IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 28 October 2022, accepted 12 November 2022, date of publication 21 November 2022,
date of current version 28 November 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3223710

RESEARCH ARTICLE

Automatic Monitoring Cheese Ripeness Using Computer Vision and Artificial Intelligence

ANDREA LODDO[1], CECILIA DI RUBERTO[1], GIULIANO ARMANO[1], AND ANDREA MANCONI[2]
[1]Department of Mathematics and Computer Science, University of Cagliari, 09124 Cagliari, Italy
[2]Institute of Biomedical Technologies, National Research Council, CNR, Segrate, 20054 Milan, Italy

Corresponding author: Andrea Loddo (andrea.loddo@unica.it)

ABSTRACT Ripening is a very important process that contributes to cheese quality, as its characteristics are determined by the biochemical changes that occur during this period. Therefore, monitoring ripening time is a fundamental task to market a quality product in a timely manner. However, it is difficult to accurately determine the degree of cheese ripeness. Although some scientific methods have also been proposed in the literature, the conventional methods adopted in dairy industries are typically based on visual and weight control. This study proposes a novel approach aimed at automatically monitoring the cheese ripening based on the analysis of cheese images acquired by a photo camera. Both computer vision and machine learning techniques have been used to deal with this task. The study is based on a dataset of 195 images (specifically collected from an Italian dairy industry), which represent Pecorino cheese forms at four degrees of ripeness. All stages but the one labeled as "day 18", which has 45 images, consist of 50 images. These images have been handled with image processing techniques and then classified according to the degree of ripening, i.e., 18, 22, 24, and 30 days. A 5-fold cross-validation strategy was used to empirically evaluate the performance of the models. During this phase, each training fold was augmented online. This strategy allowed to use 624 images for training, leaving 39 original images per fold for testing. Experimental results have demonstrated the validity of the approach, showing good performance for most of the trained models.

INDEX TERMS Cheese ripening, image analysis, image processing, machine learning, image classification, deep learning.

I. INTRODUCTION

Dairy products have a high commercial value in the food industry, even considering that they are a source of proteins, calcium, and micro-nutrients with beneficial effects on bone and muscle health. In addition, thanks to their probiotic content, they increase the health of the digestive tract and positively influence the microbiome.

Among dairy products, cheese is a popular food produced and consumed in many parts of the world. The cheese quality depends on several characteristics, including chemical components, internal structure, physical properties, and other attributes such as oxygen and dielectric properties [1]. Other important aspects that determine the quality of the final product are sensory aspects stimulated by specific properties and components of the cheese. Not incidentally, the transformations of the milk constituents that affect the cheese quality occur during the ripening phase. In particular, several biochemical changes (lipolysis and proteolysis) that occur during this process significantly affect the cheese's flavor, aroma, and texture [2]. Consequently, monitoring the degree of ripeness is a mandatory step in the product quality assurance process [3]. Unfortunately, accurately determining the degree of maturation of the cheese is not trivial, even when considering only a specific product. The point here is that this process is not entirely predictable or controllable, being influenced by many factors –including season, the origin of the milk, processing steps, and temperature of the storage places [4]. On the other hand, an error in determining the maturation level of cheese typically has many negative consequences. In particular, for soft cheeses, the decision to place them on the market must be made within hours,

The associate editor coordinating the review of this manuscript and approving it for publication was Gianluigi Ciocca.

as the ripening process usually takes a few weeks. In either case (i.e., cheese not mature enough or cheese too mature), a product of lower quality than expected is sent to the market, with potential risks for the company in terms of income or prestige (also note that keeping aged cheeses in the dairy has warehouse costs). Visual inspection and weight checking are the conventional methods adopted in dairy industries to assess cheese ripeness. In both cases, despite the training of the personnel involved, there is room for occasional errors, which can result in costs for the companies. Moreover, checking an entire batch of cheese with these methods can be very time-consuming, so most often, only some samples are checked out. For all these motivations, the dairy sector's interest is growing in adopting cutting-edge technologies capable of effectively monitoring the maturation process.

In recent years, non-invasive techniques for monitoring cheese ripening have been adopted, typically based on the analysis of physicochemical, chromatographic, and electrophoretic characteristics. However, these techniques are time-consuming and expensive [5]. Other non-invasive methods based on spectroscopic techniques have also been proposed in the literature. Without claim to be exhaustive, some works are recalled from now on. Dufour et al. [6] analyzed the mid-infrared and fluorescence of sixteen different kinds of cheese collected at four ripening times. The similarity maps obtained by applying principal component analysis (PCA) to the acquired spectra demonstrated the feasibility of discriminating the cheeses according to their ripening times. FT-NIR (Fourier Transform Near-Infrared) and FT-IR (Fourier Transform Infrared) spectroscopy have been applied in [7] to study the shelf-life of Crescenza cheese. Spectral data were acquired from cheese samples collected at different times for twenty days. The PCA applied to this data detected the decrease in the "freshness" of Crescenza and defined the critical day during the shelf-life period. Del Campo et al. [8] used mid-infrared spectroscopy to study the characterization of the ripening stages of Emmental cheeses. Using PCA, the authors could discriminate among four categories of cheese ripening. Specifically, these groups include samples ripened during i) 21, 27, and 34 days, ii) 51 and 58 days, iii) 65 days, and iv) about 85 days (i.e., the samples as found at the end of the ripening process). Soto-Barajas et al. [9] proposed a study aimed at predicting cheese ripening and the types of milk mixtures used therein. Artificial neural networks (ANNs) were designed and trained with data reporting those cheeses' fatty acid composition and NIR spectra. The ANN model was able to predict the ripening of cheeses acquired in a period of six months.

Among the various food industry-related tasks in which we can frame our work, the spectrum of tasks addressed is extremely diverse. The problem of food quality and authenticity determination is a hot topic [10]. In this context, Jahanbakhshi et al.proposed methods based on computer vision and deep learning techniques to detect adulteration in turmeric powder [11], [12], while Al-Sarayreh [13] proposed a novel method based on the combination of spectral and textural information from red meat images to identify possible adulteration. Other important topics are certainly food recognition, retrieval, and classification [14], [15], [16] as a potential help in making recipes [17], for example, or food calorie estimation [18], [19].

This work is primarily motivated by the need to implement a non-invasive approach able to automatically and accurately determine the degree of ripeness of the cheese. This approach has the advantage of automating cheese control, eliminating common human errors that can lead to placing a non-quality product on the market. All the above mentioned aspects are addressed in this work. The goal is to implement an automated system to ensure the production of a quality product.

This paper proposes a new non-invasive approach for monitoring the cheese ripening process using advanced computer vision (CV) and machine learning (ML) techniques. Specifically, cheese images acquired with an ordinary camera are processed with CV techniques and then classified using relevant ML algorithms.

Our original contribution is fivefold: *i*) four different categories of visual handcrafted (HC) descriptors were extensively studied and compared; *ii*) the classification performance of different families of ML classifiers was analyzed and compared; *iii*) ten different convolutional neural networks in an end-to-end deep learning (DL) classification were evaluated; *iv*) a novel image processing pipeline, from the image acquisition to the ripeness classification, is proposed; *v*) an extensive comparative analysis in a novel domain and an effective pipeline to solve the task is proposed.

To the best of our knowledge, this is the first study aimed at evaluating cheese ripening by analyzing images acquired through a photo camera.

The remainder of this article is organized as follows. Materials and methods are described in Section II. Experimental settings and the corresponding results are reported in Section III. A discussion of the results follows in Section IV, together with the challenges raised by this task. Finally, conclusions are drawn in Section V.

II. MATERIALS AND METHODS

The automated monitoring of the cheese ripeness was approached as a supervised classification problem, characterized by the need to correctly map a cheese image to its *actual* maturation degree. Specific strategies based on DL and machine learning have been implemented and used to tackle this classification problem. In particular, DL algorithms were used with a twofold aim: i) to build classifier models and ii) to employ them for feature extraction. Both tasks were addressed using off-the-shelf convolutional neural network (CNN) architectures proposed in the literature. Beyond the adoption of DL techniques and architectures, several classifier models were constructed from handcrafted features obtained by image processing techniques and features extracted from CNNs. As for the dataset used in this study, all images were collected in a local dairy industry.

TABLE 1. **Table reports the number of images for a ripening period of 18 days (i.e., *18_Aged*), 22 days (i.e., *22_Aged*), 24 days (i.e., *24_Aged*) and 30 days (i.e., *30_Aged*).**

Ripening time	Class label	# images
day 18	*18_Aged*	45
day 22	*22_Aged*	50
day 24	*24_Aged*	50
day 30	*30_Aged*	50

The dataset consists of cheese images representing four degrees of ripeness of a specific type of soft cheese.

This section is organized as follows: the image dataset is first described in Section II-A; then the pipeline aimed at preprocessing the dataset is described in Section II-B; and afterward the extracted features are described in Section II-C. Subsequently, the adopted DL and ML algorithms are briefly recalled in Section II-D and Section II-E. Finally, the performance measures used to assess the models are reported in Section II-F.

A. DATASET CHARACTERISTICS

The dataset used in this research was built by collecting images of a Pecorino cheese produced by a Sardinian (Italy) dairy company. The specific product is classified as soft cheese, as its maturing period reaches its completion in 20-25 days. The dataset was built with the support of the Sardinian agency for the implementation of regional agricultural and rural development programs (LAORE[1]).

The dataset consists of 195 images representing the selected Pecorino cheese forms at four degrees of ripeness, i.e., 18, 22, 24, and 30 days. All stages but the one labeled as ‘‘day 18’’, which has 45 images, consist of 50 images (see Table 1).

The highly trained staff of the dairy ensured that no acceleration or slowdown was observed in the selected forms. As a consequence, those labeled as ‘‘day k’’ represent that day in an ideal ripening process. Note that for this specific product, the ripening process completes on ‘‘day 24,’’ which means that the forms labeled as ‘‘day 18’’ and ‘‘day 22’’ should be considered insufficiently ripe. Conversely, those labeled ‘‘day 30’’ should be regarded as overripe.

The camera used to acquire images was a Nikon D750, with a CMOS 35.9 × 24.0 mm sensor and a resolution of 24 Mpixel. All dataset images have a resolution of $4,016 \times 6,016$ or $6,016 \times 4,016$. A sample image for each category is shown in Figure 1.

B. DATA PREPROCESSING

Data preprocessing consists of a pipeline of two steps, i.e., image segmentation, and cropping. As shown by the sample reported in Figure 1, a non-uniform background characterizes all the images. As not beneficial for the classification task, it has been removed by a proper segmentation process that allowed substituting the original background with a neutral and constant one.

Image segmentation - Initially represented in RGB color space, all images were first converted to HSV space, being more representative of the contrast between the cheese and background region. Then, the images were segmented by a threshold approach, taking the saturation channel as a reference. Specifically, the images were segmented (i.e., binarized) by choosing an automatic adaptive threshold based on the local average intensity (first-order statistics) in the neighborhood of each pixel [20]. Thanks to the binarization process, the object of interest, represented by the cheese region, was properly extracted by selecting the bounding box of the biggest region, i.e., the cheese region. This operation made it possible to preserve the information related to the cheese region without influences dictated by the uneven background.

Image cropping - Once the cheese region was segmented, an image was produced, whose dimensions were derived from the size of the region itself. This image is formed by a constant background (black color was chosen) and the segmented object of interest. The cropped images obtained from the ones depicted in Figure 1 are presented in Figure 2. The images thus generated are then provided to the next stage of analysis, which is the feature extraction, or directly used as input for a convolutional neural network – depending on the chosen classification approach.

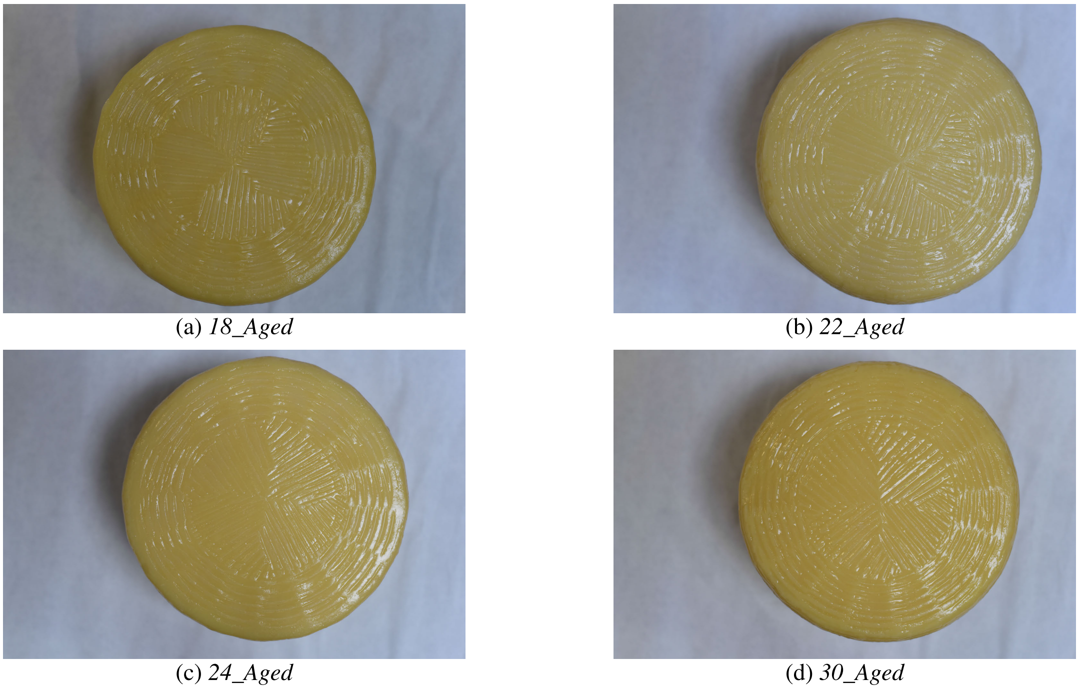
It is worth noting that an additional step has been applied while implementing the DL-based approaches. In this case, after cropping, the images have also been scaled according to the input shape (i.e., 224 × 224 or 299 × 299) of the adopted CNN architectures.

C. FEATURE EXTRACTION

Prior to this stage, a thorough image preprocessing pipeline allowed us to obtain representative images of the cheese acquired during the ripening process. Then different feature sets in various combinations were extracted and used to train the selected ML models. According to Putzu et al. [21], these features can be grouped into four main categories: *invariant moments*, *texture*, *color*, and *deep* features. The first three categories are handcrafted, while the last one includes all features extracted by means of CNNs. The selected categories represent a broad spectrum of existing descriptors in computer vision and are used in multiple activities, e.g., industry [22], biomedicine [23], [24], [25], agriculture [26], [27], video-surveillance [28], [29]. To the best of our knowledge, no ‘‘off-the-shelf’’ solution is available in the literature for this purpose. In fact, the existing methods presented in Section I address different types of cheese and ripening days. This prompted us to develop an ad hoc image processing pipeline that makes use of the above descriptors.

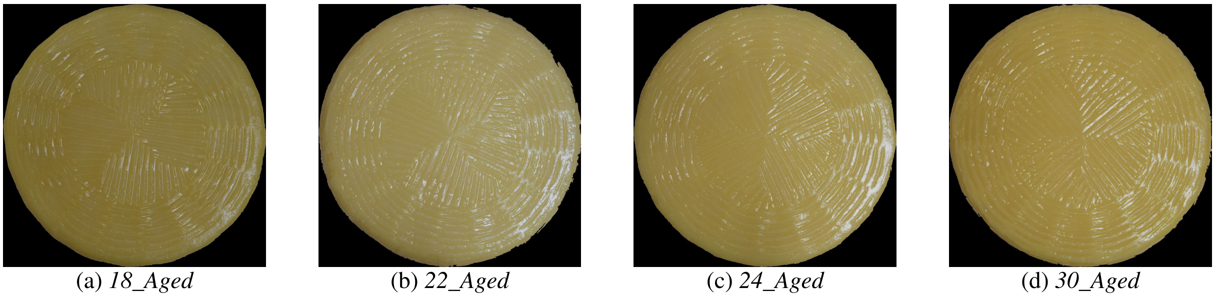
Invariant Moments - An image moment is a weighted average (i.e., the moment) of the image pixel intensities used to extract some properties from an image. Moments are used in image analysis and pattern recognition to describe objects

[1] https://www.sardegnaagricoltura.it



(a) *18_Aged* (b) *22_Aged* (c) *24_Aged* (d) *30_Aged*

FIGURE 1. Samples of cheese image for each class of ripeness.



(a) *18_Aged* (b) *22_Aged* (c) *24_Aged* (d) *30_Aged*

FIGURE 2. Samples of cropped cheese image for each class of ripeness.

after segmentation. In this work, four types of moments were used (i.e., Hu, Zernike, Legendre, and Chebyshev). Let us briefly summarize them.

- *Hu* moments (HM) have been proposed by Hu [30] to solve pattern recognition problems. They are invariant to translation, scale, and rotation changes.
- *Legendre* moments (LM) are obtained using Legendre polynomials as the kernel function. First introduced by Teague [31], they belong to the class of orthogonal moments and can be used to attain a near zero value of redundancy measure in a set of moment functions. These moments can be used to highlight independent characteristics of the image [32].
- *Zernike* moments (ZM) are based on the Zernike polynomials, an orthogonal sequence of polynomials on the unit disk. These orthogonal moments are used to represent image properties without redundancy [33].
- *Chebyshev* moments (CH) were first proposed by Mukandan et al. [34]. Unlike Zernike and Legendre moments, they belong to the class of discrete orthogonal moments. Hence, the implementation of these moments does not involve any numerical approximation. They are derived from Chebyshev polynomials and can extract global features in an image by varying moment order [35]. Here, first and second-order Chebyshev moments (denoted as CH_fi and CH_se) were extracted.

All mentioned invariant moments have been calculated with an order ranging from 3 to 10, being aware that higher orders would only increase the computation time by adding features representative of irrelevant details or noise [35].

Texture features - The texture features evaluated in this proposal were focused on fine textures. In particular, the histogram of the Local Binary Pattern (LBP), converted to a rotation invariant form, viz. LBP_ri [36], was extracted and used as the feature vector. Also, thirteen Haralick features extracted from the Gray Level Co-occurrence Matrix (GLCM) [37] converted into rotation invariant features, viz. HAR_ri (see [37]). More specifically, four kinds of GLCM were computed, all with $d = 1$ and $\theta = [0^\circ, 45^\circ, 90^\circ, 135^\circ]$. As for LBP, a map in the neighborhood characterized by r and n equal to 1 and 8, respectively, was calculated.

Color features - The color histogram (Hist) and color autocorrelogram (AC) features were extracted as color features. The former describes the overall color distribution in the image, from which seven statistical descriptors were computed: mean, standard deviation, smoothness, skewness, kurtosis, uniformity, and entropy. For ease of analysis and computations, they were calculated from images converted to grayscale. The latter is used to find the spatial correlation of identical colors by encoding their spatial color distribution [38]. More specifically, the probability of finding two pixels of the same color at a distance d is saved. Four distance values were used for our experiments: $d = 1, 2, 3, 4$. Once extracted, the four probability vectors were concatenated to generate a single feature vector.

Deep features - The term refers to the features of an image extracted from the deep layers of a CNN. The deep features were extracted from off-the-shelf CNN architectures pretrained on the well-known natural image dataset ImageNet [39]. Specifically, these features represent the activation values obtained from one of the most significant layers at the end of the network [40], [41]. In this study, according to the architecture, deep features were extracted: i) from the penultimate layer, ii) from the last fully connected layer or iii) from the last pooling layer to obtain features that represent the learned global knowledge of the network (see Table 2).

Note that the fine-tuning strategy for the classification phase was not considered to preserve the generalization ability of the networks [42], [43].

D. DEEP LEARNING STRATEGIES

CNNs are powerful architectures that are increasingly and successfully used to address a variety of image classification problems in different domains [27], [44], [45]. In addition, they transformed the manual design of feature extraction into an automated process, being able to take advantage of the activations of their layers to obtain the learned features.

In this study, different CNN architectures have been used to build classifier models to address the problem at hand and to perform the feature extraction task. Features extracted by the selected CNNs were used to train different models according to the ML algorithm described in Section II-E. In particular, regarding the end-to-end deep learning strategy, pretrained models' weights on ImageNet were considered. They were optimized by fine-tuning the pretrained models with the common practice of freezing all layers except the last (three) fully-connected layers from training. With this solution, transfer learning was adopted to reduce the problem of insufficient training data and avoid the risk of overfitting [46]. The CNN architectures used in this study (see Table 2) are briefly recalled hereinafter.

AlexNet – Proposed by Krizhevsky et al. [47], AlexNet consists of a cascade of convolutional and max-pooling layers, ended by 3 fully-connected layers. It is the shallowest among the considered architectures, containing only 5 convolutions layers.

TABLE 2. The table reports some information related to the CNNs used in this study. In particular, their names and related references, number of trainable parameters, image input size, and the layer used for feature extraction are reported for each CNN.

Reference	Parameters	Input Shape	Feature Layer
AlexNet [47]	60M	224 × 224	Pen. FC
GoogLeNet [48]	5M	224 × 224	Loss3
ResNet-18 [49]	11.7M	224 × 224	Pool5
ResNet-50 [49]	26M	224 × 224	Avg. pool
ResNet-101 [49]	44.6M	224 × 224	Pool5
Inception-v3 [50]	21.8M	299 × 299	Last FC
Inception-ResNet-v2 [51]	55M	299 × 299	Avg. pool
DarkNet-53 [52]	20.8M	224 × 224	Conv53
DenseNet-201 [53]	25.6M	224 × 224	Avg. pool
EfficientNet B0 [54]	5.3M	224 × 224	Avg. pool

ResNet – This name is used to denote a set of deep architectures based on residual learning [49], composed of skip-connections or recurrent units between blocks of convolutional and pooling layers. Furthermore, the blocks are followed by a batch normalization [55]. In this work, three versions of ResNet have been used, i.e., ResNet-18, ResNet-50, and ResNet-101 (note that the specified number represents the network depth).

GoogleNet – In 2014, Szegedy et al. [48] proposed the GoogLeNet architecture, which is based on blocks of inception layers. Each block is a set of convolution layers, while the filters used can vary from 1 × 1 to 5 × 5, thus allowing multi-scale learning. GoogLeNet uses global average pooling instead of max-pooling.

Inception-v3 – The Inception-v3 [50] architecture is also based on the concept of inception layers but improves GoogLeNet through the use of factorized, smaller, and asymmetric convolutions. The Inception models are famous for their multi-branch architectures, having a set of filters (1 × 1, 3 × 3, 5 × 5, and so forth) that are merged with concatenation in each branch.

Inception-ResNet-v2 – Devised as a combination of ResNet and Inception architectures [51], multiple-sized convolutional filters are combined with residual connections in the Inception-Resnet block. Inception-ResNet-v2 consists of 4 max-pooling and 160 convolutional layers.

DarkNet – Mostly based on the existing concepts of inception and batch normalization, one of the versions of DarkNet embeds 53 convolutional layers. This architecture is the backbone network for the object detection method You Only Look Once (YOLO) [52].

DenseNet – Proposed with $L(L+1)/2$ connections [53], this architecture was introduced to overcome the fact that traditional CNNs had some layers L equal to the number of connections. For each layer, the outputs of all previous layers are used as input to the next layer. The number of filters used in each convolutional layer varies according to the growth rate parameter (viz. k). In this study, DenseNet-201 with $k = 32$ has been adopted.

EfficientNet – Proposed by Tan et al., EfficientNet [54] uses compound scaling to uniformly and efficiently scale width, depth, and resolution of the network. Eight versions of EfficientNet exist, from B0 to B7. In this study, B0 has been adopted.

E. MACHINE LEARNING STRATEGIES

Handcrafted and deep features have been separately used to feed the chosen ML classifiers, i.e., k-Nearest Neighbor (k-NN) [56], Support Vector Machine (SVM) [57], Decision Tree (DT) [58], Random Forest (RF) [59] and a two-layer MultiLayer Perceptron (MLP), with 10 hidden layers. These ML algorithms are briefly summarized hereinafter.

k-NN – To categorize an observation, a k-NN classifier uses a local strategy that involves the k nearest neighbor training examples, together with a voting policy that yields a prediction starting from the selected neighbors. In this proposal, an image is classified according to the nearest image found in the training set, meaning that k is set to 1. Euclidean distance has been used as a distance measure. Note that with $k = 1$, no voting strategy is actually required.

SVM – SVMs are binary classifiers that categorize observations according to their position with respect to a hyperplane drawn in accordance with the so-called "support vectors". By default, an SVM is linear; however, when no hyperplane can properly discriminate between negative and positive observations, the original problem can be projected into a multidimensional space employing suitable kernel functions. In this proposal, a Gaussian radial basis function (RBF) has been used as the kernel. Note that multiclass problems can be dealt with SVMs using the *One-vs-Rest* (OvR) approach.

DT – Downstream of training, a DT can predict the category of observation by simply looking at the set of embedded *if-then* rules that have been inferred from training data. The deeper the tree, the more complex the decision rules.

RF – An RF is an ensemble of DTs. To ensure diversity among the DTs, ad-hoc rules are enforced during training. As a result, RFs are typically robust concerning the imbalance of data, allowing better control overfitting and better generalizations. In this work, the number of DTs has been set to 100.

F. PERFORMANCE MEASURES

The classification performance has been measured in terms of *accuracy*, *precision*, *specificity*, *sensitivity*, *F1-score*, and *Matthews Correlation Coefficient (MCC)*. Straightforward definitions of these measures for binary classification problems are given hereinafter, followed by their generalizations for multiclass problems.

1) STANDARD DEFINITIONS FOR BINARY CLASSIFICATION PROBLEMS

An example (say e) is characterized by a pair $\langle i, t\rangle$, where i is a list of feature values and t is the assigned category (i.e., target category). A dataset D is defined as a set of examples. When the number of target categories in D is 2, we face a binary problem. In this case, the categories found therein can be called *negative* and *positive*. To measure the performance of a binary classifier on a dataset D, each instance occurring therein will be labeled as *negative* or *positive*, depending on the classifier's output. According to the classification outcome and the actual target value, an instance will increase one of the following values:

- *true negatives (TN)* - Number of instances belonging to the *negative* class that have been correctly predicted;
- *false positives (FP)* - Number of instances belonging to the *negative* class that have been incorrectly predicted.
- *false negatives (FN)* - Number of instances belonging to the *positive* class that have been incorrectly predicted;
- *true positives (TP)* - Number of instances belonging to the *positive* class that have been correctly predicted.

According to these quantities, the above-mentioned measures can be described as follows:

- *Precision* – Fraction of positive instances correctly classified among all instances classified as positive:

$$P = \frac{TP}{TP + FP} \tag{1}$$

- *Sensitivity* (or *recall* R) – Measures the ability of the classifier to predict the positive class against FN (also called true positive rate):

$$SEN = \frac{TP}{TP + FN} \tag{2}$$

- *Specificity* – Measures the ability of the classifier to predict the negative class against FP (also called true negative rate):

$$SPE = \frac{TN}{TN + FP} \tag{3}$$

- *F1-score* – Defined as the harmonic mean between *precision* and *recall*:

$$F1 = 2 \cdot \frac{P \cdot R}{P + R} \tag{4}$$

- *Accuracy* – Ratio between the number of instances correctly classified and the total number of instances:

$$ACC = \frac{TP + TF}{TP + TF + FP + FN} \tag{5}$$

Note that a different definition of accuracy may also be adopted when working with unbalanced datasets. It is called balanced or unbiased accuracy, which is defined as the mean between specificity and sensitivity. This alternative measure completely ignores the imbalance.[2] In symbols:

$$BA = \frac{SPE + SEN}{2} \tag{6}$$

In this work, balanced accuracy is used to measure the performance of classifiers, as some experimental settings generate an imbalance among the classes.

- *MCC* – MCC combines *TN*, *FP*, *FN*, and *TP*. Ranging from -1 to $+1$, MCC provides a high score only when the classifier shows a good performance in both categories:

$$MCC = \frac{TP{\cdot}TN - FP{\cdot}FN}{\sqrt{(TP+FP){\cdot}(TP+FN){\cdot}(TN+FP){\cdot}(TN+FN)}} \tag{7}$$

2) STANDARD DEFINITIONS FOR MULTICLASS CLASSIFICATION PROBLEMS

As pointed out, the cited measures can also be generalized to deal with multiclass classifiers. A straightforward strategy to meet this need is calculating the measures for each category using an OvR approach. Downstream of this process, the average value of each binary measure is calculated, thus yielding an informative value for the multiclass model. Three different averaging methods could be used – i.e., micro, macro and weighted. In this work, macro averaging has been adopted. In summarizing, for a classification problem on K classes, the measures with macro averaging are calculated as follows:

- *Macro Average Precision* (with P_k denoting per-class k *precision*):

$$P = \frac{\sum_{k=1}^{K} P_k}{K} \tag{8}$$

- *Macro Average Sensitivity* (with SEN_k denoting per-class k *sensitivity*):

$$SEN = \frac{\sum_{k=1}^{K} SEN_k}{K} \tag{9}$$

- *Macro Average Specificity* (with SPE_k denoting per-class k *specificity*):

$$SPE = \frac{\sum_{k=1}^{K} SPE_k}{K} \tag{10}$$

- *Macro Accuracy* (with ACC_k denoting per-class k *accuracy*):

$$ACC = \frac{\sum_{k=1}^{K} ACC_k}{K} \tag{11}$$

- *Macro Average F1-score* (with P and R denoting *macro average precision-recall* and *macro average recall*):

$$F1 = 2 \cdot \frac{P \cdot R}{P + R} \tag{12}$$

Multiclass MCC and BA were not calculated with macro averaging. Hence, their reformulation is separately provided.

TABLE 3. This table reports the number of images for the three classes problem. Images acquired for a ripening period of 18 days and 22 days were grouped together to represent nonmature cheese (i.e., *unripened*). Images acquired on the 24th day of ripening represent the ideal ripeness (i.e., *ideal*). Too mature cheese is acquired on the 30th day of ripening (i.e., *over-ripened*).

Ripening time	Class label	# images
18 and 22 days	not mature yet, i.e., *unripened*	95
24 days	mature, i.e., *ideal*	50
30 days	too mature, i.e., *over ripened*	50

- MCC directly takes into account all categories of a multiclass confusion matrix.

$$MCC = \frac{c \cdot s - \sum_{k}^{K} p_k \cdot t_k}{\sqrt{(s^2 - \sum_{k}^{K} p_k^2) \cdot (s^2 - \sum_{k}^{K} t_k^2)}} \tag{13}$$

  where
  - $c = \sum_{k}^{K} C_{kk}$ represents the total instances correctly predicted for each class k;
  - $s = \sum_{i}^{K} \sum_{j}^{K} C_{ij}$; represents the total number of instances
  - $p_k = \sum_{i}^{K} C_{ki}$; represents the total prediction for the class k
  - $t_k = \sum_{i}^{K} C_{ik}$. represents the times that class k truly occurred
- *Balanced Accuracy* is calculated as follows:

$$BA = \frac{\sum_{k=1}^{K} SEN_k}{K} \tag{14}$$

  Note that SEN_k denotes the per-class k *sensitivity*. Hence, multiclass BA coincides with the definition of macro sensitivity.

III. RESULTS

In this section, experimental results are presented. The experimental setup is described in Section III-A, whereas the results achieved for ML- and DL-based strategies are reported in Section III-B and in Section III-C.

A. EXPERIMENTAL SETUP

Experiments have been carried out on a workstation equipped with the following hardware: Intel(R) Core(TM) i9-8950HK @ 2.90GHz CPU, 32 GB RAM, and an NVIDIA GTX1050 Ti GPU with 4GB of memory. All the implementations and experimental evaluations have been realized with MATLAB R2021b.

Initially, the preprocessed dataset has been used to build classifier models able to map a cheese image to one of the four ripening times (i.e., *18_Aged*, *22_Aged*, *24_Aged*, and *30_Aged*). Then, the class labels were reduced to three to better take into account the needs of the dairy company, for which the most relevant aspects were to identify whether a cheese form is not mature yet (days 18 and 22), mature (day 24), or too mature (day 30) (see Table 3). Finally, the selected ML and DL strategies have been tested for this new classification problem.

[2] Note that standard accuracy on imbalanced datasets may provide overoptimistic (or overpessimistic) estimations of the classifier's ability to discriminate between classes. To contrast this ''bias'', one may define accuracy as the mean performed over specificity and sensitivity, so that the class imbalance is not taken into account.

TABLE 4. This table reports the number of sample images used for training and test set for the four-class problem.

Class label	training samples	test samples
18_Aged	36	4
22_Aged	40	10
24_Aged	40	10
30_Aged	40	10

TABLE 5. This table reports the number of sample images used for training and test set for the three-classes problem.

Class label	training samples	test samples
unripened	76	19
ideal	40	10
over ripened	40	10

TABLE 6. Hyperparameters settings for CNNs fine-tuning.

Hyper-parameters	Setting
Solver	Adam
Max Epochs	100
Mini Batch Size	8
Initial Learn Rate	1^{e-4}
Learn Rate Drop Period	10
Learn Rate Drop Factor	0.1
L_2 Regularization	0.1

Image augmentation has also been applied before training the models to avoid overfitting. Specifically, the training set has been augmented by adding three images for each sample. The added images have been obtained by rotating the original ones with an angle of 90°, 180°, and 270°. 5-fold cross-validation has been adopted as a testing strategy. This strategy repeatedly performs training and testing on the same dataset while ensuring the statistical reliability of the results. In particular, at each step, the dataset is split into 80% for training and 20% for the test set (see Table 4 and Table 5). As for CNNs, their fine-tuning has been performed using the hyperparameters reported in Table 6.

B. MACHINE LEARNING BASED CLASSIFICATION

As introduced in Section II-E, five different classifiers have been used to conduct the ML-based experiments.

This section only reports the performance achieved by the best classifier models (i.e., the SVM with RBF kernel). The interested reader can find a complete report of the achieved performance for all the ML algorithms in Appendix . The results obtained with both the four- and three-classes classification tasks are reported in the following subsections.

1) RESULTS FOR THE FOUR-CLASSES CLASSIFICATION TASK

Apart from HM, models trained using the HC features achieved good performance with a low standard deviation for all invariant moments. As for SVM, the best performance has been achieved by training the model with LM features (see Table 7). In fact, every category of moments produced satisfactory results with a limited amount of features (66 for LM, with order 10, and for both CH, with order 8), except for HM and ZM. The former, with only 7 features, proved insufficiently representative (F1 equal to 32.2%). The second, whose best results were obtained with an order of 7 and 9 repetitions, performed satisfactorily despite performance lower than those measured for LM and CH.

TABLE 7. Best descriptor-class performance obtained with the SVM classifier on the four classes task. The table reports the performance in terms of accuracy, precision, sensitivity, specificity, F1 score, Matthew's correlation coefficient, and balanced accuracy.

Desc.	#features	ACC	P	SEN/BA	SPE	F1	MCC
HM	7	68.5±6.7	37.0±6.7	37.7±14.2	78.9±4.6	32.2±13.1	17.1±15.3
ZM	20	88.5±1.6	79.8±2.9	77.3±2.8	92.3±1.1	77.0±3.1	70.6±3.6
LM	66	**97.4±1.8**	**95.8±3.0**	**94.8±3.7**	**98.3±1.2**	**94.8±3.7**	**93.5±4.6**
CH_fi	45	96.2±3.3	93.1±5.7	92.3±6.5	97.4±2.2	92.2±6.6	90.1±8.4
CH_se	45	96.9±3.0	94.6±4.9	93.7±6.2	97.9±2.0	93.6±6.3	92.0±7.6
HAR_ri	26	74.9±6.7	53.2±11.6	50.4±13.6	83.1±4.5	48.6±12.5	34.5±17.0
AC	256	73.8±4.3	47.2±9.1	47.8±9.1	82.5±2.9	47.0±9.1	30.0±11.8

TABLE 8. Results for the four classes task obtained training the SVM classifier with features extracted from CNNs. The table reports the performance in terms of accuracy (ACC), precision (P), sensitivity (SEN), specificity (SPE), F1 score, Matthew's correlation coefficient, and balanced accuracy (BA).

Desc.	#features	ACC	P	SEN/BA	SPE	F1	MCC
AlexNet	4,096	94.6±4.6	90.4±8.4	89.3±9.0	96.4±3.0	88.7±10.0	86.2±11.9
ResNet-18	512	94.1±2.7	88.8±5.7	88.3±5.4	96.1±1.8	88.2±5.5	84.6±7.2
ResNet-50	2,048	95.6±3.5	92.0±7.0	91.3±7.0	97.1±2.3	91.1±7.4	88.7±9.3
ResNet-101	2,048	**97.4±2.0**	**95.8±3.0**	**94.9±3.9**	**98.3±1.4**	**94.9±4.0**	**93.6±4.9**
GoogLeNet	1,000	86.9±6.7	74.9±17.2	73.4±13.1	91.3±4.4	71.3±16.0	65.1±18.7
Inc.-v3	1,000	93.8±1.7	88.4±3.4	87.9±3.1	95.9±1.1	87.7±3.3	84.0±4.4
Inc.-Res.v2	1,536	94.6±1.1	90.4±1.6	89.3±2.1	96.4±0.7	89.4±2.0	86.2±2.5
DarkNet-53	1,000	95.9±3.3	92.3±6.7	91.9±6.7	97.3±2.2	91.7±6.8	89.3±8.9
DenseNet201	1,920	**97.2±1.1**	**95.0±2.0**	**94.5±2.1**	**98.1±0.7**	**94.5±2.1**	**92.9±2.7**
EffNetb0	2,048	92.6±4.0	86.6±6.2	85.2±7.7	95.0±2.7	84.6±8.2	80.8±9.7

As for the textural features, HAR_ri achieved the best performance, even though it is very far from the results obtained with the invariant moments (in fact, F1 is only 48.6%, which makes the corresponding model hard to use). Finally, the best color feature was the autocorrelogram, with the same issues described for HAR_ri. In addition, HAR_ri and AC have the highest standard deviation, as well as HM.

Table 8 reports the performance obtained by training the SVM with the features extracted by the CNNs. The best and second-best results are emphasized in bold. An SVM trained with features extracted by ResNet-101 and DenseNet201 achieved the best results. Although the SVM trained with the features extracted by ResNet-101 reached the highest absolute results, DenseNet201 gets similar (though slightly lower) results. However, it shows a significantly better standard deviation for each performance measure (about half that of ResNet-101). Finally, the number of features considered for the classification task is far higher with respect to the HC categories. In fact, ResNet-101 provided 2,048 features, and DenseNet201 1,920.

TABLE 9. Performance obtained with the SVM classifier trained with features extracted from CNNs on the three-classes task. The table reports the performance in terms of accuracy, precision, sensitivity, specificity, F1 score, Matthew's correlation coefficient, and balanced accuracy.

Desc.	#features	ACC	P	SEN/BA	SPE	F1	MCC
AlexNet	4,096	**94.9±4.8**	**93.5±7.1**	92.5±5.7	95.9±3.2	**92.4±6.9**	**89.0±9.8**
ResNet-18	512	90.1±9.0	84.7±16.8	82.2±18.3	91.0±8.8	81.7±20.6	75.5±24.4
ResNet-50	2,048	**98.6±1.4**	**98.0±2.2**	**98.0±2.2**	**98.9±1.2**	**98.0±2.1**	**96.9±3.3**
ResNet-101	2,048	94.5±6.2	92.6±7.7	**93.1±7.4**	**96.1±4.4**	92.0±8.7	88.8±12.0
GoogLeNet	1,000	90.1±5.3	85.6±7.0	85.7±8.3	92.2±4.4	85.3±7.7	77.7±12.0
Inc.-v3	1,000	92.5±3.9	89.4±5.8	89.4±5.0	94.2±2.8	88.9±5.5	83.5±8.3
Inc.-Res.v2	1,536	91.5±3.6	87.5±6.8	87.8±3.9	93.4±2.2	86.9±5.4	81.0±7.8
DarkNet-53	1,000	93.8±6.9	88.9±15.0	89.3±13.5	94.4±6.7	87.9±16.4	84.3±19.2
DenseNet201	1,920	92.5±5.6	91.2±3.2	87.9±12.1	93.7±5.9	87.8±10.2	83.3±12.3
EffNetb0	2,048	91.8±2.2	88.8±2.8	87.8±4.4	93.4±2.2	87.6±3.5	81.7±5.0

TABLE 10. Best descriptor-class performance obtained with the SVM classifier on the three-classes task. The table reports the performance in terms of accuracy, precision, sensitivity, specificity, F1 score, Matthew's correlation coefficient, and balanced accuracy.

Desc.	#features	ACC	P	SEN/BA	SPE	F1	MCC
HM	7	62.1±8.4	30.2±6.0	30.7±5.8	65.0±3.8	21.4±0.9	4.3±9.6
ZM	30	86.0±5.5	80.7±8.9	79.9±8.5	88.8±4.3	78.8±8.4	69.1±12.9
LM	66	**97.3±1.5**	**96.0±2.1**	**96.2±2.2**	**98.0±1.2**	**95.8±2.1**	**94.0±3.0**
CH_fi	66	95.6±1.5	94.3±2.3	93.5±3.5	96.4±1.6	93.3±2.4	90.4±3.3
CH_se	66	95.9±2.6	94.3±3.9	94.2±4.2	96.7±2.2	93.8±3.9	91.0±5.9
HAR_ri	26	63.4±7.3	39.1±10.0	38.5±8.2	71.4±4.2	36.4±10.0	10.6±13.3
AC	256	72.6±8.7	55.0±14.0	53.3±15.3	77.8±8.3	51.5±17.9	31.7±23.3

2) RESULTS FOR THE THREE-CLASSES CLASSIFICATION TASK

The results reported in Table 10 highlight a behavior consistent with that observed on the four-classes problem. In fact, the models trained using the LM features and CH features (both with order 10) obtained a performance above 90% for all assessed performance measures. More specifically, Legendre moments, with order 10, performed better, with F1 = 95.8%, MCC = 94.0%, and BA = 97.1%, with only 66 features. The models trained with HM and ZM features (best results with order 10 and repetitions 6) performed worse than the other invariant moments. Even with HAR_ri, they significantly worsen the results obtained on the four-classes task. Again, in this case, AC is the best color descriptor. However, it reached 51.5% of F1.

As for the results obtained by training the ML models with the deep features, the performance for this task is also relatively high, although in some cases, some performance measures have a significant drop (e.g., GoogLeNet and ResNet-18 obtained MCC of 77.7% and 75.5%, respectively). However, several architectures performed very well. Among all, the features extracted from ResNet-50 achieved the best performance (i.e., F1 = 98.0% and BA = 98.4%), with a considerably reduced standard deviation (below 2.3% for all measures, but MCC). However, AlexNet gets the second-best performance in terms of accuracy, precision, F1, and MCC, whereas ResNet-101 in terms of sensitivity, specificity, and balanced accuracy. Both best and second-best are emphasized in bold.

C. DEEP LEARNING BASED CLASSIFICATION

Section II-D introduced the CNN architectures investigated in this work. This section reports the results (see Table 11) obtained with a fine-tuning strategy adopted on the dataset under this study.

a: RESULTS FOR THE FOUR-CLASSES TASK

In this task, in which the deep features are used for training, the models perform well, with only a few exceptions. Although all CNN architectures achieved a good performance, three of them (i.e., Inception-ResNetv2, DarkNet-53, and ResNet-18) outperformed the others. Specifically, Inception-ResNetv2 and DarkNet-53 achieved the best accuracy, precision, and specificity, with the former showing a lower standard deviation for the three measures. DarkNet-53 achieved better sensitivity, MCC, and balanced accuracy, while Inception-ResNetv2 obtained the best F1 scores. In general, the performance measures of ResNet-18 were slightly lower than those obtained with Inception-ResNetv2 and DarkNet-53. However, a lower standard deviation has been obtained on all measures.

Supported by the results, ResNet-18 can be a potentially robust solution to the problem.

b: RESULTS FOR THE 3-CLASS TASK

Regarding the results obtained in this task, what immediately emerges is the very high variability among the various folds on which the CNN architectures were tested. The standard deviations have considerably high values. For example, the best architecture, Inception-v3, produced an F1 of 74.3%, with a standard deviation of 23.4%. The results of the remaining networks confirm the same behavior.

IV. DISCUSSION

The experiments showed a different behavior of the trained classifiers for the three- and four-classes tasks. In particular, the results on the four-classes task tend to be similar to those obtained for the three-classes task when approached with ML-based strategies. In fact, considering HC features, invariant moments alone allow for considerably high performance with low standard deviation. Moreover, although no single descriptor emerges as the absolute best for both tasks, it is important to point out how Legendre and Chebychev moments (both types) achieved high performance, very close to the absolute best regardless of the order examined. This aspect makes this category highly versatile, and representative of the classes of the problem studied.

The same trend for both classification tasks can be observed when features extracted by CNNs are used to train the SVM. As a general rule, the features extracted from ResNet-101 seem feasible to represent the classes of this study. More specifically, ResNet-101 features are the absolute best referring to the four-classes task, while they are the second-best in the three-classes one. However, in the last scenario, they permit comparable performance to the best features, i.e., ResNet-50. In any case, a residual network architecture may be suitable to address the problem.

Summing up the ML approach results, as seen from the performance measures presented in Appendix , the SVM is

TABLE 11. Performance obtained with the fine-tuned CNNs on the four-classes task.

Network	ACC	P	SEN/BA	SPE	F1	MCC
AlexNet	96.4±5.9	92.8±11.9	94.6±8.3	97.8±3.5	92.1±13.5	90.9±14.9
ResNet-18	**96.4±2.1**	**92.8±4.1**	**93.8±3.4**	**97.8±1.2**	**92.5±4.6**	**90.8±5.2**
ResNet-50	95.6±7.1	91.2±14.3	93.3±10.7	97.3±4.3	90.8±15.1	89.1±17.7
ResNet-101	95.9±5.8	91.8±11.9	93.7±9.0	97.5±3.5	91.3±12.8	89.7±14.8
GoogLeNet	94.6±4.3	89.2±8.3	92.2±4.2	96.9±2.1	87.8±11.0	86.4±10.4
Inc.-v3	95.9±3.6	91.8±7.0	93.5±4.6	97.5±2.0	91.3±8.2	89.7±8.6
Inc.-Res.v2	**96.7±4.8**	**93.4±9.4**	93.9±9.0	**98.0±2.8**	**92.8±10.6**	91.4±12.5
DarkNet-53	**96.7±5.4**	**93.4±10.8**	**95.2±7.3**	**98.0±3.1**	92.6±12.5	**91.7±13.5**
DenseNet201	94.6±7.8	89.1±15.8	91.2±11.8	96.8±4.5	87.9±18.4	85.9±20.4
EffNetb0	96.4±6.1	92.8±12.3	94.2±9.3	97.8±3.6	91.9±14.1	90.8±15.7

TABLE 12. Performance obtained with the fine-tuned CNNs on the three-classes task. The table reports the performance in terms of accuracy, precision, sensitivity, specificity, F1 score, Matthew's correlation coefficient, and balanced accuracy.

Desc.	ACC	P	SEN/BA	SPE	F1	MCC
AlexNet	82.2±16.3	81.1±16.8	76.3±23.9	90.2±8.8	73.3±24.7	69.2±28.1
ResNet-18	81.9±14.8	80.8±15.0	81.0±17.8	90.0±7.4	73.3±22.2	69.6±23.9
ResNet-50	81.9±16.2	81.4±16.6	76.8±22.8	90.4±8.1	73.0±24.4	69.6±27.0
ResNet-101	82.9±16.4	82.5±16.8	77.6±23.5	**91.0±8.4**	74.6±24.8	71.2±27.7
GoogLeNet	81.2±15.0	79.8±14.9	80.7±17.6	89.9±7.5	72.1±22.2	68.5±23.8
Inc.-v3	**82.6±15.6**	**82.1±16.1**	**82.0±18.7**	90.7±7.9	**74.3±23.4**	**71.3±25.5**
Inc.-Res.v2	81.9±15.6	81.1±16.1	76.4±22.3	90.2±7.7	73.1±23.7	69.2±26.1
DarkNet-53	81.9±15.0	81.4±15.4	75.4±22.3	90.1±7.1	72.9±23.3	68.8±25.2
DenseNet201	81.5±14.6	81.1±15.0	75.4±21.0	89.8±6.7	72.6±22.3	68.3±24.0
EffNetb0	82.2±15.5	81.8±15.9	76.4±22.1	90.4±7.5	73.6±23.6	69.7±25.8

the most suitable classifier when trained with HC features. The same is true for classifiers trained with features extracted from pretrained CNNs. All of the remaining four classification models can be within the performance of the SVM trained with features extracted from residual networks for both tasks.

In contrast, the results obtained with CNNs in classification significantly differ between the two tasks analyzed. In fact, as expressed above, the results of the CNNs on the four classes tend to be high, albeit with fairly high standard deviations. However, this issue does not plague ResNet-18, which has the best results from a standard deviation perspective and the second-best absolute results on various measures.

The scenario noticeably changes when analyzing the CNNs results on the three classes. In fact, unlike the previous task, standard deviations are very high, making this approach unfeasible with this task and leaving open questions and room for further investigations regarding the significance of the four different classes, how to address the problem derived from the fusion of classes *18_Aged* and *22_Aged*, and how the conditions of acquisition and subsequent preprocessing may have affected the data processed by CNNs.

Some relevant trends emerge, shifting attention to the overall results obtained by the classifiers used. First, as evidenced by Table 13 and Table 14, the best-performing classifier is the SVM in both configurations. Specifically, in the three-class configuration, the SVM obtains an F1 approaching 95% with both handcrafted and deep features, exceeding by two percentage points the F1 obtained by the best CNN, Inception-ResNetv2. In the four-class-class configuration, the SVM obtains 98.0% with features extracted from ResNet-50, and and 95.8% with Legendre moments, exceeding by more than 20 percentage points the best result obtained by the best CNN, ResNet-101 in this case. The Random Forest classifier, trained with the features extracted from DenseNet201 in the first configuration and with both types of features in the second configuration, also achieved comparable and better results than CNN. Second, Legendre moments proved to be the best handcrafted features for the SVM. In fact, in the first configuration, the performance is comparable. In one case, features extracted from ResNet-101 achieve 94.9% of F1, while Legendre's moments obtain 94.8%. However, the difference in the number of features is important. In the former case, ResNet-101 brings in 2,048 features, while Legendre's moments are just 66, with a significant computational advantage. In the second configuration, the difference is slightly more significant. Specifically, the features extracted from ResNet-50 allow an F1 of 98% with 2,048 features, while Legendre's moments stop at 95.8%, but with just 66 features. This demonstrates the high representative power of the descriptors examined, especially Legendre's moments, for the task.

The proposed method has also been compared with other solutions offered in the literature. Nevertheless, a direct comparison with the methods proposed in the literature cannot be implemented. In fact, to the best of our knowledge, the presented approach is the first designed to evaluate cheese ripening by analyzing images acquired through a photo camera. In contrast, other methods analyze chemical or spectroscopic parameters. Moreover, all the methods proposed in the literature focus on different types of cheese and ripening times.

In any case, some works have reported quantitative performance. More specifically, Del Campo et al. [8] proposed a PCA-based model that discriminates among four ripening stages of Emmental cheeses, using mid-infrared spectroscopy. This model achieved 87% cross-validation accuracy on 14 samples, and 57% test set accuracy on 14 samples. The key difference with our work is that the characterization of ripening was analyzed over a longer time frame. Also, Soto-Barajas et al. [9] proposed an ANN trained with data reporting the fatty acid composition and the NIR spectra of sixteen milk mixtures, obtaining 50% accuracy in discriminating among the types and 100% in classifying unknown cheese samples for the ripening time.

Despite the similarities that exist with other proposals, this work should be considered innovative, also for its ability to reach 98.0% classification accuracy in discriminating between the different phases of ripeness. Being non-invasive and end-to-end automatable, it is also useful for identifying problems that may arise within the cheese storage, for example, an inappropriate temperature.

As for future work, different margins of improvement are left. First, only the top layers of the CNNs were trained to take full advantage of the pretrained models and avoid overfitting issues related to the reduced size of the acquired dataset.

TABLE 13. **Best performance obtained with all the classifiers involved on the three-classes task. The table reports the performance in terms of accuracy, precision, sensitivity, specificity, F1 score, Matthew's correlation coefficient, and balanced accuracy. For each classifier, results obtained with the best handcrafted feature are shown on the top row. The result produced by using the features extracted from the best CNN are reported on the bottom row. The last row shows the best result produced by CNNs.**

Classifier	Desc./Net	#features	ACC	P	SEN/BA	SPE	F1	MCC
kNN	CH_se	15	92.3±3.0	85.9±5.7	84.6±6.2	94.9±2.0	84.4±6.3	80.0±8.0
	DarkNet-53	1,000	93.1±3.9	87.0±8.5	86.3±7.9	95.4±2.6	86.0±8.0	82.0±10.7
SVM	LM	66	**97.4±1.8**	**95.8±3.0**	**94.8±3.7**	**98.3±1.2**	**94.8±3.7**	**93.5±4.6**
	ResNet-101	2,048	**97.4±2.0**	**95.8±3.0**	**94.9±3.9**	**98.3±1.4**	**94.9±4.0**	**93.6±4.9**
RF	CH_se	15	92.3±2.4	86.6±3.9	84.7±5.1	94.9±1.6	84.2±5.4	80.3±6.0
	DenseNet201	1,920	96.4±3.3	94.0±5.6	92.9±6.5	97.6±2.2	92.7±6.8	91.0±8.3
DT	HAR_ri	26	88.5±4.7	78.6±10.3	77.1±9.4	92.3±3.1	75.2±11.4	69.8±12.9
	DenseNet201	1,920	89.0±4.8	79.6±9.6	78.1±9.4	92.6±3.2	78.2±9.2	71.4±12.6
ANN	ZM	23	73.3±0.6	39.6±1.7	47.8±1.2	82.1±0.4	34.7±1.1	29.3±1.8
	Inception-ResNetv2	1,536	73.3±1.7	42.0±1.8	47.8±3.5	82.1±1.1	35.8±2.5	30.4±4.1
CNN	Inception-ResNetv2	-	96.7±4.8	93.4±9.4	93.9±9.0	98.0±2.8	92.8±10.6	91.4±12.5

TABLE 14. **Best performance obtained with all the classifiers involved on the four-classes task. The table reports the performance in terms of accuracy, precision, sensitivity, specificity, F1 score, Matthew's correlation coefficient, and balanced accuracy. For each classifier, results obtained with the best handcrafted feature are shown on the top row. The result produced by using the features extracted from the best CNN are reported on the bottom row. The last row shows the best result produced by CNNs.**

Classifier	Desc./Net	#features	ACC	P	SEN/BA	SPE	F1	MCC
kNN	CH_se	15	90.1±2.5	85.6±4.4	84.8±3.8	91.7±2.1	84.9±4.0	77.0±6.1
	DarkNet-53	1,000	93.1±3.9	87.0±8.5	86.3±7.9	95.4±2.6	86.0±8.0	82.0±10.7
SVM	LM	66	**97.3±1.5**	**96.0±2.1**	**96.2±2.2**	**98.0±1.2**	**95.8±2.1**	**94.0±3.0**
	ResNet-50	2,048	**98.6±1.4**	**98.0±2.2**	**98.0±2.2**	**98.9±1.2**	**98.0±2.1**	**96.9±3.3**
RF	CH_se	10	90.1±7.6	86.2±10.9	84.5±12.9	91.6±6.8	84.7±12.3	77.1±18.3
	DenseNet201	1,920	95.2±2.2	93.0±4.0	92.6±3.5	96.1±1.7	92.7±3.6	88.9±5.4
DT	LBP	72	86.7±5.0	81.3±6.8	80.9±4.5	90.2±2.8	79.2±6.5	71.2±8.6
	DarkNet-53	1,000	86.7±5.7	82.3±6.5	80.0±10.4	89.7±4.6	79.5±9.6	70.5±13.1
ANN	LM	28	79.8±2.5	58.5±2.1	61.6±4.4	81.2±2.9	54.5±2.7	44.6±5.4
	ResNet-101	2,048	79.5±1.2	54.2±1.1	62.5±0.9	82.7±1.5	52.8±1.1	43.3±2.4
CNN	ResNet-101	-	82.9±16.4	82.5±16.8	77.6±23.5	91.0±8.4	74.6±24.8	71.2±27.7

Although the results were inferior to those obtained with ML classifiers, we plan to investigate targeted training on different layers of the CNNs, possibly even from scratch. Second, combining some of the extracted heterogeneous features to train the classifier models may further benefit the classifiers investigated so far. Third, a most informed feature selection strategy could greatly help the generalization process for most of the selected ML techniques. Fourth, adopting Generative Adversarial Networks (GANs) to extend the dataset could expand the dataset's variance and allow the proposed methods to be hardened against differences in texture, illumination, shape, etc. Fifth, another factor that can be taken into account is how much the background of the images may affect both feature extraction and the performance of the classification algorithms. In this way, the system could be tuned to use the most appropriate one for the task at hand.

V. CONCLUSION

The proposed study addressed the issue of implementing a novel non-invasive methodology aimed at monitoring the cheese ripening process in order to automatically detect the maturation status. A novel methodology has been presented, which synergistically makes use of CV and ML to identify the cheese maturation status by analyzing images acquired with an ordinary photo camera. This methodology has been applied to specific Pecorino soft cheese produced by a Sardinia dairy company.

A dataset consisting of images of cheeses with an ideal ripeness, in contrast with other images representing unripened and over-ripened cheeses, has been built to test the proposed methodology. Different classifier models have been built, based on ML and DL strategies. These models allowed us to perform a comparative study on *i*) heterogeneous handcrafted descriptors, *ii*) deep features extracted by different pre-trained CNNs, *iii*) classification performance of DL and ML algorithms. The best classification performance has been obtained with models trained using ML strategies, with particular reference to the SVM classifier.

As it can be easily applied to inspect an entire batch of cheese, the proposed methodology can be beneficial to abandoning the habit of sampling cheese and guessing the status of the rest of the cheese forms. For its non-invasiveness and considering that learned models can produce a result with a time order of milliseconds, it lends itself to easy integration into industrial production. Moreover, being related only to image analysis, it could be applied to different types of cheese with an easy customization process.

Taking into account what has been said so far and expressed in Section IV, we consider the proposed approach suitable for deployment in real-time systems, thanks to the short inference times. In fact, the proposed method can be

TABLE 15. **Performance obtained with the kNN classifier trained with HC and deep features for the four-classes task. The table reports the performance in terms of accuracy, precision, sensitivity, specificity, F1 score, Matthew's correlation coefficient, and balanced accuracy.**

Desc.	#features	ACC	P	SEN/BA	SPE	F1	MCC
HM	7	70.3±7.3	36.6±15.3	40.9±15.4	80.1±5.0	37.6±15.1	19.4±20.0
ZM	10	84.1±3.1	72.9±4.5	68.6±6.5	89.3±2.1	68.3±6.7	59.6±7.8
LM	28	91.3±2.1	83.8±4.2	82.4±4.5	94.2±1.4	82.1±4.7	77.1±5.7
CH_fi	21	90.3±4.1	82.1±7.5	80.4±8.5	93.5±2.7	79.7±9.2	74.5±10.7
CH_se	**15**	**92.3±3.0**	**85.9±5.7**	**84.6±6.2**	**94.9±2.0**	**84.4±6.3**	**80.0±8.0**
CLBP216	8,232	87.2±2.4	77.1±4.4	74.4±5.0	91.4±1.6	74.1±5.4	66.8±6.3
Haar	192	73.6±4.8	47.6±10.2	47.8±10.0	82.3±3.3	46.3±10.2	30.0±13.3
AlexNet	4,096	85.9±3.1	74.4±5.9	71.8±6.4	90.5±2.1	71.7±6.6	63.5±8.2
ResNet-18	512	86.9±3.9	78.1±7.7	73.4±7.8	91.2±2.6	73.1±7.3	66.4±10.1
ResNet-50	2,048	91.8±1.9	85.6±3.4	83.4±3.9	94.5±1.3	83.8±3.7	78.9±4.9
ResNet-101	2,048	91.3±2.8	83.0±6.0	82.6±5.6	94.2±1.8	82.6±5.8	77.0±7.6
GoogLeNet	1,000	80.3±5.5	64.1±9.4	60.2±10.6	86.8±3.6	60.0±10.3	48.6±13.5
Inception-v3	1,000	86.4±5.0	76.1±7.2	72.8±9.8	90.9±3.4	73.1±8.9	65.1±12.0
Inc.-ResNetv2	1,536	86.7±3.8	75.7±7.5	73.3±7.3	91.1±2.5	73.3±7.0	65.4±9.8
DarkNet-53	**1,000**	**93.1±3.9**	**87.0±8.5**	**86.3±7.9**	**95.4±2.6**	**86.0±8.0**	**82.0±10.7**
DenseNet201	1,920	90.8±2.8	85.0±4.7	81.4±5.8	93.8±1.9	81.7±5.8	76.8±7.1
EfficientNetb0	2,048	86.4±6.3	76.3±8.1	72.8±12.6	90.9±4.3	73.2±11.6	65.2±14.9

TABLE 16. **Performance obtained with the RF classifier trained with HC and deep features for the four-classes task. The table reports the performance in terms of accuracy, precision, sensitivity, specificity, F1 score, Matthew's correlation coefficient, and balanced accuracy.**

Desc.	#features	ACC	P	SEN/BA	SPE	F1	MCC
HM	7	82.6±5.6	67.0±9.6	65.4±11.6	88.4±3.8	63.9±11.1	54.2±14.1
ZM	10	85.4±3.7	72.1±8.6	71.1±7.3	90.2±2.5	70.2±7.9	61.7±10.2
LM	15	92.3±2.6	85.7±5.4	84.7±5.1	94.9±1.7	84.1±5.3	79.9±6.9
CH_fi	36	90.0±4.6	81.1±9.8	80.1±8.9	93.3±3.0	78.7±9.6	73.7±12.2
CH_se	**15**	**92.3±2.4**	**86.6±3.9**	**84.7±5.1**	**94.9±1.6**	**84.2±5.4**	**80.3±6.0**
LBP212	352	90.0±4.6	82.4±8.1	79.8±9.0	93.3±3.1	79.7±8.9	74.3±11.7
Hist	7	80.8±4.2	64.9±12.0	61.9±8.9	87.1±2.8	59.7±9.4	49.8±12.0
AlexNet	4,096	90.0±6.5	81.1±14.0	80.0±12.8	93.3±4.3	78.8±14.2	73.8±17.5
ResNet-18	512	92.3±2.4	85.6±5.0	84.8±4.6	94.8±1.6	84.6±5.1	80.0±6.4
ResNet-50	2,048	93.8±3.4	87.9±6.9	87.8±6.7	95.9±2.3	87.6±6.9	83.7±9.1
ResNet-101	2,048	93.8±2.9	89.4±4.7	87.9±5.7	95.9±1.9	87.0±6.6	84.3±7.3
GoogLeNet	1,000	84.9±6.7	69.1±16.6	69.3±13.0	89.9±4.4	67.8±14.8	59.3±18.5
Inception-v3	1,000	90.5±1.9	81.6±4.2	81.2±4.0	93.7±1.3	80.8±3.7	75.0±5.3
Inc.-ResNetv2	1,536	92.6±3.4	86.5±6.3	85.3±6.9	95.0±2.3	85.3±6.8	80.8±8.9
DarkNet-53	1,000	93.8±3.8	88.9±7.7	87.9±7.6	95.9±2.5	87.3±8.3	84.2±10.2
DenseNet201	**1,920**	**96.4±3.3**	**94.0±5.6**	**92.9±6.5**	**97.6±2.2**	**92.7±6.8**	**91.0±8.3**
EfficientNetb0	2,048	92.8±1.1	85.9±2.6	85.7±2.2	95.2±0.7	85.4±2.7	81.0±3.1

TABLE 17. **Performance obtained with the DT classifier trained with HC and deep features for the four-classes task. The table reports the performance in terms of accuracy, precision, sensitivity, specificity, F1 score, Matthew's correlation coefficient, and balanced accuracy.**

Desc.	#features	ACC	P	SEN/BA	SPE	F1	MCC
HM	7	78.7±4.7	58.5±11.3	57.6±9.8	85.8±3.1	56.4±10.3	43.7±13.3
ZM	9	79.5±2.7	59.8±7.3	59.0±5.6	86.3±1.8	57.5±6.1	45.5±7.5
LM	21	84.6±2.6	71.8±5.9	69.6±5.2	89.7±1.7	68.6±4.4	60.1±6.9
CH_fi	15	81.3±4.3	66.5±10.8	62.7±8.5	87.5±2.9	61.6±8.7	51.4±12.1
CH_se	21	83.6±4.8	69.3±13.2	67.4±9.4	89.0±3.2	66.4±10.6	57.2±13.9
HAR_ri	**26**	**88.5±4.7**	**78.6±10.3**	**77.1±9.4**	**92.3±3.1**	**75.2±11.4**	**69.8±12.9**
Hist	7	79.5±4.2	56.8±8.3	59.4±8.8	86.3±2.8	56.5±8.5	44.7±10.8
AlexNet	4,096	81.8±4.2	66.3±11.5	63.6±8.0	87.8±2.8	63.3±9.5	52.7±12.3
ResNet-18	512	82.8±3.7	66.9±8.5	65.6±7.4	88.6±2.5	65.2±7.8	54.7±10.3
ResNet-50	2,048	85.1±2.8	73.1±5.4	70.5±5.4	90.1±1.8	70.2±5.9	61.6±7.2
ResNet-101	2,048	84.1±4.7	68.2±10.7	68.3±9.3	89.4±3.2	67.5±10.1	57.7±13.0
GoogLeNet	1,000	76.4±8.4	54.4±18.8	52.6±17.0	84.3±5.6	52.2±17.5	37.6±23.2
Inception-v3	1,000	82.3±5.3	67.1±11.3	64.5±10.8	88.2±3.5	64.8±11.0	53.8±14.5
Inc.-ResNetv2	1,536	84.6±1.6	70.2±3.8	69.5±3.2	89.7±1.1	68.8±2.7	59.5±4.3
DarkNet-53	1,000	85.9±5.1	74.2±10.3	71.8±10.1	90.6±3.4	71.9±10.2	63.4±13.4
DenseNet201	**1,920**	**89.0±4.8**	**79.6±9.6**	**78.1±9.4**	**92.6±3.2**	**78.2±9.2**	**71.4±12.6**
EfficientNetb0	2,048	84.1±3.0	70.7±3.2	68.2±6.4	89.4±2.0	68.2±5.6	58.6±6.9

TABLE 18. **Performance obtained with the MLP classifier trained with HC and deep features for the four-classes task. The table reports the performance in terms of accuracy, precision, sensitivity, specificity, F1 score, Matthew's correlation coefficient, and balanced accuracy.**

Desc.	#features	ACC	P	SEN/BA	SPE	F1	MCC
HM	7	70.8±3.8	35.5±6.3	42.6±7.5	80.5±2.5	29.6±5.5	22.6±8.9
ZM	**23**	**73.3±0.6**	**39.6±1.7**	**47.8±1.2**	**82.1±0.4**	**34.7±1.1**	**29.3±1.8**
LM	28	72.8±2.1	35.5±2.1	46.8±4.5	81.9±1.5	31.9±1.8	26.0±3.6
CH_fi	66	72.6±2.7	35.5±3.8	46.2±5.8	81.8±1.9	30.9±2.6	25.6±3.3
CH_se	66	73.6±1.7	37.3±3.2	48.3±3.7	82.4±1.2	33.4±1.2	28.5±1.7
LBP212	352	72.1±3.2	40.3±4.6	45.1±6.8	81.2±2.1	33.5±5.6	27.1±8.4
Hist	7	71.0±3.8	35.6±4.0	43.0±8.1	80.7±2.7	28.9±5.7	22.5±7.0
AlexNet	4,096	70.3±4.7	33.1±6.6	41.7±9.2	80.2±3.1	28.8±6.9	20.3±11.7
ResNet-18	512	72.8±2.1	39.1±4.2	46.8±4.5	81.8±1.4	34.0±4.0	27.8±5.9
ResNet-50	2,048	72.8±1.4	39.2±1.4	46.9±2.8	81.8±1.0	34.3±1.4	27.7±3.2
ResNet-101	2,048	73.1±1.8	39.8±3.0	47.4±3.6	81.9±1.2	34.8±2.6	28.6±5.1
GoogLeNet	1,000	68.2±3.2	29.2±5.5	37.5±5.9	79.0±1.9	24.2±6.2	15.3±7.2
Inception-v3	1,000	71.3±1.5	38.4±4.1	43.6±2.8	80.7±0.9	32.0±2.8	24.7±4.9
Inc.-ResNetv2	**1,536**	**73.3±1.7**	**42.0±1.8**	**47.8±3.5**	**82.1±1.1**	**35.8±2.5**	**30.4±4.1**
DarkNet-53	1,000	71.8±3.3	36.6±5.9	44.9±6.5	81.1±2.1	32.2±5.4	24.6±9.2
DenseNet201	1,920	73.3±1.7	41.0±2.7	47.9±3.4	82.1±1.1	35.6±2.6	29.7±4.8
EfficientNetb0	2,048	72.1±1.9	39.9±4.1	45.3±3.8	81.2±1.2	33.6±3.2	26.8±5.8

TABLE 19. **Performance obtained with the kNN classifier trained with HC and deep features for the three-classes task. The table reports the performance in terms of accuracy, precision, sensitivity, specificity, F1 score, Matthew's correlation coefficient, and balanced accuracy.**

Desc.	#features	ACC	P	SEN/BA	SPE	F1	MCC
HM	7	62.7±12.1	42.5±18.4	41.5±15.2	71.0±8.9	40.7±15.6	13.3±26.0
ZM	**14**	**80.2±7.4**	**71.9±12.3**	**71.1±8.6**	**84.1±5.4**	**70.3±10.1**	**55.8±15.9**
LM	21	88.4±3.7	82.9±6.2	83.0±6.6	90.5±3.4	82.5±6.0	73.4±9.4
CH_fi	28	88.0±5.5	82.8±8.0	83.0±9.3	90.4±4.9	82.2±8.9	73.2±13.2
CH_se	10	90.1±4.7	85.8±7.9	84.8±7.6	91.7±4.1	84.9±7.4	77.1±11.6
CLBP216	8,232	87.0±3.9	79.6±6.7	79.7±7.9	90.6±3.2	78.8±7.3	70.0±10.2
Haar	192	65.8±8.3	44.0±10.4	43.8±10.0	73.5±6.1	42.6±9.3	17.7±16.7
AlexNet	4,096	82.9±4.4	76.6±5.3	74.6±8.1	86.4±4.1	74.4±6.4	61.8±10.6
ResNet-18	512	86.3±4.8	81.3±7.3	81.2±6.5	89.7±3.3	80.0±7.2	70.7±10.3
ResNet-50	2,048	89.7±3.2	85.0±5.0	85.4±5.6	92.1±2.5	84.7±5.2	77.1±7.6
ResNet-101	2,048	90.1±3.7	85.1±5.8	85.4±6.1	92.1±3.1	85.0±5.7	77.3±8.8
GoogLeNet	1,000	81.2±5.0	72.2±6.8	71.2±8.4	85.5±3.9	70.4±7.6	56.9±11.1
Inception-v3	1,000	83.2±5.5	76.7±5.7	74.6±12.0	86.8±5.5	73.8±9.5	62.1±13.9
Inc.-ResNetv2	**1,536**	**85.6±5.4**	**79.4±6.3**	**79.3±8.0**	**89.0±4.0**	**78.0±7.7**	**68.2±10.8**
DarkNet-53	1,000	90.8±5.7	87.0±7.7	88.6±6.8	93.2±4.1	86.9±8.0	80.7±11.5
DenseNet201	1,920	88.4±3.3	83.1±4.3	83.0±4.3	91.0±2.7	82.4±4.6	73.9±7.0
EfficientNetb0	2,048	82.9±6.5	76.0±6.9	74.2±12.9	86.7±6.0	73.4±11.2	61.5±15.7

TABLE 20. **Performance obtained with the RF classifier trained with HC and deep features for the three-classes task. The table reports the performance in terms of accuracy, precision, sensitivity, specificity, F1 score, Matthew's correlation coefficient, and balanced accuracy.**

Desc.	#features	ACC	P	SEN/BA	SPE	F1	MCC
HM	7	77.8±8.3	68.7±14.1	67.4±10.1	82.6±5.6	66.4±11.8	50.9±18.0
ZM	**9**	**80.2±9.7**	**73.1±15.4**	**69.9±14.5**	**83.8±7.8**	**69.6±14.7**	**55.4±22.7**
LM	15	88.7±3.3	85.7±6.6	83.4±3.6	90.9±2.1	82.8±4.6	75.5±7.2
CH_fi	15	85.6±5.6	79.0±9.8	78.6±9.9	88.3±5.2	78.1±9.6	67.0±14.7
CH_se	10	90.1±7.6	86.2±10.9	84.5±12.9	91.6±6.8	84.7±12.3	77.1±18.3
CLBP212	704	90.1±2.2	84.8±4.5	85.1±6.3	92.4±1.9	84.3±5.1	77.4±6.7
Hist	7	74.7±13.3	60.8±18.8	62.0±18.7	81.5±9.5	59.5±20.0	42.7±27.9
AlexNet	4,096	87.7±6.3	86.3±10.3	78.5±10.9	88.9±5.5	79.1±10.7	71.7±15.1
ResNet-18	512	89.7±5.8	87.3±10.6	81.3±10.7	90.8±4.8	81.9±11.1	75.7±14.9
ResNet-50	2,048	91.1±4.9	88.9±6.6	84.6±9.4	92.1±4.6	85.2±8.8	79.2±11.9
ResNet-101	2,048	91.5±5.0	89.3±7.0	85.5±9.5	92.7±4.7	85.9±8.4	80.3±11.7
GoogLeNet	1,000	89.1±6.9	85.0±7.6	83.4±10.1	91.4±5.2	83.1±9.9	75.5±14.2
Inception-v3	1,000	89.7±5.5	86.4±7.0	82.8±9.8	91.2±5.0	83.3±9.2	76.3±12.8
Inc.-ResNetv2	**1,536**	**91.8±1.4**	**88.4±1.8**	**87.5±3.8**	**93.3±1.8**	**87.2±2.2**	**81.3±3.7**
DarkNet-53	1,000	93.8±2.3	91.7±3.2	90.2±3.2	95.0±1.7	90.3±3.1	86.1±4.6
DenseNet201	1,920	95.2±2.2	93.0±4.0	92.6±3.5	96.1±1.7	92.7±3.6	88.9±5.4
EfficientNetb0	2,048	90.8±2.6	86.9±6.3	84.8±3.9	92.3±1.6	85.2±4.3	78.4±6.7

successfully used within the context of a dairy industry that needs to keep automated, near real-time checks on the ripening status of cheeses in storage. In addition, it applies to monitoring the storage situation in order to prevent or

TABLE 21. Performance obtained with the DT classifier trained with HC and deep features for the three-classes task. The table reports the performance in terms of accuracy, precision, sensitivity, specificity, F1 score, Matthew's correlation coefficient, and balanced accuracy.

Desc.	#features	ACC	P	SEN/BA	SPE	F1	MCC
HM	7	67.5±4.7	53.3±7.1	52.8±3.3	75.3±2.4	51.3±6.1	28.3±7.4
ZM	**16**	**74.0±11.7**	**63.0±19.1**	**62.0±16.8**	**79.6±9.1**	**61.0±17.7**	**42.0±26.8**
LM	28	80.9±6.2	71.6±9.7	70.9±8.8	85.2±5.1	69.6±8.5	56.4±14.0
CH_fi	10	76.8±7.0	67.3±9.0	65.4±8.9	81.3±5.3	65.0±9.3	47.8±14.2
CH_se	21	80.9±3.9	72.7±5.0	73.4±5.6	85.4±3.0	71.4±5.8	58.2±8.1
CLBP18	72	86.7±5.0	81.3±6.8	80.9±4.5	90.2±2.8	79.2±6.5	71.2±8.6
Hist	7	72.6±10.5	60.7±12.6	61.2±14.7	80.2±7.1	58.2±14.5	40.7±20.8
AlexNet	4,096	80.5±8.1	72.0±11.7	72.4±12.8	85.0±6.3	71.2±12.4	57.0±18.5
ResNet-18	512	79.8±4.6	72.7±6.0	70.1±5.2	83.7±3.1	70.4±6.1	55.1±8.8
ResNet-50	2,048	83.9±6.0	78.2±8.7	76.9±7.4	87.3±4.4	75.8±7.6	64.8±12.3
ResNet-101	2,048	83.9±7.1	79.9±12.6	75.0±8.6	86.8±4.8	75.2±10.0	64.8±14.8
GoogLeNet	1,000	80.9±3.9	70.5±6.2	70.6±7.0	85.1±2.9	70.0±6.6	55.5±9.4
Inc.-v3	1,000	79.5±3.2	69.8±4.1	68.2±5.8	83.8±3.2	68.3±4.8	52.6±8.0
Inc.-ResNetv2	**1,536**	**78.5±8.0**	**68.8±10.2**	**68.4±12.4**	**83.1±6.4**	**67.5±11.3**	**51.5±17.5**
DarkNet-53	1,000	86.7±5.7	82.3±6.5	80.0±10.4	89.7±4.6	79.5±9.6	70.5±13.1
DenseNet201	1,920	85.3±2.6	78.9±4.3	78.9±1.9	88.7±1.4	78.0±2.5	67.6±4.6
EfficientNetb0	2,048	83.6±3.1	74.3±5.2	74.9±7.8	87.0±3.2	74.0±6.2	61.6±9.2

TABLE 22. Performance obtained with the MLP classifier trained with HC and deep features for the three-classes task. The table reports the performance in terms of accuracy, precision, sensitivity, specificity, F1 score, Matthew's correlation coefficient, and balanced accuracy.

Desc.	#features	ACC	P	SEN/BA	SPE	F1	MCC
HM	7	71.6±5.1	48.0±9.1	49.7±7.8	75.7±3.4	42.8±7.4	26.7±12.0
ZM	**14**	**79.1±2.8**	**58.6±4.4**	**61.2±4.6**	**80.4±2.4**	**54.3±4.0**	**43.5±6.7**
LM	28	79.8±2.5	58.5±2.1	61.6±4.4	81.2±2.9	54.5±2.7	44.6±5.4
CH_fi	28	77.8±1.7	54.9±3.8	58.9±3.0	80.2±1.8	51.3±2.8	39.9±4.0
CH_se	28	76.1±3.6	52.8±6.8	56.2±5.9	78.4±2.3	49.1±6.0	35.7±9.1
CLBP18	72	73.7±4.1	50.9±2.7	54.0±3.6	81.3±2.4	45.1±3.4	34.9±5.4
Hist	7	76.4±4.4	51.0±4.4	56.2±6.1	82.2±3.8	47.4±4.9	37.7±8.7
AlexNet	4,096	76.4±4.1	51.2±3.4	57.8±5.2	81.1±3.1	48.5±4.6	37.5±7.5
ResNet-18	512	76.8±2.9	51.4±2.1	57.8±3.3	82.4±2.0	48.3±2.9	38.7±4.9
ResNet-50	2,048	78.8±4.9	54.3±5.3	61.5±5.7	82.3±4.4	52.1±5.6	42.4±10.0
ResNet-101	2,048	79.5±1.2	54.2±1.1	62.5±0.9	82.7±1.5	52.8±1.1	43.3±2.4
GoogLeNet	1,000	73.3±4.5	48.5±5.6	52.7±6.3	77.5±3.5	45.2±5.8	29.9±9.7
Inception-v3	1,000	76.1±3.6	49.7±4.7	56.8±7.2	80.5±3.7	47.4±5.5	35.8±9.0
Inc.-ResNetv2	**1,536**	**75.4±4.6**	**52.8±7.0**	**56.1±8.0**	**77.8±3.7**	**48.7±7.2**	**34.5±11.2**
DarkNet-53	1,000	78.1±5.3	52.9±4.9	60.2±7.1	83.5±4.2	50.1±6.0	41.8±10.6
DenseNet201	1,920	79.1±2.5	53.7±2.7	61.9±3.3	83.6±2.9	51.7±2.7	43.3±5.8
EfficientNetb0	2,048	75.4±3.5	50.5±5.6	55.5±5.0	79.3±2.0	47.3±5.2	34.8±7.5

verify potential issues, such as maintaining an appropriate temperature for ripening.

However, several areas for improvement remain, ranging from combining heterogeneous features to adopting feature selection techniques. In addition, studying the influence of the background of photos or their illumination could make an important contribution to improving the robustness of the system.

Finally, as the results obtained by the CNNs with the three-classes setup have left open questions, it must be considered that artificial intelligence applications are requested to offer a high level of accountability and transparency. Therefore, explanations for algorithm decisions and predictions can be explored to justify their reliability and provide high interpretability for the end users.

APPENDIX.
NUMERICAL RESULTS

See Tables 15–22.

REFERENCES

[1] T. Lei and D.-W. Sun, "Developments of nondestructive techniques for evaluating quality attributes of cheeses: A review," *Trends Food Sci. Technol.*, vol. 88, pp. 527–542, Jun. 2019.

[2] P. L. McSweeney, "Biochemistry of cheese ripening," *Int. J. Dairy Technol.*, vol. 57, nos. 2–3, pp. 127–144, 2004.

[3] A. Forde and G. F. Fitzgerald, "Biotechnological approaches to the understanding and improvement of mature cheese flavour," *Current Opinion Biotechnol.*, vol. 11, no. 5, pp. 484–489, Oct. 2000.

[4] P. Fox, J. Wallace, S. Morgan, C. Lynch, E. Niland, and J. Tobin, "Acceleration of cheese ripening," *Antonie Leeuwenhoek*, vol. 70, no. 2, pp. 271–297, 1996.

[5] L. Sakkas, C. S. Pappas, and G. Moatsou, "FT-MIR analysis of water-soluble extracts during the ripening of sheep milk cheese with different phospholipid content," *Dairy*, vol. 2, no. 4, pp. 530–541, Sep. 2021.

[6] E. Dufour, G. Mazerolles, M. Devaux, G. Duboz, M. Duployer, and N. M. Riou, "Phase transition of triglycerides during semi-hard cheese ripening," *Int. Dairy J.*, vol. 10, nos. 1–2, pp. 81–93, 2000.

[7] T. M. P. Cattaneo, C. Giardina, N. Sinelli, M. Riva, and R. Giangiacomo, "Application of FT-NIR and FT-IR spectroscopy to study the shelf-life of Crescenza cheese," *Int. Dairy J.*, vol. 15, nos. 6–9, pp. 693–700, Jun. 2005.

[8] S. T. Martín Del Campo, N. Bonnaire, D. Picque, and G. Corrieu, "Initial studies into the characterisation of ripening stages of emmental cheeses by mid-infrared spectroscopy," *Dairy Sci. Technol.*, vol. 89, no. 2, pp. 155–167, Mar. 2009.

[9] M. C. Soto-Barajas, M. I. González-Martín, J. Salvador-Esteban, J. M. Hernández-Hierro, V. Moreno-Rodilla, A. M. Vivar-Quintana, I. Revilla, I. L. Ortega, R. Morón-Sancho, and B. Curto-Diego, "Prediction of the type of milk and degree of ripening in cheeses by means of artificial neural networks with data concerning fatty acids and near infrared spectroscopy," *Talanta*, vol. 116, pp. 50–55, Nov. 2013.

[10] A. M. Jiménez-Carvelo, A. González-Casado, M. G. Bagur-González, and L. Cuadros-Rodríguez, "Alternative data mining/machine learning methods for the analytical evaluation of food quality and authenticity—A review," *Food Res. Int.*, vol. 122, pp. 25–39, Aug. 2019.

[11] A. Jahanbakhshi, Y. Abbaspour-Gilandeh, K. Heidarbeigi, and M. Momeny, "A novel method based on machine vision system and deep learning to detect fraud in turmeric powder," *Comput. Biol. Med.*, vol. 136, Sep. 2021, Art. no. 104728.

[12] A. Jahanbakhshi, Y. Abbaspour-Gilandeh, K. Heidarbeigi, and M. Momeny, "Detection of fraud in ginger powder using an automatic sorting system based on image processing technique and deep learning," *Comput. Biol. Med.*, vol. 136, Sep. 2021, Art. no. 104764.

[13] M. Al-Sarayreh, M. M. Reis, W. Qi Yan, and R. Klette, "Detection of red-meat adulteration by deep spectral–spatial features in hyperspectral images," *J. Imag.*, vol. 4, no. 5, p. 63, May 2018.

[14] G. Ciocca, P. Napoletano, and R. Schettini, "Learning CNN-based features for retrieval of food images," in *Proc. Int. Conf. Image Anal. Process.*, in Lecture Notes in Computer Science, vol. 10590, S. Battiato, G. M. Farinella, M. Leo, and G. Gallo, Eds. Catania, Italy: Springer, 2017, pp. 426–434.

[15] G. Ciocca, P. Napoletano, and R. Schettini, "CNN-based features for retrieval and classification of food images," *Comput. Vis. Image Understand.*, vols. 176–177, pp. 70–77, Nov. 2018.

[16] M. Shukor, G. Couairon, A. Grechka, and M. Cord, "Transformer decoders with MultiModal regularization for cross-modal food retrieval," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2022, pp. 4567–4578.

[17] H. Wang, D. Sahoo, C. Liu, E.-P. Lim, and S. C. H. Hoi, "Learning cross-modal embeddings with adversarial networks for cooking recipes and food images," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Long Beach, CA, USA, Jun. 2019, pp. 11572–11581.

[18] S. Turmchokkasam and K. Chamnongthai, "The design and implementation of an ingredient-based food calorie estimation system using nutrition knowledge and fusion of brightness and heat information," *IEEE Access*, vol. 6, pp. 46863–46876, 2018.

[19] R. D. Kumar, E. G. Julie, Y. H. Robinson, S. Vimal, and S. Seo, "Recognition of food type and calorie estimation using neural network," *J. Supercomput.*, vol. 77, no. 8, pp. 8172–8193, Aug. 2021, doi: 10.1007/s11227-021-03622-w.

[20] D. Bradley and G. Roth, "Adapting thresholding using the integral image," *J. Graph. Tools*, vol. 12, pp. 13–21, Jan. 2007.

[21] L. Putzu, A. Loddo, and C. Di Ruberto, "Invariant moments, textural and deep features for diagnostic MR and CT image retrieval," in *Proc. CAIP*, in Lecture Notes in Computer Science, vol. 13052, 2021, pp. 287–297.

[22] C. Lv, P. Zhang, and D. Wu, "Gear fault feature extraction based on fuzzy function and improved Hu invariant moments," *IEEE Access*, vol. 8, pp. 47490–47499, 2020.

[23] Z. Xia, X. Wang, M. Wang, S. Unar, C. Wang, Y. Liu, and X. Li, "Geometrically invariant color medical image null-watermarking based on precise quaternion polar harmonic Fourier moments," *IEEE Access*, vol. 7, pp. 122544–122560, 2019.

[24] A. Humeau-Heurtier, "Texture feature extraction methods: A survey," *IEEE Access*, vol. 7, pp. 8975–9000, 2019.

[25] A. K. Sharma, S. Tiwari, G. Aggarwal, N. Goenka, A. Kumar, P. Chakrabarti, T. Chakrabarti, R. Gono, Z. Leonowicz, and M. Jasinski, "Dermatologist-level classification of skin cancer using cascaded ensembling of convolutional neural network and handcrafted features based deep neural network," *IEEE Access*, vol. 10, pp. 17920–17932, 2022.

[26] A. Wang, W. Zhang, and X. Wei, "A review on weed detection using ground-based machine vision and image processing techniques," *Comput. Electron. Agricult.*, vol. 158, pp. 226–240, Mar. 2019.

[27] A. Loddo, M. Loddo, and C. Di Ruberto, "A novel deep learning based approach for seed image classification and retrieval," *Comput. Electron. Agricult.*, vol. 187, Aug. 2021, Art. no. 106269, doi: 10.1016/j.compag.2021.106269.

[28] F. Bouhlel, H. Mliki, and M. Hammami, "Abnormal crowd density estimation in aerial images based on the deep and handcrafted features fusion," *Expert Syst. Appl.*, vol. 173, Jul. 2021, Art. no. 114656.

[29] H. Hsu, T. Huang, G. Wang, J. Cai, Z. Lei, and J. Hwang, "Multi-camera tracking of vehicles based on deep features re-id and trajectory-based camera link models," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, Long Beach, CA, USA, Jun. 2019, pp. 416–424. [Online]. Available: http://openaccess.thecvf.com/content_CVPRW_2019/html/AI_City/Hsu_Multi-Camera_Tracking_of_Vehicles_based_on_Deep_Features_Re-ID_and_CVPRW_2019_paper.html

[30] M.-K. Hu, "Visual pattern recognition by moment invariants," *IRE Trans. Inf. Theory*, vol. 8, no. 2, pp. 179–187, Feb. 1962.

[31] M. R. Teague, "Image analysis via the general theory of moments," *J. Opt. Soc. Amer.*, vol. 70, no. 8, pp. 920–930, 1980.

[32] C.-H. Teh and R. T. Chin, "On image analysis by the methods of moments," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-10, no. 4, pp. 496–513, Jul. 1988.

[33] M. Oujaoura, B. Minaoui, and M. Fakir, "Image annotation by moments," in *Moments and Moment Invariants—Theory and Applications*, vol. 1. Xanthi, Greece, 2014, pp. 227–252. [Online]. Available: https://local.infobel.gr/GR101039571/science_gate_publishing-xanthi.html

[34] R. Mukundan, S. H. Ong, and P. A. Lee, "Image analysis by Tchebichef moments," *IEEE Trans. Image Process.*, vol. 10, no. 9, pp. 1357–1364, Sep. 2001.

[35] C. Di Ruberto, L. Putzu, and G. Rodriguez, "Fast and accurate computation of orthogonal moments for texture analysis," *Pattern Recognit.*, vol. 83, pp. 498–510, Nov. 2018.

[36] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary pattern," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 971–987, Jul. 2002.

[37] L. Putzu and C. Di Ruberto, "Rotation invariant co-occurrence matrix features," in *Proc. Int. Conf. ICIAP*, vol. 10484. Cham, Switzerland: Springer, 2017, pp. 391–401.

[38] J. Huang, S. R. Kumar, M. Mitra, W.-J. Zhu, and R. Zabih, "Image indexing using color correlograms," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 1997, pp. 762–768.

[39] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.

[40] M. Nixon and A. Aguado, *Feature Extraction and Image Processing for Computer Vision*. New York, NY, USA: Academic, 2019.

[41] Y. Chandola, J. Virmani, H. Bhadauria, and P. Kumar, "Hybrid computer-aided classification system design using lightweight end-to-end pre-trained CNN-based deep feature extraction and PCA-SVM classifier for chest radiographs," in *Deep Learning for Chest Radiographs* (Primers in Biomedical Imaging Devices and Systems), Y. Chandola, J. Virmani, H. Bhadauria, and P. Kumar, Eds. New York, NY, USA: Academic, 2021, pp. 197–204.

[42] L. Putzu, L. Piras, and G. Giacinto, "Convolutional neural networks for relevance feedback in content based image retrieval," *Multimedia Tools Appl.*, vol. 79, nos. 37–38, pp. 26995–27021, Oct. 2020.

[43] J. Donahue, Y. Jia, and O. Vinyals, "DeCAF: A deep convolutional activation feature for generic visual recognition," in *Proc. Int. Conf. ML*, 2014, pp. 647–655.

[44] G. Litjens, T. Kooi, B. E. Bejnordi, A. A. A. Setio, F. Ciompi, M. Ghafoorian, J. A. Van Der Laak, B. Van Ginneken, and C. I. Sánchez, "A survey on deep learning in medical image analysis," *Med. Image Anal.*, vol. 42, pp. 60–88, Jan. 2017.

[45] E. Raei, A. A. Asanjan, M. R. Nikoo, M. Sadegh, S. Pourshahabi, and J. F. Adamowski, "A deep learning image segmentation model for agricultural irrigation system classification," *Comput. Electron. Agricult.*, vol. 198, Jul. 2022, Art. no. 106977. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0168169922002940

[46] Y. Wang, D. Sun, K. Chen, F. Lai, and M. Chowdhury, "Efficient DNN training with knowledge-guided layer freezing," 2022, *arXiv:2201.06227*.

[47] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. NIPS*, vol. 1, 2012, pp. 1097–1105.

[48] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1–9.

[49] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[50] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2818–2826.

[51] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-ResNet and the impact of residual connections on learning," in *Proc. 31st AAAI Conf. Artif. Intell.*, S. Singh and S. Markovitch, Eds. San Francisco, CA, USA: AAAI Press, 2017, pp. 4278–4284.

[52] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*.

[53] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 2261–2269.

[54] M. Tan and Q. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proc. Mach. Learn. Res.*, vol. 97, K. Chaudhuri and R. Salakhutdinov, Eds. Long Beach, CA, USA: PMLR, Jun. 2019, pp. 6105–6114.

[55] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn.*, in JMLR Workshop and Conference Proceedings, vol. 37, F. R. Bach and D. M. Blei, Eds., Jul. 2015, pp. 448–456.

[56] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Trans. Inf. Theory*, vol. IT-13, no. 1, pp. 21–27, Jan. 1967.

[57] Y. Lin, F. Lv, S. Zhu, M. Yang, T. Cour, K. Yu, L. Cao, and T. S. Huang, "Large-scale image classification: Fast feature extraction and SVM training," in *Proc. 24th IEEE Conf. Comput. Vis. Pattern Recognit.*, Colorado Springs, CO, USA, Jun. 2011, pp. 1689–1696.

[58] J. R. Quinlan, "Learning efficient classification procedures and their application to chess end games," in *Machine Learning*. Berlin, Germany: Springer, 1983, pp. 463–482.

[59] L. Breiman, "Random forests," *Mach. Learn.*, vol. 4, pp. 5–32, Jan. 2001.

ANDREA LODDO received the B.Sc., M.Sc., and Ph.D. degrees from the University of Cagliari, in 2012, 2014, and 2019, respectively. His Ph.D. thesis faced blood cell image analysis and classification issues with creating new tools for automatic diagnosis to support medical research. He is currently an Assistant Professor at the Department of Mathematics and Computer Science, University of Cagliari. He is the author of more than 20 scientific manuscripts in peer-reviewed journals and international conference proceedings. His research interests include image analysis and processing, computer vision, pattern recognition, and machine and deep learning, with particular purposes for medical tasks. Currently, he is pursuing research activities for biomedical image analysis for diagnosis support systems and video surveillance applications.

CECILIA DI RUBERTO received the M.S. degree in computer science from the University of Salerno, Italy, in 1990, and the Ph.D. degree in computer science from the University of Naples, Italy, in 1995. She is currently an Associate Professor in computer science at the Department of Mathematics and Computer Science, University of Cagliari, Italy. Her research interests include computer vision, image retrieval, medical image analysis, pattern recognition, and machine learning. She has been working in microscopic image analysis, in particular in blood smear image analysis for cell counting, malaria parasites detection and classification, and leukemia detection. She is the author of more than 100 scientific papers in peer-reviewed journals and international conference proceedings.

GIULIANO ARMANO is currently an Associate Professor in computer science at the Department of Mathematics and Computer Science (DMI), University of Cagliari. With a background on expert systems and machine learning, over the years, he has investigated various kinds of learning strategies and techniques. His current research interests include performance measures for classifier systems, artificial neural networks, complex networks, and classifier ensembles. These topics have been extensively experimented on various application fields, in particular bioinformatics, information retrieval, and text categorization. He is currently teaching “programming languages for mathematics” and “laboratory of big data.” Previously, his teaching activities include “elements of bioinformatics,” “object-oriented programming languages,” and “software engineering.”

ANDREA MANCONI received the Ph.D. degree in electronics and computer engineering from the University of Cagliari, Cagliari, Italy. Since 2011, he has been a member of the Bioinformatics Laboratory, Institute of Biomedical Technologies of the National Research Council of Italy (CNR-ITB). His research interests include the development of novel tools and infrastructures for bioinformatics.