J Real-Time Image Proc
DOI 10.1007/s11554-006-0018-5

SPECIAL ISSUE

A study on design of object sorting algorithms in the industrial application using hyperspectral imaging

Pavel Paclík · Raimund Leitner · Robert P. W. Duin

Received: 27 June 2006 / Accepted: 16 October 2006

Abstract Many industrial object-sorting applications leverage benefits of hyperspectral imaging technology. Design of object sorting algorithms is a challenging pattern recognition problem due to its multi-level nature. Objects represented by sets of pixels/spectra in hyperspectral images are to be allocated into pre-specified sorting categories. Sorting categories are often defined in terms of lower-level concepts such as material or defect types. This paper illustrates the design of two-stage sorting algorithms, learning to discriminate individual pixels/spectra and fusing the per-pixel decisions into a single per-object outcome. The paper provides a case-study on algorithm design in a real-world industrial sorting problem. Four groups of algorithms are studied varying the level of prior knowledge about the sorting problem. Apart of the sorting accuracy, the algorithm execution speed is estimated assuming an ideal implementation. Relating these two performance criteria allows us to discuss the accuracy/speed trade-off of different algorithms.

Keywords Hyperspectral imaging · Object sorting · Algorithm design · Pattern recognition

P. Paclík (✉) · R. P. W. Duin
ICT Group, TU Delft, 2628 CD Delft, The Netherlands
e-mail: P.Paclik@ewi.tudelft.nl

R. P. W. Duin
e-mail: R.Duin@ieee.org

R. Leitner
CTR AG, 9524 Villach/St. Magdalen, Austria
e-mail: Raimund.Leitner@ctr.at

1 Introduction

Spectral imaging has gained importance in industrial sorting applications. Rich spectral information provided by hyperspectral sensors can capture detailed material composition. The advantages of spectral measurements are combined with the locality and resolution of image data. Using various image processing techniques, objects or defects may be localized. The object sorting applications of spectral imaging include recycling applications (polymers [9], paper [17]) and quality control (apples [1, 8], potatoes [10], tomatoes [16] or poultry [4, 7]).

Existing studies on object sorting usually discuss the design of the entire sorting system, starting from the imaging spectrometer, data calibration, and pre-processing and finishing with the sorting algorithm and the system evaluation. Due to complexity of the system design, pattern recognition algorithms and their performance evaluation usually receive only limited attention. In this paper, we attempt to remedy this situation providing the sorting community with a more detailed analysis of several advanced sorting algorithms on a real-world object sorting problem.

The contribution of this work is two-fold. Firstly, we discuss not only the sorting accuracy of the algorithms but also estimate their computational complexity in execution. In addition to such estimates given in other studies [9, 17], we visualize the relation between both performance metrics. This allows us to discuss effects of different design choices and provides ground for multi-criterial selection of algorithms in sorting problems.

The second contribution of this study lays in the systematic approach to design and evaluation of object sorting algorithms. Existing studies often design classifiers at the level of pixels/spectra assuming that the high-performance classification of pixels will result in a high-performance object sorting [9, 10, 16, 17]. This approach, however, neglects the influence of the necessary pixel-to-object fusion step. We have shown in [15], that a model, selected by optimizing the per-pixel error, may yield a high per-object error due to the used fusion rule. We therefore argue, that to meet the performance estimates in production, the sorting algorithms need to be designed optimizing the per-object performance metrics.

The paper is organized as follows. In Sect. 2, we describe the object sorting system from the pattern recognition perspective. In Sect. 3, the studied sorting algorithms are introduced. Section 4 discusses the evaluation strategy for object sorting problems. The description of the dataset used, experimental setup and results are discussed in Sect. 5. Finally, the conclusions are given.

2 Object sorting system

The object sorting system based on spectral imaging receives an image stream on its input. Objects, present in the stream, are detected and each of them is classified into one of the pre-specified sorting categories. In this study, we assume a perfect object detector and focus entirely on the design of the object classifier.

Surveying the real-world object sorting problems we can observe that the sorting categories are usually defined using lower-level concepts [9, 10]. For example, in potato sorting the high-level category *defected potato* is described by fractions (percentiles) of damage, rot, or greening present in the object. In this study, we follow the approach where the classifiers of the high-level classes or lower-level concepts are trained based on individual pixels/spectra within the objects. The per-object decision is then performed by the fusion of lower-level decisions [9, 15]. From a pattern recognition viewpoint, such sorting system exhibits multiple levels, namely pixels/spectra, objects, lower-level concepts (modes), and high-level categories (classes).

Formally, an object z_i, $i = 1,\ldots,N$ is assigned into one of the C pre-defined high-level categories (classes) $\Omega = \{\omega{:}\omega = 1,2,\ldots,C\}$. Each category $\omega \in \Omega$ is defined as a collection of concepts $\{m_j^{\omega}{:}\, j = 1,\ldots,M_{\omega}\}$, where M_{ω} denotes the number of concepts in a category ω; the total number of concepts $M = \sum_{\omega} M_{\omega}$. Each object z_i belongs to a single concept m_i^{ω} and thereby to a single class ω and is composed of K_i pixels/spectra represented as D-dimensional feature vectors $\mathbf{x}^i \in R^D$.

We assume that in a sorting problem the concepts of high-level categories are known apriori. The sorting algorithms may thereby leverage this prior knowledge i.e. account for the problem multi-modality. Otherwise, cluster analysis may be employed to define lower-level concepts in multi-modal problems [12]. Note that we use the term mode to describe lower-level concepts of sorting categories. This differs from the statistical convention where mode usually represents a unimodal peak of the probability density function. Modes of a sorting system such as material types may thereby exhibit internal statistical multi-modality.

3 Sorting algorithms

The object sorting algorithms considered in this study operate in two steps, namely at the pixel and object level. During training, a pixel-level classifier of the high-level classes is derived. For a new object to be sorted, this trained classifier is applied to all the object pixels and its crisp decisions are collected. The object is then assigned to the high-level class based on the majority vote. Because the pixel-to-object fusion rule is fixed, only the pixel-level classifier needs to be trained.

We study four groups of the pixel-level algorithms:

- State-of-the-art algorithms published in other studies
- Dissimilarity-based algorithms
- Decomposition-based descriptors (mixtures)
- Decomposition-based discriminants.

3.1 State-of-the-art algorithms

- FLD: Fisher linear discriminant directly applied to spectral data
- GLDB-FLD: spectra-specific feature extractor (a top-down Generalizer Local Discriminant Bases algorithm [5]) followed by the FLD
- PCA-QDC: Principal Component Analysis (PCA) followed by quadratic discriminant assuming normal densities (QDC) [9]
- PCA-3NN: PCA followed by the 3-nearest neighbor rule [17]
- DBC-NN: nearest mean classifier using Spectral Angle Mapper (SAM) distance [19] and mean class spectra as prototypes [9]

The PCA accounts for 99% of data variance [2].

3.2 Dissimilarity-based approaches

Four dissimilarity-based sorting algorithms are discussed varying the classification strategy and dissimilarity measure. Apart of the traditionally-used nearest neighbor classifier, also the approach based on a dissimilarity space is used [3]. The dissimilarities to prototypes are considered as new features. All the training spectra are projected into this new space and the FLD classifier is trained. Note that this approach leverages correlations between dissimilarities to different prototypes.

Two spectra-specific dissimilarity measures are employed, namely the Spectral Angle Mapper (SAM) and a derivative-based distance computing L1-norm between smoothed Gaussian derivatives of spectra [11]. The derivative-based dissimilarity adopts the Savitzky–Golay algorithm with a window of 11 wavelengths [14]. All algorithms use mean spectra of training objects as prototypes and learn to directly distinguish the high-level classes.

- SAM-NN: object prototypes, SAM, 1-NN
- SAM-FLD: object prototypes, SAM, FLD
- DerDist-NN: object prototypes, derivative dissimilarity, 1-NN
- DerDist-FLD: object prototypes, derivative dissimilarity, FLD.

3.3 Decomposition-based descriptors

The classifiers in this group describe the material types by statistical models. Typically, the allocation of training examples to modes is estimated using the EM algorithm. However, in case of the multi-level sorting problem defined in Sect. 2, the mixture model may take advantage of the known material membership of the training examples. Thus, we use a single component per material type and estimate the mixture parameters directly using the material type labels.

- MOGC: Gaussian mixture model built on original spectra
- PCA-MOGC: PCA dimensionality reduction, followed by MOGC
- mode-SIMCA: for each mode a separate PCA projection and model is built. The classification is performed based on the combination of in-model Mahalanobis and out-of-model Euclidean distance [18]
- LDA-MOGC: Linear Discriminant Analysis (LDA) on data modes; MOGC in the resulting low-D subspace.

These algorithms are chosen to illustrate a gradual increase in supervision of data representation. Starting a mixture model built in the original space, unsupervised dimensionality reduction (PCA-MOGC), per-mode reduction by mode-SIMCA, to the supervised representation built by LDA-MOGC.

3.4 Decomposition-based discriminants

The algorithms in Sect. 3.3 describe the data by modeling the class-conditional probability distributions. However, building a full class descriptor requires more statistical evidence than designing a class discriminant because the entire data domain is modeled, not only the separation boundary. Because the eventual goal of object sorting is class discrimination, we proposed in [12] to tackle the multi-modal sorting problem by a combination of simple discriminants rather than by data descriptors. A complex problem is first decomposed into two-class sub-problems involving only pairs of concepts (modes) from different classes. Note that this form of decomposition leverages the prior knowledge on the sorting problem. For each of such $(M^2 - \sum_{\omega} (M_{\omega})^2)/2$ sub-problems a linear discriminant is derived. All training examples are then processed by the sub-problem discriminants, their outputs concatenated, re-scaled and labeled by the high-level class labels. On this second-stage dataset the final linear discriminant (combiner) is trained. We have illustrated in [12] that a sigmoidal mapping of the first-stage discriminants' outputs results in a non-linear classifier.[1] The advantage of this setup is the fast execution because a non-linear classification is achieved using inexpensive linear discriminants. Note that this scheme resembles a two-layer neural network. The proposed classifier is however trained deterministically in two steps, not by an error-correction mechanism typical to neural networks.

We employ two variants of this Decomposition-based Multi-Modal Discriminant (DMMD) algorithm:

- DMMD: both the sub-problem discriminants and the combiner are FLDs
- GLDB-DMMD: for each sub-problem a specific feature representation is first derived by the GLDB extractor [5]. The FLD is used both as the first-stage classifier and as the combiner.

For the sake of comparison, we also include a naïve algorithm which builds discriminants between all pairs of modes and combines their outputs by majority voting. This algorithm neglects the prior knowledge on the sorting problem i.e. the fact that only the discrimination between materials originating from different high-

[1] We adopt the sigmoidal scaling of classifier outputs used in PRTools toolbox [2]. The sigmoid bias is fixed to zero and the slope parameter is estimated on the training set using the maximum-likelihood estimator.

level classes is of interest. This algorithm is similar to the Bayesian Pair-wise Classifier (BPC) proposed by Kumar et al. [5] for general multi-class spectral classification:

- BPC: for each pair of modes a PCA projection accounting for 95% of variance is trained and a regularized QDC is built. The per-pixel decision on the high-level class is made by voting over the $M(M-1)/2$ crisp classifiers' outputs.

4 Evaluation of object sorting algorithms

In this study, we consider two performance criteria of an object sorting algorithm related to the sorting accuracy and the speed of algorithm execution, respectively. Due to nature of hyperspectral imagery, the number of pixels/spectra handled during the sorting algorithm design may be very large (hundreds of thousands). However, the number of respective objects may be still limited to tens or hundreds. Because the algorithm performance needs to be estimated on objects unseen during the training phase, object sorting may pose a small-sample size problem. In such situation, a cross-validation procedure conducted over objects is necessary to provide realistic estimates of sorting performance.

A set of available objects is split into ten parts by sampling the high-level classes. In each fold, the nine parts are used for training of the sorting algorithm. The performance of the trained algorithm is then estimated using the remaining tenth of objects. The process is repeated ten times so each object appears in the testing stage only once. Note that all the steps required for building of a sorting algorithm such as feature-extractors and pixel classifiers are trained using the objects in the per-fold training set.

4.1 Sorting accuracy

The eventual goal of a sorting system is accurate allocation of new objects into the sorting categories (classes). In order to estimate the sorting accuracy, the mean error rate is usually adopted. This error metric, however, depends on the prior probabilities of the classes. In order to compare mean error rates, the class priors should be fixed. In this study, we estimate the mean error rate assuming the equal class priors e_{eq}.

In reality, the class priors may shift. For example, the fraction of defective potatoes in a sorting batch may change based on a particular geographical location being processed. A sorting algorithm with a high accuracy at the equal-prior operating point may loose performance when confronted with unbalanced priors [6]. In order to evaluate algorithms over the range of priors, we construct the Receiver-Operator Curve (ROC).

4.2 Execution speed

Pattern recognition algorithms are usually developed incrementally as the designer's understanding of the problem improves. The resulting implementation prototype is thus optimized for the sake of flexibility not execution speed. Measuring the speed of an algorithm prototype may be, thereby, not indicative of its true speed potential. On the other hand, maximizing the execution speed during the algorithm design phase limits the number of investigated variants. The eventual fast solution may be very sub-optimal in terms of sorting accuracy or robustness. Considering this accuracy/speed trade-off, we favor the flexible incremental design of multiple sorting algorithms and propose to estimate their execution speed based on a model of an ideal implementation. Ideal implementation assumes that all quantities that may be precomputed prior the execution are indeed precomputed.

In this study, the execution speed is estimated by the number of operations required for processing of a single pixel/spectrum in a hyperspectral image. The effect of pixel-to-object fusion by voting is neglected because the same majority voting combiner is used for all studied algorithms. The eventual object-sorting speed is proportional to the number of pixels in an average object.

5 Experiments

The dataset, used in this study, originates from the recycling application. Hyperspectral images of objects moving on the conveyor belt were acquired using N17 spectrograph from SpecIm, Ltd. and SU-128 InGaAs camera from Sensors Unlimited. The data were normalized using dark current and white background reference images.

Two sorting categories (classes) are considered, defined in terms of three or six material types (modes), respectively. Each object is entirely composed of a single material. The material type of each object is known during the training phase. Table 1 provides details on the number of objects and pixels/spectra in each material type and class. On average, an object is represented by 360 pixels/spectra.

Table 1 Number of objects and spectra for each of the material types and class of the design dataset

Class	Material type	No. of objects	No. of spectra
1	a	18	7,175
1	b	58	21,055
1	c	16	4,078
2	d	4	1,597
2	e	6	2,170
2	f	5	2,648
2	g	8	1,762
2	h	12	5,442
2	i	6	2,794

The available set of 197 objects was randomly split into a design set of 133 objects and the validation set with 64 objects. Random sampling is performed for objects of each material type separately. The validation set thereby contains objects of each material type. While the design set is used for incremental design and evaluation of the discussed sorting algorithms, the validation set was kept untouched during the design stage. Such splitting of the data limits the number of objects available during the algorithm design and thereby also the achieved sorting performance. It allows us, on the other hand, to validate the reliability of the performance estimates obtained during the design phase (Sect. 5.3).

5.1 Results on sorting accuracy

Figure 1 presents the mean ROC curves estimated by the cross-validation procedure. In the sub-figure 1a we can observe that the DBC-NN method, implementing a type of a nearest mean classifier, performs worse than the FLD-based algorithms. This class covariance-structure apparently carries important discriminatory information. The bare FLD classifier, applied directly to the spectral data, provides more accurate discrimination than the FLD classifier trained on features extracted by the GLDB method. This observation suggests that the sample size is sufficient for training a classifier directly on wavelength features and additional dimensionality reduction step is not necessary. There is little observable benefit of the non-linear PCA-3NN classifier over the linear methods.

Sub-figure 1b presents the results of the dissimilarity-based algorithms. FLD classifiers built in a dissimilarity space (dashed lines) perform better than the nearest-neighbor rules (solid lines). This result is in agreement with other studies suggesting that exploiting correlations between dissimilarities to prototypes is better classification strategy than simple ranking of dissimilarity values [3, 11]. The classifiers employing the derivate-based dissimilarity are to be preferred over the algorithms using the SAM distance. This suggests that shapes of spectra, emphasized by the derivative-based dissimilarity, bear additional discriminatory information.

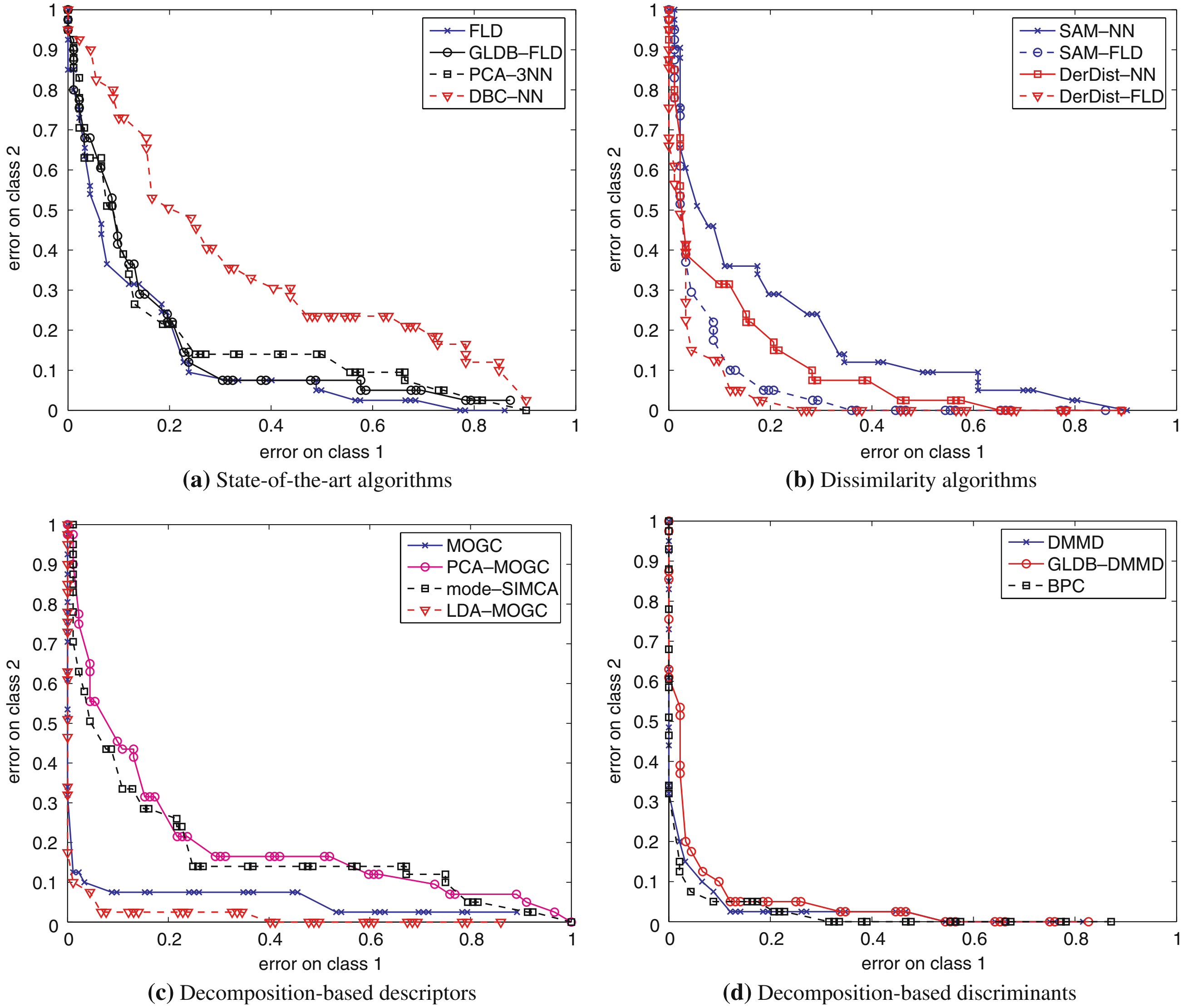
The results of decomposition-based descriptors are given in sub-figure 1c. The Gaussian mixture model built directly in the 128-dimensional feature space of wavelengths appears to perform significantly better than algorithms employing unsupervised (PCA-MOGC, circular markers) or per-mode (modal-SIMCA, square markers) dimensionality reduction. The PCA dimensionality reduction probably recovers directions preserving the overall data variance but unrelated to the class discrimination. The mode-SIMCA algorithm may also suffer from the choice of the proper distance measure combining the contribution of the in-model and out-of-model distances, as observed in [13]. On the other hand, the supervised dimensionality reduction by LDA on modes (LDA-MOGC) improves over the result of the MOGC algorithm. It is clearly beneficial to derive the low-dimensional (8D) feature space informative for mode and thereby also class separation. The LDA-MOGC yields the highest sorting accuracy of all investigated algorithms.

The decomposition-based discriminants are presented in sub-figure 1d. The BPC method (dashed line) slightly outperforms the two DMMD approaches. The DMMD classifier applying the FLD classifiers directly to spectral wavelengths provides better solution than the GLDB-DMMD algorithm employing the sub-problem GLDB extractors. However, these performance differences are not significant, as we can observe in Fig. 2 (the ROC plots in Fig. 1 omit the error bars for the sake of clarity).

5.2 Relation between sorting error and execution speed

In Fig. 2 we visualize the relation between the equal-prior classification error of sorting algorithms and their predicted execution speed in terms of number of operations (plotted on a logarithmic scale). In case of classification errors, the means and standard deviations of the means over the ten fold cross-validation folds are provided. For the number of operations, minimum and maximum values are given.

The group of state-of-the-art methods exhibits fast execution due to dimensionality reduction algorithms involved (GLDB, PCA or FLD). The only exception is the PCA-3NN algorithm, slowed down by the computation of dissimilarities to the 1000 prototype objects. None of the algorithms, however, reaches the equal-prior error lower than 15%.



(a) State-of-the-art algorithms **(b)** Dissimilarity algorithms

(c) Decomposition-based descriptors **(d)** Decomposition-based discriminants

Fig. 1 Mean ROC curves for the four algorithm groups

The accuracy improvement of classifiers built in a dissimilarity-space (SAM-FLD and DerDist-FLD) over the nearest neighbor rules (SAM-NN and DerDist-NN) does not result in a significantly slower execution. Because the same set of prototypes is used, the marginal slow-down is only due to the use of a weighted sum instead of the minimum operation on the same set of dissimilarity values [11]. Using the identical prototype sets, the derivative-based dissimilarity yields a more informative data representation than the SAM measure. It is, however, more than five times slower in execution. Note that our execution model assumes sequential implementation. By parallelizing the dissimilarity computations, the speed of algorithms could be significantly increased.

The full mixture of Gaussians built in the original space of spectral wavelengths (MOGC) provides an accurate but slow classifier. The dimensionality reduction using PCA on the entire dataset (PCA-MOGC), or on the modes (mode-SIMCA), does increase the execution speed but results in the loss of accuracy. The supervised feature extraction using LDA on the modes, followed by a mixture model (LDA-MOGC) yields the best accuracy/speed trade-off. It is apparently beneficial to use the supervision and prior knowledge on the sorting problem (mode labels) for the derivation of a data representation. The extracted 8-dimensional space (LDA projection using nine modes as classes) allows for construction of a fast non-linear classifier.

The decomposition-based discriminants provide overall high-accuracy solutions. The BPC classifier exhibits large speed variability. The PCA projections computed for all the 36 mode pairs by fixing the 95% of

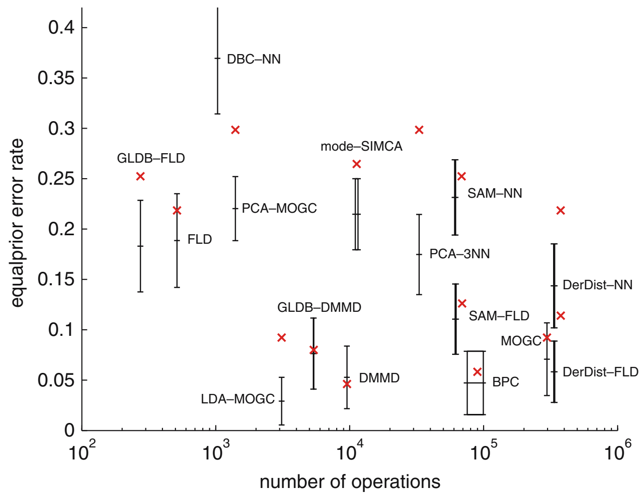


Fig. 2 The relation between equal-prior sorting error and the estimated per-pixel execution speed. *The error boxes* refer to results of cross-validation on the design set. *The cross markers* denote the results obtained by training the classifier of the full design set and estimating its performance on the entirely independent set of validation objects

preserved variance result in output dimensionalities varying from 3 to 45 dimensions where quadratic discriminants are built. While the DMMD classifier reaches similar classification error of 5%, it is based on computationally-cheap linear discriminants and derives the needed non-linear solution using sigmoids with only a single parameter for each of the 18 sub-problems. This makes it an order of magnitude faster than BPC. Deriving specific GLDB features for each sub-problem does increase the speed further but also lowers the classification accuracy.

5.3 The performance on the independent validation set

Complex sorting algorithms are usually developed incrementally using a design set of objects. The application-specific problems such as multi-modality or required level of non-linearity are understood by the designer based on of algorithms' performances estimated on the design set. Based on this knowledge, more advanced architectures are introduced. Such incremental algorithm development inevitably results in overtraining, i.e. in over-emphasizing the structure of the design set which is not necessarily repeated in the object sets met later on during production.

In this section, we use an entirely independent set of objects to assess the performance differences with respect to the expected error rates predicted during the design stage by the cross-validation procedure. Each of the studied algorithms was re-trained using the entire design set and executed on the validation set of objects. The results are presented in Fig. 2 by red cross markers. Note that although these results illustrate a realistic performance snapshot which may be observed in production, they lack any statistical significance.

It is interesting to compare the validation performances of related types of algorithms. For example, while the validation-set error rate of the bare FLD algorithm falls within the error bar predicted on the design set, the error rate of the GLDB-FLD algorithm does not. We can observe the similar effect between the LDA-MOGC algorithm which is preceded by a feature extractor and the MOGC classifier which is not. This could suggest that an additional feature-extraction step may reduce the algorithm robustness by emphasizing directions which are not informative for the class separation in case of the validation-set objects.

However, all the three decomposition-based discriminants predict the validation-set performance very well including the GLDB-DMMD algorithm which does employ an extra feature extraction in each of the sub-problems. Although the existing evidence is not sufficient to draw generalizing conclusions, we hypothesize that the decomposition-based discriminants may exhibit more robustness due to their multi-stage nature. Contrary to the LDA-MOGC algorithm sequentially extracting features and performing a non-linear classification, the DMMD and BPC algorithms combine the simpler sub-problem decisions into the single per-pixel decision. The combiner might be increasing the algorithm robustness.

The overall conclusion we draw from this experiment is that the design set of objects is apparently not well representative of the validation set. Some algorithms are severely affected by this discrepancy. In order to understand the reasons for the unexpectedly large validation-set error, a larger scale study is needed including multiple validation sets.

6 Conclusions

This paper discusses the design of pattern recognition algorithms for object sorting based on hyperspectral imaging. The design of object sorting systems is complicated by the presence of multiple levels such as pixels/spectra, objects, material types and sorting categories.

Sixteen algorithms, compared on a real-world industrial sorting problem, vary the amount of available prior knowledge, such as the existence of lower-level concepts or definition of sorting categories in terms of these concepts.

Simple classifiers learning the high-level classes by classical combination of PCA dimensionality reduction followed by a classifier provide fast but high-error solutions. The dissimilarity algorithms may reach high-accuracy sorting but are expensive in execution due to large number of prototypes involved. Sorting accuracy may be significantly improved by exchanging the nearest neighbor rule with a classifier trained in a similarity space. Execution complexity of dissimilarity-based classifiers could be reduced by parallelization of dissimilarity evaluations.

Two types of algorithms employing decomposition of the sorting problem were considered, namely the descriptors and discriminants. While the descriptors attempt to model the lower-level concepts, the discriminant merely learn to distinguish between them. The Gaussian mixture trained on the original spectra (our basic data descriptor) provides high sorting accuracy, but is very slow in execution due to high dimensionality. The best accuracy/speed trade-offs are reached by a mixture trained in a low-dimensional feature space extracted using the prior knowledge on material-types (LDA-MOGC) and by the ensemble of sub-problem specific linear discriminants (DMMD).

Finally, an independent set of objects was used to compare the performance predicted during the algorithm design with that observed in a realistic production situation. The results suggest that some algorithms (DMMD, BPC, MOGC, SAM-FLD) might be more robust to such unseen configuration of objects than others (LDA-MOGC, DerDist-FLD). Statistical assessment of algorithm robustness to varying object configuration and understanding of reasons of this variation is of great importance for sorting system design and will become a topic of our future research.

Acknowledgments The authors would like to thank Sergey Verzakov and Carmen Lai for fruitful comments on the manuscript. This research is/was supported by the Technology Foundation STW, applied science division of NWO and the technology program of the Ministry of Economic Affairs.

References

1. Cheng, X.: Hyperspectral imaging and pattern recognition technologies for real time fruit safety and quality inspection. Ph.D. thesis, University of Maryland (2004)
2. Duin, R.P.W., Juszczak, P., de Ridder, D., Paclík, P., Pekalska, E., Tax, D.M.J.: PR-Tools 4.0, a Matlab toolbox for pattern recognition. Tech. rep., ICT Group, TU Delft, The Netherlands (2004) http://www.prtools.org
3. Duin, R.P.W., de Ridder, D., Tax, D.M.J.: Experiments with object based discriminant functions; a featureless approach to pattern recognition. Pattern Recognit. Lett. **18**(11–13), 1159–1166 (1997)
4. Kong, S., Chen, Y., Kim, I., Kim, M.: Analysis of hyperspectral fluorescence images for poultry skin tumor inspection. Appl. Opt. **43**(4), 824–833 (2004)
5. Kumar, S., Gosh, J., Crawford, M.M.: Best-bases feature extraction algorithm for classification of hyperspectral data. IEEE Trans Geosci Remote Sens., **39**(7), 1368–1379 (2001)
6. Landgrebe, T., Paclík, P., Tax, D., Verzakov, S., Duin, R.: Cost-based classifier evaluation for imbalanced problems. In: Fred, A., Caelli, T., Duin, R., Campilho, A., de Ridder, D. (eds.) Structural, Syntactic, and Statistical Pattern Recognition, Proc. SSSPR2004 (Lisbon, Portugal, August 2004), LNCS, vol. 3138, pp. 762–770. Springer, Berlin Heidelberg New York (2004)
7. Lawrence, K., Windham, W., Park, B.R.J.B.: Hyperspectral imaging for poultry contaminant detection. NIR News **12**(5) (2001)
8. Leemans, V., Destain, M.: A real-time grading method of apples based on features extracted from defects. J. Food Eng. **61**, 83–89 (2004)
9. Leitner, R., Mairer, H., Kercek, A.: Real-time classification of polymers with nir spectral imaging and blob analysis. Real-Time Imaging **9**(4), 245–251 (2003)
10. Noordam, J.: Chemometrics in multispectral imaging for quality inspection of postharvest products. Ph.D. thesis, Radboud University Nijmegen (2005)
11. Paclík, P., Duin, R.P.W.: Dissimilarity-based classification of spectra: computational issues. Real-Time Imaging **9**(4), 237–244 (2003)
12. Paclík, P., Duin, R.P.W.: Designing multi-modal classifiers of spectra: a study on industrial sorting application. In: accepted for the 2nd Spectral Imaging workshop, Villach, Austria (2005)
13. Paclík, P., Tax, D., Verzakov, S., Duin, R.: Simplifying the model-based classifiers for multi-modal problems in classification of spectra. In: Nicolls, F. (eds.) Proc.of the Sixteenth Annual Symposium of the Pattern Recognition Association of South Africa, pp. 33–38. PRASA, Cape Town, RSA (2005). Langebaan, South Africa
14. Paclík, P., Verzakov, S., Duin, R.P.W.: Hypertools: the toolbox for spectral image analysis. Tech. rep., Pattern Recognition Group, TU Delft, The Netherlands (2003)
15. Paclík, P., Verzakov, S., Duin, R.P.W.: Improving the maximum-likelihood co-occurrence classifier: a study on classification of inhomogeneous rock images. In: Kalviainen, A.K.H., Parkinen, J., (eds.) Image Analysis—SCIA 2005 (Proc. 14th Scandinavian Conf. Joensuu, Finland, June 19–July 22), Lecture Notes in Computer Science, vol. 3540, pp. 998–1008. Springer, Berlin Heidelberg New York (2005)
16. Polder, G., van der Heijden, G., Young, I.: Tomato sorting using independent component analysis on specral images. Real-Time Imaging **9**(4), 253–259 (2003)
17. Tatzer, P., Wolf, M., Panner, T.: Industrial application for inline material sorting using hyperspectral imaging in the nir range. Real-Time Imaging **11**, 99–107 (2005)
18. Wold, S., Sjostrom, M.: SIMCA: a method for analysing chemical data in terms of similarity and analogy. ACS Symp. Series 52 (1977)
19. Yuhas, R.H., Goetz, A.F.H., Boardman, J.W.: Discrimination among semiarid landscape endmembers using the spectral angle mapper (SAM) algorithm. In: Summaries of the Third Annual JPL Airborne Geoscience Workshop, pp. 147–149. Jet Propulsion Laboratory, Pasadena, CA (1992)