A Decomposition Machine-learning Strategy for Automated Fruit Grading

Teo Susnjak[1], Andre Barczak[2] and Napoleon Reyes[2]

Abstract—**Automated grading of fruit is an important industrial task that is expanding rapidly in its uptake. Machine learning-based techniques are increasingly being applied to this domain in order to formulate effective solutions for complex classification tasks. The inherent variability in the visual appearance of fruit and its quality-determining features, contributes to it often being a challenging classification task with much potential for improving the predictive accuracies for many fruit varieties. Additionally, the usability of many sophisticated machine learning algorithms in the form of tunable parameters and interpretable outputs is low, thus presenting a real barrier for the uninitiated. We address these problems by decomposing the overall machine learning task into subproblems. We propose combining a more sophisticated boosting algorithm (AdaBoost.ECC) with low interpretability for the learning of fruit-surface characteristics, whose outputs can then be combined with rule induction algorithms (RIPPER and FURIA) that learns the overall fruit grading rules with outputs of high interpretability for the operators to both review and revise. Our initial experiments considered four fruit datasets. We compared the results of our approach with that from a commercial system using manual calibration of the fruit grading parameters and found that our strategy can improve the accuracy over the current industry methods while providing high usability and interpretability of outputs.**

Index Terms—**boosting, classification problem decomposition, rule-based induction, machine learning, fruit sorting.**

I. Introduction

THE combination of computer vision (CV) and machine learning (ML) solutions to the industrial problem of automating fresh produce sorting and grading is currently receiving considerable attention. As witnessed from the surveys of the relevant literature [1]–[4], this large industrial sector is now becoming one of the most active application domains for combined computer vision and ML-based solutions.

Automated inspection is important since it provides a more objective and thus consistent grading of fresh produce over manual inspection [1]. Fruit quality, for example, is commonly determined based on the extracted features representing the size, shape, color and the presence of blemishes and foreign materials [1]. The ability of manual inspection to deliver accurate grading diminishes with the increase in the number of factors that have to be considered [4], which raises the need for consistent and objective grading. Automatic grading is also more efficient since it increases the volume of produce that can be inspected, thus elevating productivity. Though the financial benefits of lowering the labor costs are certainly one driving factor towards automated grading, in some cases it is also crucial since certain fruit varieties are seasonal and grow in isolated regions where it is difficult to secure the required labor force.

Examples of recent fruit grading research involving ML and CV include: [5] citrus fruit sorting based on rottenness caused by fungi by way of hyper-spectral imaging and the use of neural networks and CART ML algorithms, [6] blueberry sorting and detection of foreign materials using near infrared (NIR) spectral imaging with linear discriminant analysis (LDA), [7] bruise detection on pears from hyper-spectral images using the Mahalanobis, Euclidean distance as well as maximum likelihood classification, [8] apple grading by classifying blemishes using the *k*-mean clustering and quadratic discriminant analysis, [9] mango grading based on maturity levels using Gaussian mixture models (GMM), [10] investigations into raisin grading comparing SVMs, ANN, Bayesian Networks, decision trees and application of SVMs to automatic grading of Chinese jujube fruit [11]. Notable solutions using hyper-spectral imaging without ML techniques have been applied to detecting citrus canker lesions on grapefruit [12] with later modifications in [13] to accomplish the same task but in real-time using only two-band spectral imaging and pomegranate aril classification [14].

Despite the advantages associated with the automated sorting and grading of fruit using CV and ML techniques, there are two obstacles preventing a greater proliferation of this technology in this industry: (1) insufficient accuracy for certain fruit varieties and (2) the usability of the ML components in classification. Automated inspection removes the subjectivity and the inconsistency in grading associated with manual inspection; nevertheless, this does not always translate to required accuracy rates in the packing houses for a number of different fruit varieties [15], [16]. The evidence for this still being an open problem in a number of areas is found in the large volume of ongoing research. Naive CV techniques like segmentation and color thresholding are sometimes sufficient to accurately determine the quality of fruit as the recent survey [3] points out. In many cases though, the critical determinant of fruit quality is the presence of multiple types of blemishes, fruit features and foreign materials, which render simplistic approaches ineffective [8] and require the usage of ML techniques. ML is perfectly suited to providing solutions for this task; however, given that the standards of what constitutes a certain type of a blemish and its degree of severity are non-uniform across different geographic locations and may even undergo re-adjustments within the same location from one crop to the next [16], the one-size-fits-all classifiers trained by ML experts off-site are not well suited and contribute to low grading accuracies. In addition, unforeseen environmental conditions produce

[1] Teo Susnjak is with the School of Engineering and Advanced Technology at Massey University Albany, New Zealand. Phone +64-9-4140800 ext.43146 T.Susnjak@massey.ac.nz
[2] Andre Barczak (A.L.Barczak@massey.ac.nz) and Napoleon Reyes (N.H.Reyes@massey.ac.nz) are with the Institute of Natural and Mathematical Sciences at Massey University Albany, New Zealand.

ISBN: 978-988-19253-1-2
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

defects like hail or frost damage for which the response in providing new classifiers must be immediate and therefore requires on-site re-training of the classifiers. The issue then becomes, which types of ML algorithms should be employed by non-ML experts for these types of problems? This question has to be carefully considered since using more complex algorithms are tempting; however, they usually come with more tunable parameters that need to be set appropriately, which makes them harder to use for the non-initiated and posses internals which are oftentimes more opaque [17].

We propose a generalizable ML solution to the challenges of fruit grading and demonstrate an appropriate strategy of applying different families of ML techniques to address the real-world industry requirements. Our solution decomposes the classification task into multiple phases in order to address the problem of maintaining classification accuracy and adaptability, as well as the usability of ML and the interpretability of their outputs. We devised a classification architecture which employs a sophisticated boosting algorithm (AdaBoost.ECC [18]) for learning blemish and surface-type features at the initial layer. The outputs of this classifier subsequently act as inputs to the next classification layer, which are combined with the global fruit surface features like colour, size and shape. The second layer is represented by a state-of-the-art rule-induction learning algorithm. Though rule-based algorithms are often not the most accurate inducers [19], they provide the advantages of high usability and the interpretability of outputs [20]. For this, we experiment with both the RIPPER [21] and FURIA [20] algorithms for generating the final fruit grading classifiers.

The novelty of our contribution lies in the unique combination of different and as yet unexplored ML techniques for this problem domain. With this we are answering calls [2] to discover new ways of combining ML techniques for addressing the classification challenges in this field. We used a commercial fruit sorting machine to extract images and features. We demonstrate the effectiveness of our proposed method by comparing the accuracies achieved by the commercial machine to sort several fruit varieties using optimal settings derived manually by a domain expert, against the accuracies of the combination of boosting for surface feature classification and RIPPER and FURIA for the overall fruit grading.

To the best of our knowledge, there are only two instances in literature recording the usage of boosting algorithms in relation to the domain of fruit grading. [16] used a boosting algorithm called RealBoost [22] to perform a pixel-wise classification of potato surfaces in order to detect blemished regions, whereas [23] experimented using several ML algorithms, of which AdaBoost [22] is one of them to explore the accuracies of differentiating stems from calyxes on apples. Although both employed boosting algorithms on problem domains concerning fresh produce, neither addressed the overall grading aspect of each individual fruit. To our best efforts, no instances of research or industrial application of rule-based induction for the purpose of fruit grading has been uncovered by the authors.

The remainder of the paper is structured as follows: Section II describes the rationale behind the strategy of decomposing a given problem into multiple learning sub-tasks. Section III provides a brief overview of the proposed machine learning algorithms for usage on this type of a problem domain, while Section IV and Section V present the methodology and the experimental results respectively, before the concluding remarks in the succeeding section.

II. Machine Learning Problem Decomposition

According to [24], *decomposition* generally describes the process of breaking down a given task or a system into smaller units. The idea is not new to machine learning and can be traced back to Samuel [25] in the 1960s with his decomposition approach application to the checkers playing programs. The motivation behind decomposition is to reduce a complex problem into more manageable sub-tasks, that can then be combined in order to solve the initial problem. The definition of a such a goal-subgoal hierarchy can serve as a powerful and effective approach to reformulating a classification problem. Although the reduction in processing complexity might seem as a primary driver for employing this strategy, research indicates that decomposing a problem can also improve the classification accuracy of existing approaches [26]. Additional advantages inherent within the decomposition strategy are the increase in the comprehensibility of the original problem, the maintenance of simpler classification models as well as the flexibility that enables the usage of different types of inducers on each of the sub-problems [27].

The complexity of a learning task often refers to it comprising of high dimensionality (features) data. The challenge of performing machine learning in high dimensionality domains is a well understood problem. The principal difficulty arises in the fact that as the dimensionality (or the number of features) of a learning problem increase, a fixed-sized training dataset covers an ever decreasing fraction of the possible sample input space. With the growth in the sample dimensionality, the generalizability on such a domain becomes exponentially more difficult [17]. For example, even when presented with a trivial problem of learning a Boolean function B, where $B = \{0,1\}$ and dimensionality $d = 50$, the total number of samples representing the input space becomes as large as 2^{50}. If the problem domain lends itself, then one possible solution is to explicitly decompose the learning task into learning sub-task h_1 and h_2, each comprising of d_1 and d_2 dimensions where $d_1 + d_2 = d$. In this case the size of the total input space for learning a given Boolean function would be considerably reduced to $2^{d_1} + 2^{d_2}$.

In a domain where it is costly in terms of time resources to gather large datasets of samples, the importance of lowering the dimensionality of the learning problem becomes even more acute. The domain of fruit sorting is one such area, since each image sample must be carefully inspected and correctly labeled with the appropriate class. The learning problem for fruit classification in this case lends itself well to this form of sub-tasking which can be reformulated into a hierarchical decomposition, where the outputs of one sub-problem become the inputs to another. In this instance, features relevant to blemish classification are extracted and used only for the learning of the blemish classifiers, whose output becomes the new input feature for the induction of the global fruit grading classifier.

III. Proposed Two-stage Classification Strategy

We propose decomposing the fruit grading problem into two classification tasks: (1) the training and classification of fruit surface features and blemishes, (2) the training and grading of the overall fruit based on the combination of the blemish-classification output and the general fruit color/appearance characteristics. We propose using a more sophisticated ensemble-based inducer (AdaBoost.ECC seen in Algorithm 1) with low interpretability for learning the blemish classifier, while employing less powerful rule-induction algorithms (RIPPER and FURIA) to generate final fruit grading classification rules with high interpretability.

Algorithm 1: AdaBoost.ECC

Given: $(x_1, y_1), ..., (x_m, y_m)$ where $x_i \in X, y_i \in Y$ to make uniform over all incorrect labels
Output: Hypothesis
$H_{final}(x) = \arg\max_{\ell \in Y} \sum_{t=1}^{T} g_t(x)\mu_t(\ell)$
Initialize $\tilde{D}_1(i,\ell) = [\![\ell \neq y_i]\!]/(m(k-1))$ where m and k are the number of samples and class labels respectively and $[\![\pi]\!]$ evaluates to 1 if proposition π holds, otherwise 0.
for t = 1 **to** T **do**
- Compute coloring $\mu : Y \to \{-1, 1\}$
- Let $U_t = \sum_{i=1}^{m} \sum_{\ell \in Y} \tilde{D}_t(i,\ell)[\![\mu_t(y_i) \neq \mu_t(\ell)]\!]$
- Let $D_i = \frac{1}{U_t} \cdot \sum_{\ell \in Y} \tilde{D}_t(i,\ell)[\![\mu_t(y_i) \neq \mu_t(\ell)]\!]$
- Train weak learner on examples $(x_1, \mu_t(y_1)), ..., (x_m, \mu_t(y_m))$ weighted according to D_t
- Get weak hypothesis $h_t : X \to \{-1, 1\}$
- Compute the weight of positive and negative votes α_t and β_t respectively
- Define: $g_t(x) = \begin{cases} \alpha_t & \text{if } h_t(x) = 1 \\ \beta_t & \text{if } h_t(x) = -1 \end{cases}$
- Update $\tilde{D}_{t+1}(i,\ell) = \frac{1}{\tilde{Z}_t} \cdot \tilde{D}_t(i,\ell) \exp\{(g_t(x_i)\mu_t(\ell) - g_t(x_i)\mu_t(y_i)) \cdot \frac{1}{2}\}$ where $\tilde{Z}_t$ is the normalization factor so that $\tilde{D}_{t+1}$ will sum to 1.

The combination of ensemble-based machine learning methods with boosting and weak underlying models, have recently experienced a widespread use due to their effectiveness at addressing many challenging classification problems. Following the success of the binary-class AdaBoost [28] algorithm, [18] proposed AdaBoost.ECC (error-correcting codes) in order to overcome the limitations of its predecessor and to thus extend boosting to multiclass scenarios. AdaBoost.ECC elegantly merges error correcting output coding (ECOC) principles with boosting. The algorithm repeatedly calls a weak learner (decision stump) on samples with variable weights, for a predetermined T rounds. A *coloring* function μ is defined which decomposes the multiclass problem into a binary one by re-labeling sample class-memberships. After each round, the *coloring* function μ then becomes the vehicle for iteratively generating the columns of the coding matrix which is used by ECOC methods for the resolution of predictions. An additional distribution $\tilde{D}$ is maintained to maximize the error correcting ability of each column in the coding matrix. The evaluation of final classifier H, on a sample x is computed as being the class label l, which receives the highest weighted vote from all class labels returned by each weak classifier $h_t(x)$.

Rule-based learning is one of the oldest and well studied paradigms within machine learning [19]. Its distinguishing feature is its high applicability to domains where the comprehensibility of the induced model is of prime importance, and where manual revision and adaptation of the induced models is necessary. RIPPER (*Repeated Incremental Pruning* to *Produce Error Reduction*) is a state-of-the-art algorithm in this genre, and recently FURIA (*Fuzzy Unordered Rule Induction Algorithm*) has been proposed as its extension and an improvement over the original. We propose conducting two sets of experiments whereby the boosting algorithm is combined alternatively with RIPPER and FURIA for generating fruit grading rules.

RIPPER constructs rules in a greedy manner. The rules consist of conjunctions of predicates and a consequent part which designates a class to which the covered instances of that rule are assigned to. RIPPER learns rules one class label at a time, beginning with the smallest class in terms of the number of samples. Samples are removed from the training set incrementally with each subsequent antecedent that covers them. The training set is divided into a growing and a pruning set that signify two phases of the rule induction process. The growing phase specializes the rule by inducing and appending each new antecedent according to the *information gain* criterion. This is then followed by the pruning phase which removes the antecedents it considers to have overfitted the data according to its rule-value metric. Both the growing and the pruning phases are repeated until all the samples of the given class are covered or until the complexity of the rules exceeds the total description length metric. Following this, a sophisticated optimization phase is executed involving the re-running of the growing and pruning steps, and replacing existing antecedents with alternative and newly generated ones.

Whereas RIPPER produces hard and inflexible decision boundaries between different classes, FURIA proposes introducing a softer transition between class boundaries through *fuzzy rules*. It also departs from its predecessor by inducing rules for each class using the one-versus-all method which frees up the classifier from a strict order in which it must be evaluated. Arguably, this increases the comprehensibility as well as the knowledge discovery quality of its rules since they no longer implicitly embody the negated conditions of the previous rules [20]. This however introduces a problem during classification of unseen samples, where a sample may not satisfy any of the generated rules. FURIA addresses this by devising a rule stretching mechanism that generalizes the rules further to ensure a maximum coverage.

A more thorough exposition of the RIPPER and FURIA algorithms can be found in [20], [21] respectively.

IV. Methodology

We used four datasets in our experiments comprising three fruit varieties. Two datasets consisted of oranges, and the remainder, of plums and gala apples. Table I outlines the details of each of the datasets.

The datasets themselves were obtained from packing houses from different locations around the world. The equip-

Fig. 1: Example of the commercial machine used to capture images and extract features for the training of classifiers.

ment and software used to capture the images and extract the features into datasets originated from a commercial fruit sorting equipment, manufactured by Compac Sorting Limited[1]. The key components of the equipment associated with the capture of the images are: (1) the conveyor belt, upon which reside the individual fruit cup holders which rotate the fruit on a single axis and making its entire surface visible, (2) the computer vision cabinet, which resides on top of the conveyor belt and contains the necessary lighting for the multiple cameras (Fig. 1). The cameras are capable of capturing and synchronizing the rotating images from both the visible and infra-red spectra (Fig. 2a-b).

Each dataset was randomly split into halves representing the training and test datasets. Following best practices [29], the splits were stratified in order to ensure an equal proportion of samples from each class in both datasets. For each fruit variety, the images from the training dataset were used for calibrating the computer vision components. This entailed manually selecting pixels representing dominant hues in order to achieve color segmentation of regions signifying the quality of a given fruit, as well as the regions that identify the background (Fig. 2c). The software then classified the remaining pixels into the selected colors based on similarity measures. For each fruit variety, key blemish types that determine the grading quality of each specimen were identified. Thereupon, within the segmented fruit surface areas, further regions of interest (ROI) were manually identified as representing these surface features (Fig. 2d). We term them generically as *blobs*, and though they signify defects of varying type and severity, they can equally represent natural surface features like stems or calyxes depending on the fruit variety. These ROI were identified by manually selecting pixels that were most representative of both the blob features and normal fruit-surface areas which the software once again used to categorize the remaining pixels based on similarity measures. Finally, using the same segmentation technique from previous steps, we proceeded to segment the blob ROI themselves into dominant colors which would then serve as input features for the blob classifiers (Fig. 2e).

The blob feature vectors were then extracted into training datasets and manually labeled with the correct class mem-

[1]Compac Sorting is one of the world industry leaders in automated fruit sorting with its headquarters in Auckland, New Zealand.

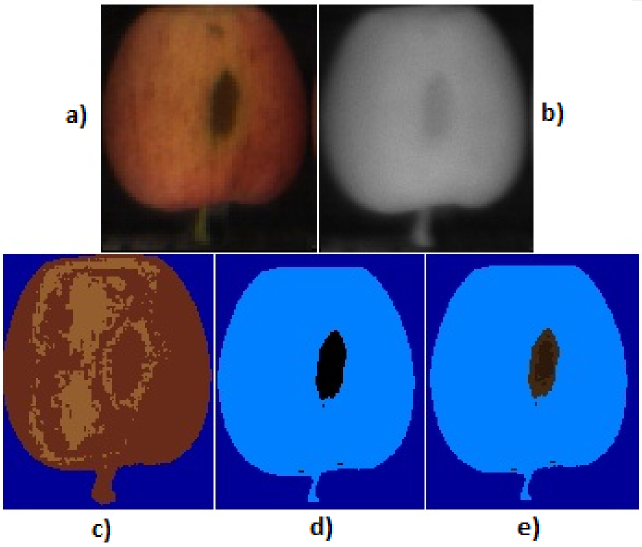

Fig. 2: Example of a Gala apple with a blemish, imaged under the (a) visible and (b) infra-red spectra. Example of the manual process of selecting dominant fruit colours and that of the conveyor belt in order to achieve segmentation (c). Identification of the blemish region within the segmented image (d) and the manual identification of the dominant colours representing the blemish region (e).

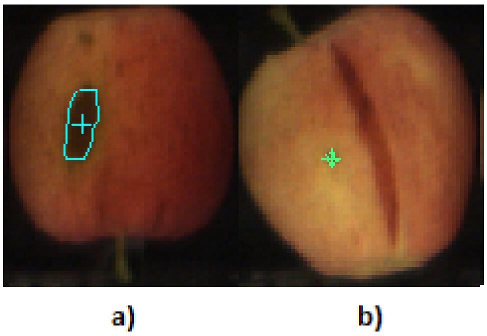

Fig. 3: Example of blob classification using the trained AdaBoost.ECC classifier Gala apples with a blemishes. Classification of (a) a severe blemish type, (b) mild blemish type.

bership. 10-20 blob samples were selected for each blob class. Using AdaBoost.ECC, we trained blob classifiers for each fruit variety with the ensemble size set to 150. We used 5-fold-cross validation to generate blob classifiers on the training dataset in order to inspect the generalizability of the problem first. Provided that the selected blob dataset was of good quality, then the entire blob training set was used to generate the final classifier. These classifiers were subsequently applied to both the fruit training and test set images (Fig. 3) in order to extract blob feature vectors that would form the training and test sets of the rule-based classifiers. The blob feature vectors comprised (1) the sum of pixels representing the classified blob type (seen in Fig. 2d), (2) the sum of pixels for each of the colors that represent the particular blob class (seen in Fig. 2e). These feature vectors were combined with additional features extracted from the datasets representing global characteristics of each fruit (eg. surface area covered by each of the selected hues in Fig. 2c).

The second phase involved training the rule-based classi-

ISBN: 978-988-19253-1-2
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

TABLE I: Dataset characteristics

	Dataset samples			Fruit dataset attributes		Blob dataset attributes	
Fruit variety	Training set	Test set	Infra-red images	Classes	Features	Classes	Features
Gala apples	78	79	yes	5	9	3	31
Plums	73	71	yes	3	8	5	31
Oranges	63	60	no	5	5	3	31
Navel-split oranges	61	62	no	3	5	5	31

fiers on the training sets and testing the generalizability of the outputted classifiers on to the test datasets. The RIPPER and FURIA algorithms were used to train these classifiers. For each algorithm type, five classifiers with different random number seeds were trained. Average classification accuracy and geometric mean measures could then be calculated together with their standard deviations over the five runs. The accuracy of these classifiers was then compared to the accuracy attained by manually designed grading configurations by domain experts. The domain experts followed the same procedure of designing the grading configurations solely from the data available to them from the training datasets, whose generalizability was subsequently evaluated against the test datasets.

We implemented our own version of AdaBoost.ECC in C++ and used the WEKA [29] machine learning toolkit for training RIPPER and FURIA classifiers.

V. Results

Given that the four test datasets consisted of a skewed number of samples for each class, we employed both the total accuracy and the geometric mean as measures of generalizability. Presenting only the accuracy has been shown to be inadequate and often misleading on class-imbalanced datasets [30], whereas the geometric mean can be a more meaningful measure of accuracy for biased class-distributions. [31] demonstrated how the geometric mean of recall values of each class i of a total of k classes can be applied to the multiclass scenario by being calculated as:

$$Geometric\ mean = \left(\prod_{i=1}^{k} Recall_i\right)^{\frac{1}{k}} \qquad (1)$$

which yields a single value from $0-1$ that presents a balanced performance of a classifier across all classes.

Table II shows the total accuracy of all the methods on datasets. On three of the four datasets, the proposed methods outperformed the manual strategy of calibrating thresholds. RIPPER outperformed FURIA on two datasets, whereas FURIA was a clear winner on the Oranges dataset. The overall accuracy measures are summarized in the form of mean ranks showing that RIPPER was the best performing algorithm on these problem sets, while the manual method was least successful.

The geometric mean measures are listed in Table III. From this perspective, the picture changes slightly. In the overall placings, the manual approach moved up to second place above FURIA. However, the manual approach also demonstrated an inconsistency in its performance which is highlighted on the Oranges dataset. On this dataset, the manual approach failed to correctly classify samples of one particular class and was thus unable to generate a valid

TABLE II: Accuracy results as a percentage, for the three methods across all datasets.

Fruit Variety	Manual	RIPPER	FURIA
Gala apples	**81**	80.5 ±0.7	74.9 ±0.6
Plums	52	**54.4 ±4**	53.5 ±3
Oranges	53	67 ±3	**71 ±2**
Navel-split oranges	50	**56 ±2**	55 ±2
Mean ranks	2.5	**1.5**	2

geometric mean figure. On the other hand, it significantly outperformed both RIPPER an FURIA on the Navel-split oranges dataset. The latter two algorithms scored poorly on this dataset since they both achieved low hit rates for one class that consequently reduced their overall scores.

TABLE III: Geometric mean results for the three methods across all datasets.

Fruit Variety	Manual	RIPPER	FURIA
Gala apples	**0.58**	**0.58 ±0.004**	0.54 ±0.01
Plums	0.35	**0.40 ±0.004**	0.34 ±0.05
Oranges	-	0.63 ±0.04	**0.65 ±0.04**
Navel-split oranges	**0.64**	0.30 ±0.03	0.30 ±0.03
Mean ranks	1.9	**1.8**	2.4

An example of some detailed classification results from the Oranges dataset is shown in Table IV in the form of confusion matrices. In Table IVa, the difficulty that the manual method had in classifying samples of class E can be seen. All samples of this class were misclassified as belonging to class A. RIPPER and FURIA on the other hand, scored highly in terms of accuracy for this class.

Both the accuracy and geometric mean measures across the four datasets indicate that the proposed decomposition strategy using the AdaBoost.ECC algorithm for learning surface features, whose output is combined with the FURIA and RIPPER algorithms for learning fruit grading rules is effective at matching and exceeding the performance of manually calibrated grading thresholds by domain experts.

The following analysis gives a typical example using the Oranges dataset, of the types of rules that were generated from all three methods in order to assess their interpretability. The manual method defined four rules for five classes:

ISBN: 978-988-19253-1-2
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

TABLE IV: Example of the confusion matrices for all algorithms on the Oranges dataset.

(a) Manual.

A	B	C	D	E	Classified as
9	1	0	0	0	A
4	**7**	0	1	0	B
1	0	**11**	0	0	C
1	6	0	**5**	0	D
10	4	0	0	**0**	E

(b) RIPPER.

A	B	C	D	E	Classified as
6	1	0	3	0	A
1	**7**	0	2	2	B
0	0	**11**	1	0	C
0	2	0	**7**	3	D
0	2	0	1	**11**	E

(c) FURIA.

A	B	C	D	E	Classified as
9	1	0	0	0	A
2	**8**	0	0	2	B
0	1	**11**	0	0	C
1	3	0	**4**	4	D
0	2	0	1	**11**	E

```
IF (CombinedBlack+Dark+LightDefectPixelArea ≤ 20) and (BlackDefectPixelArea ≤ 40)
    and (DefectStemPixelArea = 0) and (SevereBlemishPixelArea = 0)
    and (MildBlemishPixelArea ≤ 60)
    and (CombinedOrange+GreenPixelArea ≤ 65) THEN grade = A
ELSE IF (CombinedBlack+Dark+LightDefectPixelArea ≤ 90)
    and (BlackDefectPixelArea ≤ 40)
    and (DefectStemPixelArea = 0) and (SevereBlemishPixelArea = 0)
    and (MildBlemishPixelArea ≤ 60)
    and (CombinedOrange+GreenPixelArea ≤ 65) THEN grade = B
ELSE IF (CombinedBlack+Dark+LightDefectPixelArea ≤ 90)
    and (BlackDefectPixelArea ≤ 40)
    and (DefectStemPixelArea = 0) and (SevereBlemishPixelArea = 0)
    and (MildBlemishPixelArea ≤ 60) THEN grade = C
OTHERWISE grade = D
```

At classification runtime, the above rules function like RIPPER rules. An effective rule for the final class E could not be induced using the manual method. Though there are considerable overlaps between the rules for each of the classes, the entire set of rules comprises 18 antecedents.

Below is the example of rules that were generated by the RIPPER algorithm:

```
IF (OrangePixelArea ≥ 27.51) and (MildBlemishPixelArea ≥ 20.92) THEN grade = B
ELSE IF (CombinedOrange+GreenPixelArea) ≥ 69.65) THEN grade = C
ELSE IF (OrangePixelArea ≥ 67.04) THEN grade = A
ELSE IF (OrangePixelArea ≥ 14.57) THEN grade = D
OTHERWISE grade = E
```

In this example RIPPER induced five rules and seven antecedents. This is considerably less complex, without duplications and thus more readable than the manually derived rules. The final example is that of the generation of FURIA rules. FURIA induces fuzzified rules and therefore relies on a trapezoidal function that determines the degree of membership for a given value. We represent this function and its intervals using three values. The range between the positive or negative ∞ and its adjacent real number represents the values that are fully covered. The range between the two floating point numbers is the fuzzy interval that determines the degree of membership. In addition, each rule is associated with a confidence factor (CF). The class with the highest confidence factor is assigned to a candidate sample. The FURIA rule set consists of a total of seven rules and 15 antecedents.

```
CLASS WITH MAX SUPPORT:
(OrangePixelArea in [45.07, 67.04, ∞]) and
    (MildBlemishPixelArea in [−∞, 8.54, 22.29]) THEN grade = A (CF = 0.88)
(OrangePixelArea in [26.18, 27.51, ∞]) and
    (MildBlemishPixelArea in [8.54, 20.92, ∞]) THEN grade = B (CF = 0.67)
(CombinedOrange+GreenPixelArea in [63.31, 69.65, ∞]) THEN grade = C (CF = 0.88)
(MildBlemishPixelArea in [108.38, 111.64, ∞]) THEN grade = D (CF = 0.55)
(CombinedOrange+GreenPixelArea in [−∞, 46.59, 47.27]) and
    (OrangePixelArea in [12.4, 14.57, ∞]) and
    (OrangePixelArea in [−∞, 45.94, 50.8]) and
    (CombinedOrange+GreenPixelArea in [41.93, 42.79, ∞])
    THEN grade = E (CF = 0.82)
(MildBlemishPixelArea in [40.11, 42.58, ∞]) and
    (MildBlemishPixelArea in [−∞, 47.89, 63.04]) THEN grade = D (CF = 0.71)
(OrangePixelArea in [−∞, 10.74, 14.57]) and
    (CombinedOrange+GreenPixelArea in [−∞, 63.31, 69.97])
    THEN grade = E (CF = 0.89)
```

Though the FURIA rule set gives the initial impression of being as complex as the manually derived rules, arguably this is not the case. Both the manually and the RIPPER generated rules embody within themselves the negated conditions of the previous rules. This means that in order to understand a given rule, it must be combined with the preceding rules. FURIA rule sets on the other are explicit and offer more in terms of knowledge discovery.

VI. Conclusion

We addressed the problem of automated fruit sorting using machine learning. We demonstrated a novel strategy which decomposes a classification problem into two phases for this problem domain. The first phase consists of applying a multiclass boosting algorithm AdaBoost.ECC to the subproblem of classifying surface defects. The outputs of this classifier were then inputted into a rule induction algorithm that necessarily generates highly interpretable and human readable rules. We experimented with state-of-the-art RIPPER and FURIA algorithms. Our initial experiments into this area indicated that this novel combination of algorithms has shown potential to both match and improve upon the accuracy of manually expert-calibrated machines. The generated rules are less complex than the manually derived rules, and are also more likely to contribute to knowledge discovery.

Our future research will focus on compiling both greater sample sizes and numbers of datasets from different varieties of fruit in order to conduct large-scale experiments that can yield more robust and definitive conclusions.

Acknowledgment

The authors thank to the staff at Compac Sorting Ltd. for providing the experimental datasets as well as for the usage of their software for the feature extraction components. In particular the authors wish to express their gratitude to Kurt Bagby for investing time for manually designing the grading maps for each of the fruit varieties used in this research.

References

[1] T. Brosnan and D.-W. Sun, “Improving quality inspection of food products by computer vision – a review,” *Journal of Food Engineering*, vol. 61, no. 1, pp. 3 – 16, 2004.
[2] C.-J. Du and D.-W. Sun, “Learning techniques used in computer vision for food quality evaluation: a review,” *Journal of Food Engineering*, vol. 72, no. 1, pp. 39 – 55, 2006.
[3] S. Cubero, N. Aleixos, E. Molto, J. Gmez-Sanchis, and J. Blasco, “Advances in machine vision applications for automatic inspection and quality evaluation of fruits and vegetables,” *Food and Bioprocess Technology*, vol. 4, no. 4, pp. 487–504, 2011.
[4] D. Lorente, N. Aleixos, J. Gmez-Sanchis, S. Cubero, O. Garca-Navarrete, and J. Blasco, “Recent advances and applications of hyperspectral imaging for fruit and vegetable quality assessment,” *Food and Bioprocess Technology*, vol. 5, no. 4, pp. 1121–1142, 2012.
[5] J. Gmez-Sanchis, J. D. Martn-Guerrero, E. Soria-Olivas, M. Martnez-Sober, R. Magdalena-Benedito, and J. Blasco, “Detecting rottenness caused by penicillium genus fungi in citrus fruits using machine learning techniques,” *Expert Systems with Applications*, vol. 39, no. 1, pp. 780 – 785, 2012.

[6] T. Sugiyama, J. Sugiyama, M. Tsuta, K. Fujita, M. Shibata, M. Kokawa, T. Araki, H. Nabetani, and Y. Sagara, "NIR spectral imaging with discriminant analysis for detecting foreign materials among blueberries," *Journal of Food Engineering*, vol. 101, no. 3, pp. 244 – 252, 2010.

[7] J. Zhao, Q. Ouyang, Q. Chen, and J. Wang, "Detection of bruise on pear by hyperspectral imaging sensor with different classification algorithms," *Sensor Letters*, vol. 8, no. 4, pp. 570–576, 2010.

[8] V. Leemans and M.-F. Destain, "A real-time grading method of apples based on features extracted from defects," *Journal of Food Engineering*, vol. 61, no. 1, pp. 83 – 89, 2004.

[9] C. S. Nandi, B. Tudu, and C. Koley, "An automated machine vision based system for fruit sorting and grading," in *Sensing Technology (ICST), 2012 Sixth International Conference on*. IEEE, 2012, pp. 195–200.

[10] K. Mollazade, M. Omid, and A. Arefi, "Comparing data mining classifiers for grading raisins based on visual features," *Computers and Electronics in Agriculture*, vol. 84, no. 0, pp. 124 – 131, 2012.

[11] H. Zheng, H. Lu, Y. Zheng, H. Lou, and C. Chen, "Automatic sorting of chinese jujube (zizyphus jujuba mill. cv. 'hongxing') using chlorophyll fluorescence and support vector machine," *Journal of Food Engineering*, vol. 101, no. 4, pp. 402 – 408, 2010.

[12] J. Qin, T. F. Burks, M. A. Ritenour, and W. G. Bonn, "Detection of citrus canker using hyperspectral reflectance imaging with spectral information divergence," *Journal of Food Engineering*, vol. 93, no. 2, pp. 183 – 191, 2009.

[13] J. Qin, T. F. Burks, X. Zhao, N. Niphadkar, and M. A. Ritenour, "Development of a two-band spectral imaging system for real-time citrus canker detection," *Journal of Food Engineering*, vol. 108, no. 1, pp. 87 – 93, 2012.

[14] J. Blasco, S. Cubero, J. Gmez-Sanchs, P. Mira, and E. Molt, "Development of a machine for the automatic sorting of pomegranate (punica granatum) arils based on computer vision," *Journal of Food Engineering*, vol. 90, no. 1, pp. 27 – 34, 2009.

[15] W. Huang, C. Zhang, and B. Zhang, "Identifying apple surface defects based on gabor features and svm using machine vision," in *Computer and Computing Technologies in Agriculture V*, ser. IFIP Advances in Information and Communication Technology, D. Li and Y. Chen, Eds. Springer Berlin Heidelberg, 2012, vol. 370, pp. 343–350.

[16] M. Barnes, T. Duckett, G. Cielniak, G. Stroud, and G. Harper, "Visual detection of blemishes in potatoes using minimalist boosted classifiers," *Journal of Food Engineering*, vol. 98, no. 3, pp. 339 – 346, 2010.

[17] P. Domingos, "A few useful things to know about machine learning," *Commun. ACM*, vol. 55, no. 10, pp. 78–87, Oct. 2012.

[18] V. Guruswami and A. Sahai, "Multiclass learning, boosting, and error-correcting codes," in *Proceedings of the twelfth annual conference on Computational learning theory*, ser. COLT '99. New York, NY, USA: ACM, 1999, pp. 145–155.

[19] J. Fürnkranz, D. Gamberger, and N. Lavrac, *Foundations of Rule Learning*. Springer, 2012.

[20] J. Hühn and E. Hüllermeier, "Furia: an algorithm for unordered fuzzy rule induction," *Data Mining and Knowledge Discovery*, vol. 19, no. 3, pp. 293–319, 2009.

[21] W. Cohen, "Fast effective rule induction," in *In Proceedings of the Twelfth International Conference on Machine Learning*, 1995, pp. 115–123.

[22] Y. Freund and R. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of Computer and System Sciences*, vol. 55, no. 1, pp. 119–139, 1997.

[23] D. Unay and B. Gosselin, "Stem and calyx recognition on 'jonagold' apples by pattern recognition," *Journal of Food Engineering*, vol. 78, no. 2, pp. 597 – 605, 2007.

[24] O. Maimon and L. Rokach, "Decomposition methodology for knowledge discovery and data mining," *Data Mining and Knowledge Discovery Handbook*, pp. 981–1003, 2005.

[25] A. L. Samuel, "Some studies in machine learning using the game of checkers II: Recent progress," *IBM Journal of Research and Development*, vol. 11, no. 6, pp. 601–617, 1967.

[26] A. J. C. Sharkey, G. O. Chandroth, and N. E. Sharkey, "A multi-net system for the fault diagnosis of a diesel engine," *Neural Computing & Applications*, vol. 9, no. 2, pp. 152–160, 2000.

[27] L. Rokach, "Decomposition methodology for classification tasks: a meta decomposer framework," *Pattern Analysis and Applications*, vol. 9, no. 2, pp. 257–271, 2006.

[28] Y. Freund and R. E. Schapire, "A short introduction to boosting," *Journal of Japanese Society for Artificial Intelligence*, vol. 14, no. 5, pp. 771–780, 1999.

[29] I. Witten, E. Frank, and M. Hall, *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2011.

[30] F. Provost and T. Fawcett, "Analysis and visualization of classifier performance: Comparison under imprecise class and cost distributions," in *Proceedings of the third international conference on knowledge discovery and data mining*. AAAI Press, 1997, pp. 43–48.

[31] Y. Sun, M. Kamel, and Y. Wang, "Boosting for learning multiple classes with imbalanced class distribution," in *Data Mining, 2006. ICDM '06. Sixth International Conference on*, December 2006, pp. 592 –602.