond to the fact that images are grouped by 10.

4 Comparison of individual metrics and similarities

In order to somehow objectively measure the properties and accuracy of metrics and similarities of calculating a distance between matrices, a learning data-set consisting of images would be required. Unfortunately, there isn't one in our problem, there is no ground truth. An artificial data set then needs to be fabricated (see subsection 4.1) to test the metrics and similarities, since the ground truth of such data set is known. The data sets are created to resemble the texture of actual ink jet paper or to be sensitive to certain variable, that occurs in the actual paper photographs.

However, even with the artificial set of images, there are still some unresolved issues. For example: is it desired that the method will be sensitive to some condition, such as rotation? The answer for that specific example is no, because no matter how the paper was rotated, we still want to recognize it as one particular piece. The same can probably be said about scale (no matter how "close" was the picture taken), perhaps even change in standard deviation (noise shouldn't be able to compromise the results that much). On the other hand, the paper itself might have some kind of grain pattern, and suddenly the sensitivity to change in standard deviation would be desired. And we want to be able to distinguish between various frequencies very clearly - sensitivity to frequency is desired. So even with a use of fabricated data set we will only learn sensitivity of those methods for certain properties, not which one is simply the best.

In general, it is probably safe to say that we would like the methods to be reasonably sensitive to a certain condition, meaning that minor change in that condition should not affect the result of classification. Visualization of such method for evenly changing condition should look like Fig. 4.1. In ideal case this visualization would look like a Gaussian function moving along the main diagonal.

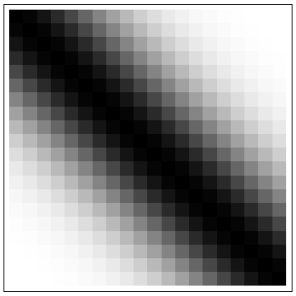


Figure 4.1: Visualization of ideal method sensitivity to certain condition.

4.1 Creating artificial set of data

In order to capture the various conditions and variables, more then just one artificial data-set is fabricated in *Wolfram Mathematica* environment [28]. All sets look similar - 20 relatively small-sized (200×200 pixel) gray scale images. Only 5 of those 20 pictures are shown in the following subsections for formatting reasons. First, fifth, tenth, fifteenth and twentieth picture are always shown.

Some of the variables and conditions are quite simple and straightforward, such as rotation, scale or frequency. Others were created in order to resemble the texture of the ink jet paper as much as possible (for example all the distance transformations).

4.1.1 Standard deviation

This data-set (Fig. 4.2) belongs to the simpler category - images are filled with noise with uniform distribution, and the range of the distribution is slowly increasing from $\langle 0.475, 0.525\rangle$ to $\langle 0, 1\rangle$.

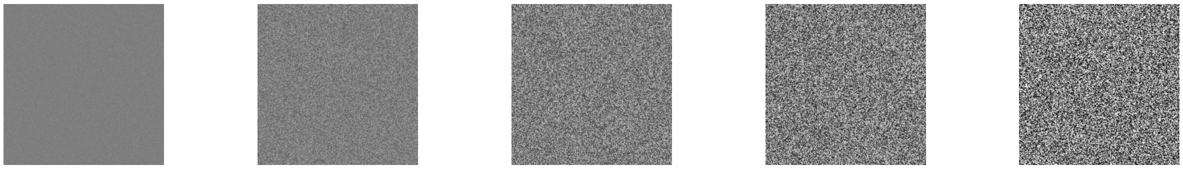


Figure 4.2: Examples from data-set based on standard deviation change

4.1.2 Frequency

Representation of increasing frequency by generating sinusoids (from range $\langle 2, 40\rangle$ changes per picture) is used to create the following data set (Fig. 4.3).

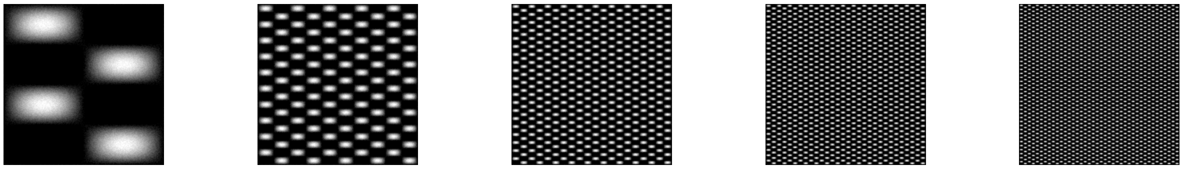


Figure 4.3: Examples from data-set based on vertical frequency

4.1.3 Scale

In order to represent scale change (Fig. 4.4), only single image filled with noise with uniform distribution (range $\langle 0, 1\rangle$) is generated. The rest of the image set is obtained by resizing the previous image by the factor of 1.5 and then cropping it back to desired 200×200 pixel size.

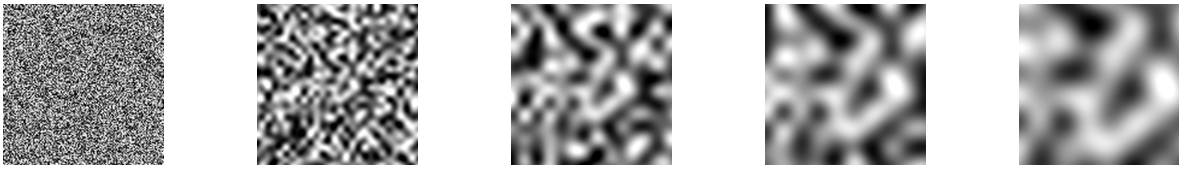


Figure 4.4: Examples from data-set based on scale change

4.1.4 Spatial frequency

A spatial frequency image set (Fig. 4.5) is represented by 2D sinusoids with frequency increasing by the factor of $\left(\frac{p}{i}\right)^2$, where i is the number of picture ($i \in \langle 1, 20\rangle$) and $p{\in}\mathbb{Z}$, $p \in \langle 1, 10\rangle$.

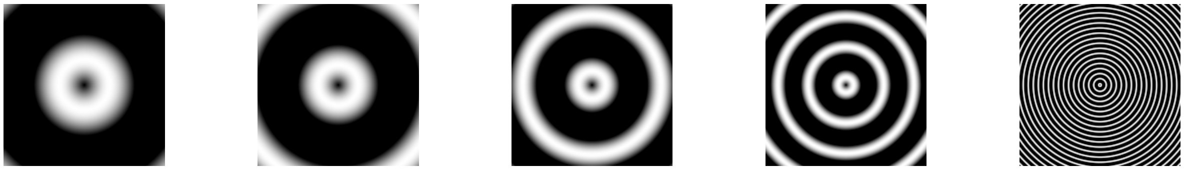


Figure 4.5: Examples from data-set based on 2D frequency change

4.1.5 Rotation

This set of images is meant for testing rotation invariance. It is created using a fifth image from the "Frequency" data set (see 4.1.2) and rotating it with the step of $\frac{\pi}{38}$. Using this step, image is rotated exactly 90° in 20 steps (Fig. 4.6).

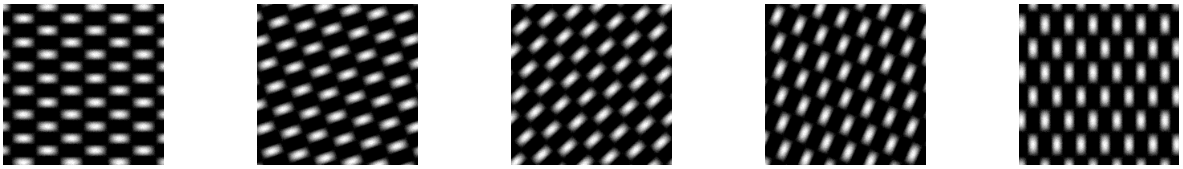


Figure 4.6: Examples from data-set based on rotation change

4.1.6 Distance transformation of points

Images with random number of white points (starting at 20 points and increasing by factor of 20) are created as a basis for this data set (Fig. 4.7). A *Wolfram Mathematica* function "DistanceTransform" is then used to calculate distance transformation - value of each pixel in the image is replaced by its distance to the nearest background (black) pixel.

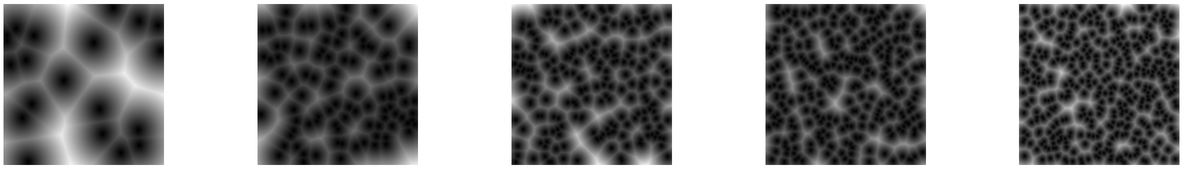


Figure 4.7: Examples from data-set based on distance transformation of points change

4.1.7 Distance transformation of lines

Almost the same as the previous data set, except this time random white lines with random length are used instead of points (Fig. 4.8). Distance transformation is then calculated via the "Distance-Transform" function.

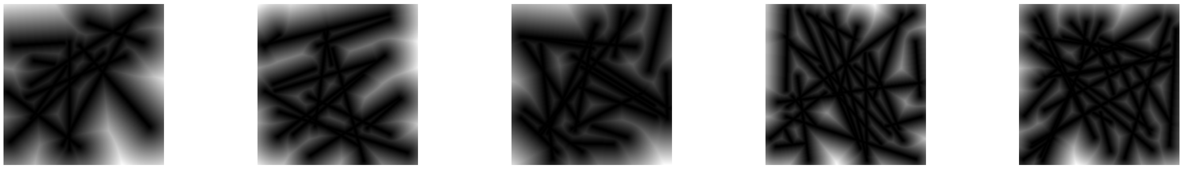


Figure 4.8: Examples from data-set based on distance transformation of lines change

4.1.8 Bandpass of noise

An image filled with noise with uniform distribution (range $\langle 0.475, 0.525\rangle$) is used as a base for this set of images (Fig. 4.9). That image is then filtered with bandpass filter with an expanding bandwidth that continuously increases its boundaries from $\{0.045, 0.05\}$ to $\{0.9, 1\}$.

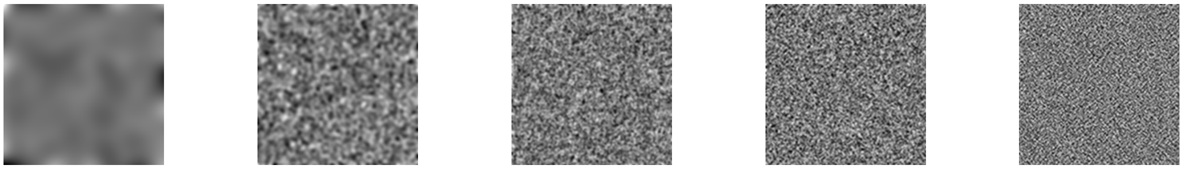


Figure 4.9: Examples from data set based on changing bandpass noise filtering

4.1.9 Distance transformation of points with random "steepness"

A grids with increasing density are generated, and a value between 0 and 1 is assigned to every nod in each grid. The grids are then interpolated to a 200×200 size, creating the gray scale images (Fig. 4.10).

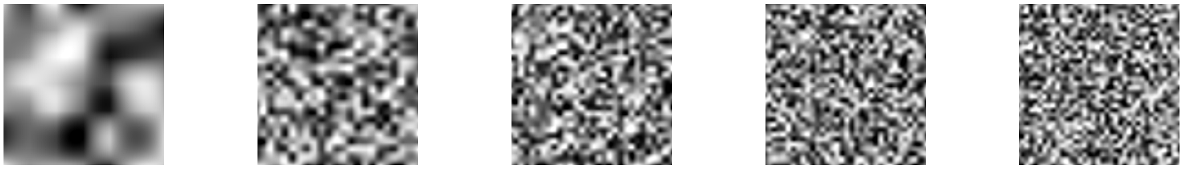


Figure 4.10: Examples from data-set based on distance transformation of points with random "steepness" change

4.2 Artificial data set based on ink jet paper

Our fabricated data set can be made from the images of ink jet paper as well. Some of the variables of the completely artificial data set can be easily transferred to the actual images, such as rotation or scaling. However, some of the variables (like the distance transformations) can't really be meaningfully used for adjusting the paper pictures.

As a base image for each of the following data sets, one small 200×200 size patch from ink jet paper picture is used (Fig. 4.11).

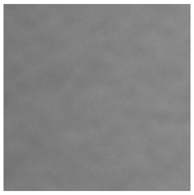


Figure 4.11: A 200×200 size patch from the ink jet paper picture

4.2.1 Standard Deviation

The first artificial data set based on the real ink jet paper pictures uses a change in standard deviation. The change is accomplished by equalizing the histogram of the images in 20 steps, using the completely artificial standard deviation data set as a reference. Examples of both the generated results and their histograms can be found in Fig. 4.12.

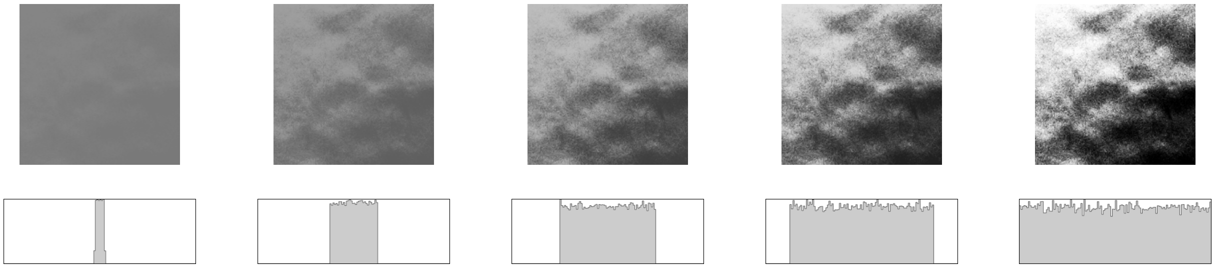


Figure 4.12: Examples of artificial data set based on standard deviation change and corresponding histograms

4.2.2 Rotation

Once again, this set of images is created using a base patch (Fig. 4.11) and rotating it with the step of $\frac{\pi}{38}$. Using this step, image is rotated exactly 90° in 20 steps (Fig. 4.13).

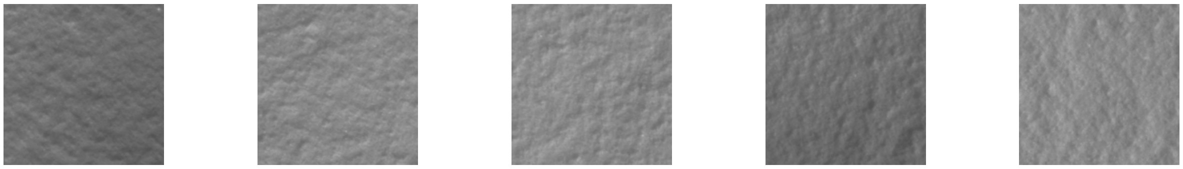


Figure 4.13: Examples of artificial data set based on rotation change

4.2.3 Scale

Scale change (Fig. 4.14), is again represented by using the base patch (Fig. 4.6). The rest of the image set is obtained by resizing the base patch by the factor of 1.5 and then cropping it back to desired 200×200 pixel size.

Figure 4.14: Examples of artificial data set based on scale change

4.2.4 Bandpass filter

Base patch image (Fig. 4.6) is filtered with bandpass filter with an expanding bandwidth that continuously increases its boundaries from $\{0.045, 0.05\}$ to $\{0.9, 1\}$ (Fig. 4.15).

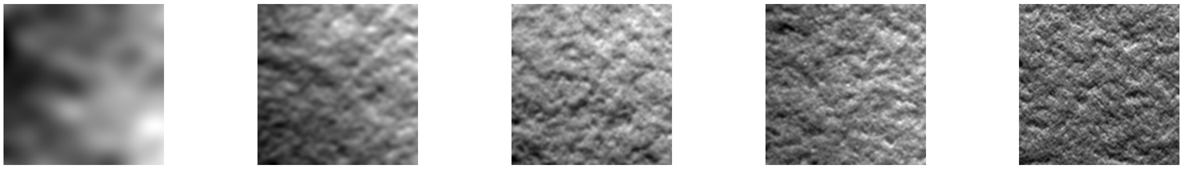


Figure 4.15: Examples of artificial data set based on changing bandpass noise filtering

4.3 Methods for testing the artificial data sets

After creating the artificial data set consisting of various texture images, we need to develop a way to test it for desired properties. The most common conditions that might affect the classification are caused by the way raking light photomicrographics are taken. A total of 6 different conditions were chosen - rotation of the paper, scale of the photograph, brightness of the photograph, exposure level, blur, and the distribution of the data, i.e. histogram. All are described below in subsections 4.3.1 to 4.3.6. Multiple images from each of the artificial data set described in subsections 4.1 and 4.2 were used as a base patch.

4.3.1 Rotation

Simulation the rotation of the paper under the imaging device was achieved in exactly the same way as in subsection 4.1.5 - rotating the base patch 90° in ε number of steps.

4.3.2 Scale

Once again, simulation of the scale change in the photograph was exactly the same as in subsection 4.1.3 - resizing the base patch by the factor of 1.5 and then cropping it back to desired 200×200 pixel size in ε number of steps.

4.3.3 Brightness

Approximation of brightness change in the photograph was achieved simply by adjusting the brightness of the base patch from -95% to 195% in ε number of steps.

4.3.4 Over-exposure

Overexposure of the base patch was simulated simply by shifting each pixel value by a factor of -0.5 to 1.5 in ε number of steps.

4.3.5 Burr

A Gaussian filter with a radius changing in ε number of steps was applied to the base patch in order to approximate the blur condition.

4.3.6 Histogram transformation

Simulation of a different data distribution in base patch was achieved via histogram transformation. Specifically, the histogram of the base patch was gradually equalized in ε number of steps.

4.4 Testing results

After applying the conditions to the base patch and creating ε different testing images, all 11 classification methods were used to obtain the resulting distance matrices. Since we have 11 different methods of classification, 13 different data sets and 6 different conditions, displaying the results in the similarity matrices as before would be overwhelming. Instead we can compute mean across each of the diagonals in the similarity matrix. For example, say dimensions of the matrix are 10 × 10. Then we would have 19 different means, first one would be just the value at position {10,1}, last one value at position {1,10}, and the tenth one would be the mean of leading diagonal.

We can further simplify the visualization by averaging means obtained by the same classification method from all 13 data sets. This will yield just 6 curves for each classification method, representing the sensitivity of given method to the 6 different conditions. Each curve can be viewed as an anti-diagonal of the similarity matrix. Results can be seen in figures 4.16, 4.17 and 4.18.

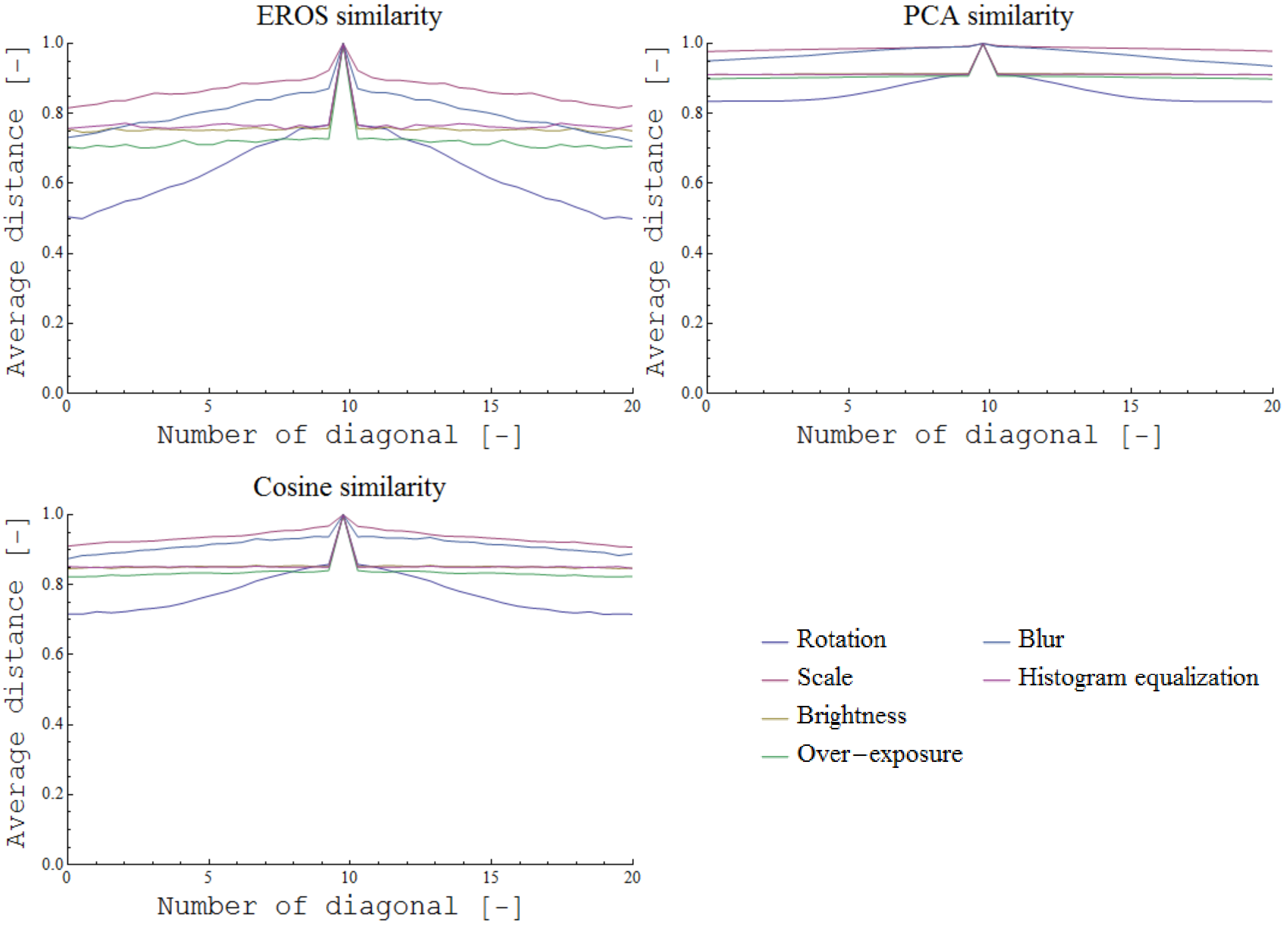


Figure 4.16: Visualization of condition testing results.

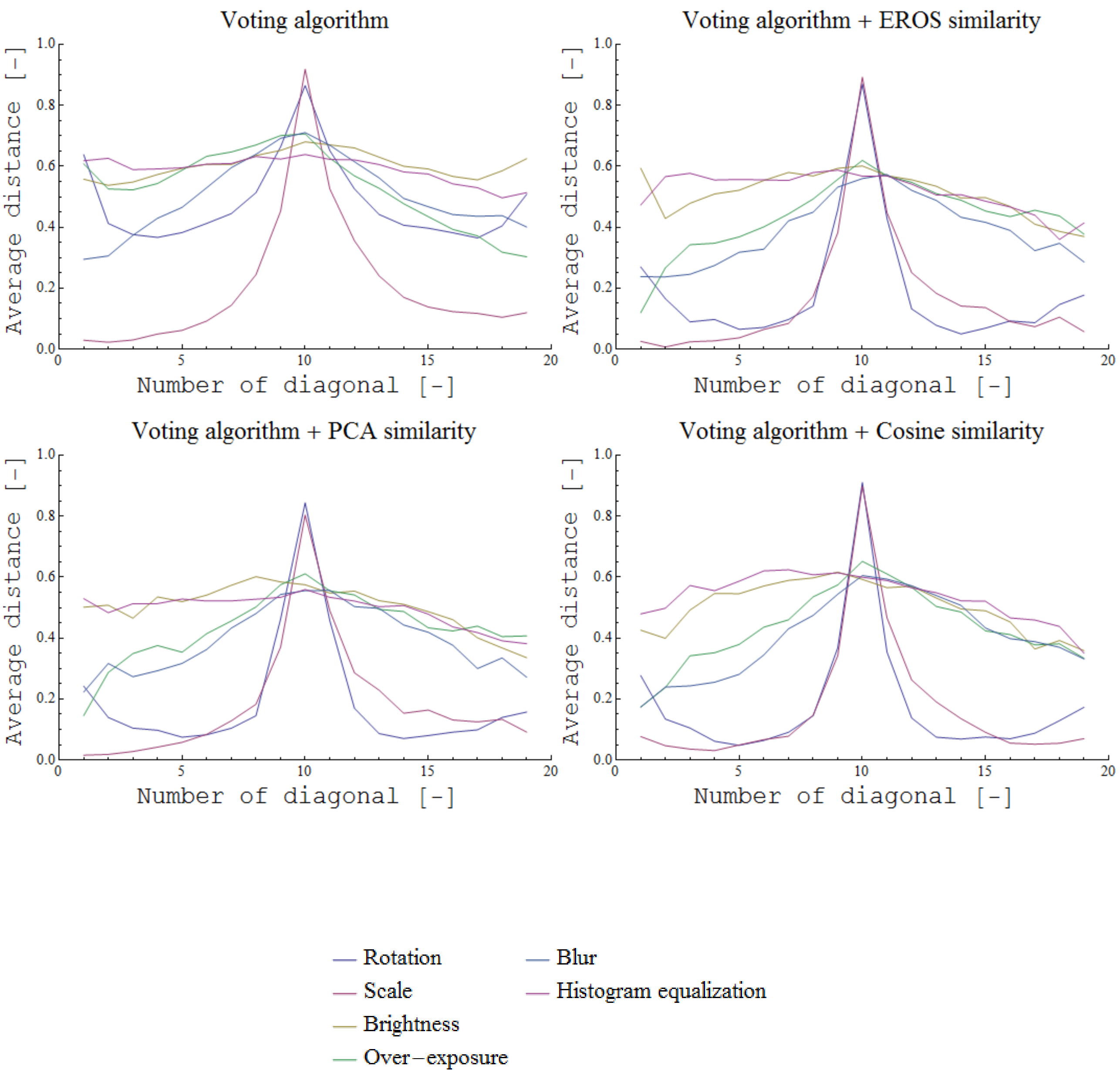


Figure 4.17: Visualization of condition testing results cont..

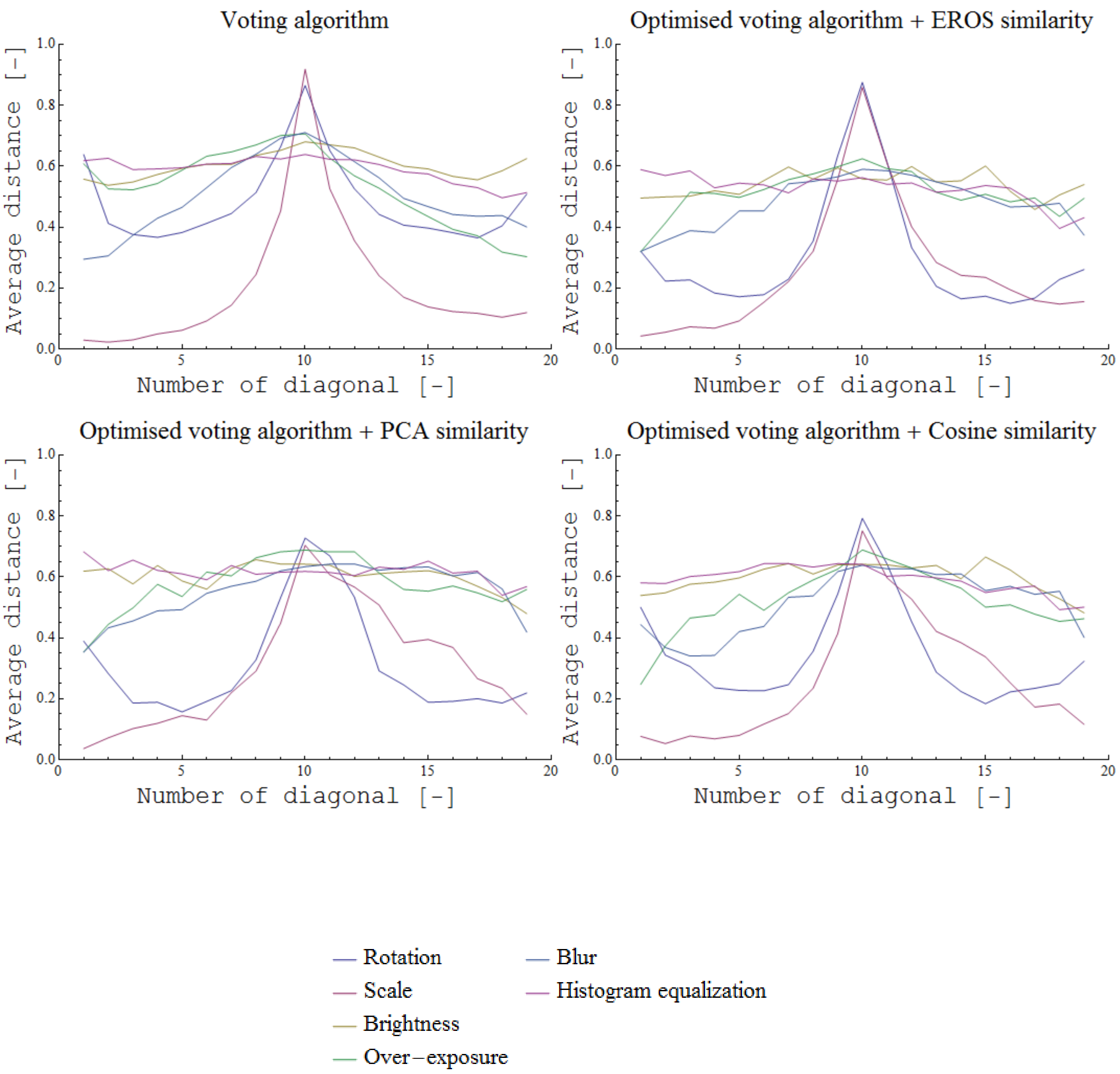
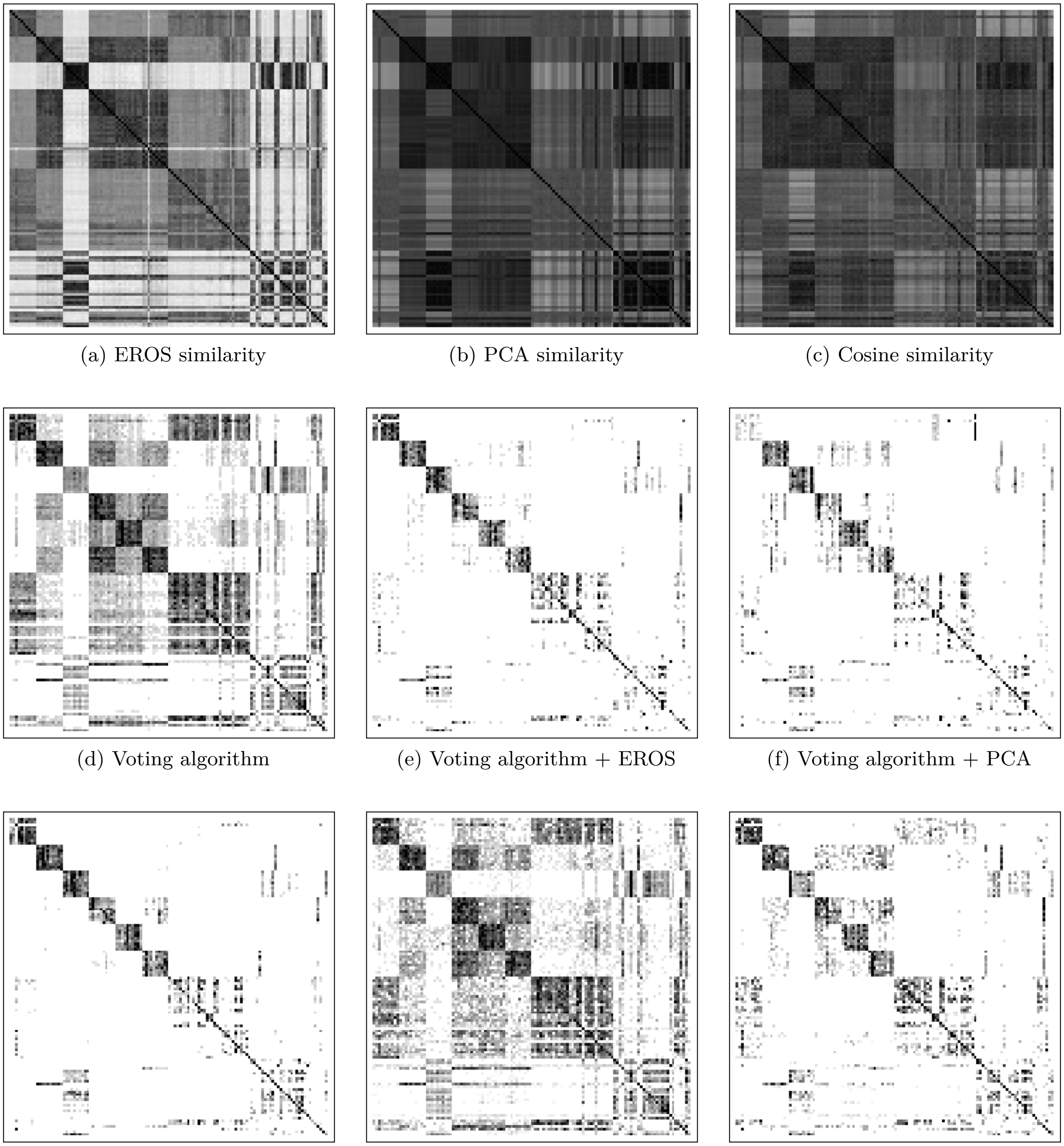


Figure 4.18: Visualization of condition testing results cont..

We can clearly see that EROS similarity, PCA similarity and Cosine similarity are hardly sensitive to any of the selected conditions. All of the voting algorithms (even those combined with EROS, PCA and Cosine similarity) are sensitive to rotation and scale change, and certainly more sensitive to the other conditions as well. Interestingly, the voting algorithms are very similar to each other in terms of sensitivity. This suggests we have a choice between non-sensitive methods (EROS similarity, PCA similarity and Cosine similarity) and sensitive methods (all voting algorithms).

5 Results

Eleven different methods for ink jet paper classification were proposed in previous sections. We learned that some are more sensitive to certain conditions such as rotation, scale, etc.. However, the methods have not been tested on the actual ink jet paper data set yet. First, lets build a similarity matrix for each of the methods using the ink jet paper data set (figures 5.1 and 5.2).



(a) EROS similarity

(b) PCA similarity

(c) Cosine similarity

(d) Voting algorithm

(e) Voting algorithm + EROS

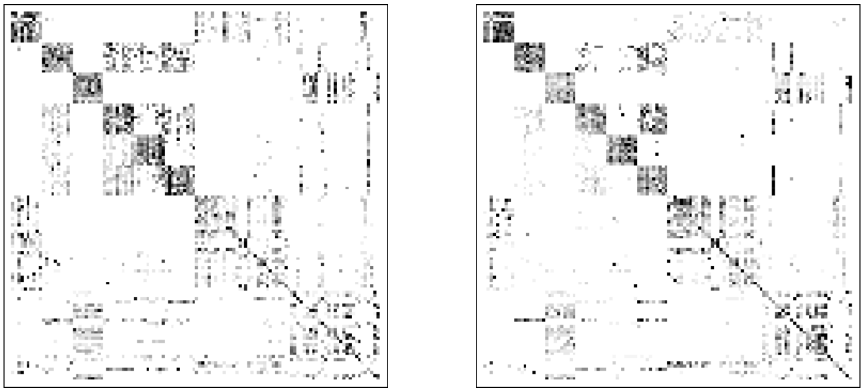
(f) Voting algorithm + PCA

(g) Voting algorithm + Cosine sim.

(h) Optimized voting algorithm

(i) Optimized voting alg. + EROS

Figure 5.1: Classifications methods and their similarity matrices build using ink jet paper.



(j) Optimized voting alg. + PCA

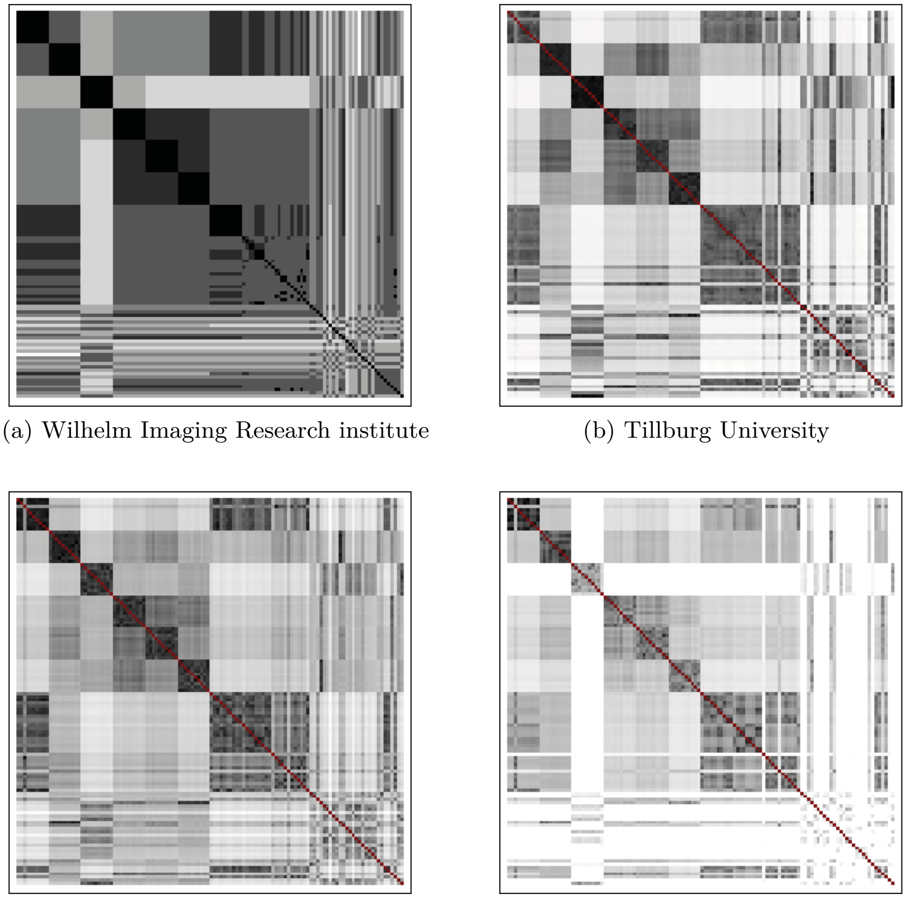
(k) Optimized voting alg. + Cosine similarity

Figure 5.2: Classifications methods and their similarity matrices build using ink jet paper cont..

We can clearly see the blocks of size 10 along the main diagonal representing images from same or very similar papers in all matrices corresponding to both the original and optimized voting algorithms. This means that the classification will most likely be successful. However, the relations to other images in the data set are not as clear as those obtained from EROS similarity, PCA similarity and Cosine similarity.

Obtained similarity measures can also be visually compared to those from other university teams. Figure 5.3 includes similarity measured from Worcester Polytechnic Institute, Tillburg University and Ecole Normale Superieure de Lyon as well as similarity matrix created in Wilhelm Imaging Research institute using expert human meta data knowledge (without actually viewing or visually comparing the ink jet paper and canvas samples).

Rate of successful classification can be measured objectively. Suppose we randomly select M classes c from the data set ($c \in M$, $c < 90$), and then classify each of them against the whole data set 100 times. Each time the probability of right classification p_c for a given method can be calculated using 3.25. The final probability p is then a simple arithmetical average of the individual probabilities p_c. We can also compute the variance of correct matches, i.e. how much are the individual probabilities p_c different from each other. Last but not least, the average time per one classification/method can be measured. The objective results are summarized in table 5.1.



(a) Wilhelm Imaging Research institute

(b) Tillburg University

(c) Ecole Normale Superieure de Lyon

(d) Worcester Polytechnic Institute

Figure 5.3: Other university teams similarity matrices. [8]

	Probability of right classification [%]	Variance of correct matches [-]	Average time for 1 classification [s]
EROS similarity	76.89	0.146	0.0436
PCA similarity	58.44	0.239	0.0224
Cosine similarity	77.77	0.243	0.0372
Voting algorithm	69.33	0.369	10.7714
Voting algorithm + EROS	97.11	0.079	38.5907
Voting algorithm + PCA	73.11	0.425	32.9480
Voting algorithm + Cosine similarity	88.89	0.333	37.7573
Optimized voting algorithm	64.67	0.324	2.2054
Optimized voting algorithm + EROS	83.56	0.226	7.0549
Optimized voting algorithm + PCA	54.22	0.340	4.1063
Optimized voting algorithm + Cosine similarity	52.44	0.402	5.4256

Table 5.1: Objective evaluation of classification methods.

Conclusion

The goal of this thesis was to create a reliable method for classifying ink jet papers. Several other university teams were invited to take part in this task initiated by the Museum of Modern Arts, de facto competing against each other. Thesis began with describing the common background for this task - non-destructive way of capturing ink jet paper surface textures and creating one data set. Each team was on it's own after this point.

We started by reducing data from the raking photomicrographics using extraction of patches and singular value decomposition. Parameters of this reduction - N, p and $n_u$were examined and chosen with respect to performance and results. A number of ways to calculate distance between two matrices was described, including distance similarities. We then described the principal behind voting algorithm and discussed it's parameters using asymptotic analysis. Then the voting algorithm was optimized, using iterations to avoid unnecessary computations and increase performance. The advantage of voting algorithms is that we have a way of comparing different versions of the algorithm without the need of training data. Both original and optimized voting algorithms were also combined with various ways of determining distance between two matrices.

All the classification methods were tested for common conditions that might affect the quality of classification via artificially created data set with known relations to chosen conditions. We concluded by examining the methods both subjectively using similarity matrices as visual representations and objectively by statistical analysis of the obtained results.

We can see from table 5.1 that there is a trade off between accuracy and computational complexity. Voting algorithm combined with EROS similarity clearly outperforms the other methods in accuracy, however it is the slowest to compute. On the other hand EROS similarity itself is very fast on it's own while delivering satisfactory results.

The three simpler methods (EROS similarity, PCA similarity, Cosine distance) are not sensitive on various texture affecting conditions. Weather or not is this desirable depends on the application of classification methods. In certain cases might sensitivity be an advantage, for example sensitivity on change in frequency of certain fibers and characteristic formations.

It is difficult to select the best method out of the 11. In order to do that, several other ink jet paper data sets would be required for testing the methods and their properties more thoroughly. However, I will continue to work on this task, so we might get more conclusive answer in the future.

References

[1] MESSIER, P. *“Les Emulsion Industrielles”*. Le Vocabulaire Technique de la Photographie, ed. A. Cartier-Bresson. Paris: Les Editions Marval. 2008. pages 454–456.

[2] MESSIER, P. *“20th Century Black and White Papers”*. [online]. Paul Messier LLC. Boston, MA USA. 2008. [quote 24.8.2014]. Accessible from: <http://notesonphotographs.org/index.php?title=Messier,_Paul._20th_Century_Black_and_White_Papers>

[3] Gavaert Company, *“The Book of Gevaert Paper Samples”*. New York, USA. Circa 1935. New York: The Gavaert Company of America.

[4] Defender Photo Supply Company, *“Velour Black Specimen Prints”*. Rochester, USA. Circa 1935. New York: Defender Photo Supply Company.

[5] Mimosa, *“Mimosa Papiere”*. Dresden, Germany. Circa 1935. Dresden: Mimosa AG.

[6] Eastman Kodak Company, *“Commercial and Illustrative Photographic Papers”*. Rochester, USA. Circa 1935. New York: The Eastman Kodak Company.

[7] Eastman Kodak Company, *“Surface Characteristics of Kodak Photographic Papers”*, Rochester, USA. Circa 1935. New York: The Eastman Kodak Company.

[8] MESSIER, P., JOHNSON, R., WILHELM, H., SETHARES, W., KLEIN, A.G., ABRY, P., JAFFARD, S., WENDT, H., ROUX, S., PUSTELNIK, N., VAN NOORD, N., POSTMA., N. *“Automated Surface Texture Classification of Ink jet and Photographic Media”*. Cornell University, New York. 2013, Accessible from: <http://people.ece.cornell.edu/johnson/IST_NIP29_PM_RJ_HW_2013_09_30.pdf>

[9] *”As vintage photographs rise in price, fakes are becoming more common”* [online]. The Economist Group, 2001. [quote 25/8/2014]. Accessible from: <http://www.economist.com/node/682985 >

[10] Museum of Modern Arts, *“Modern Photographs from the Thomas Walther Collection, 1909-1949”.* New York, USA. 2014. Exhibition organized by Department of Photography, MoMA.

[11] ”*Paper Texture ID Challenge*” [online]. Paul Messier, 2013. [quote 15/10/2013]. Accessible from <http://www.papertextureid.org/>

[12] GONZALEZ R. C. WOODS, R.E. “*Digital Image Processing*”, Third edition. Upper Saddle River, NJ: Prentice Hall. 2008. ISBN: 013168728X

[13] DEVORE, R. A., KONYAGIN, S.V., TEMLYAKOV, V.N. “*Hyperbolic wavelet approximation*”. Constructive Approximation. 14:1–26, 1998.

[14] ROUX, S. G., CLAUSEL, M., VEDEL, B., JAFFARD, S., ABRY, P. *"Self-Similar Anisotropic Texture Analysis: The Hyperbolic Wavelet Transform Contribution*", 2013. Accessible From: <http://arxiv.org/abs/1305.4384v1>

[15] MALLAT, S. "*A Wavelet Tour of Signal Processing*", Third Edition: The Sparse Way. Academic Press. 2008. ISBN: 0080922023

[16] KLEIN, A., DO, A., BROWN, C., KLAUSMEYER, P. "*Texture Classification via Area-Scale Analysis of Raking Light Images*". Worcester Polytechnic Institute. Submitted to ASOMILAR conference 2014.

[17] LIU, L., FIEGUTH, P.H. "*Texture classification from random features*". IEEE Transactions on Pattern Analysis and Machine Intelligence, 34(3): 574–86. 2012.

[18] VARMA, M., ZISSERMAN, A. "*A Statistical Approach to Material Classification Using Image Patch Exemplars*". IEEE Transactions on Pattern Analysis and Machine Intelligence. 31(11): 2032–2047. 2009.

[19] MOON, T. K., STIRLING, W.C. "*Mathematical Methods and Algorithms for Signal Processing*". Prentice Hall: 3.4:138-141 and 7:369-395. 2000.

[20] VAN DE GEER, J. P. *"Some Aspects of Minkowski Distances*". [online]. Leiden: Leiden University, 1995. [quote 24/11/2013]. Accessible from: <http://www.datatheory.nl/pdfs/95/95_03.pdf>

[21] ZUO, W. M., WANG, K. Q., ZHANG, D. "*Assembled matrix distance metric for 2DPCA-based face and palmprint recognition*". [online]. Harbin: Harbin Institute of Technology, 2005. [quote 27/11/2013]. Accessible from: <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?reload=true&arnumber=1527800>

[22] YANG, J., ZHANG, D., FRANGI, A.F., YANG, J. Y. *Two dimensional PCA: a new approach to appearance-based face representation and recognition.* IEEE Trans. PAMI, vol. 26, no. 1, pp. 131-137, 2004.

[23] YANG, K., SHAHABI, C. "*A Multilevel Distance-based Index Structure for Multivariate Time Series*". [online]. Los Angeles: University of Southern California, 2005. [quote 18/11/2013]. Accessible from: <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?tp=&arnumber=1443353>

[24] KRZANOWSKI, W.J. "*Between-Groups Comparison of Principal Components*". Journal of the American Statistical Association, vol. 74, no. 367, pp. 703-707, 1979

[25] YANG, L., ZHIQING, S. *"The Similarity Calculation of Concepts from Different Ontologies Based on Cosine"*. ICIII, 2010 International Conference, vol. 4, pp. 130-134, 2010. Accessible from: <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?tp=&arnumber=5694864>

[26] HARDY, G. H., LITTLEWOOD, J. E., PÓLYA, G. "*Tchebychef's Inequality*" §2.17 and §5.8 in Inequalities, 2nd ed. Cambridge, England: Cambridge University Press, pp. 43-45 and 123, 1988.

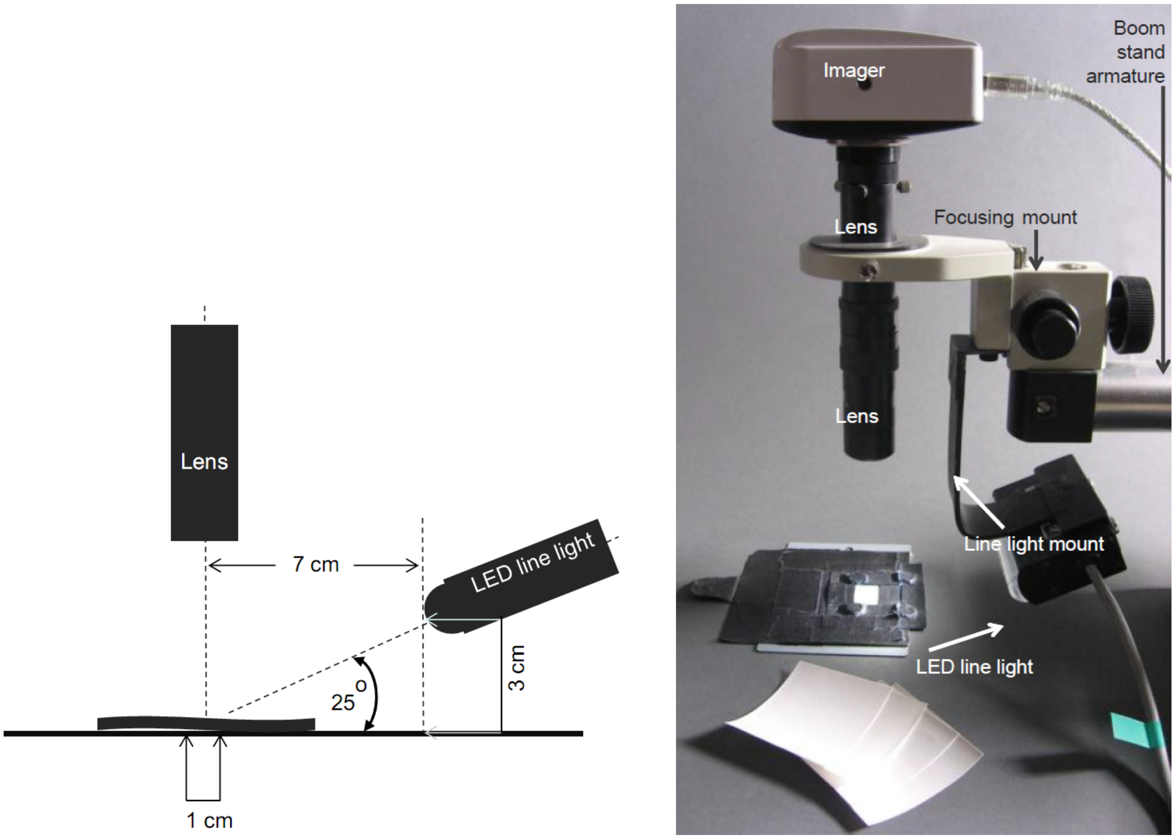
[27] SETHARES, W., INGLE, A., KRČ. T., WOOD, S. "*Eigentextures: An SVD Approach to Automated Paper Classification*" University of Wisconsin, United States. Submitted to ASOMILAR conference 2014.

[28] WOLFRAM MATHEMATICA - *Technical Computing Software* [online]. Official web pages. [quote 14/12/2013]. Accessible from: <http://www.wolfram.com/mathematica/>

Appendix

A Imager set-up

Fig. (A.1) shows the details of hardware set-up for obtaining the raking light images.



(a) (b)

Figure A.1: Set-up of the raking light imager: (a) schematic, (b) actual device. [8]

B Ink jet paper and canvas samples used in the data set

The number is the sequential numbering system suggested by the teams following image processing. The papers are identified by manufacturer, brand, manufacturer location and date of acquisition of papers. Other descriptors (such as surface finish details) are taken directly from the manufacturer packaging. All samples were drawn from the *Wilhelm Analog and Digital Color Print Materials Reference Collection* [11].

10 samples from the same sheet (3 sheets)

001-010. Canon Platinum Pro, Smooth, Glossy: Japan, Purchased 4/2012
011-020. Ilford Gallerie Gold Fibre Silk, Smooth, Glossy: Germany, ~ 2009 21-30.
021-030. Hahnemuhle Fine Art William Turner, Textured, Matte: Germany, ~2009

10 samples from the same package (3 packages)

031-040. Epson Premium Luster Photo Paper (roll) , Smooth, Semi-Glossy: Japan, Purchased 8/2002
041-050. Epson Ultra Premium Photo Paper Luster (Formerly called Epson Premium Photo Paper Luster), Smooth, Semi-Glossy: Japan, Purchased 7/2011
051-060. Epson Sample Roll Premium Luster Photo Paper, Smooth, Semi- Glossy: Japan, ~ 2001

10 samples from the same (or similar) manufacturing standard (3 sets)

061. HP Premium Plus Photo Paper, High gloss, Smooth Glossy: United States, 2006
062. HP Premium Plus, Glossy Photo Paper, Smooth, Glossy: Switzerland, 2002
063. HP Premium Plus Photo Paper, glossy, Smooth, Glossy: UK, 2001
064. HP Premium Plus Photo Paper, High gloss, Smooth, Glossy: United States, 2005
065. HP Premium Plus High Gloss Photo Paper, Smooth, Glossy: Switzerland, 2004
066. HP Premium Plus Photo Paper, High gloss, Smooth, Glossy: Switzerland, 2005
067. HP Premium Plus Photo Paper, High gloss, Smooth, Glossy: Switzerland, 2006
068. HP Premium Plus Photo Paper, High gloss, Smooth, Glossy: United States, 2007
069. HP Premium Plus Photo Paper, glossy, Smooth, Glossy: Switzerland, 2002
070. HP Premium Plus Photo Paper, High gloss, Smooth, Glossy: Switzerland, 2007
071. Epson Photo Quality Glossy Film, Smooth, Glossy: Japan, ~ 1996
072. Epson Photo Paper Glossy, Smooth, Glossy: Germany, Purchased 02/2008
073. Epson Premium Photo Paper Glossy (Formerly Premium Glossy Photo Paper), Smooth, Glossy: Japan, Purchased 03/29/2008
074. Epson Premium Photo Paper Glossy (Formerly Premium Glossy Photo Paper), Smooth, Glossy: Japan, Purchased 03/08/2008
075. Epson Ultra Premium Glossy Photo Paper, Smooth, Glossy: Japan, Purchased 03/2007
076. Epson Ultra Premium Photo Paper Glossy (Formerly Ultra Premium Glossy Photo Paper), Smooth, Glossy: Japan, Purchased 02/2008

077. Epson Ultra Premium Glossy Photo Paper, Smooth, Glossy, Japan, Purchased 02/2007

078. Epson Photo Paper Glossy (Formerly Glossy Photo Paper), Smooth, Glossy: Germany, Purchased 02/2007

079. Epson Premium Glossy Photo Paper, Smooth, Glossy: Japan, Purchased 06/2004

080. Epson Photo Paper Glossy, Smooth, Glossy: Germany, Purchased 05/2007

081. Kodak Ultima Picture Paper, High Gloss, Smooth, Glossy: Canada, Purchased 12/2003

082. Kodak Ultra Premium Photo Paper, High Gloss, Smooth, Glossy: Germany, Purchased 11/2011

083. Kodak Premium Photo Paper, Gloss, Smooth, Glossy: Germany, Purchased 06/2011

084. Kodak Premium Picture Paper, High Gloss, Smooth, Glossy: Canada, Purchased 12/203

085. Kodak Photo Paper, Gloss, Smooth, Glossy: Germany, Purchased 03/2009

086. Kodak Ultima Picture Paper, High Gloss, Smooth, Glossy: Canada, ~ 2002

087. Kodak Professional Ink jet Photo Paper, Smooth, Glossy: USA, Purchased 7/2004

088. Kodak Premium Picture Paper, High Gloss, Smooth, Glossy: Canada, Purchased 04/2003

089. Kodak Ultima Picture Paper, ultra Glossy, Smooth, Glossy: Canada/UK, Purchased 03/2004

090. Kodak Premium Photo Paper, Gloss, Smooth, Glossy: Germany/USA, Purchased 03/2007

30 samples showing diversity

091. Epson Ultra Premium Glossy Photo Paper, Smooth, Glossy: Japan, ~ 2005

092. Epson Ultra Premium Presentation Paper, Smooth, Matte: Japan, Purchased 4/2007

093. Canon Fine Art Paper Premium Matte, Smooth, Matte: Japan, ~ 2006

094. Canon Photo Paper Pro II, Smooth, Glossy: Japan, Purchased 12/2008

095. Epson Sample Roll Premium Luster Photo Paper, Smooth, Semi-Glossy: Japan, ~ 2001

096. Canson BFK Rives, Textured, Matte: France, Purchased 4/2008

097. Canson Rag Photographique, Smooth, Matte: France, Purchased 4/2008

098. Canson Museum Canvas Water Resistant Matte, Canvas/Textured, Matte: France, Purchased 4/2008

099. Canson Velin Museum Raf, Smooth, Matte: France, Purchased 4/2008

100. Canson Arches Aquarelle Rag, Textured, Matte: France, Purchased 4/2008

101. Ilford Gallerie Gold Fibre Silk, Smooth, Glossy: Germany, ~ 2009

102. Epson Exhibition Fibre Paper, Smooth, Soft-Gloss: Japan, Purchased 11/1/2007

103. Canson Artist Canvas Water Resistant Matte, Canvas, Matte: France, Purchased 04/2008

104. Epson Water Color Paper-Radiant White, Textured, Matte: Japan, ~ 2000

105. Canson Artist Canvas Professional Gloss, Canvas, Glossy: France, Purchased 04/2008

106. Canson Mi-Teintes, Honeycomb, Matte: France, Purchased 04/2008

107. Canson Edition Etching Rag, Smooth, Matte: France, Purchased 04/2008

108. Canson Montval-Torchon, Textured, Matte: France, Purchased 03/2008

109. Epson Cold Press Bright, Cold-Press Textured, Matte: Italy, Purchased 8/2010

110. Epson Hot Press Bright, Hot-Press Smooth, Matte: Italy, Purchased 03/2011

111. Epson Cold Press Natural, Cold-Press Texture, Matte: Italy, Purchased 8/2010

112. Epson Hot Press Natural, Hot-Press Smooth, Matte: Italy, Purchased 3/2011

113. Canon Photo Paper Pro (PR-101), Smooth, Glossy, Japan, ~ 2006

114. Canon Matte Photo Paper (MP-101), Smooth, Matte: Japan, ~ 2007

115. Ilford Galerie Smooth Gloss, Smooth, Glossy: Switzerland, ~ 2009

116. HP Premium Plus Photo Paper, High Gloss, Smooth, Glossy: Switzerland, ~ 2005

117. Epson ColorLife Photo Paper Semi Gloss, Smooth, Semi-Glossy: Switzerland, Purchased 2/2004

118. Kodak Ultima Picture Paper, Satin, Smooth, Glossy: Canada, ~ 2000

119. Epson Premier Art Matte Scrapbook Photo Paper, Smooth, Matte: Japan, ~ 2003

120. Hahnemuhle Fine Art William Turner, Textured, Matte: Germany, ~ 2009

C DVD

The attached DVD contains text of this thesis in PDF format, source code for the implementation of all 11 classification measures as well as source code for any calculation performed in the text. There is also a folder containing ink jet paper data set (described in detail in appendix B).