Active Grading Ensembles for Learning Visual Quality Control from Multiple Humans

Davy Sannen and Hendrik Van Brussel

Katholieke Universiteit Leuven, Department of Mechanical Engineering
Celestijnenlaan 300B, B-3001 Heverlee (Leuven), Belgium
{Davy.Sannen,Hendrik.VanBrussel}@mech.kuleuven.be

Abstract. When applying Machine Learning technology to real-world applications, such as visual quality inspection, several practical issues need to be taken care of. One problem is posed by the reality that usually there are multiple human operators doing the inspection, who will inevitable contradict each other occasionally. In this paper a framework is proposed which is able to deal with this issue, based on trained ensembles of classifiers. Most ensemble techniques have however difficulties learning in these circumstances. Therefore several novel enhancements to the *Grading* ensemble technique are proposed within this framework – called *Active Grading*. The Active Grading algorithm is evaluated on data obtained from a real-world industrial system for visual quality inspection of the printing of labels on CDs, which was labelled independently by four different human operators and their supervisor, and compared to the standard Grading algorithm and a range of other ensemble (classifier fusion) techniques.

Keywords: Ensemble learning, grading, classifier fusion, visual quality control, learning from multiple humans.

1 Introduction

The most effective and flexible way to reproduce the human cognitive abilities needed to automate the required complex decision tasks in production processes, such as visual quality inspection, is by learning these tasks from human experts [1]. Traditionally, this is done using supervised learning, the data for which is provided by one selected person. The learning system is trained on this single set of data items, each of which has a unique label assigned to it. There may be some minor inconsistencies within the data, but these are usually considered as being random and each label is considered to be the ground truth.

However, quality inspection systems nowadays require the highest possible flexibility (due to e.g. changing customer demands, slight changes in the production line, new products to be inspected, etc.) [2]. This requires the human operators, currently performing their task manually, to be able to directly train and adapt the system without too much intervention of their supervisor. A typical situation is that there are three shifts and one operator per shift is working

J.A. Benediktsson, J. Kittler, and F. Roli (Eds.): MCS 2009, LNCS 5519, pp. 122–131, 2009.

on the system. Their supervisor would like to be in control of their decisions as much as possible, but does not perform the inspection him-/herself.

Visual quality inspection is difficult because it is based on human evaluations which cannot be converted (easily) into mathematical rules. The literature shows that the effectiveness of human visual quality inspection lies around 80% [3]. This means that in 20% of the cases the decision a human operator makes is different from his/her supervisor. These can be caused by a number of factors, such as different levels of experience and training or fatigue and stress, caused by the typically very strict time restrictions. Therefore techniques are needed which can deal with these contradictions and inconsistencies in a systematic way if we want the human operators to train the system themselves.

This paper proposes an approach for this kind of problem based on ensembles of classifiers. Each of the operators will train their own personal classifier, which are afterwards combined by an ensemble method, trained to represent the supervisor's decisions as well as possible. The main difficulty is that for a substantial part of the data (about 20%), systematically *none* of the operators agrees with their supervisor (and hence also not the decisions of the classifiers each of the operators trains). Most ensemble methods cannot cope well with such a setting. To solve this problem an extension of the *Grading* ensemble method [4] will be presented which is able to combine the decisions of the classifiers, trained by the different operators, in an appropriate way.

The remainder of this paper is organised as follows. A general framework for learning visual quality inspection from multiple humans is proposed in Section 2, in which each of the operators trains his/her own personal classifier, which are afterwards combined by an ensemble method. A novel ensemble method – called *Active Grading* – which is able to effectively combine the decisions of the different operators is formulated in Section 3 as a generalisation of the *Grading* ensemble method [4]. This ensemble method is able to learn in the setting of the application in this paper, i.e. when for a substantial part of the data none of the classifiers in the ensemble provides the correct classification. Experiments were done using real-world data obtained from an industrial visual quality control application for CD imprints, described in Section 4 together with the obtained results. Finally, a conclusion is formulated in Section 5.

2 Architecture

In Figure 1 a generic framework for learning visual quality inspection from multiple human operators is shown. Starting from the original image of the product which is to be inspected (left-hand side of the figure), a "deviation image" is calculated. The grey-level value of each pixel in this image correlates to the degree of deviation from the "optimal" image of the product. Usually the image is mostly black, with the potentially defective parts highlighted by non-black groups of pixels. The contrast image is used to eliminate application-specific elements from subsequent processing steps. From the contrast image Regions Of Interest (ROIs) are extracted. Essentially this is a grouping of the non-black

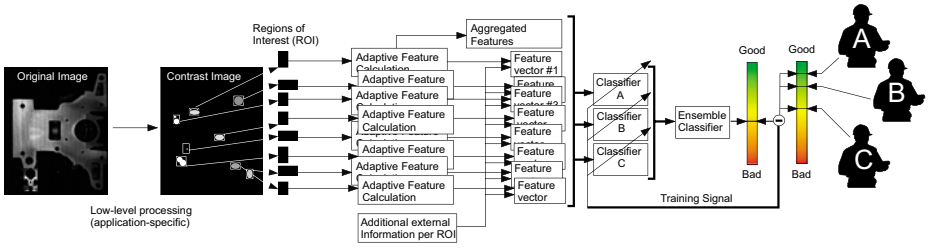

Fig. 1. General classification framework for visual quality inspection which can be trained by different human operators

pixels in the image into one or more distinct groups (called "objects"), each of which is a potential defect. The features of each object are calculated and can be complemented by three additional data sources: information about the ROIs, information about the status of the production process and aggregate features, characterising the images as a whole (information about all objects together) [5]. The feature vectors are processed by an operator-specific trainable classifier which generates a gradual good/bad decision for the entire image. This result is compared to the input of the human quality operator and a feedback loop, in the form of an (incremental) learning process, adapts the classification system. The operators, each training their own classifiers this way (during the initial training of the system or possible further adaptation), will inevitably provide different inputs to the system for some of the images – and thus their personal classifiers will also produce different classifications. These contradicting decisions are resolved using ensemble methods. This combination can be done using fixed rules or, if a supervisor labels the data as well, trainable ensemble methods, to better represent the decisions of the supervisor. The decision the ensemble makes is the final decision of the classification system.

Each of the operators will thus train their own personal classifier as they think would be best, according to their experience and expertise. Two levels of contradiction in the operators' decisions can be distinguished. The *inter-operator* contradictions are the *systematic* contradictions between the decisions of different operators. They can be caused e.g. by different levels of experience, training, skill, etc. The *intra-operator contradictions* are the contradictions an operator makes with decisions he has made himself. They can be caused by personal factors (such as the level of fatigue, attention, stress and boredom), environmental factors (such as a changed quality policy of the company and recent complaints of customers), etc. (see e.g. [6]).

The intra-operator contradictions, which are assumed to be basically random, are dealt with by the classifiers themselves. Several learning techniques can naturally handle noisy data (see e.g. [7]). The systematic inter-operator contradictions will be handled by the ensemble by combining the outputs of the different classifiers in a suitable and systematic way (see Section 3).

This architecture has several important advantages over other architectures which cannot deal with training input from different human operators. The classifiers trained independently by the operators will be easier to train, as they only need to handle the (non-systematic) intra-operator contradictions. The inter-operator contradictions are dealt with by the ensembles. Furthermore, it can be clearly distinguished what has been taught by which operator, enabling the system to give operator-specific feedback. The knowledge of each of the operators separately can be captured by the system.

3 The Active Grading Ensemble Framework

3.1 Combining the Decisions of Different Humans

There are two main requirements when selecting appropriate ensemble methods to be used in the architecture described in Section 2: (i) they have to combine an existing set of trained classifiers (trained by the operators); and (ii) the classifiers are no local experts in some parts of the feature space, but are trained over the entire features space (the classifiers are trained by the operators on the data provided by the inspection system, which cannot be influenced). The ensemble algorithms within the class of *classifier fusion* fulfill exactly these requirements. Classifier fusion techniques will be not explained in detail here; for reviews and detailed discussions see e.g. [8, 9]. They will however be used for comparison in the evaluation in Section 4. Also another ensemble technique closely related to classifier fusion, called *Grading* [4], fits these requirements. This technique will be described in Section 3.2. To effectively tackle the problem in this paper the Grading algorithm will be reformulated in a novel "*Active Grading*" framework in Section 3.3, which is a generalisation of the original Grading technique. Within this framework, several enhancements to the original algorithm are described, which will enable the algorithm to learn from different humans.

3.2 Grading

Let us consider the case in which there is a diverse set of N^D trained classifiers available, $D_1, \ldots, D_{N^D}$, each trained to classify their own training data set into N^C different classes. Creating diversity in this set of classifiers can be done in different ways: different training samples can be selected to train the classifiers, different feature subsets can be selected, the target output can be changed, etc. (see e.g. [9, 10]). Therefore, the training sets of the different classifiers can, but need not be of the same size or contain the same number of features. Note that in our application the diversity comes from changing the target output: each of the classifiers is trained by one of the human operators, which provides his own labelling for the training data. Assuming the training set of each of the classifiers D_i has a number of N_i^A attributes, the classifiers can be considered a mapping $D_i : \mathbb{R}^{N_i^A} \mapsto \mathbb{R}^{N^C}$, where the features are mapped to the classifier's confidence for each of the N^C different classes. Let us denote the confidence of classifier D_i

for class j when classifying a data item $\mathbf{x}$ as $D_{i,j}(\mathbf{x})$.[1] Without loss of generality we can assume $D_{i,j}(\mathbf{x}) \in [0,1]$.

In the Grading ensemble method [4], for each of the (level-1) classifiers D_i a level-2 "*Grading classifier*" G_i is trained, which predicts whether or not the level-1 classifier will provide the correct prediction, based on the *original* classifier feature space. The Grading classifiers are thus mappings $G_i : \mathbb{R}^{N_i^A} \mapsto [0,1]$, where 0 means the Grading classifier is perfectly sure the classifier will err; 1 means the Grading classifiers is perfectly sure the classifier will provide the correct classification. The training sets for the Grading classifiers are easily constructed by comparing the "crisp" classifier outputs with the target classifications. The crisp outputs of classifier D_i for a data item $\mathbf{x}$, $D_i^{\mathrm{Cr}}(\mathbf{x})$, can be obtained as follows:

$$\forall j : D_{i,j}^{\mathrm{Cr}}(\mathbf{x}) = \begin{cases} 1, \text{ if } j = \arg\max_k D_{i,k}(\mathbf{x}); \\ 0, \text{ else.} \end{cases} \tag{1}$$

Note that in the application in this paper, the target classifications are the labelling provided by the supervisor for (a part of) the training data.

When a new data item $\mathbf{x}$ is to be classified, each of the level-1 classifiers' predictions are obtained. The evaluation of the Grading classifiers is also obtained, indicating which of the level-1 classifiers is estimated to be correct. The final prediction of the Grading ensemble is obtained only from the level-1 classifiers which are estimated to be correct. If at least one classifier is estimated to be correct, the final prediction is calculated using the following formula [4]:

$$\forall j : Grad(\mathbf{x})_j = \sum_{i=1}^{N^D} \left\{ G_i(\mathbf{x}) | D_{i,j}^{\mathrm{Cr}}(\mathbf{x}) = 1 \wedge G_i(\mathbf{x}) > 0.5 \right\} , \tag{2}$$

where $Grad(\mathbf{x})_j$ is the final confidence of the Grading algorithm for class j.

If none of the classifiers is estimated to be correct the same procedure as above is applied, using $(1-G_i(\mathbf{x}))$ instead of $G_i(\mathbf{x})$ in (2). This comes down to using the classifications of all classifiers, even though they are estimated to be incorrect. In [4] this is described as being a "rare case" – although this may be true in the case of "standard" pattern recognition applications, this will not be the case in the application in this paper. The operators will contradict their supervisor *systematically* in some parts of the feature space, meaning that for significant parts of the feature space *none* of the classifiers (trained by the operators) will be correct (as will be shown in Section 4). Combining these outputs in the way the Grading algorithm does will result in incorrect classifications in these regions of the feature space. This is the problem the *Active Grading* approach will tackle, as will be explained in Section 3.3.

3.3 Active Grading

In this section the Grading algorithm [4] as described in Section 3.2 will be reformulated in a new "Active Grading" framework. Afterwards, enhancements

[1] $\mathbf{x}$ will be used to denote a data item, described by the appropriate features for the current classifier (if the classifiers are trained using different features).

to the Grading algorithm will be proposed within this framework, which will enable learning in the context of this paper – namely when none of the classifiers provide the correct classification for a significant part of the feature space.

Let us assume, like in Section 3.2, that we have a set of N^D trained classifiers, $D_1, \ldots, D_{N^D}$, and a set of N^D Grading classifiers, $G_1, \ldots, G_{N^D}$, each of which predicts whether its corresponding classifier will provide the correct classification for some new data item.

When a new data item $\mathbf{x}$ is to be classified, in the Active Grading framework a number of operations will be performed as follows. First, the classifier outputs are obtained and made crisp according to (1), resulting in $D_1^{\mathrm{Cr}}(\mathbf{x}), \ldots, D_{N^D}^{\mathrm{Cr}}(\mathbf{x})$. Also the outputs of the Grading classifiers are obtained (predicting whether the corresponding classifiers correctly classify $\mathbf{x}$), resulting in $G_1(\mathbf{x}), \ldots, G_{N^D}(\mathbf{x})$.

Next, for each classifier D_i, a *correction* operation is performed to correct the classifiers' outputs based on the outputs of the Grading classifiers, resulting in $D_i^{\mathrm{Corr}}(\mathbf{x})$. For the standard Grading algorithm described in Section 3.2 this operation is given by the following equation:

$$\forall i : D_i^{\mathrm{Corr}}(\mathbf{x}) = \begin{cases} G_i(\mathbf{x}) D_i^{\mathrm{Cr}}(\mathbf{x}), & \text{if } G_i(\mathbf{x}) > 0.5; \\ [1 - G_i(\mathbf{x})]\, D_i^{\mathrm{Cr}}(\mathbf{x}), & \text{else.} \end{cases} \tag{3}$$

Note that this operation is nothing more than a reweighing, which will be used further on in the case when none of the classifiers is estimated to be correct. At the end of this section an enhancement will be proposed which does perform a real correction of the classifier outputs, and which will form the heart of the Active Grading approach.

After the classifier outputs are "corrected", for each of the classifiers D_i an *inclusion* operation is performed to indicate which of the classifiers will participate in the final prediction, resulting in $I_i(\mathbf{x})$. For the standard Grading algorithm this operation is given by the following equation:

$$\forall i : I_i(\mathbf{x}) = \begin{cases} 0, \text{ if } G_i(\mathbf{x}) \leq 0.5 \wedge \exists k : G_k(\mathbf{x}) > 0.5; \\ 1, \text{ else.} \end{cases} \tag{4}$$

The final step in the Active Grading framework is the actual classifier fusion. The fusion of the Grading algorithm can now be simply written as follows:

$$\forall j : Grad_j(\mathbf{x}) = \sum_{i=1}^{N^D} \left\{ D_{i,j}^{\mathrm{Corr}}(\mathbf{x}) | I_i(\mathbf{x}) = 1 \right\} \ , \tag{5}$$

where $D_{i,j}^{\mathrm{Corr}}(\mathbf{x})$ denotes the confidence for class j of $D_i^{\mathrm{Corr}}(\mathbf{x})$ and $Grad_j(\mathbf{x})$ denotes the final predicted confidence of the Grading algorithm for class j.

The above formulation of the Grading algorithm does exactly the same thing as the algorithm described in Section 3.2. However, it provides a convenient framework for enhancements to this algorithm. As mentioned above, the main problem is how to handle situations in which none of the classifiers (are estimated to) provide the correct classification. Within the Active Grading framework introduced in this paper, we will propose an enhanced correction operation with

respect to the one used in the standard Grading algorithm. When a classifier is estimated to be incorrect by its corresponding Grading classifier, we propose to effectively modify the output of the classifier in such a way that another class is predicted than the one which was initially predicted by the classifier. More formally, we propose to change (3) into the following equation:

$$\forall i : D_i^{\text{Corr}}(\mathbf{x}) = \begin{cases} G_i(\mathbf{x}) D_i^{\text{Cr}}(\mathbf{x}), & \text{if } G_i(\mathbf{x}) > 0.5; \\ \left[1 - G_i(\mathbf{x})\right] \left[1 - D_i^{\text{Cr}}(\mathbf{x})\right], & \text{else.} \end{cases} \tag{6}$$

Note that although at first glance this might seem to be a small modification, it has significant consequences. In the case that none of the classifiers is estimated to be correct, the standard Grading algorithm uses the (weighted) outputs of all of the classifiers without changing their "winning" classes. This will most likely provide incorrect classifications in this case. The Active Grading algorithm, however, effectively changes the "winning" classes the classifiers predict. By doing so, it actively uses the information provided by the Grading classifiers and modifies the classifier outputs accordingly. To the authors' knowledge, this is the first ensemble algorithm which changes the classifier outputs in such a way.

A second modification to the standard Grading algorithm is motivated by the idea that the corrected classifier outputs can be used, regardless whether the other classifiers are estimated to be correct or not. As the classifiers' outputs are corrected when the initial prediction of the classifiers is estimated to be incorrect, all classifiers can contribute valuable information to the ensemble. This can be very easily incorporated into the framework by using the following equation instead of (4): $\forall i : I_i(\mathbf{x}) = 1$. Intuitively, we estimate whether the classifiers are correct; if they are then their predictions are used, if they are not then their predictions are modified and these modified predictions are used.

4 Experimental Results

As discussed in Section 2, the proposed architecture for teaching the quality inspection to the system by multiple human quality control operators clearly has many advantages. By only taking into account the predictions of the operators' classifiers, we want to model the decisions of the supervisor. Of course, the accuracy of this system should not drop compared to a system in which one single classifier would only be trained on the data provided by the supervisor.

For the experiments in this paper a data set obtained from an industrial visual inspection system used for checking the quality of the labels printed on CDs is used. This data set contains 1534 samples and was independently labelled by 4 different operators and their supervisor into 2 classes: "good" and "bad". As discussed in Section 2, from the images obtained from the vision system 74 generic features (e.g. the number of objects detected, the area of the largest object, the maximum brightness of an object, etc.), describing each of the images, are derived [5]. Analysis of these data sets has shown that the operators make about the same decisions for the entire feature space, while in some part of the feature space (about 20% of the data) the supervisor makes different decisions

Table 1. Mean accuracy (in %) of CART classifiers for the CD data sets: the first row shows the evaluation of each of these classifiers for the operator's own (test) data; the second row shows the evaluation of these classifiers for the supervisor's data

Evaluation data provided by	Training data provided by				
	Operator01	Operator02	Operator03	Operator04	Supervisor
Same as training	94.77	96.60	97.39	96.01	94.38
Supervisor	75.16	70.00	73.86	73.66	94.38

Table 2. Mean accuracy (in %) of the different ensemble methods when combining the outputs of the classifiers, trained by the different operators, to model the supervisor's decisions for the CD data sets

Classifier fusion methods						Grading methods		
Vote	AC	FI	DT	DS	DDS	Grad	AGrad-N	AGrad-S
72.81	78.56	73.27	78.24	73.66	73.66	79.54	93.01	94.77

than *all* of the operators. Interestingly, this is about the error rate of human visual quality inspection reported elsewhere [3].

For the data sets provided by each of the operators a *CART* decision tree classifier [11] was trained. The accuracy of these classifiers was determined for all 5 data sets (using 10-fold cross-validation), the results of which can be found in the first row of Table 1. From these results it is clear that the classifiers are well trained for the data provided to them (ranging from 94.38% to 97.39%). However, when the operators' classifiers are evaluated on the data provided by the supervisor the accuracy drops significantly, ranging from 70% to 75.16% (the first 4 values in the second row of the Table 1), which are much lower than the 94.38% of the classifier trained by the supervisor himself. It is, however, the ensemble's job to combine the first four classifiers and to obtain an accuracy comparable to a classifier trained specifically on the supervisor's data.

The ensemble methods are trained on the same training data as the classifiers, so the outputs of the classifiers for their own training data are used as input to the ensembles. To combine the decisions of these classifiers the standard Grading [4] (*Grad*) and two variants of the proposed Active Grading approach are evaluated (Active Grading applied when none of the classifiers is estimated to be correct (*AGrad-N*) and applied for each of the classifiers separately when estimated to be incorrect (*AGrad-S*) – as detailed in Section 3.3). The *CART* decision tree classifier [11] was also used as Grading classifier. For comparison, a number of the most effective classifier fusion techniques (for detailed discussions, see e.g. [8,9]) were evaluated as well: Voting (*Vote*) [12], a number of simple Algebraic Connectives (*AC*) such as the Maximum, Minimum, Product, Mean and Median rules [13], Fuzzy Integral (*FI*) [14], Decision Templates (*DT*) [15], Dempster-Shafer combination (*DS*) [16] and its extension Discounted Dempster-Shafer

combination (*DDS*) [17]. The results of these algorithms can be found in Table 2 (only the result of the best Algebraic Connective, the Product rule, is shown).

From the results in Table 2 it can be seen clearly that the classifier fusion algorithms do not perform well for this task. Their accuracies lie in the range of 72.81% to 78.56%. The standard Grading algorithm performs already slightly better with 79.54%, but this level of accuracy still is not enough for industrial applications. In contrast, the two proposed Active Grading approaches perform very well. *AGrad-N* and *AGrad-S* achieve accuracies of 93.01% and 94.77%, respectively. This means an improvement of 13.47% and 15.23% compared to the best of the other ensemble methods which were evaluated. It should be noted that the result of AGrad-S is even slightly better than a classifier trained specifically on the supervisor's data (see Table 1). This confirms that the decisions of the supervisor can effectively be modelled by the ensemble, if the classifiers trained by the different operators are combined in an appropriate way.

The reason the classifier fusion methods are not performing very well for this kind of problem is that they are trained on the *classifier outputs*, rather than on the original feature space. As for this application the majority of the data is correctly classified by the classifiers and the other part is *systematically* misclassified by each of the classifiers within the ensemble (trained by the operators), these methods will not be able to increase the performance of the system very much. In order to do this, information about the original feature space is required, which is used by the Grading methods. The standard Grading method can detect which of the classifiers will provide an incorrect prediction, but does not contain any mechanism to use this information in a constructive way. This is exactly what the Active Grading methods do: they actively modify the classifier outputs, so that they become useful for the combination process.

5 Conclusion

In this paper a framework for dealing with the reality that multiple human operators might be training a visual quality inspection system is proposed, in which the operators train their own personal classifier, the predictions of which are combined by an ensemble method. The operators' decisions are however not perfect and will systematically contradict their supervisor's decisions. This poses a problem for most ensemble techniques, which cannot cope well with the situation in which none of its member classifiers outputs the correct prediction. Therefore, the *Grading* ensemble technique is extended to a more general *Active Grading* framework, in which some extensions to the standard Grading method are proposed which make it able to learn in these circumstances. This technique is evaluated on data obtained from a real-world industrial system for visual quality inspection of the printing of labels on CDs, which was labelled independently by four different human operators and their supervisor, and compared to the standard Grading algorithm and a range of other classifier fusion algorithms. The experimental results show a performance boost of over 15% compared to the best other ensemble method.

Acknowledgments. This work was partly supported by the European Commission (project Contract No. STRP016429, acronym DynaVis). This publication reflects only the authors' view.

References

1. Castillo, E., Alvarez, E.: Expert Systems: Uncertainty and Learning. Springer, New York (2007)
2. Malamas, E., Petrakis, E., Zervakis, M., Petit, L., Legat, J.D.: A survey on industrial vision systems, applications and tools. Image and Vision Computing 21 (2003)
3. Juran, J., Gryna, F.: Juran's Quality Control Handbook, 4th edn. McGraw-Hill, New York (1988)
4. Seewald, A., Fürnkranz, J.: An evaluation of grading classifiers. In: Hoffmann, F., Adams, N., Fisher, D., Guimarães, G., Hand, D.J. (eds.) IDA 2001. LNCS, vol. 2189, pp. 115–124. Springer, Heidelberg (2001)
5. Sannen, D., Nuttin, M., Smith, J., Tahir, M.A., Caleb-Solly, P., Lughofer, E., Eitzinger, C.: An on-line interactive self-adaptive image classification framework. In: Gasteratos, A., Vincze, M., Tsotsos, J. (eds.) ICVS 2008. LNCS, vol. 5008, pp. 171–180. Springer, Heidelberg (2008)
6. Govindaraju, M., Pennathur, A., Mital, A.: Quality improvement in manufacturing through human performance enhancement. Integrated Manufacturing Systems 12(5) (2001)
7. Duda, R., Hart, P., Stork, D.: Pattern Classification, 2nd edn. John Wiley & Sons, New York (2000)
8. Kuncheva, L.: Combining Pattern Classifiers: Methods and Algorithms. Wiley, Chichester (2004)
9. Polikar, R.: Ensemble based systems in decision making. IEEE Circuits and Systems Magazine 6(3), 21–45 (2006)
10. Brown, G., Wyatt, J., Harris, R., Yao, X.: Diversity creation methods: A survey and categorisation. Information Fusion 6(1), 5–20 (2005)
11. Breiman, L., Friedman, J., Olshen, R., Stone, C.: Classification and Regression Trees. Wadsworth International Group, Belmont (1984)
12. Kuncheva, L., Whitaker, C., Shipp, C., Duin, R.: Limits on the majority vote accuracy in classifier fusion. Pattern Analysis & Applications 6(1), 22–31 (2003)
13. Kittler, J., Hatef, M., Duin, R., Matas, J.: On combining classifiers. IEEE Transactions on Pattern Analysis and Machine Intelligence 20(3), 226–239 (1998)
14. Cho, S., Kim, J.: Combining multiple neural networks by fuzzy integral for robust classification. IEEE Transactions on Systems, Man, and Cybernetics 25(2), 380–384 (1995)
15. Kuncheva, L., Bezdek, J., Duin, R.: Decision templates for multiple classifier fusion: An experimental comparison. Pattern Recognition 34(2), 299–314 (2001)
16. Rogova, G.: Combining the results of several neural network classifiers. Neural Networks 7(5), 777–781 (1994)
17. Sannen, D., Van Brussel, H., Nuttin, M.: Classifier fusion using Discounted Dempster-Shafer combination. In: Poster Proceedings of the 5th International Conference on Machine Learning and Data Mining, pp. 216–230 (2007)