Automatic Visual Quality Assessment of Biscuits Using Machine Learning

Mardlla de Sousa Silva[1], Luigi Freitas Cruz[1], Pedro Henrique Bugatti[1], and Priscila Tiemi Maeda Saito[1,2](✉)

[1] Department of Computing, Federal University of Technology - Paraná, Cornélio Procópio, Curitiba, Brazil
{mardlla,luigicruz}@alunos.utfpr.edu.br
{pbugatti,psaito}@utfpr.edu.br
[2] Institute of Computing, University of Campinas, Campinas, Brazil

Abstract. Considering the great competition among industries, one of the main factors that make companies market leaders is the quality of their products. However, the techniques applied to quality control are often flawed or inefficient, due to the great dependence on the human factor, which leads to a tiresome process and highly susceptible to errors. Besides that, in the context of the industry 4.0 the use of technologies to improve the evaluation of these products becomes increasingly essential. Hence, this work aims to find the most appropriate automatic classification method of non-standard food products allowing its deployment in a real biscuit industry. To do so, we evaluate different image descriptors and classifiers, based on deep learning (end-to-end and deep features) and traditional techniques (handcrafted features). From the obtained results, we can see that the proposed methodology can provide more effective quality control for the company, reaching an accuracy of up to 99%. This testifies that it can avoid offering non-compliant products in the market, improving the credibility of the brand with the consumer, its profitability and consequently its competitiveness.

Keywords: Image classification · Computer vision · Machine learning · Deep learning · Food industry

1 Introduction

Nowadays, any manufacturing process aims to make use of the most effective machine/information technology, mainly taking advantage of machine learning methods. Hence, there is great potential for improvement of the production activity to provide higher quality products. From the increasingly automated production process, it becomes important to have a more efficient process of checking non-standard quality products to keep up with the advances and the speed of the large-scale production.

In the food industry, the bakery sector can be considered one of the most important [23]. It includes the production of biscuits, pasta and bread. However,

© Springer Nature Switzerland AG 2020
L. Rutkowski et al. (Eds.): ICAISC 2020, LNAI 12416, pp. 59–70, 2020.
https://doi.org/10.1007/978-3-030-61534-5_6

to keep the sector constantly growing, technological innovation is essential in the production process. The quality of what is produced becomes vital, because it is intrinsically linked to customers' satisfaction, which is the key to the success or failure of a company. Non-standard products (i.e. regarding color, size, texture or flavor) can generate consumer complaints [14]. Consumers disappointed with the quality of products can generate losses and compromise the competitiveness of the company.

The great problem is that many quality control procedures are performed through human analysis (e.g. visual analysis). The visual quality analysis performed by employees is inefficient and unfeasible, considering that a large number of products are produced daily in an industry.

In this context, automating the quality inspection processes to provide higher quality products may require technological investments and, consequently, it demands increased costs. In this sense, it is important to evaluate the most appropriate automatic learning strategies capable to providing good and reliable results.

Taking into account the described scenario, this work aims to learn descriptors and pattern classifiers, improving the classification and the quality control of biscuits in a real food industry. Hence, in summary, our contributions in this paper are twofold: (i) we introduced an approach based on transfer learning capable of better identifying biscuits corresponding (or not) to the standards required by the company; (ii) we performed an extensive comparison between handcrafted and deep features with several traditional supervised classifiers, and against end-to-end state-of-the-art convolutional neural network architectures.

2 Background

It is extremely important to automate the process of identifying products that do not meet the standards established by the company. In these cases, the use of computer vision techniques has presented significant results for the classification of products [5,9,19,22,27,28]. According to [22] one of the main problems, regarding to the biscuit classification in real time, refers to the high volume of computing at high speed, which requires high performance computing vision techniques. Although there are many papers in the literature related to food quality inspection [8,18,27], few of them can be used in high-speed product classification.

In [22,23,28] the authors analyze only one type of defect (e.g. detection of cracks) in a specific kind of biscuit (only one type). Differently, our work deals with not only the classification of different types of biscuits, but also with their standard or non-standard quality.

2.1 Image Description and Classification

Ideally, image descriptors should perform the extraction of relevant characteristics of a given image in a similar manner to a human observer. However, despite some advances, there is still much to be explored and improved, given the current knowledge regarding vision, cognition and human emotion.

Among the most used features to describe an image are those defined as primitive (low-level) derived from three fundamental elements of the image: distribution of intensities (colors), texture and shape.

Color-based features are widely used in numerous applications, due to low computational cost and invariance to operations such as rotations and translations. The color description is usually used to build a color histogram. In spite of presenting linear cost w.r.t. the number of *pixels*, histograms present reduced discrimination capacity and do not provide information about the spatial distribution of colors in an image.

Several proposals have been made to address such problem, including a combination of color and texture features. Unlike color, texture occurs over a certain region rather than a point (*pixel*). In addition, since it has a certain periodicity and scale, it can be described in terms of direction, roughness, contrast, among others. The shape-based features, although in general involve non-trivial processes (which can generate a higher computational cost), are also interesting when used in some application domains.

All the aforementioned image description processes (i.e. feature extraction) are based on the so-called handcrafted features (i.e. they are intrinsically bounded to the problem's context). Hence, they can present some generalization issues. To diminish such problem, nowadays, there is an expansion of image description methods based on deep learning architectures like convolutional neural networks (CNN) [21]. This kind of architecture is capable of learning the features regarding a problem through an hierarchical representation of features from low-level to high-level ones.

It can be used not only to generate "generic" visual features (deep features), but also to perform the classification step at the same process (end-to-end learning). However, the computational cost of such techniques (requiring great computational power and volume of data) can impairs their use in some real scenarios. There are different CNN architectures in the literature. For instance, ResNet [24] presents residual blocks to reduce the training time. It introduces a "shortcut" connection, which skips one or more layers. Another one is Inception-v3 [26] that applies the so-called bottlenecks' layers also trying to reduce the cost.

Besides the aforementioned possibilities of cost reduction, there is the transfer learning technique. Through this process we can reuse a CNN that was pretrained in a "generic" context (e.g. ImageNet dataset [29]) and apply it to solve a more specific problem. To do so, we just need to retrain a considerably smaller portion of the original architecture (e.g. dense layers) and freeze the remaining ones (e.g. convolutional layers).

3 Proposed Methodology

In this section, we present our proposed methodology for classification and quality control in a food industry focused on the production of biscuits. For this, it is important to obtain the best learning approaches for description and classification. Then, different feature extractors and classifiers should be analyzed both for i-) identification of biscuits corresponding (or not) to the standards required by the company (i.e. binary classification), and ii-) identification of different types of biscuits (i.e. multiclass classification). Figure 1 illustrates the pipeline of our proposed methodology.

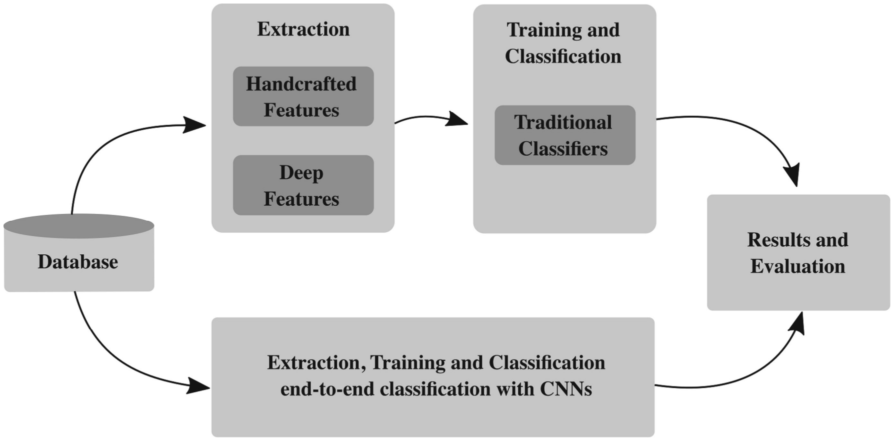

Extraction
Handcrafted Features
Deep Features
Training and Classification
Traditional Classifiers
Database
Results and Evaluation
Extraction, Training and Classification end-to-end classification with CNNs

Fig. 1. Pipeline of the proposed methodology.

Initially, we obtained the image database from a production line in an industry. After organizing the datasets (see description in Sect. 4.1), there are two main flows. Through the first flow, it is possible to perform the extraction of (handcrafted and deep) features, using the traditional and the CNN architectures, respectively. Considering the deep features, we can obtain them from a given layer (generally the last dense layer) or fine-tune the pre-trained CNN model to work with a new domain. Different CNN architectures can be used. In this case, we applied the transfer learning technique and used the pre-trained models, which generally were trained on a bigger dataset from a general domain (i.e. not related to biscuits). We considered the learned weights (i.e. trainable parameters) from the CNNs as deep features. Therefore, these (handcrafted and deep) extracted features can be evaluated by the traditional classifiers.

Regarding the second flow, in the fine-tuning step, we evaluated the end-to-end classification process. In this case, we retrain the fully connected layer and replace the final classification layer (e.g. softmax) to output the correct number of probabilities, according to our dataset.

Afterwards, our methodology enables performing analyses between different types of extractors and classifiers, and evaluating the more appropriate setting (pair extractor/classifier) to the classification of biscuits. Algorithm 1 presents details of our methodology.

Algorithm 1: Proposed methodology

```
input       : image dataset $\mathcal{D}$
output      : best learning model $M^{\Omega}$ obtained through maximum accuracies
auxiliaries: $E$: set of feature extractors; $H$: set of pre-trained CNN
              architectures; Feats: hand-crafted and deep feature sets; TrainSets
              and TestSets: training and test sets; perTrain and perTest:
              percentages of the training and test sets; nsplits: number of splits;
              $\mathcal{C}$: set of traditional classifiers; ModelSets: learning model sets;
              AccSets: mean accuracies; TrainSetIds and TestSetIds: identifiers of
              the training and testing samples from each split, $MeanAcc^{\Omega}$:
              maximum accuracies.
HandCraftedFeatures ← getHCFeatures($\mathcal{D}$, $E$);
DeepFeatures ← getDeepFeatures($\mathcal{D}$, $H$);
Feats ← HandCraftedFeatures $\bigcup$ DeepFeatures;
for each $i \in Feats_i,\ i = 1, ..., nf$ do
    TrainSets$_i$, TestSets$_i$ ← stratifiedSplits(Feats$_i$, perTrain, perTest, nsplits);
    for each $j \in \mathcal{C}_j$ do
        ModelSets$_{ij}$ ← generateModels(TrainSets$_i$, $\mathcal{C}_j$);
        AccSets$_{ij}$ ← testModels(TestSets$_i$, ModelSets$_{ij}$);
    end
end
for each $i \in H_i,\ i = 1, ..., nh$ do
    AccSets ← AccSets $\bigcup$ end2end(TrainSetIds, TestSetIds, H$_i$);
end
MeanAcc$^{\Omega}$ ← findMaxAcc(AccSets);
```

4 Experiments

4.1 Dataset Description

The initial stage of this work was to build the image dataset. To do so, we chose different types of biscuits (e.g. strawberry, vanilla, among others) presenting standard and non-standard quality according to the factory policy. The dataset was divided into 8 classes, regarding each biscuit flavor and its quality level (standard or non-standard). We collected $1,000$ samples for each class, composing a set of $8,000$ samples for the experiment. All samples were preprocessed changing the background of the image from white to blue background, which simulates the color of the treadmill used in the manufacturing process. Table 1 shows each image class and its respective description and number of samples.

Employees of the production line were needed to collect these biscuits, since it is not possible to frequently visit the production line by unauthorized people. The employees collected the samples once a week, because the biscuits are produced on demand. Thus, a given type of biscuit are not produced every day. The image acquisition was performed using an 8 megapixel camera with a resolution of 3264 × 2448 pixels Full HD (1920 × 1080 pixels) with 60 fps. Figure 2 shows image examples of the dataset's classes (each type of biscuit and its quality level - standard and non-standard).

Table 1. Description of the biscuit dataset.

Class	Description	Total
C_1	Standard Vanilla	1000
C_2	Non-standard Vanilla	1000
C_3	Standard Chocolate	1000
C_4	Non-standard Chocolate	1000
C_5	Standard Candy	1000
C_6	Non-standard Candy	1000
C_7	Standard Strawberry	1000
C_8	Non-standard Strawberry	1000

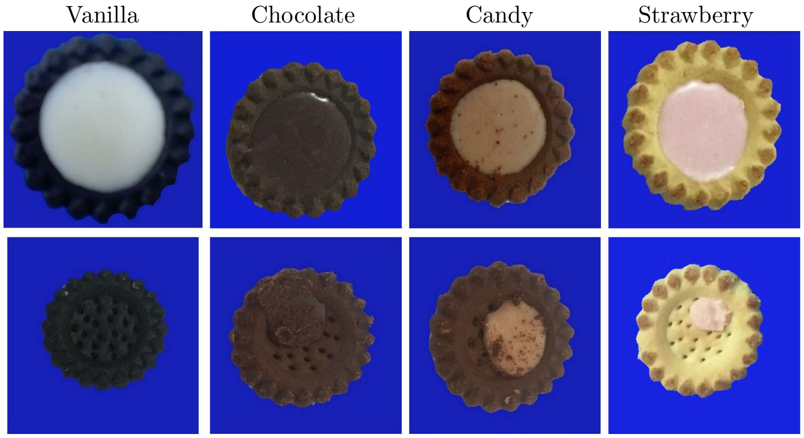


Fig. 2. Biscuit samples from each class. Upper images correspond to standard samples and lower images refer to non-standard samples.

4.2 Scenarios

To perform the experiments we extracted handcrafted features using traditional image descriptors and deep features from two different state-of-the-art CNN architectures, called Inception-v3 [2,26] and Resnet-v2 [24]. We applied the transfer learning technique for both of them (pre-trained on ImageNet dataset [29]). This provides a considerable reduction of data and training cost. We used

Table 2. Description of the extractors of handcrafted and deep features.

Extractor	Category	Features
ACC [13]	Color	768
BIC [3]	Color	128
CEDD [7]	Color	144
FCTH [6]	Color and Texture	192
Gabor [30]	Texture	60
GCH [32]	Color	255
Haralick [20]	Texture	14
JCD [17]	Color and Texture	336
LBP [12]	Texture	256
LCH [32]	Color	135
Moments [30]	Texture	4
MPO [10]	Texture	6
MPOC [10]	Texture	18
PHOG [33]	Texture	40
RCS [31]	Color	77
Tamura [15]	Texture	18
Inception-v3 [29]	Generic	2048
ResNet-v2 [24]	Generic	1536

both architectures not only to extract deep features from our image dataset, but also to perform the end-to-end process (retraining just the dense layers and freezing the others). Table 2 shows the traditional extractors that we used, with their respective types (e.g. color, texture, generic) and number of features.

Moreover, we used different traditional supervised classifiers considering each type of features (handcrafted and deep ones), and compared them against the end-to-end process with transfer learning. To the evaluation process we used the hold-out protocol. To do so, we split our dataset into 80% for training and 20% for testing. We generated 10 stratified splits. Regarding the traditional supervised classifiers we used k-NN [25], J48 [1], RF [4] and SVM [11], all of them with their default literature parameters.

Considering the Inception-v3 and ResNet-v2 architectures, as required by them, the images were resized to 299 × 299 pixels. Both were trained using 50 epochs, and we used a batch size of 32 samples, an initial learning rate of 10^{-4}, a learning rate decay factor equal to 0.7, a number of epochs before decay of 2, and the Adam optimizer [16].

4.3 Results

Table 2 shows the results obtained by each combination of feature extractor and traditional classifiers. From the results, it is possible to observe and obtain the

Table 3. Mean accuracies ± standard deviation presented by descriptors and classifiers.

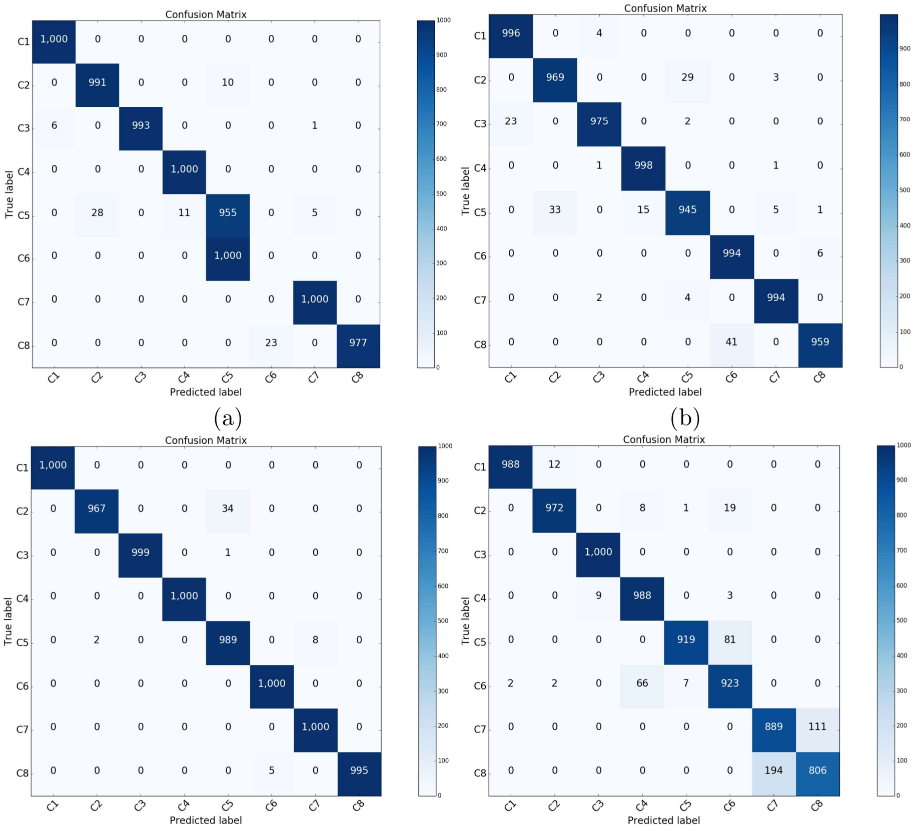
	k-NN	J48	RF	SVM
ACC	**98.75 ± 0.40***	97.56 ± 0.58	**99.03 ± 0.37***	85.28 ± 1.09
BIC	**98.88 ± 0.42***	97.83 ± 0.60	**99.33 ± 0.28***	85.47 ± 2.24
CEDD	**97.84 ± 0.55**	96.08 ± 0.70	**98.69 ± 0.37***	90.81 ± 0.96
FCTH	**96.43 ± 0.55**	94.29 ± 0.74	**96.66 ± 0.51**	85.67 ± 1.03
Gabor	**96.33 ± 0.67**	90.70 ± 2.00	**96.21 ± 0.58**	66.82 ± 1.26
GCH	**99.08 ± 0.38***	97.59 ± 0.56	**99.20 ± 0.34***	84.66 ± 2.55
Haralick	**96.56 ± 0.54**	94.43 ± 1.29	**96.97 ± 0.54**	52.41 ± 6.04
JCD	**98.23 ± 0.50**	96.70 ± 0.60	**99.07 ± 0.36***	91.17 ± 0.95
LBP	**98.56 ± 0.44***	95.36 ± 0.82	**98.10 ± 0.40**	**98.20 ± 0.52**
LCH	**98.95 ± 0.41***	97.29 ± 0.61	**99.17 ± 0.35***	**98.55 ± 0.40***
Moments	95.04 ± 0.80	94.88 ± 0.90	**97.14 ± 0.62**	65.63 ± 4.17
MPO	**97.50 ± 0.57**	96.07 ± 0.69	**97.95 ± 0.54**	66.74 ± 4.46
MPOC	**98.23 ± 0.48**	96.72 ± 0.62	**98.40 ± 0.40**	82.91 ± 2.79
PHOG	93.45 ± 0.72	89.53 ± 0.96	**96.08 ± 0.61**	60.72 ± 1.35
RCS	**97.82 ± 0.53**	96.34 ± 0.67	**98.16 ± 0.52**	69.45 ± 1.23
Tamura	**96.50 ± 0.67**	94.73 ± 0.67	**97.28 ± 0.47**	78.75 ± 1.49
Inception-v3	**98.92 ± 0.38***	96.92 ± 0.68	**99.04 ± 0.40***	**99.24 ± 0.34***
Resnet-v2	**98.65 ± 0.41***	96.12 ± 0.65	**98.84 ± 0.37***	**99.09 ± 0.36***

best feature extractor for each classifier (see underlined values, Table 3). For example, using the *k*-NN classifier, the most appropriate extractors were ACC, BIC, GCH, LBP, LCH, Inception-v3 and Resnet-v2. LCH and Inception-v3 were the best extractors for all classifiers.

Analyzing each extractor, the most suitable classifiers (bold values) were RF, *k*-NN and SVM. The RF classifier presented the best accuracies for all feature extractors. It is also possible to observe the best combinations (extractor and classifier pairs) through the highest accuracy results (highlighted by an asterisk).

The best combinations were ACC using *k*-NN and RF; BIC using *k*-NN and RF; CEDD using RF; GCH using *k*-NN and RF; JCD using RF; LBP using *k*-NN; LCH using *k*-NN, RF and SVM; Inception-v3 using *k*-NN, RF and SVM; Resnet-v2 using *k*-NN, RF and SVM. Such pairs have equivalent results in terms of accuracy. However, analyzing the dimensionality of the feature vectors, we can see that the best pairs would be BIC with kNN and BIC with RF. The BIC extractor enables to obtain high (or equivalent) accuracies with a smaller number of features than the others.

We also compared the best result obtained by (handcrafted or deep) features combined with traditional classifiers, against the results achieved by the end-to-end classification process. Regarding the traditional classifiers, the best accuracy was up to 99.33% obtained by the BIC extractor with the RF classifier (see Table 3). The end-to-end classification processes reached accuracies of 99.89%



(a)

True label \ Predicted label	C1	C2	C3	C4	C5	C6	C7	C8
C1	1,000	0	0	0	0	0	0	0
C2	0	991	0	0	10	0	0	0
C3	6	0	993	0	0	0	1	0
C4	0	0	0	1,000	0	0	0	0
C5	0	28	0	11	955	0	5	0
C6	0	0	0	0	1,000	0	0	0
C7	0	0	0	0	0	0	1,000	0
C8	0	0	0	0	0	23	0	977

(b)

True label \ Predicted label	C1	C2	C3	C4	C5	C6	C7	C8
C1	996	0	4	0	0	0	0	0
C2	0	969	0	0	29	0	3	0
C3	23	0	975	0	2	0	0	0
C4	0	0	1	998	0	0	1	0
C5	0	33	0	15	945	0	5	1
C6	0	0	0	0	0	994	0	6
C7	0	0	2	0	4	0	994	0
C8	0	0	0	0	0	41	0	959

(c)

True label \ Predicted label	C1	C2	C3	C4	C5	C6	C7	C8
C1	1,000	0	0	0	0	0	0	0
C2	0	967	0	0	34	0	0	0
C3	0	0	999	0	1	0	0	0
C4	0	0	0	1,000	0	0	0	0
C5	0	2	0	0	989	0	8	0
C6	0	0	0	0	0	1,000	0	0
C7	0	0	0	0	0	0	1,000	0
C8	0	0	0	0	0	5	0	995

(d)

True label \ Predicted label	C1	C2	C3	C4	C5	C6	C7	C8
C1	988	12	0	0	0	0	0	0
C2	0	972	0	8	1	19	0	0
C3	0	0	1,000	0	0	0	0	0
C4	0	0	9	988	0	3	0	0
C5	0	0	0	0	919	81	0	0
C6	2	2	0	66	7	923	0	0
C7	0	0	0	0	0	0	889	111
C8	0	0	0	0	0	0	194	806

Fig. 3. Confusion matrices using the extractor-classifier pairs: (a) GCH-k-NN, (b) BIC-J48, (c) BIC-RF, (d) Inception-v3-SVM.

and 99.71% by the Inception-v3 and Resnet-v2, respectively. However, they are more costly because they have to learn a higher number of weights.

Analyzing the confusion matrices (Figs. 3(a)–(d)), it is possible to note that all pairs (descriptor/classifier) presented good behavior. The confusion matrix using the BIC extractor and the RF classifier (see Fig. 3(c)) shows one of the best results. There is still some confusion between classes C_2 (Non-standard Vanilla) and C_5 (Standard Candy) as presented by the other matrices, but in overall its accuracy is more robust (i.e. higher concentration of values in the main diagonal of the matrix). For instance, Fig. 3(a) (pair GCH/k-NN) predicted wrong all samples from the C_6 class (i.e. Non-standard Candy, labeling them as class C_5 – Standard Candy). Other matrices like the one illustrated in Fig. 3(d) presented confusion between classes ($trueLabel - predictedLabel$), such as: C_7-C_8, C_5-C_6, C_6-C_4. However, these were minor misclassifications (i.e. few samples among one thousand) and we were capable of obtaining good accuracies.

5 Conclusion

We presented a methodology for application in a real food industry, more specifically related to the analysis and quality control in the production of biscuits. To evaluate our proposed methodology, we performed an extensive experimental analysis considering several handcrafted and deep features with different traditional supervised classifiers, and against end-to-end convolutional neural network architectures.

From the results, we can observe that our methodology achieves accuracies of up to 99%. Despite the hype and the results presented by the state-of-the-art CNN architectures, analyzing the most cost-effective techniques (e.g. in terms of accuracy and dimensionality of the feature vectors), the most successful combination was using the BIC extractor and the RF classifier.

As future work, we intend to use other image databases, considering different types of classes (i.e. distinct types of biscuits and quality standards - damages). We also intend to use other CNN architectures and deep learning techniques. For example, data augmentation and GANs to provide more samples and improve the learning process.

Acknowledgements. This work has been supported by CNPq (grants #431668/2016-7, #422811/2016-5); CAPES; Fundação Araucária; SETI; PPGI; and UTFPR.

References

1. Arwan, A.: Determining basis test paths using genetic algorithm and J48. Int. J. Electr. Comput. Eng. **8**(5), 3333–3340 (2018)
2. Bidoia, F., Sabatelli, M., Shantia, A., Wiering, M.A., Schomaker, L.: A deep convolutional neural network for location recognition and geometry based information. In: Proceedings of the 7th International Conference on Pattern Recognition Applications and Methods, pp. 27–36. SciTePress (2018)
3. Biship, C.M.: Pattern Recognition and Machine Learning (Information Science and Statistics), 1st edn. Springer, New York (2007)
4. Breiman, L.: Random forests. Mach. Learn. **45**(1), 5–32 (2001). https://doi.org/10.1023/a:1010933404324
5. Brosnan, T., Sun, D.-W.: Improving quality inspection of food products by computer vision-a review. J. Food Eng. **61**(1), 3–16 (2004)
6. Chatzichristofis, S.A., Boutalis, Y.S.: Fcth: fuzzy color and texture histogram - a low level feature for accurate image retrieval. In: 9th International Workshop on Image Analysis for Multimedia Interactive Services, pp. 191–196 (2008)
7. Chatzichristofis, S.A., Boutalis, Y.S.: CEDD: color and edge directivity descriptor: a compact descriptor for image indexing and retrieval. In: Gasteratos, A., Vincze, M., Tsotsos, J.K. (eds.) ICVS 2008. LNCS, vol. 5008, pp. 312–322. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-79547-6_30
8. Cubeddu, A., Rauh, C., Delgado, A.: Hybrid artificial neural network for prediction and control of process variables in food extrusion. Innov. Food Sci. Emerg. Technol. **21**, 142–150 (2014)