Classifiers for Matrix Normal Images: Derivation and Testing

Ewaryst Rafajłowicz(✉)

Faculty of Electronics, Wrocław University of Science and Technology,
Wrocław, Poland
ewaryst.rafajlowicz@pwr.edu.pl

Abstract. We propose a modified classifier that is based on the maximum a posteriori probability principle that is applied to images having the matrix normal distributions. These distributions have a special covariance structure, which is interpretable and easier to estimate than general covariance matrices. The modification is applicable when the estimated covariance matrices are still not well-conditioned. The proposed classifier is tested on synthetic images and on images of gas burner flames. The results of comparisons with other classifiers are also provided.

Keywords: Matrix normal distribution · Bayesian classifier
Classification of flames

1 Introduction

Our aim is to discuss classifiers dedicated to image recognition. We consider images as matrices of their gray levels. In opposite to the mainstream literature in the field, we avoid a feature selection as the first step before image recognition. Conversely, we consider images to be recognized as whole entities. Additionally, we avoid the vectorization of matrices because for large images (10 MPix, say) it is not only inconvenient but also destroys their covariance and neighborhood structures.

We adopt random matrices as models of images. We confine our attention to random matrices having Gaussian distributions since their description requires the knowledge of the mean and covariance matrices. However, we have to confine even more, because the covariance matrix of a 100×100 pixels image has about 10^8 elements and their estimation would require a huge amount of data. Thus, we need a class of random matrices with a more specialized covariance structure. The so-called matrix normal distribution (see [7]) seems to be a good choice, because their overall covariance matrices have a special structure that can be fully described by two matrices of covariances between rows and columns, respectively. These matrices are much smaller and easier to estimate (see Section... for details). Furthermore, it seems that inter-row and inter-column covariances are dominating for a sufficiently broad class of real-life images.

L. Rutkowski et al. (Eds.): ICAISC 2018, LNAI 10841, pp. 668–679, 2018.
https://doi.org/10.1007/978-3-319-91253-0_62

Clearly, images as inputs of classifiers re-appeared in the literature starting from the perceptron for classifying hand-written digits ([3] and the bibliography therein). Also later on random matrices with spatially Markov structure, generated by cliques, were extensively studied (see [5]). However, according to our knowledge, random normal matrices as models of images to be recognized are not considered. The research that is closest to ours was done by Krzysko [4] who considered the problem of recognizing random normal matrices that arise by stacking vectors, having their own interpretation, into a matrix. This idea was extended in [8] to recognizing sequences of images by stacking them into longer matrices.

The paper is organized as follows.

- In Sect. 2 we state the problem and describe basic properties of random normal matrices for the reader's convenience.
- The derivation of the Bayes recognizer is presented in Sect. 2 and then its empirical version is proposed, which is based on the plug-in principle, using the matrix learning sequence for the maximum likelihood estimation of inter-row and inter-column covariances.
- The quality of the proposed algorithm is compared with the well known methods: the nearest neighbor (NN) and the support vector machine (SVM). The comparison is done by extensive simulations of the Monte-Carlo (MC) type, assuming that the assumed covariance structure is an adequate one.
- Finally, the proposed algorithm is compared with NN classifying a sequence of real-life images of gas burner flames. This time the method of cross-validation is used for comparisons and we are not sure whether the assumed covariance structure is an adequate one.

We notice that a similar sequence of gas burner flames was used for comparisons in [10,11]. In the latter paper a more traditional classifier – based on features extraction – was used. In the former, the classifier is based on whole images (without features extraction), but a special structure of the covariance matrix is not assumed. Instead, firstly medoid-based clustering is done. Thus, the results are incomparable with presented here, because of different assumptions.

2 Problem Statement

MND as Class Densities and Their Estimation. We assume that probability distributions of gray-level images from class $j = 1, 2, \ldots, J$ have MND with probability density functions (p.d.f.'s) of the form:

$$f_j(\mathbf{X}) = \frac{1}{c_j} \exp\left[-\frac{1}{2}\,\mathrm{tr}[U_j^{-1}(\mathbf{X}-\mathbf{M}_j)\,V_j^{-1}\,(\mathbf{X}-\mathbf{M}_j)^T]\right], \tag{1}$$

where ^{T} stands for the transposition and det[.] denotes the determinant of a matrix in the brackets, while F the normalization constants are given by:

$$c_j \overset{def}{=} (2\,\pi)^{0.5\,n\,m}\,\det[U_j]^{0.5\,n}\,\det[V_j]^{0.5\,m}, \tag{2}$$

where $n \times m$ matrices M_j's denote the class means matrices. Concerning the covariance structure of MND class densities:

1. $n \times n$ matrix U_j denotes the covariance matrix between rows of an image from j-th class,
2. $m \times m$ matrix V_j stands for the covariance matrix between columns of an image from j-th class.

The above definitions are meaningful only when $\det[U_j] > 0$, $\det[V_j] > 0$. Later on, we shall write shortly,

$$\mathbf{X} \sim \mathcal{N}_{n,\,m}(\mathbf{M}_j,\, U_j,\, V_j), \text{ for } j = 1,\, 2, \ldots,\, J \tag{3}$$

Remark 1. *In order to reveal to what extent MND is a special case of a general class of Gaussian p.d.f.'s let us notice that the formally equivalent description of MND is the following:*

$$vec(\mathbf{X}) \sim \mathcal{N}_{n\,m}(vec(\mathbf{M}_j),\, \Sigma_j),\ for\ j = 1,\, 2, \ldots,\, J, \tag{4}$$

where $vec(\mathbf{X})$ is the operation of stacking columns of matrix $\mathbf{X}$, while Σ_j is a $n\,m \times n\,m$ covariance matrix of j-th class, which is the Kronecker product (denoted as $\otimes$) of U_j and V_j, i.e.,

$$\Sigma_j \overset{def}{=} U_j \otimes V_j, \quad j = 1,\, 2, \ldots,\, J. \tag{5}$$

Thus, MND densities have special covariance structure in comparison to a general multivariate Gaussian densities. Namely, their covariance matrices do not have inter rows-columns covariances, which makes them much easier to estimate.

Now, let us consider the case when $\mathbf{M}_j$, U_j, V_j,s are unknown and they have to be estimated from learning sequences.

Further on, we assume that we have J learning sequences of the following form: $\mathbf{X}_i^{(j)}$, $i = 1,\, 2, \ldots N_j$, $j = 1,\, 2, \ldots,\, J$. They are used for calculating estimates, further denoted by the same letter with the hat $\hat{}$.

We estimate the class mean matrices and a priori class probabilities in a classical way, i.e.,

$$\hat{M}_j = N_j^{-1} \sum_{i=1}^{N_j} \mathbf{X}_i^{(j)}, \quad \hat{p}_j = N_j/N, \quad j = 1,\, 2, \ldots,\, J. \tag{6}$$

The necessary condition for estimating the covariance matrices is to have a sufficient number of observations. In [6] it was proved that the following requirements should be satisfied:

$$N_j \geq \max\left\{\frac{n}{m},\, \frac{m}{n}\right\} + 1, \quad j = 1,\, 2, \ldots,\, J. \tag{7}$$

These conditions are less demanding than the classic one: $N_j > m\,n$. The reason is that the overall covariance matrix has a special structure.

The maximum likelihood estimates (MLE) of the inter-row and inter-column covariance matrices for classes are coupled by the following set of equations (see [6,12]):

$$\hat{U}_j = \frac{1}{N_j\, m} \sum_{i=1}^{N_j} (\mathbf{X}_i - \hat{\mathbf{M}}_j)\, \hat{V}_j^{-1}\, (\mathbf{X}_i - \hat{\mathbf{M}}_j)^T, \tag{8}$$

$$\hat{V}_j = \frac{1}{N_j\, n} \sum_{i=1}^{N_j} (\mathbf{X}_i - \hat{\mathbf{M}}_j)^T\, \hat{U}_j^{-1}\, (\mathbf{X}_i - \hat{\mathbf{M}}_j) \tag{9}$$

for $j = 1,\, 2, \ldots,\, J$. Equations (8) and (9) can be solved by the freezing method. In more detail:

Step 1. $\hat{U}_j$ in (8) is calculated replacing $\hat{V}_j^{-1}$ by the identity matrix,
Step 2. $\hat{U}_j$ is substituted into (9) and $\hat{V}_j$ is calculated,
Step 3. $\hat{V}_j$ is substituted into (8) and return to Step 2 (until convergence).

It was proved in [12] that one can perform only one iteration between Step 3 and Step 2 to obtain the efficient estimates of U_j and V_j.

When the estimates $\hat{U}_j$ and $\hat{V}_j$ are obtained, then one can easily define and calculate estimates $\hat{c}_j$ for c_j's. Indeed, the empirical, plug-in type, version of (2) is the following:

$$\hat{c}_j = (2\,\pi)^{0.5\,n\,m}\, \det[\hat{U}_j]^{0.5\,n}\, \det[\hat{V}_j]^{0.5\,m}. \tag{10}$$

Thus, we have estimators for all the unknown parameters of the matrix normal class densities.

A Modified MAP Classifier for Matrices. In this section we firstly derive the maximum a priori probability (MAP) classifier, assuming for a while that f_j's are known. Then we consider its empirical version of the plug-in type. Finally, we add an important modification, namely occasionally switching to the nearest mean classifier, when the estimates of the covariance matrices are ill-conditioned. When new images appear and improve the estimated covariance matrices, then we return to the proposed classifier, which is later called the matrix classifier (MCL).

Denote by $p_j > 0$, $j = 1,\, 2, \ldots,\, J$, $\sum p_j = 1$, a priori class probabilities. It is well known that in a general case the MAP classifier assigns $\mathbf{X}$ to class j^* such that

$$j^* = \arg\max_j [p_j\, f_j(\mathbf{X})], \tag{11}$$

where $\arg\max_j$ [ ] stands for the argument for which the maximum is attained. It is also well known that this rule is the optimal one when the 0–1 loss function is used (see, e.g., [2]). Applying this rule to (1) and using the fact that the logarithm is a strictly increasing function we obtain the following decision rule:

When $\mathbf{M}_j$ and U_j, V_j, p_j, $j = 1, 2, \ldots, J$ are known, then from (11) and (1) we obtain: classify matrix (image) $\mathbf{X}$ to Class j^* if

$$j^* = \arg\min_j \left[\frac{1}{2}\, \mathrm{tr}[U_j^{-1}(\mathbf{X} - \mathbf{M}_j)\, V_j^{-1}\, (\mathbf{X} - \mathbf{M}_j)^T] \right] - \log(p_j/c_j) \tag{12}$$

The expressions in the brackets play the role of the Mahalanobis distances. The matrices U_j^{-1} and V_j^{-1} de-correlate rows and columns of an image, respectively. Thus, in a general case, the optimal classifier is quadratic in $\mathbf{X}$ and we have to know (or to estimate) all parameters: $\mathbf{M}_j$ and U_j, V_j, p_j, $j = 1, 2, \ldots J$. Their estimators were proposed in the previous subsection. Substituting them into (12), we obtain the following empirical MAP classifier: classify image (matrix) $\mathbf{X}$ to the class $\hat{j}$, where

$$\hat{j} = \arg \min_{1 \le j \le J} \left[\frac{1}{2}\, \mathrm{tr}[\hat{U}_j^{-1}(\mathbf{X} - \hat{\mathbf{M}}_j)\, V_j^{-1}\, (\mathbf{X} - \hat{\mathbf{M}}_j)^T] \right] - \log(\hat{p}_j/\hat{c}_j). \tag{13}$$

Although this classifier is the empirical version of the optimal one, it does not give a full guarantee that it works well in practice. The reasons are (at least) the following:

1. Condition (7) is the minimal one in the sense that it yields the existence of the inverse matrices $\hat{U}_j^{-1}$ and $\hat{V}_j^{-1}$. However, we even do not know whether $\hat{U}_j$ and $\hat{V}_j$ are sufficiently well-conditioned, which is a prerequisite for obtaining their inverses in a numerically stable way.
2. It is not clear to what extent (13) retains its good properties in the case when underlying class distributions are not matrix normal.

We tackle problem 1 in this section, while problem 2 is discussed in the last section.

Recall that for symmetric and positive definite matrix A, its (Euclidean) condition number κ is defined as follows:

$$\kappa = \frac{\lambda_{max}(A)}{\lambda_{min}(A)}, \tag{14}$$

where $\lambda_{max}(A)$ and $\lambda_{min}(A)$ denote the largest and the smallest eigenvalue of A, that are real and positive. It is also known that κ acts as an amplifier of errors that occur during matrix inversion. When κ is in the range between 1 and about 100, one can expect low numerical errors in matrix inversion algorithms. In practice, especially for large matrices, easily computable bounds for κ are used. However, in our computational experiments, reported in the section before last, we used directly (14) for matrices of a moderate size.

Taking the above discussion into account, we propose the following algorithm for classifying images (matrices) having the matrix normal distribution with class densities given by (1).

Matrix Classifier (MCL)

Step 0. Select $0 < \kappa_{max} < 100$.

Step 1. Initial learning phase: calculate $\hat{p}_j$, $\hat{U}_j$, $\hat{V}_j$, $\hat{c}_j$, $\hat{M}_j$, $j = 1, 2, \ldots, J$ for the learning sequence.

Step 2. Check whether all the inequalities:

$$\hat{\kappa}_j^{(1)} \overset{def}{=} \frac{\lambda_{max}(U_j)}{\lambda_{min}(U_j)} < \kappa_{max}, \; j = 1, 2, \ldots, J \tag{15}$$

as well as

$$\hat{\kappa}_j^{(2)} \overset{def}{=} \frac{\lambda_{max}(V_j)}{\lambda_{min}(V_j)} < \kappa_{max}, \; j = 1, 2, \ldots, J \tag{16}$$

are fulfilled. If so, go to Step 3, otherwise, go to Step 4.

Step 3. Classify new image (matrix) $\mathbf{X}$ according to the rule described as (13), acquire the next image (matrix) for classification and repeat Step 3.

Step 4. Classify new image (matrix) $\mathbf{X}$ according to the nearest mean rule, i.e., classify it to the class

$$\tilde{j} = \arg\min_j ||\mathbf{X} - \hat{M}_j||^2, \tag{17}$$

where the squared distance $||\mathbf{X} - \hat{M}_j||^2$ is defined as follows:

$$||\mathbf{X} - \hat{M}_j||^2 \; = \; \mathrm{tr}[(\mathbf{X} - \hat{\mathbf{M}}_j)\,(\mathbf{X} - \hat{\mathbf{M}}_j)^T]. \tag{18}$$

Update the estimates of $\hat{U}_{\tilde{j}}$ and $\hat{V}_{\tilde{j}}$ by adding current $\mathbf{X}$ to the learning sequence as $(\mathbf{X}, \tilde{j})$ and go to Step 2.

Notice that the update of $\hat{U}_{\tilde{j}}$ and $\hat{V}_{\tilde{j}}$ can be done recursively. A distinguishing feature of the MCL algorithm is that it has not only the learning phase (as usual), but also it learns at the recognition phase when the result of classification may be unreliable. Conditions (15) and (16) serve as indicators of this. The additional learning is supported by a surrogate classifier, which can be arbitrary. Here, we use the nearest mean classifier for this purpose as it is closest to the MAP classifier and does not need to store the whole learning sequence.

3 Performance Evaluation by Monte Carlo Experiments

In this section, we provide the results of extensive Monte Carlo (MC) simulations for evaluating the performance of MCL. We take advantage that in MC simulations we can repeat 1000 times the learning and the testing phase, having proper classifications for comparisons with MCL outputs. The results of these simulations are also used for comparisons of MCL performance with the classification errors committed by the nearest neighbor (NN) and the support vector machine (SVM) classifiers, implemented in the Wolfram's Mathematica ver. 11.

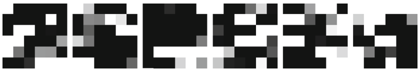

Fig. 1. Excerpt of samples from Class 1.

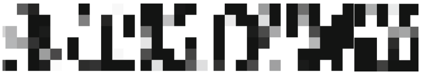

Fig. 2. Excerpt of samples from Class 2.

The following parameters were used during simulations. Classified 6 × 6 matrices (images) were drawn at random from two classes, each having the MND with

- mean matrices M_1 being a 6×6 matrix of zeros, while M_2 is a 6×6 constant matrix with 0.05 as elements.
- covariance matrices: $U_1 = 0.1\, I_6$, V_1 is a 6×6 tridiagonal matrix with 2 at the main diagonal and 1 along the bands running above and below the main diagonal, $U_2 = V_1$, $V_2 = I_6$, where I_6 is a 6×6 identity matrix,
- $p_1 = p_2 = 0.5$.

Samples of the simulated learning sequences are shown in Figs. 1 and 2 for classes 1 and 2, respectively. As one can notice, class densities were selected in such a way that the samples are not easily distinguishable. The results of testing are shown in Figs. 3 and 4. Each point at these figures was obtained from 1000 MC simulation runs. They present statistics of the percentage of errors (% Err) committed by MCL, SVM and NN classifiers as a function of the learning sequence length N_d, i.e., $N_1 = N_2 = N_d$. From Fig. 3 it is clear that MCL provides a much smaller mean and median of errors than those of NN and SVM classifiers. Furthermore, also maximal classification errors, calculated as the maximal % of errors from 1000, is essentially smaller for MCL (see the left panel in Fig. 4). The variability of % Err, measured by the dispersion, is also smaller for MCL than for NN and SVM classifiers (see the right panel in Fig. 4). Thus, one can expect a more reliable classification for different learning sequences.

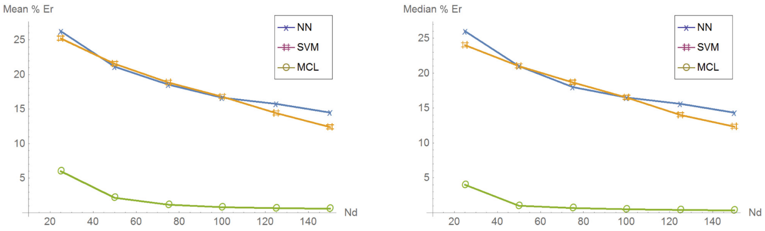

Fig. 3. The mean and the median of % of classification errors.

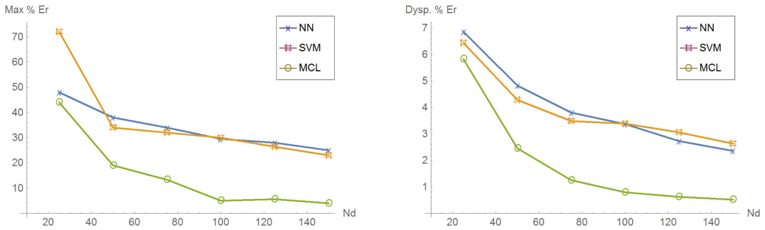

Fig. 4. The maximal % Err (Left), the dispersion of % Err (right).

As the conclusion from the above MC simulations, one can say that the MCL classifier outperforms the NN and SVM classifiers when MND assumptions hold.

4 Testing MCL on Images of Flames of a Gas Burner

Our aim in this section is to test the MCL on real data, namely on images of flames of an industrial gas burner. We also compare its performance with the 10 nearest neighbors (10–NN) classifier, where the number of neighbors was selected as the best after a series of preliminary experiments. The images of flames are classified to three classes (see Fig. 6). Namely,

Class 1 – improper combustion – too high air supply rate,
Class 2 – proper combustion – proper air supply rate,
Class 3 – improper combustion – too low air supply rate.

The result of the classification of a current flame image can be used to decide whether to decrease, leave unchanged or increase the air supply rate, respectively. This classification is rather rough and serves mainly for illustration purposes. We refer the reader to [9,13] for more detailed explanations of relationships between flames images and the combustion process.

Preparation of Flames Images. The original images of flames were stored as 170×352 RGB images. Before using them for classification, they were converted to $[0, 1]$ gray-levels scale and their size is reduced to 17×36 by 10 times downsampling. An example of an original image and its downsampled version is shown in Fig. 5.

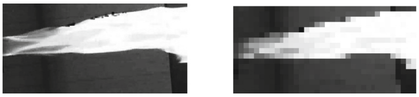

Fig. 5. An original image and its down-sampled version.

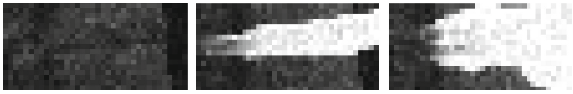

Fig. 6. Examples of images: Classes 1–3 (from the left).

Learning, Testing and Comparing Classifiers. The sequence of flames images that was at our disposal consists of 145 items. They were classified to the above-mentioned classes by the author. The resulting labeled sequence will be further denoted as H_{145}.

The Methodology of Cross-Validation Testing

In opposite to the MC testing, here we do not have the freedom of generating learning and testing examples freely. Instead, we used the well-known cross-validation (CV) methodology of testing:

Step 1. Select from H_{145} (at random with the same probabilities) a learning sequence of the length 45 and denote it as L_{45}. The rest of H_{145} denote as T_{100} and use it for testing.

Step 2. Learn MCL and 10-NN classifiers, using the L_{45} sequence.

Step 3. Test the classifiers from Step 2, applying them to T_{100}, calculate and store the percentages of errors committed by them.

Step 4. Repeat Steps 1–3 1000 times.

Notice that this is an intensive testing procedure because we have to estimate three matrices of the means and six covariance matrices 1000 times when learning MCL. Also, 10-NN method is time-consuming at the testing phase.

In order to illustrate the testing procedure, we provide the results of learning from only one of its runs.

A priori probabilities, estimated from the frequencies of occurrence in this run, are the following $\hat{p}_1 = 0.37$, $\hat{p}_2 = 0.34$, $\hat{p}_3 = 0.29$. In other CV runs this pattern is similar.

Mean images for each class in this particular CV run are shown in Fig. 7 and again, in other runs these images are similar. In Fig. 8 the inter-column 36×36 covariance matrices for the same CV run and for all three classes are shown as images (black pixels correspond to larger (co-)variance). In the case of covariance matrices, the variability between CV runs is larger than for the mean, but the general pattern remains similar. Namely, Class 1 images have almost uncorrelated columns and the correlation grows for Classes 2 and 3. In Fig. 9 the inter-column 17×17 covariance matrices for the same CV run and for all three classes are shown as images (black pixels correspond to larger (co-)variance). As above, the variability between CV runs is larger than for the mean, but the general pattern, although different than for columns, remains similar. Namely, for Class 1 almost all rows are similarly correlated, while for Class 2 mainly the central rows have higher correlations. In opposite, for Class 3 almost all rows are correlated. This justifies the hope that MCL will have a low classification error also in this case.

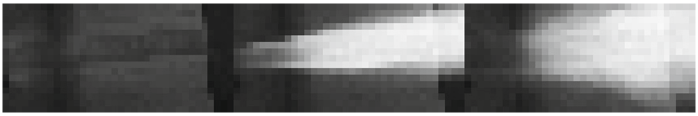

Fig. 7. Mean images for classes in one of the CV runs.

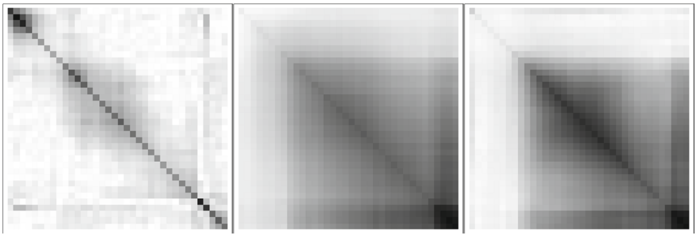

Fig. 8. Inter-column 36 × 36 covariance matrices in one CV run.

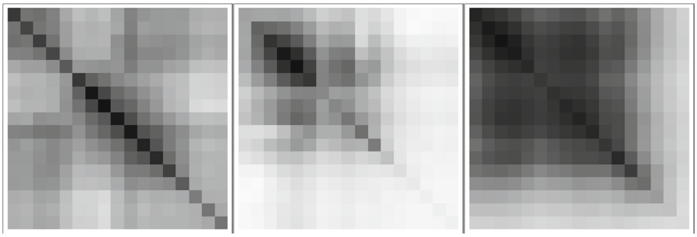

Fig. 9. Inter-rows 17 × 17 covariance matrices in one CV run.

The Results of Cross-Validation Testing. The results of testing of the MCL (with $\kappa_{max} = 50$) and 10-NN algorithm, using the above described CV methodology, are shown in Table 1. The analysis of Table 1 is equivocal. The mean % Err of the 10-NN method is smaller than that of the MCL, but the comparison of the medians yields the opposite results. A partial explanation of this behaviour is shown in Fig. 10. As one can notice, % Err of the MCL is much more concentrated around the mean than that of 10-NN (although the estimated dispersions suggest the opposite). Another partial explanation of this behavior is presented in the next subsection.

Table 1. The CV comparison of MCL and 10-NN % errors.

% Err.	Mean	Med.
10-NN	12.5	12.2
MCL	15	10

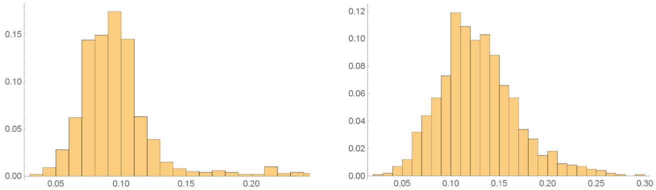

Fig. 10. The histograms of % Err for the MCL (left) and the 10-NN (right).

Baringhaus-Henze Test for Classes. The Baringhaus-Henze test (see, e.g., [1]) for the multivariate normality of observations from Classes 1–3 was performed. The results are summarized in Table 2. As one can observe, for observations from Classes 2 and 3 the test rejects the hypothesis that observations are multivariate normal. In fact, this test should be performed before MCL was used. We provide its results here, as one more partial explanation that the 10-NN classifier behaves comparably well or slightly better than the MCL, which is optimal, but only when observations have multivariate normal distributions with the special structure of the covariance matrix.

Table 2. The results of testing the multivariate normality of classes.

	Statistic	P-value
Class 1	0.999997	0.379991
Class 2	1.00041	0
Class 3	1.00006	0

5 Recommendations and Conclusions

A modified MAP classifier for images (matrices) having matrix normal distribution was derived and extensively tested both on synthetic MND data as well on down-sampled images of gas burner flames. Conclusions and recommendations from testing MCL are the following.

- When images (matrices) come from classes having matrix normal distributions, then the MCL essentially outperforms the SVM and the NN classifiers and can safely be recommended. The Baringhaus-Henze test can be used in order to test MND for classes.
- When for the observations the Baringhaus-Henze test fails, then the classification errors of the MCL and the 10-NN are comparable and it can be more reliable to apply the xx-NN or another nonparametric classifier (see [2]).

It can be of interest to develop nonparametric classifiers dedicated to classifying images (matrices).

References

1. Baringhaus, L., Henze, N.: A consistent test for multivariate normality based on the empirical characteristic function. Metrika **35**(1), 339–348 (1988)
2. Devroye, L., Gyorfi, L., Lugosi, G.: A Probabilistic Theory of Pattern Recognition. Springer, Berlin (2013)
3. Haykin, S.S., et al.: Neural Networks and Learning Machines. Pearson, Upper Saddle River (2009)
4. Krzysko, M., Skorzybut, M., Wolynski ,W.: Classifiers for doubly multivariate data. Discussiones Mathematicae: Probability & Statistics, p. 31 (2011)
5. Li, S.Z.: Markov Random Field Modeling in Image Analysis. Springer, Berlin (2009). https://doi.org/10.1007/978-1-84800-279-1
6. Manceur, A.M., Dutilleul, P.: Maximum likelihood estimation for the tensor normal distribution: algorithm, minimum sample size, and empirical bias and dispersion. J. Comput. Appl. Math. **239**, 37–49 (2013)
7. Ohlson, M., Ahmad, M.R., Von Rosen, D.: The multilinear normal distribution: Introduction and some basic properties. J. Multivar. Anal. **113**, 37–47 (2013)
8. Rafajłowicz, E.: Data structures for pattern and image recognition with application to quality control Acta Polytechnica Hungarica, Informatics (under review)
9. Rafajłowicz, E., Pawlak-Kruczek, H., Rafajłowicz, W.: Statistical classifier with ordered decisions as an image based controller with application to gas burners. In: Rutkowski, L., Korytkowski, M., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2014. LNCS (LNAI), vol. 8467, pp. 586–597. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-07173-2_50
10. Rafajłowicz, E., Rafajłowicz, W.: Image-driven decision making with application to control gas burners. In: Saeed, K., Homenda, W., Chaki, R. (eds.) CISIM 2017. LNCS, vol. 10244, pp. 436–446. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-59105-6_37
11. Skubalska-Rafajłowicz, E.: Sparse random projections of camera images for monitoring of a combustion process in a gas burner. In: Saeed, K., Homenda, W., Chaki, R. (eds.) CISIM 2017. LNCS, vol. 10244, pp. 447–456. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-59105-6_38
12. Werner, K., Jansson, M., Stoica, P.: On estimation of covariance matrices with Kronecker product structure. IEEE Trans. Sig. Process. **56**(2), 478–491 (2008)
13. Wójcik, W., Kotyra, A.: Combustion diagnosis by image processing. Photonics Lett. Pol. **1**(1), 40–42 (2009)