On Human–Machine Interaction During On-line Image Classifier Training *

Edwin Lughofer
Department of Knowledge-based Mathematical Systems, Johannes Kepler University
A-4040 Linz, Austria, edwin.lughofer@jku.at

Jim Smith, Praminda Caleb-Solly and Muhammad Atif Tahir
Department of Computer Science
University of the West of England, Bristol, UK, james.smith@uwe.ac.uk

Christian Eitzinger
Profactor GmbH
A-4407 Steyr-Gleink, Austria, christian.eitzinger@profactor.at

Davy Sannen and Hendrik van Brussel
Department of Mechanical Engineering, Katholieke Universiteit Leuven
Celestijnenlaan 300B, B-3001 Leuven (Heverlee), Belgium, davy.sannen@mech.kuleuven.be

Abstract

This paper considers on a number of issues that arise when a trainable machine vision system learns directly from humans, rather than from a "cleaned" data set, i.e. data which is perfectly labelled with complete accuracy. This is done within the context of a generic system for the visual surface inspection of manufactured parts, however, the issues treated are relevant not only to wider computer vision applications, but also to classification more generally. Some of these issues arise from the nature of humans themselves: they will be not only internally inconsistent, but will often not be completely confident about their decisions, especially if they are making decisions rapidly. People will also often differ systematically from each other in the decisions they make. Other issues may arise from the nature of the process, which may require the machine learning to have the capacity for real-time, online adaptation in response to users' input. It may be that the users cannot always provide input to a consistent level of detail. We describe how all of these issues may be tackled within a coherent methodology. Using a range of classifiers trained on real data sets from a CD imprint production process, we will present results which show that most of these issues may actually lead to improved performance.

* This work was funded by the EC under grant no. 016429, project DynaVis and the Upper Austrian Technology and Research Promotion. It reflects only the authors' views.

1 Introduction

In many machine vision applications, such as inspection tasks for quality control, an automatic system tries to reproduce human cognitive abilities. The most efficient and flexible way to achieve this is to learn the task from a human expert [5], either by supervised data or by knowledge acquisition from the human operators in form of rule bases. Typically, Machine Learning systems are trained in supervised batch mode from a set of example data items each of which has a unique label. However, as Machine Learning technology moves from research laboratories to practical applications such as Machine Vision, a range of issues arise concerning how humans relate to, and interact with such systems [9] [6]. Not only does this question the feasibility, or even relevance of considering "cleaned" data sets, there is an increasing demand for systems to operate in situations where off-line batch-mode processing is not appropriate [7]. This can occur if data is hard, time-consuming or costly to obtain, or if the underlying processes change fairly rapidly, requiring re-configuration. Both of these cases lead to the need for an element of incremental on-line training [10], which prompts a renewed interest in the nature of the human interaction with adaptive ML systems [3] [1].

In this paper we focus on a number of issues relating to human-machine interaction in the context of a generic system for the visual surface inspection of manufactured parts. Section 2 describes the basic architecture of our generic system, the data sets used in this work and the experimental

framework. Section 3 deals with the issues arising when the nature of the application demands real-time on-line learning after an initial batch-mode phase. Section 4 deals with the fact that different users will often differ systematically from each other, and considers how best to incorporate this diversity of information. Other issues may arise from the fact that humans cannot always work as fast as the underlying applications. For example, Section 5 considers how demand for rapid user responses may reduce the level of detail in the feedback they can produce, and suggests some alternative ways for dealing with this. In Section 6 we consider that for a number of reasons, the operator(s) may not be completely confident in their decisions and show how a suitable change in the human-machine interface used for online labelling can be exploited to capture this information and lead to performance improvements. We end by drawing some conclusions from this work, and highlighting areas that require further research attention (Section 7).

2 Architecture and Data Sets

The whole framework is shown in Figure 1. Starting from the original image (left) a so-called "contrast image" is calculated, where the gray value of each pixel correlates to the degree of deviation from the normal appearance of the surface. This contrast image just serves as an interface to the subsequent processing steps in order to remove the application-dependent elements. From the contrast image regions of interest (ROI) are extracted, each of which contains a single object which may or may not be a fault. From the segmented ROIs a large number of object features are calculated such as area, brightness, homogeneity or roundness of objects characterizing their shape, size etc. These are complemented by aggregate features characterizing images as a whole. The feature vectors are then processed by a trained classifier system that generates a final good/bad decision for the whole image. For off-line training the classifiers we exploited basically four different methods, namely: the decision tree-based classifiers *CART* [2], and *C4.5* [14]; k-Nearest Neighbours (*kNN*) [8]; and two incremental learning algorithms *eVQ-Class* [12] and *FLEXFIS-Class* [13]. When applying these classification algorithms on the standard aggregated feature sets (containing 17 predefined features) to real-world data from an on-line CD imprint production process, we achieved accuracies between 87% and 93% as estimated by 10-fold cross-validation [16]. Even though the accuracies lie in a reasonable range, they fall short of the original goals for a very high-performance and robust system.

Hence, one goal of the enhanced human-machine interaction issues discussed in this paper is to guide the classifiers towards 98% accuracy. Another goal is to widen the applicability and usability of the whole system. The specific issues for human-machine interaction are highlighted in Figure 1, where the labels HMI 1-4 refer to the issues dealt with in sections 3,4,5 and 6 respectively.

3 Incremental Classifiers based on Operators' Feedback

On-line incremental training comes with adaptation of parameters and evolving structures (e.g. evolving neurons, rules etc.) and is required whenever the operator gives a feedback upon the classifier(s) decisions during on-line production mode. This is because a periodic rebuilding of the classifier using all the samples seen so far is impractical as it slows down the training process too much. On the other hand, if the classifier(s) would not be updated at all during the on-line production mode the classifier could not refine its parameters, react on changing operating conditions or system behaviors and hence end up in not satisfying performance. From a Human-Machine-Interface perspective, on-line learning highlights some interesting points. In a batch-mode model the user is asked to perform repeated interactions for image labelling without and feedback or reward. This can make the process seem time-consuming and possibly pointless. In contrast to this, on-line training means that the user can "see" the system learning from their input, building a progressively more accurate model, which can help to motivate them and increase their focus and attention. For dealing with the on-line learning problem based on operator's feedback we exploited an evolving clustering-based classifier (*eVQ-Class* [12]) and an evolving fuzzy rule-based classifier (*FLEXFIS-Class* [13]). The first one is based on vector quantization and incorporates an on-line split-and-merge strategy for adapting the cluster partitions. The second one evolves multiple Takagi-Sugeno fuzzy (regression) models. Both take into account the class labels supplied during the incremental clustering process for forming the classifiers and are sufficiently flexible to integrate new operating conditions (such as new image types) and newly arising fault classes into the structure of the classifier.

N-fold cross-validation assumes a fixed data set, and so is not an appropriate measure here. Instead the CD imprint data set was split into three. The first third of the images is used for initial off-line training. The middle third is used to simulate incremental on-line training of the classifier, and is sent sample per sample into *eVQ-Class* and *FLEXFIS-Class*. The final third is used as a test set for evaluating the trained classifiers. This is an appropriate way of estimating the true on-line accuracy as the whole CD imprint data was stored in the same order as recorded on-line. Table 1 shows the performance of the incremental classifiers vs. their corresponding batch versions, i.e. trained in initial off-line mode with the first batch of data and not further up-

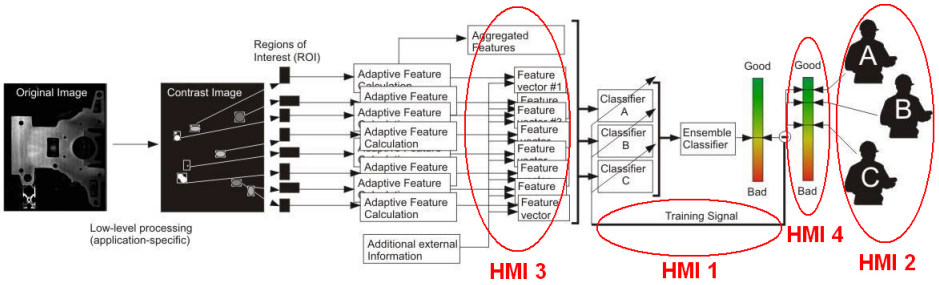

Figure 1. Classification framework for classifying images into good and bad, the four major HMI issues marked with red ellipsoids.

Table 1. Performance of incremental on-line vs. static (an re-trained) batch classifiers

Dataset	Operator01	Operator02	Operator03	Operator04
CART static	70.94	78.82	80.00	76.08
CART re-trained	82.74	90.59	90.39	88.24
eVQ-Class static	68.82	76.67	76.27	74.71
eVQ-Class inc.	83.33	88.82	88.43	88.43
FLEXFIS-Class static	68.63	85.29	85.68	78.24
FLEXFIS-Class inc.	84.12	88.06	89.92	84.90

dated (kept static): it can be seen that by doing an adaptation during on-line mode the performance on new unseen samples significantly increase by 10 to 15% over all operators. Furthermore, the third row shows us that, when re-training a batch classifier (the well known CART algorithm) on all training samples, the accuracy on the new unseen samples is not really better than for the incrementally trained approaches.

4 Handling Input from Multiple Users

The idea of *classifier ensembles* is to train a whole set (ensemble) of classifiers. Most research has considered the case where there is a single data set. Here these ensemble methods are used in a different context: the different operators train their individual classifiers as they think would be best and the contradictions among these operators are then resolved using an ensemble method. There are generally two ways to combine the decisions of classifiers in ensembles: classifier selection and classifier fusion [17]. The assumption in *classifier selection* is that each classifier is "an expert" in some local area of the feature space. *Classifier fusion* assumes that all classifiers are trained over the whole feature space. For our application the latter is appropriate since the operators train the system with the data which is provided by the vision system. The fusion of the outputs of the different classifiers (trained by the different operators) can be done using fixed or, if a "supervisor" has labelled the data, trainable classifier fusion methods. Note that this scenario is relevant in many companies: typically different operators work on the inspection systems in different shifts, while a supervisor, which is not working on the system, still somehow wants to be in control and have the operators make decisions similar to what he would do. Classifier fusion methods (for a detailed survey see e.g. [11]) include 1.) *Voting*; 2.) *Algebraic connectives* such as *maximum*, *minimum*, *product*, *mean* and *median*; 3.) *Fuzzy Integral*; 4.) *Decision Templates*; 5.) *Dempster-Shafer combination*; and 6.) *Discounted Dempster-Shafer combination* (an extension of 5.) recently proposed in [15]). Also the *Oracle*, a *hypothetical* ensemble scheme that outputs the correct classification if at least one of the classifiers in the ensemble outputs the correct classification, was considered. The accuracy of the Oracle can be seen as a "soft bound" on the accuracy which can be achieved by the classifiers and classifier fusion methods.

The CD data with 17 aggregate features described in Section 2 was labelled by 5 different operators. In the experiments described in this section each of these operators is considered as the "supervisor" in turn. Classifiers are trained for the other operators and these classifiers are then combined by the ensembles in order to better model the decisions of the supervisor. In Table 2 the first five rows show the effect of training a classifier with the input from one operator (row) then evaluating using the input from another

Table 2. Mean accuracy (in %) of classifiers when predicting labels provided by different users (columns). First five rows show single classifier trained by one operator. Last four rows show different methods for combining four classifiers trained by different operators to predict labels provided by fifth (column).

Test Data	Operator01	Operator02	Operator03	Operator04	Operator05
Operator01	90.57	89.27	85.33	88.94	71.42
Operator02	88.83	95.38	91.88	93.10	69.42
Operator03	86.50	93.42	93.90	92.68	70.59
Operator04	88.82	93.67	92.08	94.38	71.91
Operator05	72.42	71.64	71.73	73.58	91.25
Fuzzy Integral	88.50	94.42	91.61	93.69	71.25
Decision Templates	88.19	94.42	89.16	91.59	74.75
Disc. Dempster-Shafer	88.31	94.42	92.24	93.54	71.75
Hypothetical Oracle	95.83	98.89	97.07	98.60	78.90

(column). We can see that three operators make very similar decisions (Operators 02, 03 and 04), one operator differs slightly from these three operators (Operator01), and one operator makes decisions which are very different from all the other operators (Operator05). Note that the results of about 90% to 95% on the diagonal of this table denote the evaluation of the classifiers on the same data they were trained on. The last set of rows show the effect of training classifier using four different operators, and then combining them to predict the labels provided by a fifth. For reasons of space, only the best performing methods are shown, which are the trainable methods Fuzzy Integral (FI), Decision Templates (DT)and Discounted Dempster-Shafer (DDS) together with hypothetical Oracle. From these results we can see that the ensembles are able to represent the "supervisor" better than the individual classifiers in all cases, except when Operator01 is considered to be the supervisor. These improvements go up to close to 3% when Operator05 is considered as the supervisor. In general, FI and DDS are the best combination methods when several of the operators make decisions similar to the the supervisor; DT is the best combination method when none of the operators agree well with the supervisor (this is the case when Operator05 is considered as the supervisor). In every case, except when Operator01 is the supervisor, the performance of the ensembles are also relatively close to the hypothetical Oracle. Note that if the operators do not agree very well with the supervisor a drop in the accuracy is recorded - e.g. approximately 20% of the decisions of Operator05 do not agree with *any* of the other operators. In this case hypothetical Oracle bounds the achievable accuracy below 80%. From these results we can conclude that the ensemble methods can be effectively used to combine the decisions of different users to model the decisions of a supervisor, with improvements of up to 3%.

5 Handling Variable Levels of Detail in User Inputs

A major problem of image classification problems is the fact that it is not known in advance how many regions of interests may be segmented from images occurring in the future, and yet most classification algorithms assume a fixed-size input data space. The most straightforward way to tackle this is to preprocess the object feature vectors through a learning system, and then present the outputs of that system as an additional image-level information. For example, if the data is labelled at the object-level, then supervised object-level classifiers can be built [4] or alternatively if object labels are not available, unsupervised clustering methods can be used to reduce the dimensionality [3]. Supervised learning methods are highly useful if the training images contain labels for each object, but obtaining this information requires significant operator input which may not be available off-line, or may simply be infeasible on-line due to the speed of production.

In this case a Grapical User Interface (GUI) was designed to permit rapid annotation of images so that each operators could label all of the ROIs (regions of interest). After a series of interviews it was found that the users discriminated between 7 different types of "pseudo-defect" and six types of defect. From the 1534 images used, a total of 4500 objects were segmented and labelled by operators 1-4. Based on these operators-assigned labels supervised object level classifiers were constructed. Their outputs –i.e. the number of each type of object present on an image– were added to the aggregate image data features and classifiers trained and tested in a n-fold cross-validation regime. This was repeated for each operator independently. The motivation of this approach was to include information about the distribution of regions of interests among the different

Table 3. Classification accuracy using two-level approach

Dataset	Op01	Op02	Op03	Op04
CART	93.9	96.7	96.1	96.4
C4.5	93.7	96.1	97.2	95.6
1NN	93.7	96.2	95.0	95.9
9NN	92.6	96.0	94.7	95.0
eVQ-Class	92.3	95.2	92.6	92.7
C4.5-C12	94.5	97.4	96.2	96.8

defect classes for a better characterization of the whole images. We also used an unsupervised approach, whereby we applied a clustering algorithm to find C clusters in our object training data, and then objects in the test data were each assigned to the nearest cluster centroid - creating an image level data set with $17 + C$ features from which supervised classifiers could be trained. This approach has a similar motivation as the one before, but has the advantage that it can be also used when no labels on the single objects are provided (which is often the case because of workload saving reasons). Table 3 shows the classification accuracy obtained for different operators. The first four rows show the results using the supervised approach with different types of classifier. The final row shows the result of taking the unsupervised cluster based to create extra features to be used by a C4.5 classifier trained at the image level. Experimentation showed that a value of $C = 12$ gave the best results - which interestingly is almost the same as the number of classes defined by the users. Here, training an object classifier is a complex multi-classification approach, where just an accuracy of about 80% can be achieved. However, the accuracy as well as the miss-detection rates on the whole images can be again improved when taking the outputs from the object classifiers as inputs to the aggregated features (leading to 30 features in sum). More importantly, the results demonstrate that the unsupervised approach actually does as well, if not better, than the supervised approaches.

6 Accomodating Partial Confidence of Operators

During the setup phase of an image classification framework, the labelling of several images can be a difficult task for the operators, especially in cases where real faults are hard to distinguish among themselves or between so-called pseudo-errors This problem can become even worse when the operators are not working in the relative calm of an off-line setting, but are providing real-time decisions at a speed driven by other factors. In this sense, it is promising, sometimes even necessary for the operator(s) to provide information how confident they were when assigning the labels to certain images or objects. Here, only the confidences in the whole image labels are taken into account. The simplest way is to represent the users's confidence as a value in range 0.0 (very unconfident) to 1.0 (very confident). This raises two issues: with what precision should the confidence be used, and how should this information be obtained from the users? Thankfully there is a body of work related to how opinions can best be gathered, which favors the so-called "Likert" scale used in questionnaires $\{$*strongly agree,. . .,strongly disagree*$\}$. Similarly here, rather than asking users to spend time thinking of an exact value to assign, we ask them for one of five distinct values, i.e. $\{20\%, 40\%, 80\%, 100\%\}$ confidence, partially driven by the needs of the GUI, resulting from an intensive round of discussion and design iteration with the industrial users. The next question is how to incorporate this extra information into the learning system. We evaluated two principle approaches. The first approach treats the task as a regression problem rather than a classification one. User's decisions are transformed as $score = 0.5 \cdot (1.0 + / - confidence)$, with $+$ or $-$ for ok and deceptive decisions respectively. However, results showed that this approach did not significantly improve over the two-level approaches from earlier sections. One reason for this could be that regression modelling more or less just washes up the quite crisp decision boundaries achieved by classifiers on the good/bad labels, where in the end a threshold value of 0.5 again ends up in a crisp decision boundary, which does not generalize better as one which is directly learned on the class labels. It is also only applicable to two-class problems, and so is not suitable for more generic learning problems.

The second approach test is based on duplicating the (extended) aggregated feature vectors according to the assigned confidence values. Thus a feature vector corresponding to an image which is labelled with 1.0 confidence is duplicated five times, another one labelled with 0.8 confidence is duplicated four times etc. In this sense, feature vectors which are labelled with a higher confidence are higher weighted in the training process than those labelled with a lower confidence. Applying this approach on the two-level CD feature data set gives the results shown in Table 4. As can be seen, the results improve for all classifiers except 1NN. This latter is to be expected as of course duplication has no effect when only one instance is considered to make each decision. In contrast, when a larger groups of neighbors are used (9NN - column 5) the increase can be dramatic as "confident" images outvote others. Not only does this technique give improvements for all the different types of classifiers, it does so for all operators: towards 98% for operator #2 and operator #3 and to 97% for operator #8. This was the original goal in as outlined in Section 2.

Table 4. Classification accuracy using two-level approach and duplicating feature vectors according to confidence levels of operators. Results in bracket show improvement in percentage by each classifier over two-level approach.

Method	CART	C4.5	1NN	9NN	eVQ-Class
Operator01 Acc.	94.4 (+0.5)	94.0 (+0.3)	93.7 (+0.0)	94.2 (+1.6)	93.4 (+1.2)
Operator02 Acc.	97.7 (+1.0)	96.9 (+0.8)	96.2 (+0.0)	96.2 (+0.2)	96.7 (+1.5)
Operator03 Acc.	98.0 (+1.1)	97.3 (+0.1)	95.0 (+0.0)	95.2 (+0.5)	96.5 (+3.9)
Operator04 Acc.	97.0 (+0.6)	96.7 (+1.1)	95.9 (+0.0)	95.9 (+0.9)	95.6 (+2.9)

7 Conclusion and Outlook

As machine learning systems move out of the laboratory and into real-world applications such as vision and image processing, it is valuable to reconsider some of the assumptions that have been made about how such systems can best learn from users. In this paper we have discussed some of the more important human-machine interaction problems, and suggested how they might be handled. Experiments conducted with 'real' data within the context of a generic image processing system show that when properly handled, the human factors can represent an additional form of information to these systems for improving performance and may widen the applicability and usability, rather than to be a disagreeable source of noise. Key issues of these factors include on-line guidance and feedback, a diversity of user skills, uncertainties as well as different levels of know-how and detail in users' input. The improvements are made possible by recent advances in the speed with which GUIs can operate. The next generation of user-interaction devices offers the potential to build on this research, creating much richer human-machine learning interaction.

References

[1] H. Bekel, I. Bax, G. Heidemann, and H. Ritter. Adaptive computer vision: online learning for object recognition. In *Proceedings of the 26th DAGM Symposium on Pattern Recognition*, pages 447–454. Springer-Verlag Berlin, 2004.

[2] L. Breiman, J. Friedman, C. Stone, and R. Olshen. *Classification and Regression Trees*. Chapman and Hall, Boca Raton, 1993.

[3] P. Caleb-Solly and J. Smith. Adaptive surface inspection via interactive evolution. *Image and Vision Computing*, 25(7):1058–1072, 2007.

[4] P. Caleb-Solly and M. Steuer. Classification of surface defects on hot rolled steel using adaptive learning methods. In *Proc. of the IEEE Fourth International Conference on Knowledge-Based Intelligent Engineering Systems and Allied Technologies,vol. 1*, pages 103–108.

[5] E. Castillo and E. Alvarez. *Expert Systems: Uncertainty and Learning*. Springer Verlag New York Inc., New York, USA, 2007.

[6] R. Cipolla and A. P. (editors). *Computer Vision for Human-Machine Interaction*. Cambridge University Press, Cambridge, UK, 1998.

[7] F. Gayubo, J. Gonzalez, E. D. L. Fuente, F. Miguel, and J. Peran. On-line machine vision system for detect split defects in sheet-metal forming processes. In *Proc. of the 18th International Conference on Pattern Recognition*. IEEE Comput. Soc., Los Alamitos, CA, USA, 2006.

[8] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference and Prediction*. Springer Verlag, New York, Berlin, Heidelberg, Germany, 2001.

[9] A. Jaimes and N. Sebe. Multimodal human computer interaction: A survey. In *Proceedings of the International Workshop on Human-Computer Interaction, HCI/ICCV 2005*, pages 1–15. Springer Verlag, 2005.

[10] N. Kasabov. *Evolving Connectionist Systems - Methods and Applications in Bioinformatics, Brain Study and Intelligent Machines*. Springer Verlag, London, 2002.

[11] L. I. Kuncheva. *Combining pattern classifiers: Methods and algorithms*. Wiley, 2004.

[12] E. Lughofer. Extensions of vector quantization for incremental clustering. *Pattern Recognition*, 41(3):995–1011, 2008.

[13] E. Lughofer, P. Angelov, and X. Zhou. Evolving single- and multi-model fuzzy classifiers with FLEXFIS-Class. In *Proceedings of FUZZ-IEEE 2007*, pages 363–368, London, UK, 2007.

[14] J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers Inc, U.S.A., 1993.

[15] D. Sannen, H. V. Brussel, and M. Nuttin. Classifier fusion using discounted dempster-shafer combination. In *Proc. of International Conferernce on Machine Learning and Data Mining (MLDM 2007)*.

[16] D. Sannen, M. Nuttin, J. Smith, M. Tahir, E. Lughofer, and C. Eitzinger. An interactive self-adaptive on-line image classification framework. In A. Gasteratos, M. Vincze, and J. Tsotsos, editors, *Proceedings of ICVS 2008*, volume 5008 of *LNCS*, pages 173–180. Springer, Santorini Island, Greece, 2008.

[17] K. Woods, W. P. Kegelmeyer, and K. Bowyer. Combination of multiple classifiers using local accuracy estimates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19:405–410, 1997.