Automatic Classifier Fusion for Produce Recognition

Fabio Augusto Faria, Jefersson Alex dos Santos, Anderson Rocha and Ricardo da S. Torres
RECOD Lab
Institute of Computing
University of Campinas
Campinas, São Paulo, Brazil
{ffaria,jsantos,rocha,rtorres}@ic.unicamp.br

www.ic.unicamp.br/∼{ffaria,jsantos,rocha,rtorres}

***Abstract*—Recognizing different kinds of fruits and vegetables is a common task in supermarkets. This task, however, poses several challenges as it requires the identification of different species of a particular produce and also its variety. Usually, existing computer-based recognition approaches are not automatic and demand long-term and laborious prior training sessions. This paper presents a novel framework for classifier fusion aiming at supporting the automatic recognition of fruits and vegetables in a supermarket environment. The objective is to provide an effective mechanism for combining low-cost classifiers trained for specific classes of interest. The experiments performed demonstrate that the proposed framework yields better results than several related work found in the literature and represents a step forward automatic produce recognition in cashiers of supermarkets.**

***Keywords*-Produce Recognition; Ensemble of Classifiers; Diversity Measures;**

I. Introduction

Fruit and vegetable recognition is a recurrent task in supermarkets [1], [2]. One common application is concerned with the definition of the price of a produce, given its identification. This is a challenging problem as it deals with both different species of fruits and vegetables (e.g., apple, orange, potatoes) and many varieties of a single produce species (for example, Golden Delicious, Akane, Gala, and Fuji are different varieties of apples) [2].

Usually, existing recognition approaches are not automatic and demand long-term and laborious previous training sessions. One attempt to address that problem concerns with the use of barcodes that are assigned to packages of fruits/vegetables. A drawback of this solution relies on the lack of freedom on choosing the produce of interest. Another solution consists in using booklets containing photos of fruits/vegetables that are browsed to properly determine their price. That solution, however, poses new challenges related to the memorization and the subjectivity in the recognition process.

The automatic recognition of fruits and vegetables based on computer vision and image processing techniques represents a suitable alternative for the problem. In these methods, algorithms are used to encode visual properties (e.g., color, texture, and shape) of produce images into feature vectors, and machine learning techniques are employed to classify those fruits/vegetables considering their features. Bolle et al. [1], for example, proposed the *VeggieVision* system, the first supermarket produce recognition system that used different visual properties (e.g., color, texture, density). The authors reported a recognition rate of 95%, but considered the top four responses of the recognition system. Rocha et al. [2], in turn, proposed the use of fusion techniques that consider classifiers associated with different image descriptors for automatic produce recognition. Their fusion approach consisted in dividing the recognition task into multiple two-class problems. Good recognition rates ($\cong 97\%$) are reported, but the solution uses a fixed number of classifiers. It combines outcomes of $\binom{N}{2} = O(N^2)$ SVM classifiers, where N is the number of classes. Arivazhagan et. al. [3] also addressed the produce recognition problem by using a classifier based on a Minimum Distance Criterion. They used the same dataset released by [2], but the reported results are worse.

The target application demands real-time and high recognition rate. Usually, however, existing works fail to address both requirements at the same time. Other challenges involving automatic produce recognition using computer vision methods rely on dealing with existing different appearance variations of fruits and vegetables, as well as pose and illumination changes during image acquisition. This paper aims at diminishing the impact of such problems by presenting a novel framework for non-linear fusion of classifiers aiming at supporting the automatic recognition of fruits and vegetables. The objective is to provide an effective mechanism for combining efficiently low-cost base classifiers trained for specific classes of interest.

The novelty of the proposed work relies on the use of diversity measures to automatically assess the correlation of classifiers and then to determine the more appropriate ones to be combined. The proposed framework fits well the fruits and vegetables recognition task as it allows a continuous learning of suitable classifiers over time. The performed experiments demonstrate that the proposed framework yields better results than several related works found in the literature.

The remainder of this paper is organized as follows. Section II presents related concepts necessary to understand

this paper. Section III describes the steps of the proposed framework for fusion of classifiers and for selecting the most appropriate classifiers based on diversity measures. Section IV shows the experimental protocol we devised to validate our work while Section V discusses the results. Finally, Section VI states our conclusions and future research directions.

II. Related Concepts

The following subsections describe related concepts necessary to understand this paper.

A. Adaboost (BOOST)

The AdaBoost algorithm was proposed by Schapire [4]. It constructs a ensemble system (strong classifier) by repetitive evaluation of *weak classifiers*[1] in a series of rounds ($t = 1, \ldots N$). In this section we breafly describe the binary AdaBoost proposed in [4]. The multiclass AdaBoost [5] is a variation of this strategy.

Let A and B be the training and the validation sets ($A \cup B = T$), and let x be a sample (image). The strategy consists in keeping a set of weights $W_t(x)$ over T, where t is the current round. These weights can be interpreted as a measure of the difficulty level to classify each training/validation sample. At the beginning, all the samples have the same weight, but in each round, the weights of the misclassified samples are increased. Thus, in the next rounds the weak classifiers are forced to classify the harder samples.

For each round, the algorithm selects the best weak classifier $h_t(x)$ and computes a coefficient α_t that indicates the degree of importance of $h_t(x)$ in the final strong classifier. It is given by:

$$\alpha_t = \frac{1}{2} ln\Big(\frac{1 + r_t}{1 - r_t}\Big) \tag{1}$$

where $r_t = \sum_x cT(x)h_t(x)$.

In our implementation, the weak classifier is trained by using the training set A. The best weak classifier is selected based on the error on the validation set B. Therefore, the weights W_{t+1} are computed for both A and B sets based on the current weights W_t:

$$W_{t+1}(x) \leftarrow \frac{W_t(x) \exp\left(-\alpha_t T(x) h_t(x)\right)}{\sum_x W_t(x) \exp\left(-\alpha_t T(x) h_t(x)\right)} \tag{2}$$

The classification error of classifier h is given by:

$$Err(h) = \sum_{x | h(x) B(x) < 0} W(x) \tag{3}$$

At the end of N rounds, the strong classifier is given by a linear combination of N weak classifiers $h_t(x)$ and its coefficient α_t:

$$f_{boost}(x) = \text{sign}\Big(\sum_{t=1}^{T} \alpha_t h_t(x)\Big) \tag{4}$$

[1]A weak classifier is the one that produces classification results slightly better than chance.

B. Bootstrap Aggregation (BAGG)

Bootstrap aggregation (Bagging) approach is a machine learning ensemble algorithm which aims at evaluating the predictions on a sampling collection (bootstrap samples). Formally, let T be an initial training set which is divided into B equal parts $Z^i, i = 1, 2, ..., B$ [6]. Each sample set is used on the training of B classifiers. After training, each classifier obtains a coefficient (α) that will be used in the classification step. Given a new data point to test, each classifier coefficient is used to assign a class to it, and the final class will be a majority voting among the B classifiers [7]. For each element x, a prediction $\vec{f^i}(x)$ for each classifier is obtained and the result is calculated:

$$\vec{f_{bag}}(x) = \frac{1}{B} \sum_{i=1}^{B} \vec{f^i}(x) \tag{5}$$

Figure 1 illustrates the training and the classification steps of the Bagging approach.

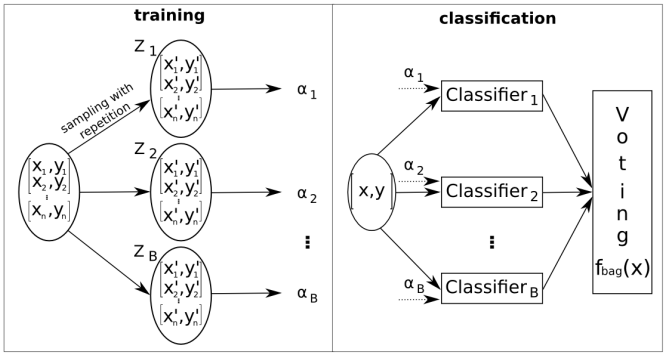

Fig. 1. Training and classification steps using a Bagging approach.

C. Support Vector Machine (SVM)

Support Vector Machine is a machine learning technique, introduced by [8]. It requires data points previously annotated to build a classification model. The goal is to construct an optimum margin decision hyperplane, which can be used to separate an n-dimensional space. The decision hyperplane is calculated such that it maximizes the margin among two classes (the standard SVM is a two-class classifier).

The margin can be seen as the minimum distance of one point of one class to the other. It can be interpreted as a separation measure among two classes and represents the separability degree among them (quality measure of classification). The points on borders among the classes are called support vectors.

When it is not possible to find a linear separator among the classes, the data are mapped on-the-fly onto higher dimensional spaces through a non-linear mapping using the kernel trick [9]. According to Cover [10], every data point which is not separable in a space can be mapped onto other higher-dimensional space and become linearly separable.

Figure 2 illustrates the use of SVM to separate two classes. More details about this technique can be found in [8], [11]–[14].

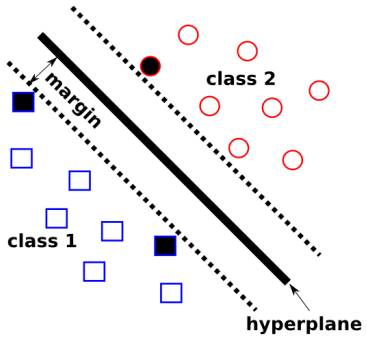

Fig. 2. The SVM classifier builds a maximum margin decision hyperplane to separate two classes (squares and circles).

III. Classifier Fusion Framework

Section III-A presents an overview of the proposed framework. Section III-B presents our approach based on diversity measures [15] for automatically selecting base classifiers that are good candidates to be combined.

A. Overview

Let $\mathcal{L}$ be a set of learning methods (e.g., Decision Tree, Naïve Bayes, kNN, SVM, etc.) and $\mathcal{F}$ be a set of image descriptors (e.g., Color Histogram). Suppose that base classifiers are created by combining each available learning method with each image descriptor. For example, three classifiers could be created by combining the learning methods Decision Tree, Naïve Bayes and kNN with the Color Histogram descriptor. Let $\mathcal{C}$ be the set of classifiers created by that combination, where $|\mathcal{C}| = |\mathcal{L}| \times |\mathcal{F}|$.

In our problem, let $\mathcal{S}$ be a set of produce images, where the class of $s_i \in S$ $(1 < i \leq |\mathcal{S}|)$ is known. The set $\mathcal{S}$ is used to construct both the training (T) and validation (V) sets, where $T \cup V = \mathcal{S}$ and $T \cap V = \emptyset$. As we consider a supervised learning scenario, the actual classes for training and validation data points are known *a priori*.

Initially, all base classifiers $c_j \in \mathcal{C}$ $(1 < j \leq |\mathcal{C}|)$ are trained on set T. Next, the performance of each classifier on the validation set V is computed and stored into a matrix M_V, where $|M_V| = |V| \times |\mathcal{C}|$.

In the following, M_V is used to select the set $\mathcal{C}^* \subset \mathcal{C}$ of classifiers that are good candidates to be combined. In our approach, diversity measures are employed to determine $\mathcal{C}^*$ (see Section III-B). Note that a new matrix $M_V^* \subset M_V$ can be created by using the selected classifiers in $\mathcal{C}^*$.

Given a new produce image I, we use each classifier $c_k \in \mathcal{C}^*$ $(1 < k \leq |\mathcal{C}^*|)$ to determine the class of I, producing k responses. The k outcomes are used as input of a fusion technique (e.g., majority voting, SVM, etc.) that takes the final decision regarding the definition of the class of I. In the case of a fusion technique that requires prior training (e.g., SVM), M_V^* is used.

Figure 3 illustrates the proposed framework for combining classifiers.

B. Diversity Measures for Selecting Base Classifiers

Diversity in classification tasks is related to the agreement and disagreement of classifiers with regard to a same set of inputs.

Consider the previously defined $\mathcal{C}$ (set of classifiers) and M_V (a matrix where $|M_V| = |V| \times |\mathcal{C}|$), containing the outcomes of classifiers $c_j \in \mathcal{C}$ on the validation set V. Let $\mathcal{D}$ be a set of diversity measures.

Each diversity measure $d_l \in \mathcal{D}$ is used to compute the agreement/disagreement between two classifiers $c_{j_n}, c_{j_m} \in \mathcal{C}$, considering all possible combinations of classifiers. Let $\mathcal{R}_{d_l} = \{(c_{j_n}, c_{j_m}), score_{d_l}(c_{j_n}, c_{j_m})\}$ be the ranked list of pairs of classifiers defined by the score of the diversity measure d_l.

Let $\mathcal{R} = \{\mathcal{R}_{d_1}, \mathcal{R}_{d_2} \ldots \mathcal{R}_{d_{|\mathcal{D}|}}\}$ be the set of ranked lists defined for each available diversity measure. Let $\mathcal{R}^t$ be a set of ranked lists, where each ranked list contains the top t pairs of classifiers (t pairs of classifiers with the lowest diversity scores) and $\mathcal{H}$ be a histogram that counts the number of occurrences of a classifier in all ranked lists of $\mathcal{R}^t$. The set $\mathcal{C}^*$ of classifiers that are combined by our fusion approach is the $h = |\mathcal{C}^*|$ most frequent base classifiers in $\mathcal{H}$ (See Figure 4).

Algorithm 1 describes the proposed steps for selecting base classifiers, by taking into account diversity measures.

Algorithm 1 *Selection of base classifiers*

Input: set $\mathcal{D}$ of diversity measures, set $\mathcal{C}$ of classifiers, and the outcomes of classifiers on validation set V encoded in M_V.

1: $\mathcal{R} \leftarrow \emptyset$
2: **for** each $d_l \in \mathcal{D}$ **do**
3: $\quad \mathcal{R}_{d_l} \leftarrow \emptyset$
4: $\quad$ **for** each pair $(c_{j_n}, c_{j_m}) \in \mathcal{C} \times \mathcal{C}$ **do**
5: $\quad\quad score_{d_l}(c_{j_n}, c_{j_m}) \leftarrow d_l(c_{j_n}, c_{j_m})$
6: $\quad\quad \mathcal{R}_{d_l} \leftarrow \mathcal{R}_{d_l} \cup \{((c_{j_n}, c_{j_m}), score_{d_l}(c_{j_n}, c_{j_m}))\}$
7: $\quad$ **end for**
8: $\quad$ Sort $\mathcal{R}_{d_l}$ with regard to $score_{d_l}$
9: $\quad \mathcal{R} \leftarrow \mathcal{R} \cup \mathcal{R}_{d_l}$
10: **end for**
11: $\mathcal{R}^t \leftarrow$ select the top t ranked pairs of classifiers for each ranked list in $\mathcal{R}$
12: **for** each $c_j \in \mathcal{C}$ **do**
13: $\quad \mathcal{H}(c_j) \leftarrow 0$
14: **end for**
15: **for** each $d_l \in \mathcal{D}$ **do**
16: $\quad$ **for** each $((c_{j_n}, c_{j_m}), score_{d_l}(c_{j_n}, c_{j_m})) \in \mathcal{R}_{d_l}^t$ **do**
17: $\quad\quad \mathcal{H}(c_{j_n})_{++}$
18: $\quad\quad \mathcal{H}(c_{j_m})_{++}$
19: $\quad$ **end for**
20: **end for**
21: $\mathcal{C}^* \leftarrow \{c_{j_n} \in \mathcal{C}$, such as $|\mathcal{C}^*| = h$ and $\forall c_{j_m} \in \mathcal{C} \setminus \mathcal{C}^*$, $\mathcal{H}(c_{j_n}) > \mathcal{H}(c_{j_m})\}$

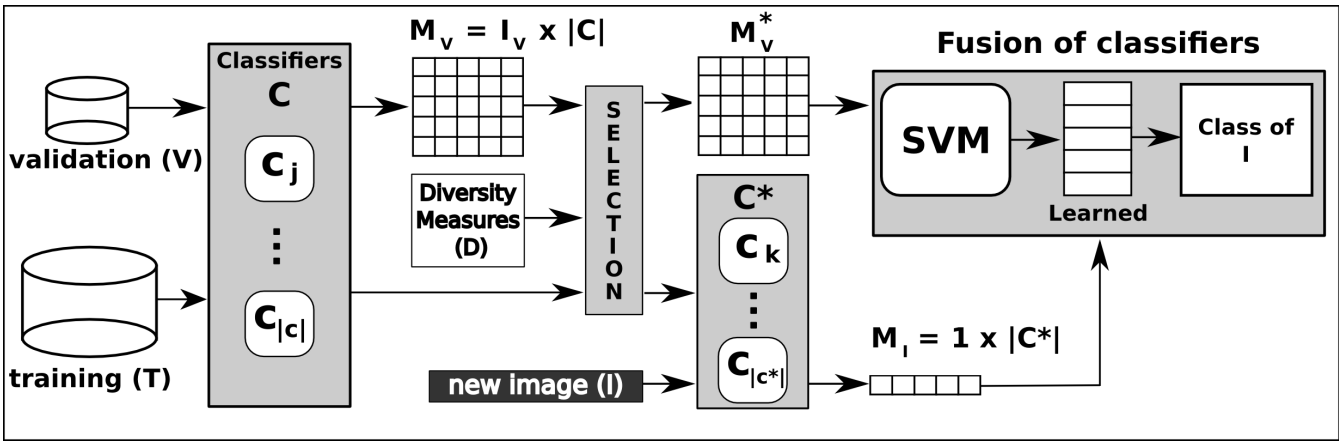

Fig. 3. Proposed framework for classifier fusion.

Fig. 4. The five steps for classifiers selection are: (a) Validation matrix M_V; (b) $\mathcal{R}$ lists sorted by diversity measures scores; (c) $\mathcal{R}^t$ lists with top t; (d) counts the number of occurrences of each classifier; (e) Selected classifiers $|\mathcal{C}^*|$.

IV. Experimental Protocol

This section presents the used image dataset, image descriptor, cross validation protocol, learning methods, diversity measures, and baselines.

A. Supermarket Produce Dataset

We have used freely available *supermarket produce dataset*[2] proposed in [2]. That dataset comprises 2,633 images divided into 15 different categories: Plum (264), Agata Potato (201), Asterix Potato (182), Cashew (210), Onion (75), Orange (103), Taiti Lime (106), Kiwi (171), Fuji Apple (212), Granny-Smith Apple (155), Watermelon (192), Honeydew Melon (145), Nectarine (247), Williams Pear (159), and Diamond Peach (211). Figure 5 depicts some images of the dataset.

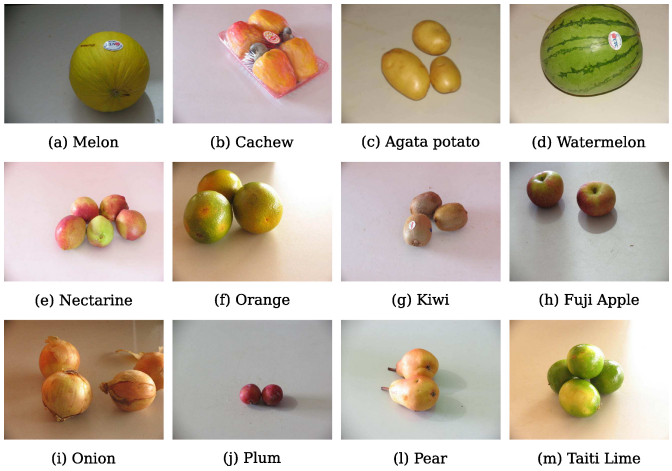

Fig. 5. *Supermarket Produce data set.*

B. Cross validation protocol

In this paper, we consider a k-fold cross-validation protocol for all experiments we perform. In this protocol, the original dataset is randomly separated into k non-overlapping subsets. A subset is chosen for testing set, and the $k-1$ subsets are used for training a learning technique. The cross-validation process is repeated k times (rounds) and each subset is used only once as test set. The final result (the classification accuracy) from this process can be the arithmetic mean among all subsets. The main goal of this protocol is to test the entire dataset and reduce the variability among rounds (result of each round must be approximately equal). In our experiments, we have considered a 5-fold cross-validation protocol. Each training set (consisting of four rounds) can be further divided into validation and actual training (for instance, three folds can be used for training and the fourth for assessing the classifier being developed). In this sense, we use the 5-fold cross-validation protocol again to further divide the training set into validation (one fold) and actual training (three folds).

[2]http://www.ic.unicamp.br/~rocha/pub/downloads/tropical-fruits-DB-1024x768.tar.gz (As of May, 2012).

C. Image Descriptors

Table I presents the color, texture, and shape descriptors we considered in our experiments. Given the produce recognition problem, the objective is to use the most complementary features as possible and rely on an effective combination

technique.

Descriptor	Type
ACC [16]	Color
BIC [17]	Color
CCV [18]	Color
GCH [19]	Color
LAS [20]	Texture
QCCH [21]	Texture
EOAC [22]	Shape

TABLE I
IMAGE DESCRIPTORS USED IN OUR EXPERIMENTS.

1) Color Autocorrelogram (ACC) [16]: The role of this descriptor is to map the spatial information of colors by pixel correlations at different distances. It computes the probability of finding in the image two pixels with color C at distance d from each other. For each distance d, m probabilities are computed, where m represents the number of colors in the quantized space. The implemented version quantized the color space into 64 bins and considered 4 distance values (1, 3, 5, and 7).

2) Border/Interior Pixel Classification (BIC) [17]: has been successful in many applications. The first step of the feature vector extraction process relies on the classification of image pixels into *border* or *interior* ones. When a pixel has the same spectral value in the quantized space as its four neighbors (the ones which are above, below, on the right, and on the left), it is classified as *interior*. Otherwise, the pixel is classified as *border*. Two histograms are computed after the classification: one for the interior pixels and another for the border ones. Both histograms are merged to compose the feature vector. The implemented version quantized the color space into 64 bins.

3) Color Coherence Vector (CCV) [18]: like GCH, it is recurrent in the literature. It uses an extraction algorithm that classifies the image pixels as "coherent" or "incoherent" pixels. This classification takes into consideration whether the pixel belongs or not to a region with similar colors, that is, coherent regions. Two color histograms are computed after quantization: one for coherent pixels and another for incoherent ones. Both histograms are merged to compose the feature vector. In our experiments, the color space was quantized into 64 bins.

4) Global Color Histogram (GCH) [19]: is one of the most commonly used descriptors, it uses an extraction algorithm which quantizes the color space in a uniform way and it scans the image computing the number of pixels belonging to each bin. The size of the feature vector depends on the quantization used. In the present work, the color space was split into 64 bins, thus, the feature vector has 64 values.

5) Local Activity Spectrum (LAS) [20]: this descriptor captures textures spatial activity in four different directions separately: horizontal, vertical, diagonal, and anti-diagonal. The four activity measures are computed for a pixel (i, j) by considering the values of neighboring in the four directions. The values obtained are used to compute a histogram that is called *local activity spectrum*. Each component g_i is quantized independently. In our experiments, each component was non-uniformly quantized into 4 bins, leading to a histogram with 256 bins.

6) Quantized Compound Change Histogram (QCCH) [21]: It uses the relation between pixels and their neighbors to encode texture information. This descriptor generates a representation invariant to rotation and translation. Its extraction algorithm scans the image with a square window. For each position in the image, the average gray value of the window is computed. Four variation rates are then computed by taking into consideration the average gray values in four directions: horizontal, vertical, diagonal, and anti-diagonal directions. The average of these four variations is calculated for each window position, they are grouped into 40 bins and a histogram of these values is computed.

7) Edge Orientation Autocorrelogram(EOAC) [22]: This is a shape descriptor. We chose this descriptor because it does not depend on segmentation to extract features. Its strategy is to classify the image edges according to two aspects: the edge orientation and the correlation between neighbor edges. The first step is to compute the image gradient from the input image. Then, the algorithm computes an edge orientation auto-correlogram. The feature vector is composed by the values from this auto-correlogram. In this implementation, we use angle quantization in 72 segments of 5° degrees each one; four distance values (1, 3, 5, and 7); the Sobel operator to compute the image gradient; and a gradient threshold equal to 25, as suggested in [22]. The final vector is comprised by 288 values.

D. Learning Methods

We have used seven learning methods in our framework: Naïve Bayes (NB), Decision Tree (DT), Simple Logistic (SL), Naïve Bayes Tree (NBT), k-Nearest Neighbors (kNN), using $k = 1$, $k = 3$, and $k = 5$. Such methods are simple and fast, being suitable to be combined in a real time recognition system. The idea of using different learning methods is that some descriptor might be better with specific methods. In this sense, a support vector machine was avoided here due to its known slow training time. Even though SVMs have sub linear time for testing, in a multi-class scenario it would need several binary SVMs to perform the multi-class classification as reported in [23]–[25].

The proposed framework aims at automatically finding suitable combinations of classifiers formed by descriptors and learning methods. We have used the implementation of those learning methods available in the WEKA[3] data mining library. All learning techniques were used with default parameters.

E. Diversity Measures

Let $\mathcal{M}$ be a matrix containing the relationship between a pair of classifiers with percentage of concordance. Table II

[3]http://www.cs.waikato.ac.nz/~ml/weka (As of May, 2012).

shows a relationship matrix $\mathcal{M}$ with percentage of hit and miss for two classifiers c_i and c_j. The value a is the percentage of images that both classifiers c_i and c_j classified correctly in the validation set. Values b and c are the percentage of images that c_j hit and c_i missed and vice-versa. The value d is the percentage of images that both classifiers missed.

	Hit c_i	Miss c_i
Hit c_j	a	b
Miss c_j	c	d

TABLE II
RELATIONSHIP MATRIX $\mathcal{M}$ BETWEEN TWO CLASSIFIERS c_i AND c_j.

In [15], Kuncheva et al. present several measures to assess diversity, considering pairs of classifiers. Following their work, in our experiments, we have used *Double-Fault Measure* (DFM), *Q-Statistic* ($QSTAT$), and *Interrater Agreement k* (IA). Those measures are defined as follows.

$$DFM_{i,j} = d, \tag{6}$$

$$QSTAT_{i,j} = \frac{ad - bc}{ad + bc}, \tag{7}$$

$$IA_{i,j} = \frac{2(ac - bd)}{(a + b)(c + d) + (a + c)(b + d)}. \tag{8}$$

The diversity is greater if the measures *Double-Fault Measure*, *Q-Statistic* and *Interrater Agreement k* are lower among pairs of classifiers c_i and c_j [15].

F. Baselines

We have used seven different approaches as baselines: BAGG-3, BAGG-17, SVM-PK, SVM-RBF, BOOST, MV-BOOST-7, and OVA-BOOST. We describe each baseline in the following.

BAGG-3 and BAGG-17 rely on the bagging approach (Section II-B) using $k = 3$ and $k = 7$ iterations, respectively. Both are tested with BIC descriptor (Section IV-C). The configuration of the bagging-based methods are the same used in [2].

SVM-PK and SVM-RBF are classifiers based on support vector machines (Section II-C) using one image descriptor. SVM-PK uses polynomial kernels and SVM-RBF uses RBF kernels. The parameters used can be found [2].

BOOST and MV-BOOST-7 implement one multi-class adaboost (Section II-A) for each descriptor. The weak learner used is an SVM with polynomial kernel. In MV-BOOST-7, the descriptor results are combined by using the majority voting (MV) scheme. OVA-BOOST combines all descriptors using one binary Adaboost (Section II-A) for each dataset class. The binary Adaboost uses linear SVMs as weak learners. We used the One-vs-All (OVA) strategy to combine the binary classifiers. That strategy relies on training the i_{th} classifier by using all patterns of class i as positive (+1) examples and the remaining class patterns as negative (-1) examples. We classify an input example x to the class with the highest response between the binary Adaboost classifiers.

We also report the results of the methods described in [2] and [3], as they considered the same dataset in their experiments. In the case of [2], we consider two methods named *Rocha BLDA* and *Rocha SVM-Fusion* that use, respectively, Bagging of Linear Discriminant Analysis and SVM in the classification process.

V. EXPERIMENTS

This section discusses the results regarding the effectiveness and efficiency of the proposed framework.

In the experiments, three different fusion approaches were conducted: SVM with RBF kernel considering $h = 49$ (FSVM-RBF-49) and $h = 10$ (FSVM-RBF-10) classifiers, and majority voting (MV-49). Note that the use of $h = 49$ refers to the use of all available base classifiers in the fusion process. We have used $t = 10$ in our experiments (see Line 11 of Algorithm 1) as we impose a constraint to classify each input as fast as possible for deployment of the system in a supermarket scenario similar to the one imposed in [2].

A. Effectiveness

Table III presents the results organized in three parts: (1) Fusion Techniques; (2) Baselines form the literature which used the same dataset in their experiments; and (3) Baselines using just one image descriptor. All results consider the average classification accuracy considering a 5-fold cross-validation protocol.

As expected, FSVM-RBF-49 outperforms the baselines as it uses all configurations of classifiers and available descriptors. However, FSVM-RBF-10 performance is close to the one observed for FSVM-RBF-49, which means that the proposed selection framework based on diversity measures were able to select the most suitable classifiers (in this case, only 10 classifiers) to be combined without sacrificing much of the classification quality. Using less classifiers impacts the overall efficiency of the recognition system as discussed in the next section.

FSVM-RBF-49 also yields better results than MV-49. One possible reason for that relies on its ability on producing a non-linear combination of classifiers. For a better visualization, Figure 6 shows all results sorted by classification accuracy.

B. Efficiency

We conducted experiments on a 2.4GHz virtual machine with 1GB of RAM (running Linux) to assess the recognition time of the proposed framework.

Our approach is composed by three steps: (1) extraction of feature vectors for all image descriptors; (2) classification using $|C^*|$ classifiers; and (3) combination of $|C^*|$ classifier outcomes with SVM. The complexity of Step 1 depends on the employed descriptors. In our case, that step takes $\cong 0.5s$. Step 2 takes less than $1s$ on average, since it uses low-cost

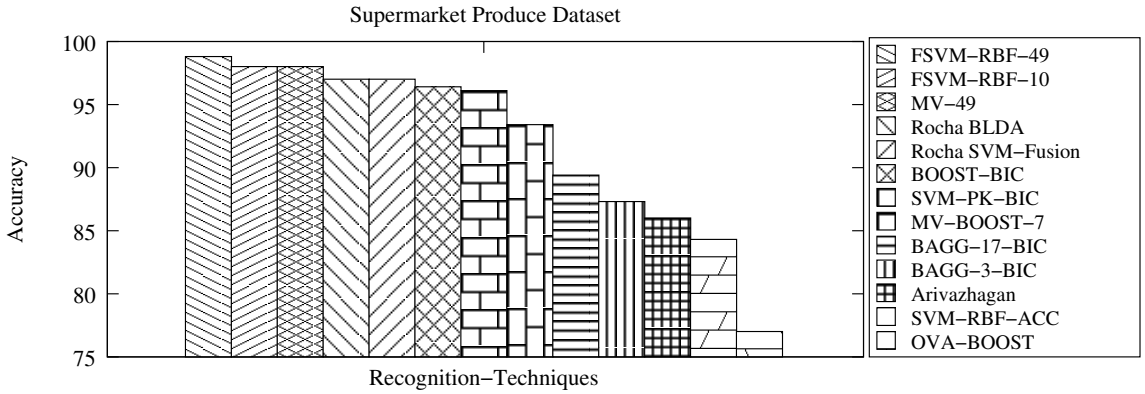

Fig. 6. *Recognition techniques sorted by classification accuracy.*

Recognition Technique	Accuracy
FSVM-RBF-49	98.8% ± 0.9
FSVM-RBF-10	98.0% ± 0.9
MV-49	98.0% ± 1.1
MV-BOOST-7	93.4% ± 1.4
OVA-BOOST	77.3% ± 0.8
Rocha BLDA [2]	97.0% ± 0.6
Rocha SVM-Fusion [2]	97.0% ± 0.4
Arivazhagan [3]	≅ 86.0%
BOOST-BIC	96.4% ± 1.0
SVM-PK-BIC	96.1% ± 1.8
BAGG-17-BIC	89.4% ± 1.8
BAGG-3-BIC	87.3% ± 1.7
SVM-RBF-ACC	84.3% ± 2.7

TABLE III
CLASSIFICATION EFFECTIVENESS OF THE PROPOSED FRAMEWORK AND BASELINES, WITH THEIR RESPECTIVE STANDARD DEVIATIONS.

classifiers. Step 3 takes less than $0.1s$, since it combines a few number of classifiers. Therefore, the average recognition time is less than $2s$, for the FSVM-RBF-10, which considers $h = |C^*| = 10$ selected base classifiers. Note that Rocha et al. [2] reported that their method takes $\cong 5s$ to recognize a new produce image, using a 2.1GHz machine with 2GB of RAM.

VI. Final Remarks and Future Work

This paper presented a new framework to combine classifiers aiming at supporting the deployment of produce recognition systems. The novelty of this work relies on the use of diversity measures to determine which base classifiers are suitable to be combined.

The experiment results show that the proposed framework yields high classification accuracy rates in a reduced time. In fact, our framework is able to combine classifiers more effectively than baselines. Different from other approaches, our method is able to not only select classifiers, but also *learn*, indirectly, which descriptors (and therefore visual properties) are more appropriate for the target application. To keep the a high recognition rate with the minimum computational effort, our classifier selection strategy exploits the use of diversity measures, which allow the combination of non-correlated, highly-effective, and low-cost base classifiers. Our approach is suitable for real-time produce recognition due to two reasons: first, all used base classifiers are of low cost in terms of computational efforts; and second, only a reduced number of effective classifiers are combined.

Future work includes the investigation of the use of other diversity measures and fusion techniques, as well as the development of an automatic way to find the final number of base classifiers to combine based on classification quality and time constraints imposed by the client. We have used part of the proposed framework for classifying remote sensing images and preliminary results are promising [26]. Therefore, we plan to continue investigating the use of the framework in other domains.

Acknowledgement

The authors are grateful to CAPES, CNPq, FAPESP (grants 2010/14910-0, 2010/05647-4, 2009/18438-7, 2009/18438-7, and 2008/58528-2), and Microsoft Research for the financial support.

References

[1] R. M. Bolle, J. H. Connell, N. Haas, R. Mohan, and G. Taubin, "Veggievision: A produce recognition system," in *IEEE WACV*, Sarasota, USA, 1996, pp. 1–8.

[2] A. Rocha, D. C. Hauagge, J. Wainer, and S. Goldenstein, "Automatic fruit and vegetable classification from images," *Elsevier COMPAG*, vol. 70, no. 1, pp. 96 – 104, 2010. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S016816990900180X

[3] S. Arivazhagan, R. N. Shebiah, S. S. Nidhyanandhan, and L. Ganesan, "Fruit recognition using color and texture features," *CIS Journal*, 2010.

[4] R. E. Schapire, "A brief introduction to boosting," in *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, ser. IJCAI '99, 1999, pp. 1401–1406.

[5] Y. Freund and R. E. Schapire, "Experiments with a new boosting algorithm," *Update*, pp. 148–156, 1996.

[6] J. Friedman, T. Hastie, and R. Tibshirani, *The Elements of Statistical Learning*, 1st ed. Springer, 2001.

[7] A. Rocha and S. Goldenstein, "Randomização progressiva para esteganálise," Master's thesis, Campinas, SP, Brazil, 2006.

[8] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," in *Proceedings of the fifth annual workshop on Computational learning theory*, ser. COLT '92, 1992, pp. 144–152.

[9] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press, 2000.

[10] T. M. Cover, "Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition," *Electronic Computers, IEEE Transactions on*, vol. EC-14, no. 3, pp. 326–334, 1965.

[11] R. Herbrich, T. Graepel, and K. Obermayer, *Large Margin Rank Boundaries for Ordinal Regression*. Cambridge, MA: MIT Press, 2000, pp. 115–132. [Online]. Available: http://stat.cs.tu-berlin.de/publications/papers/herobergrae99.ps.gz

[12] T. Joachims, "Optimizing search engines using clickthrough data," in *International Conference on Knowledge Discovery and Data Mining*, 2002, pp. 133–142.

[13] Y. Cao, X. Jun, T. Liu, H. Li, Y. Huang, and H. Hon, "Adapting ranking svm to document retrieval," in *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, 2006, pp. 186–193.

[14] P. Hong, Q. Tian, and T. S. Huang, "Incorporate support vector machines to content-based image retrieval with relevant feedback," in *International Conference on Image Processing*, 2000, pp. 750–753.

[15] L. I. Kuncheva and C. J. Whitaker, "Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy," *Machine Learning*, vol. 51, pp. 181–207, May 2003.

[16] J. Huang, R. Kumar, M. Mitra, W. Zhu, and R. Zabih, "Image indexing using color correlograms," in *IEEE CVPR*, 1997, pp. 762–768.

[17] R. Stehling, M. Nascimento, and A. Falcao, "A compact and efficient image retrieval approach based on border/interior pixel classification," in *ACM CIKM*, 2002, pp. 102–109.

[18] G. Pass, R. Zabih, and J. Miller, "Comparing images using color coherence vectors," in *ACM Multimedia*, 1996, pp. 65–73.

[19] M. Swain and D. Ballard, "Color indexing," *IJCV*, vol. 7, no. 1, pp. 11–32, 1991.

[20] B. Tao and B. Dickinson, "Texture recognition and image retrieval using gradient indexing," *JVCIR*, vol. 11, no. 3, pp. 327–342, 2000.

[21] C. Huang and Q. Liu, "An orientation independent texture descriptor for image retireval," in *ICCCS*, 2007, pp. 772–776.

[22] F. Mahmoudi, J. Shanbehzadeh, A. Eftekhari-Moghadam, and H. Soltanian-Zadeh, "Image retrieval based on shape similarity by edge orientation autocorrelogram," *Elsevier Pattern Recog.*, vol. 36, no. 8, pp. 1725–1736, 2003.

[23] C. M. Bishop, *Pattern Recognition and Machine Learning*, 1st ed. Springer, 2006.

[24] A. Passerini, M. Pontil, and P. Frasconi, "New results on error correcting output codes of kernel machines," *IEEE TNN*, vol. 15, no. 1, pp. 45–54, January 2004.

[25] A. Rocha and S. Goldenstein, "From binary to multi-class: divide to conquer," in *Visapp*, Lisbon, Portugal, February 2009, pp. 323–330.

[26] F. A. Faria, J. A. Santos, R. da S. Torres, A. Rocha, and A. Falcão, "Automatic fusion of region-based classifiers for coffee crop recognition," in *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, Munique, Alemanha, July 2012.