

Data Structures for Pattern and Image Recognition and Application to Quality Control

Ewaryst Rafajłowicz

Faculty of Electronics, Wrocław University of Science and Technology,
ul. Janiszewskiego Street 11/17, 50-372 Wrocław, Poland
ewaryst.rafajłowicz@pwr.edu.pl

Abstract: Our aim is to propose a systematics (taxonomy) of data structures that arise in classifying patterns and images, starting from unrelated vectors and ending with matrix and tensors for storing video sequences. Then, we discuss possibilities of classifying such structures under matrix (tensor) normal distribution assumptions. Finally, we provide a case study of using classifiers for quality control of laser-based additive manufacturing.

Keywords: pattern recognition; image recognition; data structures; additive manufacturing; image-based control; matrix normal distribution; Kronecker product; covariance structure estimation; cloud storage

1 Introduction

Pattern classification (recognition) is one of the oldest tools of the artificial intelligence. It has been developing for more than fifty years (see [1]). The main emphasis of researchers was and still is put on developing methods and algorithms of learning classifiers [2]. The mainstream of research is concentrated on recognizing patterns that are modeled as random vectors in the Euclidean space.

When images are recognized, the typical approach is based on their preprocessing in order to extract relevant features from them and to form vectors, which are then classified using classifiers dedicated to vector input data. A success of such an approach depends not only on a selected classifier and its learning but mainly on selecting proper features. Clearly, at the beginning of the era of computers, this approach was the only possible. Even at the beginning of the nineties, a typical PC had troubles with processing a moderate size image. In recent twenty years, however, the speed of computers and mainly a rapid growth of storage devices are developing so quickly that we are able to cluster and recognize images as a whole, without laborious (and risky) process of defining and extracting relevant features.

1.1 Motivation

Our main motivations come from computer science and decision-making theory. However, putting an emphasis on data structures for images and image sequences recognition one has also immediate associations and questions about how brain stores images. There are a large number of papers on these topics (see [3], [4], [5], [6] for an excerpt of those which are close to the topics of this paper). It is also known (see [7]) that process of memorizing and the retrieval of images in our brain is very complicated with many feedbacks. Having this in mind, we would like to touch only one aspect of the memorizing images in the long-term memory, namely, how our brain copes with a very common kind of redundancy caused by different illumination of the same object (see Figure 1 for the author's photo). We certainly are not able to answer this question, but one of the mathematical tools discussed in this paper, namely, the Kronecker product of matrices provides a simple model for coping with this kind of redundancy. In fact, images shown in Figure 1 have been obtained as the Kronecker product of the original image and the vector $[1, 0.6, 0.4]$ (see [8] for more facts concerning the Kronecker product, tensors, and operations on them).



Figure 1

One of our motivations for considering the Kronecker product structure for image sequences. The sequence of images that are taken with different illuminations can be stored as the Kronecker product of the first of them and the vector $[1, 0.6, 0.4]$.

On the other hand, one can already meet databases containing one trillion images (see [9]) and one can expect that – due to cloud resources – even larger databases can be virtually organized. Recent examples which indicate that there are needs for cloud image databases and for image classification, grouping, clustering etc. are provided in [10], [11], [12]. From the viewpoint of an image cloud organization, managing and retrieval it is of importance to standardize data structures. In Section 2 we provide a brief review of data and images structures that are convenient for their classification.

The results of classifying images can be used as input for data mining in the so-called Big Data context (see [13] for the recent survey on these topics). However, in many cases, the results of classifying images can be applied directly to decision making, as it is illustrated in this paper. Namely, we propose and describe briefly a decision-making system for additive manufacturing, which is based on detecting changes between short image sequences.

As it follows from the following excerpt of papers on industrial image processing: [14], [15], [16], [17], one can expect that needs for storing and processing huge, dedicated databases of images will be growing and cloud facilities can be an adequate answer for these needs.

Concerning possible applications of the results presented in this paper, they are directed to image-based decision-making that is based on learning. In particular, the image-based quality control is our main concern. As an illustration, we provide – in Section 6 -- an example of quality control of a laser additive manufacturing. Another example of possible applications is discussed in [18]. In [18] the states of an industrial gas burner are observed by a camera and used for decision-making. Notice that in opposite to the present paper, in [18] images are clustered, i.e., the learning without a teacher is applied.

1.2 Organization of this Paper

Our first aim is to provide a brief review of data structures that have already appeared in pattern recognition literature. The need for such a data structures for pattern and image recognition review stems from the fact that the topics of data structures for pattern recognition and/or clustering are discussed much less frequently than those of learning classifiers and they are scattered in the literature. Furthermore, relationships between data structures and the corresponding classifiers are frequently neglected. In our review we take into account the following features of data structures:

- An algebraic representation of patterns (images) as (vectors, matrices, tensors)
- Importance (or not) of ordering in time
- Relationships (dependencies) between class labels in a learning sequence
- An internal correlation structure of patterns (images) as well as possible correlations between them

Then, we shall discuss one more face of the "curse of dimensionality" that appears when we consider the estimation problem of correlation (covariance) structures of images and their sequences. This discussion leads to the need of imposing simplifying assumptions on class densities of patterns, images and their sequences. As an adequate set of class distributions, we select the matrix (tensor) normal distributions that have a special, the Kronecker product, structure of patterns (images) covariance structure.

In Section 4, we derive the Bayes decision rule for the matrix normal distribution (MND) Our next step is to discuss how to estimate the covariance matrices of MND and how to use them to plug-in into the Bayes optimal decision rules.

Finally, we consider an application of classifiers as change detectors in image sequences. It occurred that even the simplifying MND covariance assumption is not sufficient for estimating the covariance structure of sequences of images to be classified and the "competition" is won by a simple 5 nearest neighbor (5-NN) classifier, which neglects (at least partially) the covariance structure of sequences of images. But, as it is demonstrated at the last part of the paper, it is sufficient for a proper decision making in the additive manufacturing example.

Clearly, change detection is not the only application of structured image data classification. In fact, the emerging "data-intensive science", considered as a part of cloud databases (see [19]), will need classification and clustering of structured image sets even more than earlier.

Summarizing, the paper is structured as follows:

- Our main goal is to detect a change in image sequences – considered as the Bayes classification problem -- is discussed in Section 1.3.
- As the first step toward its solution, in Section 2 we provide the review of data structures for classification, taking into account not only data organization but also their correlation dependences. As the result, the class of matrices (or tensors) having the multivariate normal distributions is selected as a sufficiently general model for our purposes.
- In Section 3, the most important features of the MND's are summarized for the reader convenience since this specific class of probability distributions is not so widely known as the general class of multivariate normal distributions (GCMND).
- The Bayes classifier for MND classes is derived in Section 4. Although the Bayes classifier for GCMND is well known for many decades, its version for MND requires a re-derivation. The reason is in that MND is a sub-class of GCMND that has specific features, which should be reflected in a structure of the Bayes classifier and in the way of its learning.
- The learning procedure is proposed in Section 5. It takes into account both the specific structure of the Bayes classifiers for MND data and the specific way of estimating inter-row and inter-column covariance matrices of MNDs.
- Finally, in Section 6, we provide the results of testing the empirical Bayes classifier for MND images that arise in a laser additive manufacturing process.

1.3 Change Detection from Images

The idea of applying classifiers as change detectors in a sequence of images is depicted in Figure 2. It looks simple, but difficulties of its application depend on:

- 1) A priori knowledge about class densities (parametric or nonparametric),
- 2) A data structure (vectors, matrices, tensors),
- 3) Correlations inside each image and between them.

Success depends on a proper combination of 1) and 3). We refer the reader to [20], [21] for other approaches to change detection in image sequences.

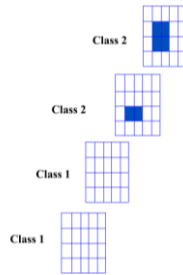


Figure 2

Main idea: change detection in an image sequence as the Bayes classification problem. At the first two frames (from below) changes are not present – they are classified to Class 1. When changes occur (upper two frames) and if they are correctly detected, then these frames are classified to Class 2 and the change is declared.

2 Data Structures for Classification

In this section we review data structures that are used in classification tasks, putting the main emphasis on sequences of images to be classified.

2.1 Classic Data Structure

In the classic problem statement, a pattern to be classified is a vector $x \in R^d$, say. The learning sequence $(x_i, L_i), i = 1, 2, \dots$ consists of such vectors and class labels L_i attach to them (see Figure 3). Usually, L_i 's are positive integers. In the standard, setting pairs $(x_i, L_i), i = 1, 2, \dots$ are assumed to be random and mutually independent. Their ordering in time is not taken into account when classifications are made. Within elements of x_i vectors correlations (or more complicated statistical dependence) are allowed.

2.2 Data Structures Arranged according to Class Labels

As far as we know, the first attempts of imposing a structure on the learning sequence of vectors $(x_i, L_i), i = 1, 2, \dots$ concentrated on subsequent class labels. Namely, it was observed that combinations of letters in words appear with different frequencies in a given natural language. This and other almost classic structures are listed below.

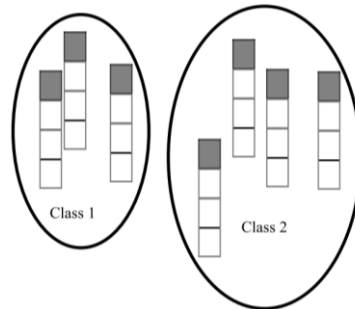


Figure 3

Classic structure: independent and identically distributed (inside classes) vectors of features with correlated elements plus class labels (gray), ordering in time – not taken into account

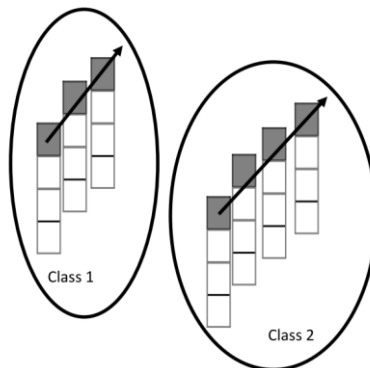


Figure 4

Almost classic structure: independent and identically distributed (inside classes) vectors of features with correlated elements plus class labels (gray), forming the Markov chain, ordering in time is taken into account, see [16]

- Markov chain dependence of labels: the result of the previous classification (e.g., a letter in a word) influences the next classification (e.g., the next letter, see [22]).
- Hierarchy of class labels – patterns arranged into classes, then – inside each class – organized into subclasses. The corresponding classifiers are

also hierarchical. The first attempts can be traced back to the eighties [23], [24] and this stream of research is still continued (see [25]).

In the Markov chain case, ordering in time of the learning sequence is important (see Figure 4). We mention the Markov chain of class labels for historical reasons only since it was one of the first attempts of imposing a structure on vectors of features to be classified. We shall not use this structure later on because it does not take into account correlations between vectors of features.

2.3 Non-Classic Matrix Structure - Repeated Observations of Patterns

An interesting, important for practice and theoretically appealing pattern recognition problem, is discussed in [26] and [27]. Namely, patterns to be classified are vectors, but this time, the learning sequence contains repeated observations of the same object. These observations are corrupted by noises (random errors). Also a new item to be classified consists of several noisy copies of the same object. Elements of each vector can be correlated. Additionally, batches of observed object can also be correlated (see Figure 5 in which possible correlations are depicted as curly brackets). Again, ordering in time appears as the important factor of this kind of data structure.

In [26] it is additionally assumed that data vectors have the normal distribution. This, in addition to the above-mentioned correlation structure, leads the authors of [26] to the conclusion that the overall structure of the learning sequence has the so-called matrix normal distribution (MND), which has a special form of the covariance matrix. Namely, its covariance matrix is the Kronecker product of covariance matrices between elements of feature vectors and between repeated observations of the same object.

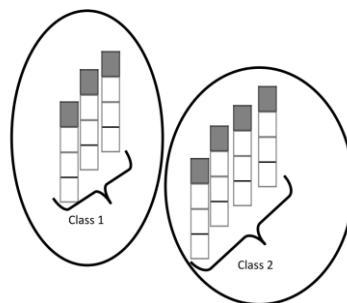


Figure 5

Non-classic matrix structure: correlated in time (curly brackets) vectors of features with correlated elements plus class labels (gray), ordering in time is taken into account, [26]. Two covariance matrices – the Kronecker product structure of the overall covariance matrix.

We shall discuss MND in more detail in the next section since we shall use it to describe image sequences. There are formal similarities between our development and [26], but there are also differences arising from the fact that in [26] random matrices arise by stacking together repeated observations of the same object, while in our case matrices are just images, coded in the gray-level convention.

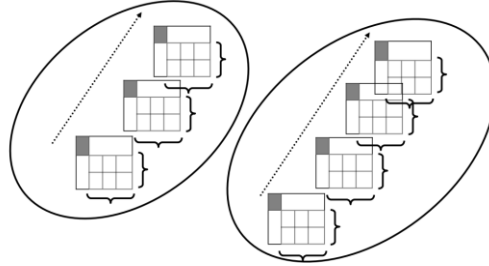


Figure 6

Basic matrix structure: uncorrelated in time matrices of features (gray levels) with correlated columns and rows plus class labels (gray). The covariance matrix is the Kronecker product structure of the row and columns covariance matrices. There is no a dependence structure imposed on class labels.

2.4 Why We Need Matrices and Tensors as Data Structures for Classification?

In this subsection, we pause our systematics of data structures for a while in order to explain why it is expedient to keep images as matrices and their sequences as tensors.

Formally, we can express matrices and tensors as vectors. Then, why it is important to keep images and tensors for classification in their original form?

- 1) A typical image has about 10 MPix and it is inconvenient to consider it as a vector. Indeed, when “Truecolor” images are stored, each of ten millions of pixels is represented by 24 bits.
- 2) The same is true for image sequences, where a vector containing a video would be rather ridiculous. A convenient structure for storing image sequences is the 3D tensor (see [8] for the definition and the fundamental operations on tensors).
- 3) A correlation structure is easier to impose when images are kept in its “natural” form since we can provide an interpretation to the following notions: between-rows and between-columns correlation matrices (see [26], [27]).

The last statement is explained in more detail in Section 3.2.

2.5 Basic Matrix Structure for Classifying Images

The structure described here is our main focus in this paper. It is well suited for classifying images with the main emphasis on detecting changes in their sequences. From this point of view ordering of images is important, but – in this model – previous decisions are not taken into account when a new classification is made. For example, when images of a properly produced item were recognized several times, this does not change the probability of classifying the next item as improper. Thus, there is no dependence between subsequent class labels. Each image is stored as a matrix with elements representing the gray levels of pixels. It is assumed that these matrices are normal random matrices. Their covariance structure is the Kronecker product of rows and columns covariance matrices (see Figure 6). This structure is described in Section 3.3.

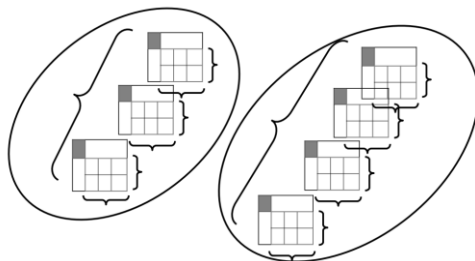


Figure 7

Tensor product structure: Correlated in time matrices of features with correlated columns and rows plus class labels (gray), ordering in time IS TAKEN into account. Three covariance matrices - the Kronecker product structure of the row and columns covariance matrices and between images covariance. There is no a dependence structure imposed on class labels.

2.6 Extended Basic Structure

The next step in the hierarchy of data structures is the one similar to that described in the previous section (see Figure 7), but additionally admitting a correlation dependence between matrices (images) in a sequence. Thus, we have three covariance matrices: between rows, columns and between matrices (images). Their Kronecker product forms the overall covariance matrix of MND. Clearly, the time ordering is important, but we do not assume a dependence between labels.

When classification is made for change detection – we classify each image but taking between images correlation into account. We omit a discussion of this case later.

2.7 Data Structures for Detecting Changes in Video Sequences

Up to now, elements of learning sequences were either vectors or matrices ordered in time (or not), correlated along different directions (or not). The next level in our hierarchy of data structures consists of sequences of matrices (tensors) that are ordered in time. In particular, this structure can describe ordered sequences of images, i.e., video sequences. Classifying such objects is as important as difficult. Notice that this time we classify all video and when we want to detect changes, we must take into account all images in the sequences. In other words, objects to be classified are 3D tensors.

This structure is much more data demanding to learn a classifier. We comment on how to reduce the amount of data in the last section, but the trick applied there can be used for short image sequences only.

2.8 Outside the Systematics

The above systematics of data structures was done from the point of view of classifying objects. For this reason, not only their algebraic description as vectors, matrices and 3D tensors was taken into account, but also importance (or not) of time ordering and a correlation structure.

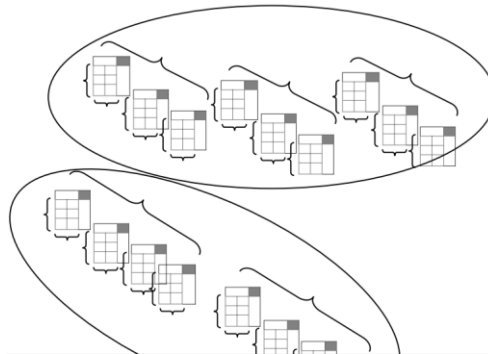


Figure 8

Change detection in video sequences: no correlation between video sequences, correlated in time matrices of features with correlated columns and rows plus class labels (gray). There is no a dependence structure imposed on class labels. This is outside our scope today.

This systematics is neither exhaustive nor complete. For example, we confined ourselves to gray level images. By adding colors (e.g., in RGB format) one can easily extend the proposed taxonomy. On the other hand, this systematics takes

only main factors influencing pattern recognition into account. Additional factors that may influence the result of classification include.

- Outer context (see [28], [18]) which is not a feature of an object to be classified, but influences the result of classification (e.g., a lighting of a scene).
- Ordered labels with different losses attached, depending on how far are current decisions from the proper one (see [29], [30], [31]).
- Topology in the space of labels (e.g., rectangular net for objects localization [32]).

Outside this systematics remains also an interesting approach proposed in [33] for semi-supervised learning. The data structure considered in this paper consists of initial labeled data and data labeled in the co-training process.

3 Bayesian Framework for Classifying Images

Our aim in this section is to provide a Bayesian framework for classifying images and – in particular – to apply it for change detection in an image sequence by classifying each image in it. Clearly, Bayesian classifiers are widely used for image classification at least from 1960', but the main stream of research and applications follows the scheme depicted in Figure 9, i.e., firstly relevant features are defined and extracted from images. Then, a vector of features is classified. The main difference between this approach and the approach considered in this paper is in that we consider images (matrices) as whole entities and they are classified as such. The present approach should not be confused with the one proposed in [21], where changes in an image sequence were detected by tracking, separately, gray levels of each pixel along the time axis (see Figure 10 for a sketch of this idea).



Figure 9

The most common approach: features extraction from each image and then apply a classifier or a change detector. Applicable when one can define features relevant to classes (changes).

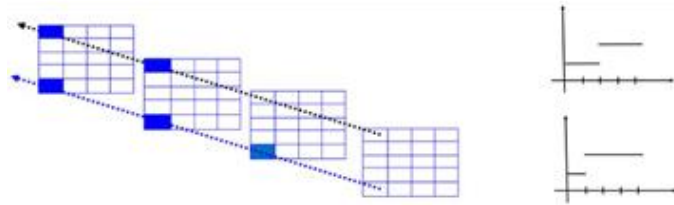


Figure 10

Spatio-temporal change detection: Changes in gray levels of each spatial location (pixel) are tracked separately, the out-of-control state is declared when a group of pixels changed.

3.1 Bayesian Classifier for Matrices (images)

Denote by X an $n \times m$ random matrix with the probability density function either $f_1(X)$ when X is drawn from Class 1 (e.g., in-control behavior) or $f_2(X)$ when X is drawn from Class 2 (e.g., out-of-control).

Remark: We confine to X 's from two classes for simplicity. Immediate generalization is possible for more than one scenario of out-of-control behavior.

Let $p_1 > 0, p_2 > 0, p_1 + p_2 = 1$ be a priori probabilities that X comes from class 1 or 2. Selecting the so-called 0-1 loss function, the optimal classifier (minimizing the Bayes risk) is of the form (see [1], [2]):

classify X to Class 1 if $p_1 f_1(x) > p_2 f_2(x)$ (1)

and to Class 2, otherwise.

3.2 Lack of Data for Learning a Matrix Classifier

In practice, f_1 and f_2 unknown, but we have two learning sequences: $X_i^{(1)}, i = 1, 2, \dots, N_1$ for estimating f_1 and $X_i^{(2)}, i = 1, 2, \dots, N_2$ for estimating f_2 . The classification of learning examples is assumed to be correct (done by an expert).

Data structures for pattern and image recognition. It is customary to distinguish two main approaches to learning classifiers:

- D) A nonparametric approach: f_1 and f_2 are unknown and they are estimated (e.g., by the Parzen kernel method). Application of this approach to image sequences is impossible since for a typical 1 Mpix \approx

$10^3 \times 10^3$ image matrix X one would need hundreds millions of learning examples.

- II) A parametric approach: f_1 and f_2 are assumed to be members of a parametric family of probability density functions, e.g., the Gaussian one. Still (almost) impossible to apply, because the covariance matrix would be as large as $10^6 \times 10^6$ for 1 MPix image. Again hundreds of millions of learning examples would be needed to estimate it.

What can we do ?

- a) To apply a heuristic classifier.
- b) To assume that f_1, f_2 are Gaussian and completely neglect the covariance structure (known as naive Bayes).
- c) To assume that f_1, f_2 are Gaussian, but to impose "a reasonable" structure on the covariance matrix.

Such an appropriate structure of the covariance matrix possess random matrices having the probability distribution function, which is known as the matrix normal distribution (MND) and – for larger dimensions – known as multilinear normal distribution (see [34]).

3.3 Basic Facts about MND

Further, we assume that class densities are MND and they have the probability density functions of the form: for $j = 1, 2$

$$f_j(X) = \frac{1}{c_j} \exp \left[-\frac{1}{2} \text{tr} \left[U_j^{-1} (X - M_j) V_j^{-1} (X - M_j)^T \right] \right] \quad (2)$$

where T stands for the transposition and $\det[\cdot]$ denotes the determinant of a matrix in the brackets. For the normalization constants we have:

$$c_j \stackrel{\text{def}}{=} (2\pi)^{0.5nm} \det[U_j]^{0.5n} \det[V_j]^{0.5m}, \quad (3)$$

where $n \times m$ matrices M_j denote the class means, while $n \times n$ matrix U_j and $m \times m$ matrices V_j are the rows and columns covariance matrices of the classes, respectively, assuming that $\det[U_j] > 0$, $\det[V_j] > 0$. Further, we shall write shortly,

$$X \sim N_{n,m}(M_j, U_j, V_j) \text{ for } j = 1 \text{ or } j = 2 \quad (4)$$

It is well known, that the formally equivalent description of MND is the following:

$$X \sim N_{n,m}(\text{vec}(M_j), \Sigma_j) \text{ for } j=1 \text{ or } j=2 \quad (5)$$

where Σ_j is $n \times m \times n \times m$ covariance matrix of j -th class, which is the Kronecker product (denoted as \otimes) of U_j and V_j , i.e.,

$$\Sigma_j \stackrel{\text{def}}{=} U_j \otimes V_j, \quad j=1,2 \quad (6)$$

Above, $(\text{vec}(X))$ stands for the operation of stacking columns of matrix X .

4 Bayes Classifier for Classes having Matrix Normal Distribution

In this section we assume that X is drawn from $N_{n,m}(M_j, U_j, V_j)$, for $j=1$ or $j=2$. For a while, we also assume that we know $M_j, U_j, V_j, j=2$. Our aim is to derive the Bayes classifier under the 0-1 loss function. As we shall see, the derivations closely follow those calculations that are well known for vectors with differences in algebraic manipulations.

4.1 General Case

Proposition 1. If X to be recognized is drawn from $N_{n,m}(M_j, U_j, V_j)$, for $j=1$ or $j=2$, then the Bayes classifier has the form: classify matrix (image) X to Class 1, if

$$\left[\frac{1}{2} \text{tr} \left[U_1^{-1} (X - M_1) V_1^{-1} (X - M_1)^T \right] \right] - \log(p_1 / c_1) < \left[\frac{1}{2} \text{tr} \left[U_2^{-1} (X - M_2) V_2^{-1} (X - M_2)^T \right] \right] - \log(p_2 / c_2) \quad (7)$$

and to Class 2, otherwise.

Proof. When M_j and $U_j, V_j, p_j, j=1,2$ are known, then from (1) and (2) we directly obtain (7).

The expressions in the brackets in (7) play the role of the Mahalanobis distance. 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 X and we have to know (or to estimate) all parameters: M_j and $U_j, V_j, p_j, j=1,2$. Their estimation is discussed in Section 5.

4.2 (Very) Special Case – Uncorrelated Matrix Elements

Let us assume that U_j, V_j are identity matrices (no correlations at all) and $p_1 = p_2 = 0.5$. Then, (7) reduces to the following: classify matrix (image) X to Class 1 if

$$\|X - M_1\|_F^2 < \|X - M_2\|_F^2 \quad (8)$$

and to Class 2, otherwise, where $\|A\|_F$ for matrix A stands for its Frobenius norm: $\|A\|_F = (\text{tr}[A^T A])^{1/2}$. In other words, classify a new matrix to the class, which mean is closer -- in the terms of the Frobenius norm.

Remark: it looks like a quadratic classifier, but in fact, it is linear in X (this will be clear later).

This is the so-called "naive Bayes classifier" and -- in spite of its simplicity-- it occurs to be very useful when we have very large vectors (matrices) of features.

4.3 (Less) Special Case – the Same Class Covariance Matrices

As is well known, in the case of classifying Gaussian random vectors with the same class covariance matrices, the Bayes classifier is linear. In this section, we show that it is also the case for classifying matrices.

Proposition 2. Let us assume that $U_1 = U_2 = U$, $V_1 = V_2 = V$, i.e., we have the same covariance structure in both classes. Define:

$$C = \frac{1}{2} \text{tr} \left[U^{-1} (M_1 V^{-1} M_1 - M_2 V^{-1} M_2) \right] + \log \left(\frac{p_1}{p_2} \right) \quad (9)$$

Then, the Bayes decision rule is: classify matrix (image) X to Class 1 if

$$\text{tr} \left[X V^{-1} (M_2 - M_1)^T U^{-1} \right] < C \quad (10)$$

and to Class 2, otherwise.

The proof follows from (7), by direct algebraic manipulations.

Apparently, (10) is linear in X and it can be rewritten as:

$$\text{tr}[XW] < C, \quad W \stackrel{\text{def}}{=} V^{-1}(M_2 - M_1)^T U^{-1}, \quad (11)$$

In order to interpret the result, let us rewrite (10) as follows:

$$\text{tr}\left[\left(U^{-T/2} X V^{-1/2}\right)\left(V^{-T/2}(M_2 - M_1)^T U^{-1/2}\right)\right] < C \quad (12)$$

where $U^{-T/2}$ stands for $(U^{-1/2})^T$. Hence, the decision rule is the inner product of:

- a) de-correlated pattern X and
- b) de-correlated difference of the class means $(M_2 - M_1)^T$.

One can consider (12) as the justification of the class of bi-linear (in weighting matrices) classifier proposed in [35].

Remark: We do not impose the Kronecker product structure on M_1 and M_2 matrices. This seems to be an excessive requirement, leading to the assumption that we have a matrix of matrices of (almost) the same elements – images. This is outside the scope of this paper.

5 Learning the Classifier – Plug-in Method

When M_j, U_j, V_j are unknown, we have two learning sequences: $X_i^{(j)}$, $i = 1, 2, \dots, N_j$, $j = 1, 2$ for estimating them. For $N = N_1 + N_2$ the estimation of the mean matrices and a priori probabilities is obvious:

$$\hat{M}_j = N_j^{-1} \sum_{i=1}^{N_j} X_i^{(j)}, \quad \hat{p} = N_j / N, \quad j = 1, 2 \quad (13)$$

Estimating covariance matrices: U_j, V_j is not so easy task. The fact is their maximum likelihood estimates (MLE) are not unique, i.e., they can be estimated up to a constant multiplier, does not lead to troubles since U_j, V_j appear as multiplicative pairs.

Maximum likelihood estimators (MLE) \hat{U}_j, \hat{V}_j can be calculated if (see [36])

$$N_j \geq \max\left\{\frac{n}{m}, \frac{m}{n}\right\} + 1, j = 1, 2. \quad (14)$$

Thus, it is not necessary to have: $N_j \geq mn$. This is the main advantage of imposing the Kronecker product structure on the class covariance matrices. For $m = n$ we need at least 2 images to calculate MLE's of rows and columns covariance matrices, which does not mean that for two samples we obtain good estimates.

5.2 MLE Estimators of U_j and V_j

According to [37], MLE estimators \hat{U}_j, \hat{V}_j have to be calculated by solving the simultaneous set of equations:

$$\hat{U}_j = \frac{1}{N_j m} \sum_{i=1}^{N_j} (X_i - \hat{M}_j) \hat{V}_j^{-1} (X_i - \hat{M}_j)^T \quad (15)$$

$$\hat{V}_j = \frac{1}{N_j n} \sum_{i=1}^{N_j} (X_i - \hat{M}_j) \hat{U}_j^{-1} (X_i - \hat{M}_j) \quad (16)$$

for $j = 1, 2$, They form the pair of matrix equations, which are usually solved as follows.

The flip-flop method:

1. Instead of \hat{U}_j, \hat{V}_j , use the unit matrices at r.h.s. of (15) and (16),
2. Calculate the left-hand sides of (15) and (16),
3. Re-substitute the results of the previous step into right-hand side of (15) and (16),
4. Repeat Step 2 and Step 3 until convergence.

Lemma 1. One flip-flop iteration of the above method is sufficient in order to obtain the consistent (convergent in the probability) and asymptotically efficient estimators of U_j, V_j as the number of observations from the two classes grows to infinity.

For the proof see [38]. This result forms the base for proving that the empirical classifier is asymptotically optimal.

5.2 Empirical Classifiers for Matrix Normal Class Distributions

In order to convert the Bayes classifier into empirical one, substitute $M_j \leftarrow \hat{M}_j$, $U_j \leftarrow \hat{U}_j$, e.t.c., into (7) to get the following classifier:

classify matrix (image) X to Class 1 if

$$\left[\frac{1}{2} \text{tr} \left[\hat{U}_1^{-1} (X - \hat{M}_1) \hat{V}_1^{-1} (X - \hat{M}_1)^T \right] \right] - \log(\hat{p}_1 / \hat{c}_1) < \left[\frac{1}{2} \text{tr} \left[\hat{U}_2^{-1} (X - \hat{M}_2) \hat{V}_2^{-1} (X - \hat{M}_2)^T \right] \right] - \log(\hat{p}_2 / \hat{c}_2) \quad (17)$$

and to Class 2, otherwise, where

$$\hat{c}_j \stackrel{\text{def}}{=} (2\pi)^{0.5nm} \det[\hat{U}_j]^{0.5n} \det[\hat{V}_j]^{0.5m}. \quad (18)$$

Proposition 3. If the row and column covariance matrices are estimated by the flip-flop method and a priori probabilities and the class means are estimated as in (13), then, for each fixed X , the left and the right hand sides of the empirical classifier, described as rule (17), is convergent in the probability to the left and the right hand sides of the optimal classifier (7), respectively, as the number of observations from the both classes approaches to infinity.

Proof. The consistency of the estimators in (13) is well known and it follows from the law of large numbers. The consistency of the row and column covariance matrices follows from Lemma 1. The convergence of the left- and the right-hand sides of (17) to those of (7) immediately follows from the well-known Slutsky's theorems since these expressions are either rational or continuous functions of the consistent estimators (13) or those described in Lemma 1.

5.3 Empirical Classifiers – Special Cases

1) **The empirical version of the "naive Bayes" classifier** is particularly simple:

Classify matrix (image) X to Class 1 if

$$\|X - \hat{M}_1\|_F^2 < \|X - \hat{M}_2\|_F^2 \quad (19)$$

and to Class 2, otherwise.

2) **The case of the same class covariance matrices.** It is expedient to consider two possible approaches:

A) Plug-in approach: classify X to Class 1 if

$$\hat{W} \stackrel{\text{def}}{=} \hat{V}^{-1}(\hat{M}_2 - \hat{M}_1)^T \hat{U}^{-1} \quad (20)$$

and \hat{C} is defined analogously.

B) A direct learning of the weight matrix W . Our starting point is again the Bayes decision rule: $\text{tr}[XW] < C$. Notice that this rule is not uniquely defined (we can multiply W and C by an arbitrary constant). Thus, later we take $C=1$. Let (X_i, y_i) , $i=1,2,\dots,N=N_1+N_2$ (both classes) be the learning sequence with class labels $y_i = -1$ for Class 1 and $y_i = +1$ for Class 2. Then, the recurrent update that minimizes one-step ahead error $(y_i - (\text{tr}[X_i W] - 1))^2$ with respect to W is of the form:

$$W_{i+1} = W_i - \gamma(y_i - (\text{tr}[X_i W] - 1))X_i^T \quad (21)$$

where $\gamma > 0$ is a small learning constant. After stopping (21) with \hat{W} , the decision is made according to $\text{sgn}[\text{tr}[X\hat{W}] - 1]$.

5.4 Classifying whole Image Sequences

Let $q > 1$ be the length of an image sequence denoted by X , which is $n \times m \times q$ tensor. Assume that class densities of $x = \text{vec}(X)$ have the tensor normal distribution with the same covariance $U \otimes X \otimes Z$, where Z is $q \times q$ inter-frame covariance matrix. The classes have different means $m_j \stackrel{\text{def}}{=} \text{vec}(M_j)$, $j=1,2$. Then, it can be shown that the Bayes classifier is again linear in x .

MLE for estimating U, V, Z consists of three sets of equations (see [36]) and it can be solved by a flip-flop like algorithm, but – in spite of the Kronecker product covariance structure – a large amount of data is required.

Hence, a simple – heuristic – classifiers should also be taken into account to classify image sequences, as it will be demonstrated in the next section.

6 A Case Study – Quality Control of an Additive Manufacturing Process using a Camera

We shall use a classifier as change detector in a sequence of short (3 images) videos, but instead of modeling them as 3D (tensor) structures we "glue" batches consisting of 3 subsequent images into one, larger, image and then, they will be classified as changes in one image sequence.

Caution: Applying a classifier as change detector one has to take into account an inherent difficulty of such an approach. Namely, the phenomenon that is known as the class imbalance (see, e.g., [39]), which appears here because, usually we have a much larger number of examples (images) of in-control examples than those out-of-control. Special actions (e.g., choice of the classifier or undersampling of the in-control images) have to be undertaken.

6.1 A Practical Problem to Solve

An additive manufacturing is a class of modern production processes. A large number of technics and technologies are used in this area, see [40], where the optimization of computer-aided screen printing design is discussed and [41] for the life cycle optimization of such processes. We refer the reader to the survey papers [42], [43], [44] on additive manufacturing processes.

As a vehicle for presenting possible applications of image classifiers in decision making, we selected the process known as the selective laser melting, which produces items (roughly) as follows:

- A metallic powder is poured in a precisely controlled way
- Simultaneously the powder is melted by a laser beam
- After hardening – it forms a part of a 3D body to be constructed
- The laser head, together with the powder supply nozzle, moves to the next place (in fact, phases of moving and pouring and melting the powder run simultaneously and continuously).

For a more detailed description of this kind of production processes, the reader is referred to [45]. This technology is expected to be developing in the future and it is therefore expedient to attempt to improve it to the perfection.

One of the problems is that the laser head stays longer near end-points (turning off) of a produced item (e.g., a wall). This results in a too wide ends for the produced wall (see Figure 11).

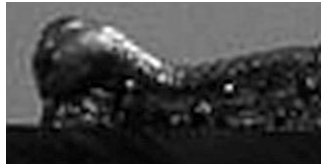


Figure 11

The left endpoint of the built wall: visible part of the wall is too wide and too high

Proposed remedy: recognize from images that the laser head is near the end point and reduce the laser power near the ends, then recognize again middle points and increase the laser power.

Many attempts were recently undertaken to cope with this problem (see [25], [46], [47]). The main difference between the approaches proposed in the papers cited above and the present one is in that here we consider the recognition that the laser head is in the near end position from short video sequences (triples of images), treated as a whole entity. Additionally, we take into account that the frequency of being in these states is much rarer than being in the "normal" state, i.e., in a middle of the wall. To illustrate the role of the class imbalance in this case, we mention that in our laboratory experiments the wall had 600 mm, while the near end zone had 2-4 mm.

6.2 Learning Sequence of Images



Figure 12

Examples of original images of the produced wall – view from above. The left end laser head position – too thick wall end is visible and the middle one – has a proper wide of the wall.

We had about 900 images of the produced wall that were taken from above (almost) along the laser beam. Examples of original images are shown in Figure 12. These images were cropped to keep only parts of the wall and then they were grouped into new images with three elements in the way that each triple overlapped with the previous one, having two common original images. In this way, the sequence of the total length 898 was obtained. In this sequence, we have distinguished 104 triples that were labeled as "BAD" since they contain the wall endpoints (usually too wide) and 794 triples marked as "OK" since they a middle part of the wall, which is of the proper width. Examples of these triples are shown in Figure 13. The next step – available data were divided into two halves: the learning and testing sequence, keeping about 10% examples from "BAD" class.

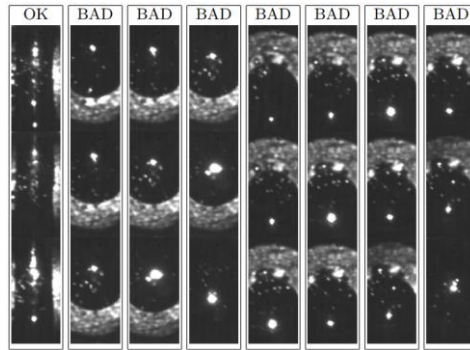


Figure 13

Examples of triples of "glued" images to be classified as "BAD" or "OK". By "OK" triples we mean those that have a proper width – they are located in the middle of the wall. By "BAD" triples we understand those that are near the endpoints of the wall – they usually are too thick. These triples are fed as inputs for classifying algorithms in order to make a decision whether to keep the laser power at the nominal value or to decrease it near the endpoints.

6.3 Naive Learning - Neglecting Class Imbalance

In this section, we provide examples of positive and negative results of learning classifiers. The goals of presenting also negative results are the following:

- To warn the reader that the task of change detection in sequences of images is nontrivial.
- To document that classifiers that are believed to be the "golden standard", such as support vector machines (SVM) may fail when the class imbalance appears in the learning sequence.

SVM classifier provided 88% correct classifications, when (after learning) was applied to the testing sequence. Unexpectedly, all triples classified by an expert as "BAD" were classified to "OK" class by the SVM classifier. Notice that seemingly good result of 88% correct classifications was obtained, because the testing sequence contained only 12% of "BAD" items and all of them were misclassified (see Table 1). The classifier had zero sensitivity (recall) to "BAD" class, also F-score was zero.

Table 1
Confusion matrix of the SVM classifier

Actual class	Predicted class		sum
	BAD	OK	
BAD	0	54	54
OK	0	395	395
sum	0	449	

"Naive Bayes" classifier provided 68% correct classifications when applied to the testing sequence. This time, almost all "BAD" items were correctly classified, but at the expense of 1/3 "OK" examples classified erroneously. The probability of detection (sensitivity, recall) of "BAD" class is still rather low, namely, 0.26. The following classifiers were also tested: logistic regression and random forest (with 50 trees). The results were somewhat better than that for SVM and Naive Bayes, but still not satisfactory.

6.4 5-NN Classifier Robust against "Naive" Learning

Satisfactory results (without editing the learning and/or testing sequence for class imbalance) were obtained for 5 Nearest Neighbors (5-NN) classifier. Namely, it provided 98% correct classifications, simultaneously, 80% of "BAD" testing examples were correctly classified. Furthermore, there were zero false alarms, as one can check from the confusion matrix in Table 2. Thus, 5-NN classifier occurred to be robust against naive learning in the class imbalance case.

The only – well-known – drawback of this classifier is the necessity of storing the whole learning sequence, but storage resources of clouds reduce it considerably

Table 2
The confusion matrix of 5-NN classifier

Actual class	Predicted class			sum
		BAD	OK	
□ BAD		43	11	54
□ OK		0	395	395
sum		43	406	

6.5 MND Classifier and Comparisons

Satisfactory results were also obtained for the MND classifier. They are summarized in Table 3. The MND classifier provided 96.2% correct classifications, simultaneously, 78% of "BAD" testing examples were correctly classified. Furthermore, there were only 1% of false alarms. Thus, also MND classifier occurred to be robust against naive learning in the class imbalance case.

Table 3
The confusion matrix of MND classifier

Actual class	Predicted class			sum
		BAD	OK	
□ BAD		42	12	54
□ OK		5	390	395
sum		47	402	

Table 4 contains the summary of testing classifiers. As one can observe, the popular SVM and naïve Bayes classifiers provide unexpectedly bad results. The reason is in that they do not take into account the class imbalance. In opposite, 5-NN and MND classifier give quite good results since they are – to some extent – insensitive to the class imbalance. Their confusion matrices (see Table 2 and 3) are almost the same.

Table 4

Comparison of classifiers: SVM, NB – naïve Bayes, 5 NN and MND classifier, according to % of correct and % of misclassifying BAD as OK

Classifier	SVM	NB	5 NN	MND cl.
% correct	88	68	98	96.2
% BAD as OK	100	0	20	22

6.6 Decision Making

After the learning phase, the 5-NN classifier can be used for making control decisions, as shown below. Let X denote current triple of images.

- 1) Classify X to class "BAD" or "OK".
- 2) If $X \in$ "BAD", reduce the laser power (by a pre-specified amount) so as to attain the temperature of the melted lake about 2140 C (this is done by the PI controller).
- 3) If $X \in$ "OK", keep the nominal laser power (or return to it, if previously $X \in$ "BAD"). The nominal laser power corresponded to the lake temperature 2445 C.
- 4) Acquire new image and form new X by adding it to X and throwing out the oldest one from it. Go to 1).

6.7 Laboratory Experiment

In order to check to what extent one can reduce unpleasant "end effects", the wall was first built with a constant laser power. In the upper panel of Figure 14, one can notice to wide ends of the wall. When the laser power was reduced each time when the laser head was near one of the endpoints (see Sec. 6.5) the resulting wall has more proper endpoints (see the lower panel of this figure). The wall has the length of about 60 mm. The speed of the laser head was about 10 mm/sec., while a stainless steel powder was supplied with the feed rate at 0.06 g/sec.

In fact, the wall at the lower panel of Figure 16 was obtained under more subtle, gradual change of the laser power, but this aspect is outside the scope of this paper.

Conclusions

Our first step was an attempt to provide some systematics for images and image sequences, from the viewpoint of their classification. At this stage, the class of images and image sequences having matrix (tensor) normal distribution was selected as sufficiently general, but still, a manageable class distribution. The MND class distributions have the covariance matrices that take into account only the inter-row and the inter-column covariances. Therefore, they are easier to estimate than in a general case. However, a specialized form of the covariance matrices leads to more specific classifiers than in the general case. Their structure was derived and their empirical forms were proposed as the classifiers for further investigations.

Finally, these classifiers were tested on the problem of detecting, from short image sequences, whether a laser head is near the endpoints of a cladding wall. In other words, the proposed classifier is used in the problem of change detection from image sequences. Its performance is quite satisfactory. Its behavior was also compared with a general purpose and widespread classifiers that do not take into account a special covariance structure or the class imbalance. As it was documented by the laboratory images, only 5-NN classifier can be comparable with the proposed approach since it is – to some extent – robust against a naïve learning.

Clearly, one can consider other methods for image feature representation and classification, e.g., in [47] the spectral and wavelet analysis as feature extraction techniques were employed, in [48] the feature extraction is based on a combination of a self-organized map used for image vector quantization and those generated by a neural network, a kernel sparse representation, which produces discriminative sparse codes to represent features in a high-dimensional feature space, is proposed in [49], while in [50] non-conventional approaches to feature extraction were proposed. A feature extraction is a common focal point of all these approaches. It is laborious, human-invented and dedicated to a particular application. In opposite, we stress that the proposed approach does not need a feature extraction step. Instead, “raw” images are supplied as inputs for a classifier, providing an acceptable level of proper classifications. This approach is less laborious, but its applicability is limited to cases when there is no need to consider very subtle differences between images.

The proposed approach may be useful, at least, at one more area of applications, namely, in using classifiers to detect states of industrial gas burners from image sequences (see [39]). It seems that further efforts are necessary in order to sketch a wider class of applications for which the proposed approach outperforms a general purpose classifiers when they are applied to image sequences.

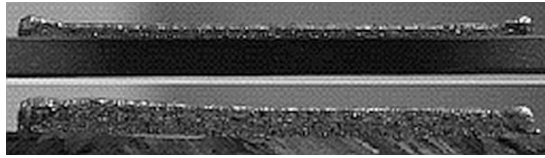


Figure 14

Upper panel – the wall produced with constant laser power along the wall length. Lower panel – the wall produced with controlled laser power trajectory along the pass.

Acknowledgement

This research has been supported by the National Science Center under grant: 2012/07/B/ST7/01216.

Special thanks are addressed to Professor J. Reiner and to MSc. P. Jurewicz from the Faculty of Mechanical Engineering, Wrocław University of Technology for common research on laser power control for additive manufacturing.

The author express his thanks to the anonymous reviewers for many suggestions, leading to the improvements of the presentation.

References

- [1] Fukunaga K.: “Introduction to Statistical Pattern Recognition”, Academic Press, 2013
- [2] Devroye L., Györfi L., Lugosi G.: “A Probabilistic Theory of Pattern Recognition”, Springer Science & Business Media, 2013
- [3] Han J., Chen C., Shao L., Hu X., Han J., Liu T.: “Learning Computational Models of Video Memorability from fMRI Brain Imaging”, IEEE Trans. on Cybernetics, 45(8), 2015, pp. 1692-1703
- [4] Ninio J.: “Testing sequence effects in visual memory: clues for a structural model”, Acta Psychologica, 116(3), 2004, pp. 263-283
- [5] Schyns P. G., Gosselin F., Smith M. L.: “Information processing algorithms in the brain”, Trends in Cognitive Sciences, 13(1), 2009, pp. 20-26
- [6] Stadler W., Schubotz R. I., von Cramon D. Y., Springer A., Graf M., Prinz W.: “Predicting and memorizing observed action: differential premotor cortex involvement”, Human Brain Mapping, 32(5), 2011, pp. 677-687
- [7] Fulton, J. T., “Biological vision”, Trafford, 2004
- [8] Lee, N., Cichocki, A., “Fundamental tensor operations for large-scale data analysis using tensor network formats”, Multidimensional Systems and Signal Processing, 29(2), 2018, pp. 921-960

-
- [9] Sean A., Jason L.: “Building and using a database of one trillion natural-image patches”, *IEEE Trans. Computer Graphics and Applications* 31(1), 2011, pp. 9-19
- [10] Tsybmal A., Meissner E., Kelm M., Kramer M.: “Towards cloud-based image-integrated similarity search in big data”, *Biomedical and Health Informatics (BHI)*, 2014 IEEE-EMBS International Conference on, IEEE, 2014, pp. 593-596
- [11] Assent I.: “Clustering high dimensional data”, *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 2(4), 2012, pp. 340-350
- [12] Lin F., Chung L., Wang C, Ku, W., and Chou T.: “Storage and Processing of Massive Remote Sensing Images Using a Novel Cloud Computing Platform”, *GIScience & Remote Sensing*, 50(3), 2013, pp. 322-336
- [13] Yaqoob I. et al.: “Big data: From beginning to future”, *Int. J. Information Management*, 36(6), 2016, pp. 1231-1247
- [14] Megahed F. M., Woodall W. H., Camelio J. A.: “A review and perspective on control charting with image data”, *J. Quality Technology*, 43(2), 2011, pp. 83-98
- [15] Duchesne C., Liu J. J., MacGregor J. F.: “Multivariate image analysis in the process industries: A review”, *Chemometrics and Intelligent Laboratory Systems*, 117, 2012, pp. 116-128
- [16] Bharati M. H., MacGregor J. F.: “Multivariate image analysis for real-time process monitoring and control”, *Industrial & Engineering Chemistry Research*, 37(12), 1998, pp. 4715-4724
- [17] Zou C., Wang Z., Tsung F.: “A spatial rank-based multivariate EWMA control chart”, *Naval Research Logistics*, 59(2), 2012, pp. 91-110
- [18] Rafajłowicz E., Rafajłowicz W.: “Image-driven decision making with application to control gas burners”, *IFIP International Conference on Computer Information Systems and Industrial Management*, Springer, 2017, pp. 436-446
- [19] Lenhardt C., Conway M., Scott E., Blanton B., Krishnamurthy A., Hadzikadic M., Vouk M., Wilson A.: “Cross-institutional Research Cyber Infrastructure for Data Intensive Science”, *High Performance Extreme Computing Conference (HPEC)*, IEEE, 2016, pp. 1-6
- [20] Prause A., Steland A.: “Detecting changes in spatial-temporal image data based on quadratic forms”, *Stochastic Models, Statistics and Their Applications*, Springer, 2015, pp. 139-147

- [21] Rafajłowicz E.: “Detection of essential changes in spatio-temporal processes with applications to camera based quality control”, *Stochastic Models, Statistics and Their Applications*, Springer, 2015, pp. 433-440
- [22] Kurzynski M.: “On the identity of optimal strategies for multistage classifiers”, *Pattern Recognition Letters*, 10(1), 1989, pp. 39-46
- [23] Kurzynski M.: “The optimal strategy of a tree classifier”, *Pattern Recognition*, 16(1), 1983, pp. 81-87
- [24] Schuermann J., Doster W.: “A decision theoretic approach to hierarchical classifier design”, *Pattern Recognition*, 17(3), 1984, pp. 359-369
- [25] Rafajłowicz E.: “Image-driven, model-free control of repetitive processes based on machine learning”, *Multidimensional (nD) Systems (nDS)*, 2017 10th International Workshop on, IEEE, 2017, pp. 1-6
- [26] Krzyśko M., Skorzybut M.: “Discriminant Analysis of Multivariate Repeated Measures Data with a Kronecker Product Structured Covariance Matrices”, *Statistical Papers*, 50(4), 2009, pp. 817-835
- [27] Krzyśko M., Skorzybut M., Wolynski W.: “Classifiers for Doubly Multivariate Data”, *Discussiones Mathematicae: Probability & Statistics*, 31, 2011
- [28] Rafajłowicz E.: “Classifiers sensitive to external context-theory and applications to video sequences”, *Expert Systems*, 29(1), 2012, pp. 84-104
- [29] Skubalska-Rafajłowicz E., Krzyzak A., Rafajłowicz E.: “Dimensionality reduction using external context in pattern recognition problems with ordered labels”, *Artificial Intelligence and Soft Computing*, Springer, 2012, pp. 430-438
- [30] Rafajłowicz E., Wietrzych J.: “Recognition of nite structures with application to moving objects identification”, *International Conference on Artificial Intelligence and Soft Computing*, Springer, 2010, pp. 453-461
- [31] Rafajłowicz E., Krzyzak A.: “Pattern recognition with ordered labels”, *Nonlinear Analysis: Theory, Methods & Applications*, 71(12), 2009, pp. 1437-1441
- [32] Rafajłowicz E.: “RBF nets in faults localization”, *Artificial Intelligence and Soft Computing-ICAISC*, 2006, pp. 113-122
- [33] Slivka J., Kovacevic A., Konjovic Z.: “Combining co-training with ensemble learning for application on single-view natural language datasets”. *Acta Polytechnica Hungarica*, 10(2), 2013, pp. 133-152
- [34] Ohlson M. Ahmad M. R., Von Rosen D.: “The multilinear normal distribution: Introduction and some basic properties”, *J. Multivariate Analysis*, 113, 2013, pp. 37-47

- [35] Kobayashi T.: “Low-rank Bilinear Classification: Efficient Convex Optimization and Extensions”, *Int. J. of Computer Vision*, 110(3), 2014, pp. 308-327
- [36] Manceur A. M., Dutilleul P.: “Maximum Likelihood Estimation for the Tensor Normal Distribution: Algorithm, Minimum Sample Size, and Empirical Bias and Dispersion”, *J. Computational and Applied Mathematics*, 239, 2013, pp. 37-49
- [37] Dutilleul P.: “The MLE algorithm for the matrix normal distribution”, *Journal of Statistical Computation and Simulation*, 64(2), 2012, pp. 105-123
- [38] Werner K., Jansson M., Stoica P.: “On estimation of covariance matrices with Kronecker product structure”, *IEEE Transactions on Signal Processing*, 56(2), 2008, pp. 478-491
- [39] Liu X., Wu J., Zhou Z.: “Exploratory Under-sampling for Class-imbalance Learning”, *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 39(2), 2009, pp. 539-550
- [40] Horvath E., et al.: “Optimisation of Computer-aided Screen Printing Design”. *Acta Polytechnica Hungarica*, 11(8), 2014, pp. 29-44
- [41] Horvath, E., Harsanyi, G., Henap, G., Torok: “A mechanical modelling and life cycle optimisation of screen printing”, *Journal of Theoretical and Applied Mechanics*, 50(4), 2012, pp. 1025-1036
- [42] Frazier, W. E.: “Metal additive manufacturing: a review”, *Journal of Materials Engineering and Performance*, 2014, 23(6), pp. 1917-1928
- [43] Murr L. E., et al.: “Fabrication of metal and alloy components by additive manufacturing: examples of 3D materials science”, *Journal of Materials Research and Technology*, 1(1), 2012, pp. 42-54
- [44] Chianrabutra V. M., Mellor S., Yang, S.: “Multiple material additive manufacturing—Part 1: a review”, *Virtual and Physical Prototyping*, 8(1), 2013, pp. 19-50
- [45] Baraniecki T., Chlebus E., Dziatkiewicz M., Kedzia J., Reiner J., Wiercioch M.: “System for laser microsurfacing of metal powders”, *Welding International*, 30(2), 2016, pp. 98-102
- [46] Rafajłowicz E., Rafajłowicz W.: “Camera in the control loop -- methods and selected industrial applications”, *Polish Control Conference*, Springer, 2017, pp. 261-270
- [47] Jurewicz P., Rafajłowicz W., Reiner J., Rafajłowicz E.: “Simulations for Tuning a Laser Power Control System of the Cladding Process”, *IFIP International Conference on Computer Information Systems and Industrial Management*, Springer, 2016, pp. 218-229

- [48] Somogyi B. B., Paláncz B.: “Classification of Cerebral Blood Flow Oscillation”, *Acta Polytechnica Hungarica*, 3(1), 2006
- [49] Chen, X., Nguyen, B. P., Chui, C. K., Ong, S. H. “An Automated Framework for Multi-label Brain Tumor Segmentation based on Kernel Sparse Representation”, *Acta Polytechnica Hungarica*, 14(1), 2017
- [50] Ban, J., F., Oravec, M., Pavlovicova, J.: “Non-conventional approaches to feature extraction for face recognition”, *Acta Polytechnica Hungarica*, 8(4), 2011, pp. 75-90