Learning-based automatic classification of lichens from images

Alberto Presta [a], *Felice Andrea Pellegrino* [b,*], *Stefano Martellos* [c]

[a] *Department of Mathematics and Geosciences, University of Trieste, Via Valerio 12/1, Trieste, 34127, Italy*
[b] *Department of Engineering and Architecture, University of Trieste, Via Valerio 6/1, Trieste, 34127, Italy*
[c] *Department of Life Sciences, University of Trieste, Via Weiss 2, Trieste, 34127, Italy*

ARTICLE INFO

Accepted 27 November 2021

Keywords:
Computer vision
Machine learning
Neural network
Species recognition
Lichen
In-field classification

Biomonitoring plays a crucial role in the assessment of air quality, as it allows to estimate the presence of pollutants, by measuring deviations from normality of the components of an ecosystem. Lichens are among the organisms most commonly used as bioindicators. The present study deals with the classification of lichen taxa from images, by means of a machine learning process based on patch classification. A given image is divided in non-overlapping patches, and each of them undergoes feature extraction and classification, eventually being associated to a category. Three different methods for extracting patch descriptors are investigated: (i) handcrafted descriptors based on classical feature extractor algorithms, (ii) convolutional neural networks employed as feature extractors, and (iii) scattering networks, which combine wavelet convolutions and nonlinear operators. For each of these methods, the descriptors are used as inputs for a classification algorithm. The whole process is evaluated in terms of classification accuracy, empirically determining the most appropriate parameters for the different models implemented. By using the dataset of lichens of this study, best results (~ 0.89 accuracy) are obtained with a specific handcrafted descriptor (dense SIFT), thus providing insights on the kind of representation which is most suitable for the task.

1. Introduction

Lichens are a common component of ecosystems, and different species can occur everywhere on the planet, even in the Antarctica (Castello et al., 2006). They are frequently used for monitoring air quality, since they are specially sensitive to phytotoxic gasses, especially NO_x and SO_2 (Cislaghi & Nimis, 1997), by means of a well-established, standardised methodology (Anpa, 2001; Nimis et al., 2002). For monitoring air quality, the total diversity of epiphytic lichens is calculated on the basis of the number of different species (i.e., classes) which occur in a well delimited portion of a tree trunk. Given the difficulty of identifying lichens (the task can be performed by trained operators only), and given that observer error is intrinsic in any vegetation survey (Morrison, 2016), an automated classification system could be extremely useful for supporting the monitoring of air quality carried out by environmental agencies. Furthermore, such a system, if ported on

* *Corresponding author.*
E-mail address: fapellegrino@units.it (F.A. Pellegrino).

Nomenclature

Adam	adaptive moment estimation algorithm
BGPs	binary Gabor patterns
BOVW	bag of visual words
CNN	convolutional neural network
CV	computer vision
DA	data augmentation
FN	false negative
FP	false positive
LBI	lichen biodiversity index
LBPs	local binary patterns
LSCs	localized scattering coefficients
ML	machine learning
ReLU	rectified linear unit
SIFT	scale-invariant feature transform
SVM	support vector machine
TL	transfer learning
TP	true positive

mobile devices, could be useful for monitoring air quality even through a citizen science approach. This study aims at building a process for the automatic classification, from images, of different lichen taxa growing on trees, thus allowing the automated computation of the so called *lichen biodiversity index*, or LBI (Ammann et al., 1987; Bini et al., 2001). Such index is a well-established technique for assessing air pollution, and especially the presence of phytotoxic gasses, which are related to anthropic disturbance (Nimis & Purvis, 2002; Cislaghi & Nimis, 1997). Computer vision (CV) offers several tools for recognising the content of digital images, most being based on machine learning (ML). Given that several different lichens species can appear next to each other in the same portion of tree bark (Fig. 1), the following different ML-based approaches could be pursued:

- end-to-end: the LBI itself is estimated directly from an image;
- object detection: lichens or their distinctive parts are treated as "objects" and the image is searched for such objects. Then, the detected objects, their extent, and the labels, are employed for computing the LBI;
- semantic segmentation: each image is partitioned in disjoint areas which are semantically meaningful, i.e., associated to different species. Then, the areas and the corresponding labels are employed for computing the LBI.

Fig. 1 – Example of several species of lichens appearing in the same portion of a tree bark.

The end-to-end approach has been discarded due to absence of a suitable dataset: indeed, a considerable amount of pairs consisting of an image and the corresponding LBI is needed. For instance, in Dyrmann et al. (2016), 10 413 images containing 22 weed and crop species where employed to train a model from scratch. As detailed below, we deal with 5–6 images only, per species. The inspection of some images reveals that the object detection approach is not suitable as well. Indeed, although some lichens are associated to distinctive parts that can be treated as objects, some others are mainly characterised by a textured appearance, which prevents the identification of object-like features. Thus, a semantic segmentation approach has been chosen. Specifically, motivated by the availability of a dataset of isolated instances of different species (collected for other purposes than training an ML system), a *patch-based* classification approach has been selected: given an image, it is divided into non-overlapping rectangular patches, each to be associated with a specific category. Thus, the problem is reduced to an image classification problem, to be solved for each of the patches. Figure 2 reports an example, where three patches are associated with three distinct species of lichens. The chosen approach has two valuable consequences: A) it implicitly augments the dataset, since several patches can be extracted from each available image, and employed for training and evaluating the classifier, and B) it provides information (a set of labeled patches) that can be used directly to estimate the LBI.[1] In this paper, the problem of associating a class label to a single patch is considered, and the terms "patch" and "image" are used as synonyms unless otherwise specified. The problem is faced by means of a classifier operating on a descriptor extracted from the image. As for the descriptor extraction, three different approaches are investigated: classical descriptors, convolutional neural networks (employed as feature extractors) and scattering networks.

2. Related work

Image classification is a process that assigns a label to a whole image. It is different from other CV processes, such as object detection (finding occurrences of a given kind of object in a an image), object recognition (finding a specific instance of an object), and semantic segmentation. Image classification is a well investigated field, and one of the most successful applications of CV (see for instance (Szeliski, 2010) for the classical approaches, and (Goodfellow et al., 2016) for some of the more recent). Traditionally, the problem of image classification has been faced by means of ML. A through review of the different approaches to the problem is outside the scope of the present paper. The basic principle, shared by most of the approaches is that the problem

[1] provided that some additional information about the actual size of the captured area is available.

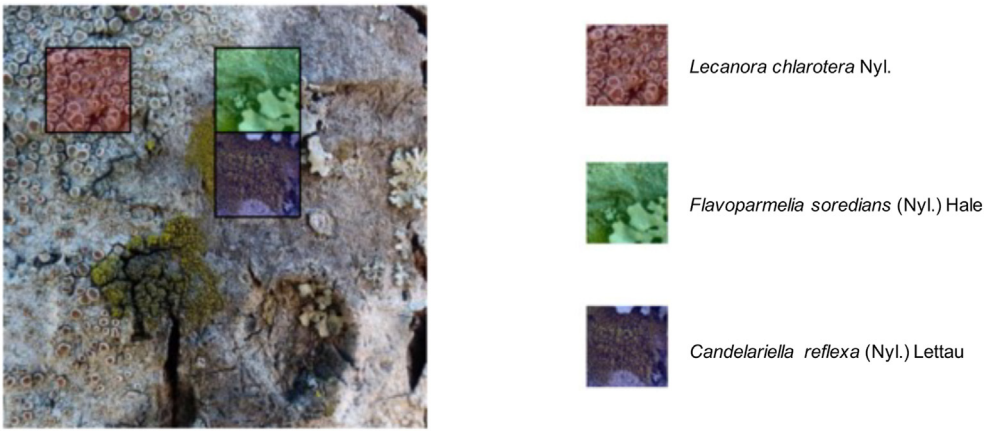

Fig. 2 – An example of patch-based classification approach, in which three different patches are shown: a single label is associated to each of them.

is formulated in terms of *supervised learning*, i.e. a learning problem in which a labeled dataset is available, containing examples of images along with the associated label. A learning machine (classifier) is thus trained based on that dataset in order to learn an inference rule for associating the proper output (label) to the input. As for the input, it can be a descriptor of the image, or the image itself. Classical approaches employ a variety of descriptors, which can be handcrafted, e.g (Lowe, 2004), or automatically built from the dataset itself, such as visual codebooks (Fei–Fei & Perona, 2005). More recent approaches, based on convolutional neural networks (CNNs), do not rely on explicit descriptors, but, instead, delegate the classifier itself to build an internal representation of the input (the image), to be employed for the classification (Rawat & Wang, 2017). The activation values of the neurons belonging to a layer, in correspondence to a certain input, can be thought of as features extracted from that input; in other words, they can be employed as descriptors for feeding any classifier (even if the network has been trained on a different dataset). Based on that, several schemes of *transfer learning* (TL) (Tan et al., 2018) are possible, including fine-tuning a given pre-trained network to a specific dataset. As of today, image classification has been employed in many fields ranging from medicine to industrial manufacturing. Image classification has been applied to botany as well, mostly aiming at identifying organisms. For instance, in Soon Jye et al. (2017), artificial neural networks and support vector machine classifiers, both trained on handcrafted descriptors, were exploited to perform classification of three different species of the genus *Ficus*. However, not only classification issues are addressed. As an example, Saleem et al. (2020) performed detection of plant disease by using different models based on CNNs. The aim was that of identifying sick leaves, and damaged portions of them. Although there are several studies on vascular plants, which often led to the publication of apps for classification through smartphones, studies focusing on lichens are scarce. In Kanmani and Rajiv Kannan (2017), a pattern recognition methodology was used to extract descriptors for training an artificial neural network. Another study (Galanty et al., 2021), used a pre-trained CNN, called SqueezeNet (Iandola et al., 2016), to perform classification of 12 different lichen species, achieving an accuracy of $\sim$ 0.61. Lichens are important to monitor air quality also because they change their morphology, and can also die, on the basis of the presence and abundance of several pollutants. Preetha et al. (2021) used this fact to monitor the level of pollutants present around smart cities. In particular, they exploited ResNet architecture (He et al., 2016) to classify lichen images (all images refer to the genus *Xanthoria*) into four different pollution levels: Not Polluted, Moderately Polluted, Heavily Polluted, Extremely Polluted. They achieved an accuracy of $\sim$ 0.953; however, they did not actually face the problem of lichen species classification.

3. Dataset

3.1. *Data collection*

Several lichen species were selected on the basis of: i) frequency in surveys aimed at estimating the LBI; ii) amount of available high quality images in the image archive of ITALIC (Nimis & Martellos, 2002), the Information System on Italian Lichens (http://italic.units.it); the mentioned archive has been chosen for two main reasons: first, the present work is part of a research project focused on lichens occurring in Italy, and, second, ITALIC is known to be reliable in terms of correct identification of each of the images of the archive; iii) an expert assessment for selecting species with different growth form (crustose, fruticose and foliose lichens), reproductive strategy (sorediate, or reproducing sexually, by apothecia), and colors.

After a first selection, the list was limited to 20 species, for which at least 5 high quality images were available. Nomenclature follows Nimis (2016).

3.2. *Data processing*

In summary, there are 20 different classes, representing the 20 species, each with a number of images ranging from 5 to 6, as shown in Table 1. Since the dataset is small in terms of available number of images per class, a *patch classification* approach has been chosen. Each image belonging to original dataset was divided into rectangular pieces, called *patches*. Each patch was then labelled with the same class as the image

Table 1 – Content of the dataset, in terms of number of images for each lichen species. For each species, scientific name, number of images initially available, and number of images obtained after data processing are reported. The column called *acronym* associates an acronym to each species.

Species	Original dataset	Patch-based dataset	Acronym
Lecanora chlarotera (Nyl).	6	92	Lc
Caloplaca cerina (Hedw.) Th. Fr.	6	91	Cc
Physconia grisea (Lam.) Poelt.	5	80	Pg
Lecanora argentata (Ach.) Malme	5	80	La
Ramalina fastigiata (Pers.) Ach.	5	80	Rfas
Phaeophyscia orbicularis (Nech.) Moberg	5	80	Po
Candelariella xanthostigma (Ach.) Lettau	5	80	Cx
Flavoparmelia caperata (L.) Hale	5	80	Fc
Chrysothrix candelaris (L.) J.R. Laundon	5	80	Cca
Ramalina farinacea (L.) Ach.	5	79	Rfar
Melanelixia glabratula (Lamy) Sandler & Arup	5	79	Mg
Physcia biziana (A. Massal.) Zahlbr.	5	78	Pb
Arthonia radiata (Pers.) Ach.	5	78	Ar
Xanthomendoza fallax (Hepp) Søchting, Kärnefelt. & S.Y. Kondr.	5	78	Xfa
Candelariella reflexa (Nyl.) Lettau	5	78	Cr
Flavoparmelia soredians (Nyl.) Hale	5	78	Fs
Xanthomendoza fulva (Hoffm.) Søchting, Kärnefelt. & S.Y. Kondr.	5	77	Xfu
Hyperphyscia adglutinata (Flørke) H. Mayrhofer & Poelt	5	75	Ha
Lecidella alaechroma (Ach.) M. Choisy	5	72	La
Gyalolechia flavorubescens (Huds.) Søchting, Frøden & Arup	5	64	Gf
Total	102	1579	

from which the patch belonged. That approach produced a larger dataset, consisting of a collection of individual patches. More specifically, for each original image, the following steps were performed:

1. the original image was resized (using bilinear interpolation), to 1000 × 1000 pixel, in order to obtain images of the same size (same width and height). In doing this, aspect ratio was not preserved. However, given the aspect ratio of the original images (having a mean of about 1.26 ≈ 5/4, and a standard deviation of 0.30), that change did not lead to exaggerated distortions, which could misrepresent the image of the lichen;
2. the regions closer to the borders were trimmed out, since the lichens are mostly located in the center of the images; Thus, intermediate images of dimension 800 × 800 were obtained;
3. from the intermediate images, 16 non-overlapping patches, each of size 200 × 200, were extracted.

An example of patch extraction is represented in Fig. 3. The final dataset was obtained after a further selection, to reject patches not containing the specific lichen. The content of the dataset is summarised in Table 1. A total of 1579 patches were obtained, from 102 original images. The number of patches per species ranges from 64 to 92.

4. Methods

4.1. *Classic methods*

Classic methods are those based on handcrafted descriptors. As opposed to learned descriptors, they are designed in advance, possibly exploiting a priori knowledge on the specific domain of application. It has been shown that such methods can be quite effective with small datasets (Napoletano, 2017), thus they have been employed in the present study.

4.1.1. *Overview of the process*

The whole process relies on two functional blocks, namely a feature extractor and a classifier, and encompasses two phases, i.e. training and inference phase. The feature extractor computes a descriptor for each image belonging to the

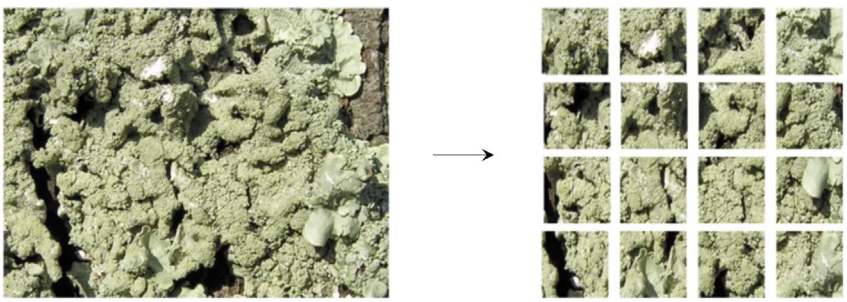

Fig. 3 – An example of patch extraction, in which 16 patches are created from an initial image.

training set. Descriptors, along with the associated labels, are employed for training the classifier. During the inference phase, the feature extractor computes the descriptor of an input image, and the trained classifier employs that descriptor to predict a label. An overview of the entire process is represented in Fig. 4, for the training phase (top) and the inference phase (bottom).

The employed descriptors can be broadly categorised as:

- *global descriptors*, which describe an image as a whole (in terms of color, texture and other characteristics). As they focus on global characteristics of an image, these kind of descriptors are not able to capture its local aspects;
- *local descriptors*, which describe neighbourhoods of properly chosen points within an image. A set of vectors is obtained from each image, each vector being the description of a neighbourhood.

When employing local descriptors, it is necessary to collect all the local features into a unique descriptor of the image, to be compared with descriptors of other images. To this end, a well-known and effective approach is the *bag of visual words* (BOVW) technique (Sivic & Zisserman, 2003). The main idea of BOVW is to apply clustering, to quantize local descriptors of the whole dataset into visual words (the centroids of the resulting clusters). For each image, the final descriptor is the frequency histogram of the visual words.

As for the classifier, a Support Vector Machine (SVM), has been employed, as it is well-known to be effective and efficient in pattern recognition (Burges, 1998).

4.1.2. *Descriptors*

The employed descriptors are:

- *Local binary patterns* (LBPs) (Ojala et al., 2002), which are based on the comparisons of each pixel of interest with its surrounding neighbourhood, and extracting values called LBPs. The whole descriptor is the frequency histogram of these values, across a set of image locations, and it is rotation invariant (but not scale invariant).
- *Scale Invariant Feature Transform (SIFT)* (Lowe, 2004), which extracts features for salient points of the image, called *keypoints*: for each of them, the process computes a descriptor based on gradients statistics of a neighbourhood of the specific point. That algorithm manages to achieve both rotation and scale invariance.
- *Dense SIFT* (Rassem & Khoo, 2011), similar to SIFT, but descriptors are extracted from regularly spaced points. Rotation invariance is preserved, while scale invariance is lost: to obtain the latter, descriptors are extracted at multiple scales, and all the obtained results are employed in the BOVW technique.
- *Gabor-based descriptor* (Aljahdali et al., 2012), based on Gabor filtering. It exploits convolution with elements from a Gabor filters bank, encompassing different orientations and scales, to extract a feature vector.
- *Binary Gabor patterns* (BGPs) (Zhang et al., 2012), which can be seen as a mixture between LBPs and Gabor filtering. For each point of interest, the process applies convolutions with elements from a Gabor filters bank (in which only different orientations are considered) to its neighbourhood, and extracts a value, which is rotation invariant. To obtain invariance to scale, Gabor filters banks at different scales are considered, and their outputs are concatenated.

In order to deal with color images, channels are considered separately, extracting a feature vector from each one, and concatenating them together.

4.2. *CNNs-based method*

CNNs (LeCun et al., 2010) operate on an image by applying sets of linear spatial filters, followed by some nonlinear and/or undersampling (pooling) operators. The set of filters are organised in subsequent layers, and their coefficients are

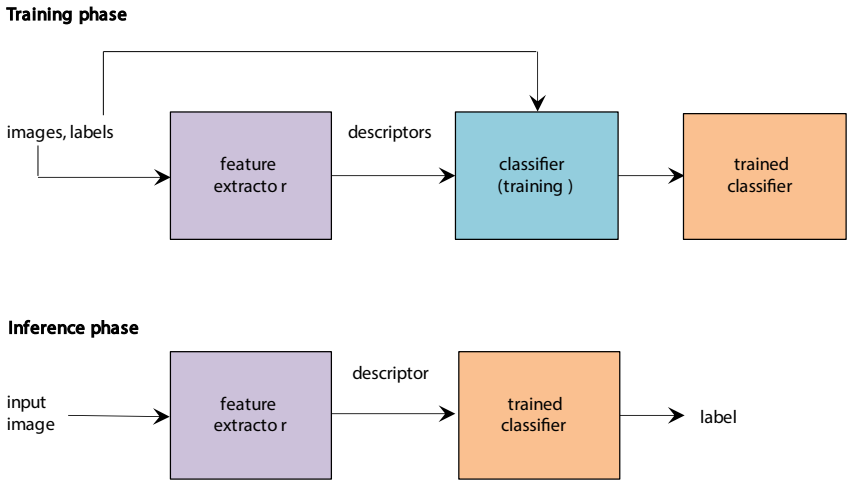

Fig. 4 – Overview of the process when using classic methods, for the training phase (top) and the inference phase (bottom). While the feature extractor varies according to the chosen descriptor, the classifier is always an SVM equipped with a Gaussian kernel.

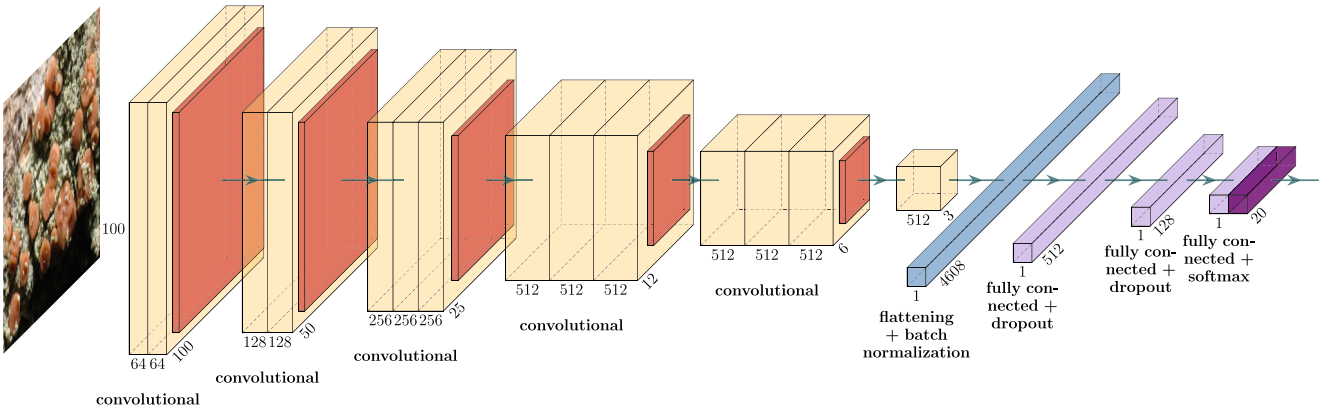

Fig. 5 – Overview of the convolutional neural network used to perform classification of lichen species.

learnt during the training of the network, which consists in the minimisation of a cost function evaluated on the training set. Figure 5 shows an example of CNN architecture. As it is well-known, CNNs can perform remarkably well in image classification tasks, often surpassing classic methods. However, when the dataset is small (as in the case at hand), CNNs often struggle to provide high generalisation capability and the risk of overfitting is significant (Goodfellow et al., 2016). To overcome these issues, several techniques have been proposed in the past, such as *data augmentation* and transfer learning.

4.2.1. *Data augmentation*

Data augmentation (DA) aims at increasing the size of the dataset, by applying some transformations to the original images (Shorten & Khoshgoftaar, 2019). DA supports in reducing the overfitting, and achieving specific invariances. In the case of lichens images, rotation, horizontal flipping, (moderate) shearing and (moderate) rescaling can be considered harmless, but useful, transformations. Thus, the dataset has been augmented by performing:

- random rotation by a uniformly sampled value from the range $[-40°, 40°]$;
- horizontal flipping;
- random shearing, with shear range equal to 0.2, which is the shear angle in counter-clockwise direction in degrees;
- random rescaling, with a rescaling factor ranging from 0.8 to 1.2.

To implement these techniques, *ImageDataGenerator* from Tensorflow (Abadi et al., 2015) has been used: it ensures that the model receives new variations of the images at each iteration of the training phase, but it only returns the transformed images and does not add it to the original corpus of images.

4.2.2. *Model architecture*

Given the heavy overparameterization of the CNNs, with a small dataset, it is almost impossible to build a model from scratch: in this context, transfer learning can be employed. TL is not restricted to neural networks. Rather, it is a general approach, whose basic idea is that some learned features are important regardless the specific task. In practice, part of a network, trained on a large generic dataset, is used as a component of a task-specific architecture.

The model employed in the present study is based on the well-known architecture VGG16 (Simonyan & Zisserman, 2014), where the filters have been set to a very small receptive field: 3×3. Only the convolutional part of the net was exploited, built in the following way:

- 2 blocks formed by 2 convolutional layers and a max-pooling layer;
- 3 blocks formed by 3 convolutional layers and a max-pooling layer.

In Fig. 5, the convolutional layers are represented in light orange, while the pooling layers in dark orange. The Tensorflow implementation of VGG16 (Abadi et al., 2015), trained on ImageNet (Deng et al., 2009), has been adopted. In order to perform classification, the following layers (in blue and purple in Fig. 5) were appended to the convolutional part of the base net:

- a flattening layer;
- a batch-normalization layer;
- a dense layer with 512 units and rectified linear unit (ReLU) activation function;
- a dropout layer with dropout probability $p = 0.5$;
- a dense layer with 128 units and ReLU activation function;
- a dropout layer with $p = 0.3$;
- an output layer with 20 units (each corresponding to a class) and Softmax activation function.

4.3. *Scattering networks*

Scattering networks (Bruna & Mallat, 2013), or *Scatnets*, are the last method employed and evaluated in this study. Contrary to CNNs, they do not exhibit heavy overparameterization, hence they are a reasonable choice in the case of data scarcity. Scatnets can be thought of as a mixture of classic methods and data-driven methods: their structure resembles that of

CNNs, in the sense that they are based on cascade of convolutions with spatial filters. However, differently from CNNs, filters are fixed, thus their coefficients are not learnt during training. More precisely, filters are fixed waveforms of small spatial support, called wavelets, and their application is followed by a non linear operator (the modulus); through these operations, Scatnets extract a set of coefficients called *localized scattering coefficients* (LSCs), which are used to perform classification. The fixed filters, as in (Bruna & Mallat, 2013), have the form of a *complex Morlet wavelet*, which is essentially a complex periodic function modulated by a Gaussian. More precisely, the filters have the form:

$$\psi(u) = (\cos(\omega u) + i\sin(\omega u) - K)e^{-|u|^2/(2\sigma^2)} \quad (1)$$

where i is the imaginary unit, σ defines the width of the Gaussian window, ω is a spatial frequency, and K is chosen to ensure zero mean. A family of rotated and spatially scaled version of the wavelet (1) is built; notice that, although the spatial frequency ω of the base wavelet is fixed, thanks to the spatial scaling, the family encompasses elements of different spatial frequencies. Figure 6 reports the modulus of some elements of the family of filters. For each element of that family, the *scatter transform* with the image is computed, which is defined as

$$\forall u \in \mathbb{R}^2, \quad (x \star \psi_\lambda)(u) = \iint_{\mathbb{R}^2} x(v)\psi_\lambda^*(u-v)dv$$

where $\star$ and * denote convolution and complex conjugation, respectively. Then, the L^1-norm is calculated, obtaining the so called *scattering coefficients*:

$$\int |x \star \psi_\lambda| du = \|x \star \psi_\lambda\|_1$$

where ψ_λ denotes the specific element of the wavelet family. The L^1-norm is a representation of the scatter transform which is both translation-invariant and stable to both additive noise and deformations. In practice, the LSCs are extracted, by filtering the complex modulus of the scatter transform with a scaled Gaussian (Bruna & Mallat, 2013), and then by computing the integral. More coefficients are computed by composing these operations several times, thus obtaining a cascade structure. The Scatnet structure is depicted in Fig. 7, in which the following operators are defined:

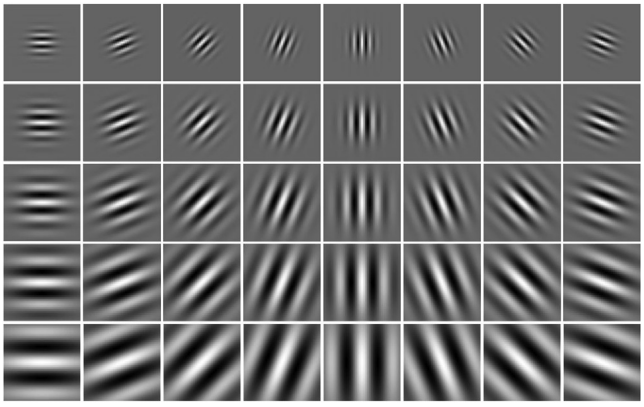

Fig. 6 – The modulus of some elements of the family of oriented filters employed in the Scatnet approach.

- given a path $p = (\lambda_1, \lambda_2, \ldots, \lambda_n)$, where each element λ_i represents a specific wavelet, the compositions of scatter transforms and modulus operations are defined as the U operator:

$$U[p]x = |\ldots||x \star \psi_{\lambda_1}| \star \psi_{\lambda_2}| \ldots | \star \psi_{\lambda_n}|;$$

  Notice that $U[\varnothing]x = x$.

- then, the LSCs are computed, represented by the convolution of the output of the U operator with a scaled Gaussian filter φ_{2^J}

$$S[p]x(u) = \int U[p]x(v) \times \varphi_{2^J}(u-v)_{2^J} dv = U[p]x \star \varphi_{2^J}(u),$$

where

$$\varphi_{2^J} = 2^{-2J}\varphi(2^{-J}u),$$

and parameter J represents the size of the spatial windows, to be fixed in the experimentation phase.

The main characteristics of the Scatnets are:

- they do not need too much data, as they are based on fixed filters;
- they give a representation of the image which is invariant to rotations and stable to deformations (Mallat, 2012).

4.3.1. *Model architecture*

LSCs are exploited as input to a CNN. As said above, the reason for such an architecture is to merge a handcrafted approach, based on Scatnets, and a data-driven one, based on CNNs. The whole model is depicted in Fig. 8; it is a deep network, in which the first part (the olive colored block on the left) is fixed, while the second part is trained based on data. To deal with RGB images, LSCs are extracted for each channel, and the obtained results are concatenated together. The CNN part of the network is composed as follows:

- a block (named "first convolutional block" in Fig. 8) formed by a convolutional layer with 256 filters of size 3 and ReLU activation function, followed by a batch-normalization layer;
- a max-pooling layer with both stride and size equal to 2;
- two convolutional blocks, each formed by a convolutional layer with 512 filters of size 3 and ReLU activation function, followed by a batch-normalization layer;
- an adaptive pooling layer, that reduces dimension of filters to $512 \times 2 \times 2$;
- a flattening layer;
- a final output layer with Log-Softmax activation function.

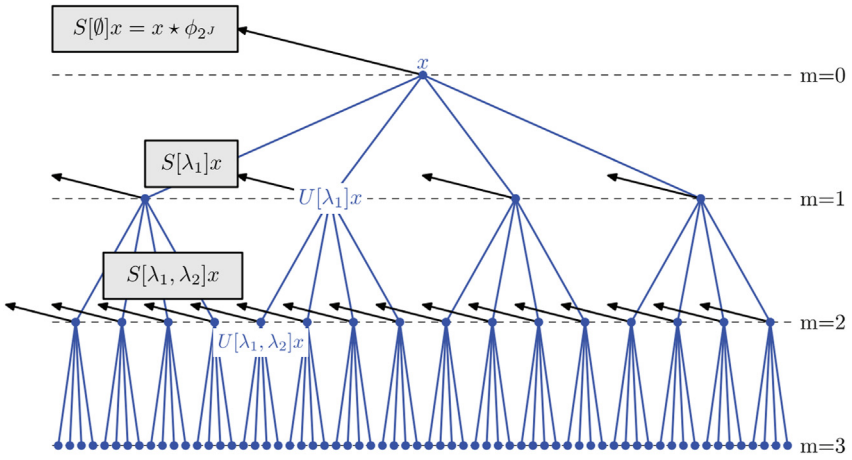

Fig. 7 – Overview of the structure of a Scatnet (image from Bruna & Mallat (2013)). Starting from an initial image, in the first layer the *U* operator is computed for each element of the Wavelet family, and then all the LSCs are extracted. The process is then repeated in following layers, until the maximum depth of the net (fixed a priori) is reached.

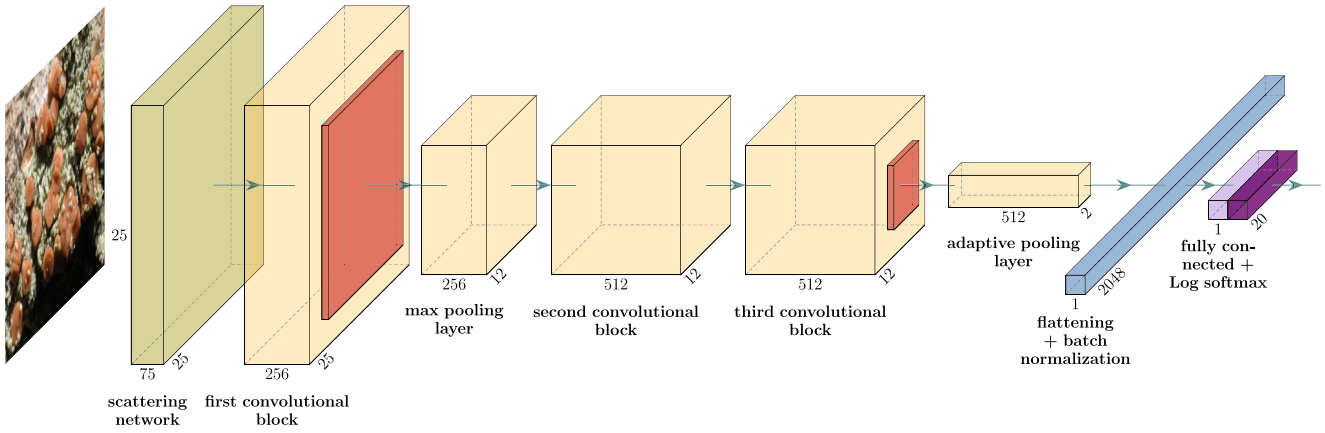

Fig. 8 – Overview of the structure used to perform classification of lichens trough LSCs, extracted with the scattering network. Each convolutional block includes also a batch normalization layer.

5. Experiments and results

The outcome of each method is analyzed individually. Then they are compared, highlighting their advantages and disadvantages. The dataset has been divided into train data (70%), and test data (30%). Models are evaluated on the basis of their *accuracy*, which ranges from 0 (no element from test dataset was correctly classified) to 1 (all elements from test dataset were correctly classified). Other useful and well-known metrics to evaluate the results are *precision*, *recall*, and *F1-score*, defined as:

$$\text{precision} = \frac{TP}{TP + FP}, \qquad \text{recall} = \frac{TP}{TP + FN},$$

$$F1 - \text{score} = 2\frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}},$$

where *TP* stands for true positive, *FP* for false positive, and *FN* for false negative. Useful insights can be obtained by observing how they vary for the different species of lichens. Thus, for some models, metrics distribution is discussed as well.

5.1. *Classic methods*

The parameter values for each descriptor were set empirically, by exploiting a random search among a reasonable range for each parameter, with the aim of finding the ones maximising accuracy. As a result, the parameters were set as follows:

- for *LBPs*, for each pixel of interest the number of points that forms the neighbourhood was fixed to 8, and the distance from the central point (the radius) to 2;
- for *SIFT*, the number of visual words was fixed to 600;
- for *dense SIFT*: the number of visual words was set to 600, and grid density, i.e., the distance at which points are taken to extract descriptors, was set to 4. Through convolutions with Gaussian filters with different standard deviation, images at several scales were taken into account, in order to partially achieve invariance to

scale: standard deviations were set empirically to $\sigma \in \{0.53, 1.06, 1.6, 2.13, 2.7\}$;
- for *Gabor based descriptors*, 8 different orientations were considered, calculated as

$$\theta = \frac{j}{m}\pi \qquad j = 0, \ldots, 7$$

and 3 scales defined by the parameters

$$\Omega = \frac{0.25}{\left(\sqrt{2}\right)^{k}} \qquad k = 0, \ldots, 2 \qquad \sigma = \frac{\Omega}{\sqrt{2}},$$

where Ω refers to the frequency of the Gabor function, while σ is the standard deviation of the Gaussian envelope.

- For *BGPs*, the settings of Zhang et al. (2012), were adopted, i.e. 8 orientations

$$\theta = \{\theta_j = j\pi / 8 : j = 0, 1, \ldots, 7\}$$

and 3 different scales $\{(\Omega_k, \sigma_k)$: $k = 0, 1, 2\}$:

$$(\Omega_0, \sigma_0) = (1.3, 1.42) \qquad (\Omega_1, \sigma_1) = (5.2, 0.4)$$

$$(\Omega_2, \sigma_2) = (22, 0.22),$$

where σ_k and Ω_k have the same meaning of the previous point.

The best result in terms of accuracy (Table 2) was achieved by *dense SIFT*, which outperformed other approaches, and especially its sparse version (the SIFT). Since both the approaches rely on the same kind of descriptor, the only difference being the locations of the interest points, it seems reasonable to conclude that for facing the particular classification problem considered in this study, a large set of uniformly sampled feature points, rather than a small set of keypoints, is preferable. Furthermore, the dense SIFT achieved remarkably consistent performance over the different classes. Indeed, the standard deviation of the accuracy with respect to different species is much smaller than the value obtained with other descriptors. This is also clear from the confusion matrix in Fig. 9, in which almost all the species are well recognised, with a few exceptions. *Candelariella Reflexa* (Cr) has an accuracy of 0.65 only, as evidenced also by low values of precision and recall (Fig. 10).

Table 2 – Results obtained by employing classic methods, with different descriptors. The rightmost column reports the standard deviation of the accuracy across the 20 different species of lichens.

Method	Accuracy	Precision	Recall	Standard deviation
LBPs	0.54	0.51	0.50	0.18
SIFT	0.52	0.54	0.51	0.2
dense SIFT	**0.89**	**0.89**	**0.9**	**0.09**
Gabor-based	0.83	0.77	0.78	0.16
BGPs	0.82	0.85	0.84	0.17

For each column, the best result (highest accuracy, precision, recall, and lowest standard deviation) is reported in bold.

5.2. *CNN-based model*

As far as input data are concerned, the net was forced to accept RGB images of size 100×100. The CNN used to perform lichens classification was trained using two different approaches:

- by training only the dense layers that were attached to the base VGG16 net, thus leaving unaltered the weights relative to the convolutional part of the structure, for a total of 24 37 268 trainable parameters.
- by training the whole structure, for a total of 1 71 51 956 trainable parameters. In this case, parameters related to the convolutional part of the structure have been fine-tuned, while the ones related to the dense layers are trained from scratch.

Categorical crossentropy was employed as loss function, while Adam optimiser was adopted, with learning rate equal to 2×10^{-5}. The model was trained for a maximum of 100 epochs, using early stopping as termination criterion. The term *epoch* refers to one cycle of training, in which the network is presented all the training data. In an epoch, all the data are used exactly once. The training is essentially a gradient-descent based numerical optimisation in which the network weights are adjusted in an iterative manner, to reduce the overall loss; it requires pass-throughs of the whole training set into the network since, to enforce convergence of the weights, only small adjustments are performed at each iteration. The best result (Table 3) was achieved with the second approach (albeit with a slightly increased standard deviation). This suggests that, in order to build an effective model for the classification of lichens, specific features need to be extracted: pre-trained weights obtained from generic datasets are not enough to reach acceptable results. The need for specific features is confirmed by the analysis of Figs. 11 and 12, which show how accuracy and loss vary through epochs. When the last dense layers only are tuned, test performance are always better than train performance, both in accuracy and loss. In other words, the chosen model and the employed training algorithm are not capable of overfitting the training data. As for the model, a possible explanation for that observation is that the features extracted by the convolutional part of the net, are not sufficiently expressive for describing properly the lichens. Thus, pre-trained networks are of limited use in this case.

5.3. *Scattering networks*

The initial wavelet filter (1) was set to have $\sigma = 0.85$ and $\omega = \frac{3\pi}{4}$ as parameters. In order to create the wavelet filter bank, four parameters were tuned:

- number of the scales, fixed empirically to 2;
- number of rotations, fixed empirically to 4;
- spatial windows of the Gaussian filter, fixed empirically to be the same value of the number of scales.
- depth of the net, in terms of number of layers: this value has been fixed to 2, as it has been proven that it is enough to extract the majority of information from an image (Bruna & Mallat, 2013).

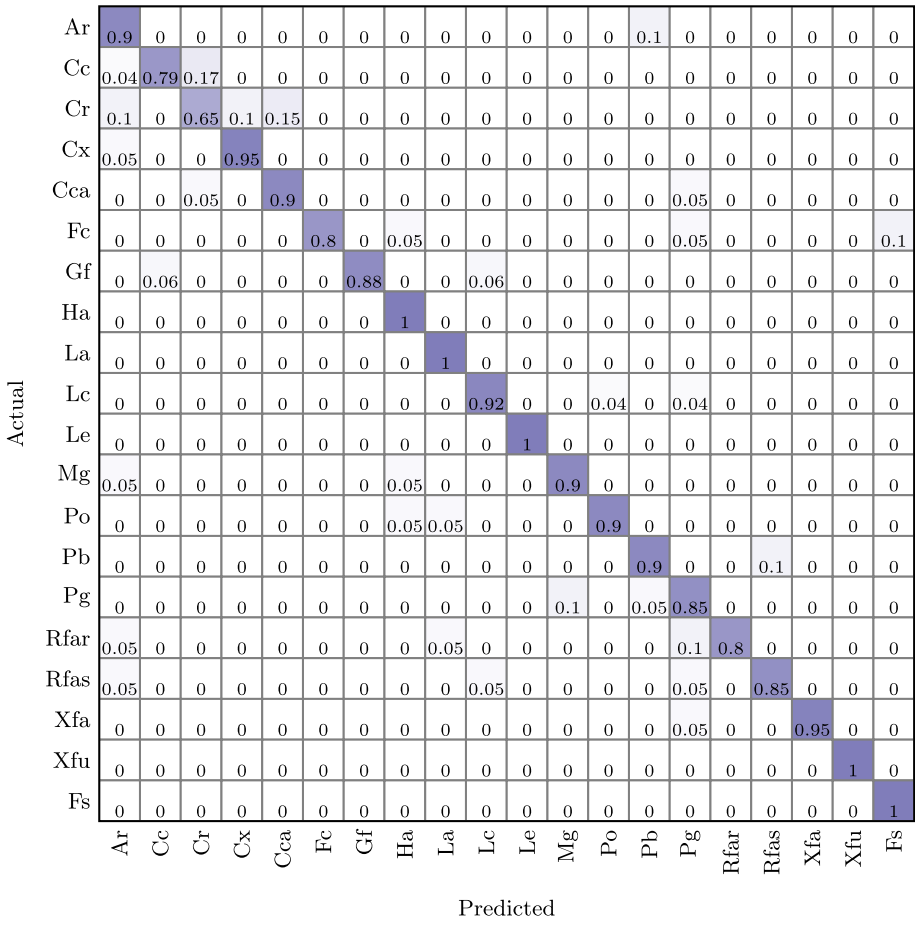

Fig. 9 – Confusion matrix obtained using the model based on Dense Sift descriptor.

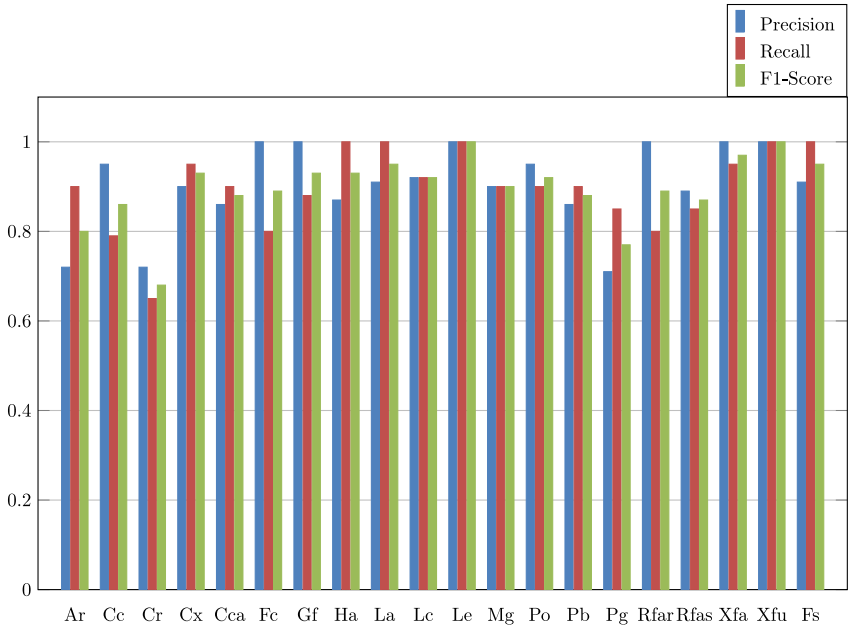

Fig. 10 – Distribution of the precision, recall, and F1-score among different species, using dense SIFT method to extract descriptors.

Table 3 – Results obtained using CNN based model.

Training	Accuracy	Precision	Recall	Standard deviation
dense layers	0.73	0.72	0.74	**0.16**
whole net	**0.82**	**0.79**	**0.84**	0.18

For each column, the best result (highest accuracy, precision, recall, and lowest standard deviation) is reported in bold.

The resulting output, for a fixed image of the dataset, was of dimension (25, 25, 75), and served as input for the CNN (Section 4.3.1). For what concerns the training phase, cross-entropy was used as loss function, with stochastic gradient descent as optimiser, with momentum equal to 0.9, and initial learning rate equal to 0.01, decreased by a factor of 0.2 every 20 epochs. The model was trained for a maximum of 100 epochs, using early stopping as termination criterion. The same data augmentation as in Section 4.2.1 was applied. Kymatio package (Andreux et al., 2020) was used to implement Scatnet, and Pytorch (Paszke et al., 2019) for the entire net. The average accuracy obtained was ~ 0.83, with both precision and recall ~ 0.84, and a standard deviation ~ 0.16: thus, the Scatnet approach outperformed the one based on transfer learning and CNN. Almost all the species were quite well discriminated (Fig. 13), with a few exceptions. As an example, *Gyalolechia Flavorubescens* was predicted correctly only 33% of the time, and this means that this species has a high percentage of false negatives, and thus a low recall value (Fig. 14). In general, there is much more fluctuation in the quality of the classification in comparison to the model based on dense SIFT descriptor.

6. Discussion

The best result was obtained using a classical method, namely the dense SIFT descriptor, with an accuracy of ~ 0.89. Remarkably, a similar approach, i.e., the sparse SIFT descriptor, achieved quite poor results, when compared to the dense version of the same algorithm, thus suggesting that a dense feature extraction from images is needed to obtain relevant information for the task of identifying lichens. The model based on CNNs did not outperform classic methods, although it performed well. A possible reason is data scarcity, which has not been sufficiently mitigated for by the data augmentation techniques applied. The model based on LSCs achieved quite good results in terms of performance and stability, but it did not outperform dense SIFT descriptor. However, its structure is appealing from a computational point of view, since it is based on fixed filter and has few trainable parameters. A recent study (Galanty et al., 2021), considered a classification problem involving 12 different species, achieving an accuracy of ~ 0.61. Such accuracy is hardly comparable to the one achieved in the present paper, since Galanty et al. (2021) worked on whole images instead of patches (as in the present study). Nevertheless, the rather low accuracy achieved by Galanty et al. (2021) testifies that the problem is indeed a difficult one, and advocates for specific techniques.

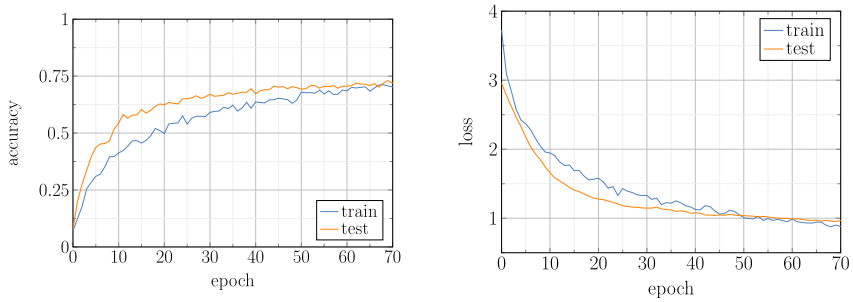

Fig. 11 – Variation of accuracy (left) and loss (right) through epochs, when fine-tuning only the dense part of the CNN: the training performance is reported in blue, the test one in orange. (For interpretation of the references to color in this figure legend, the reader is referred to the Web version of this article.)

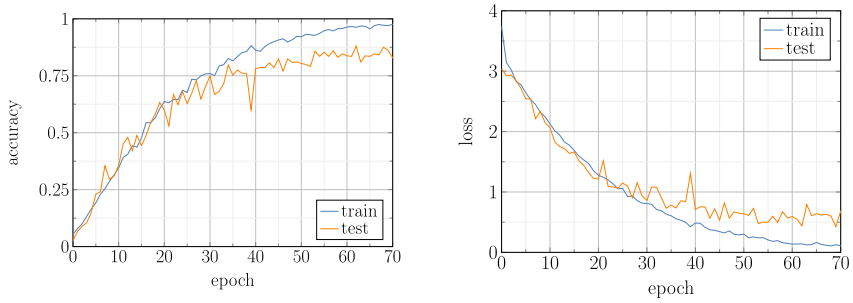

Fig. 12 – Variation of accuracy (left) and loss (right) through epochs, when we train the whole net: the training performance is reported in blue, the test one in orange. (For interpretation of the references to color in this figure legend, the reader is referred to the Web version of this article.)

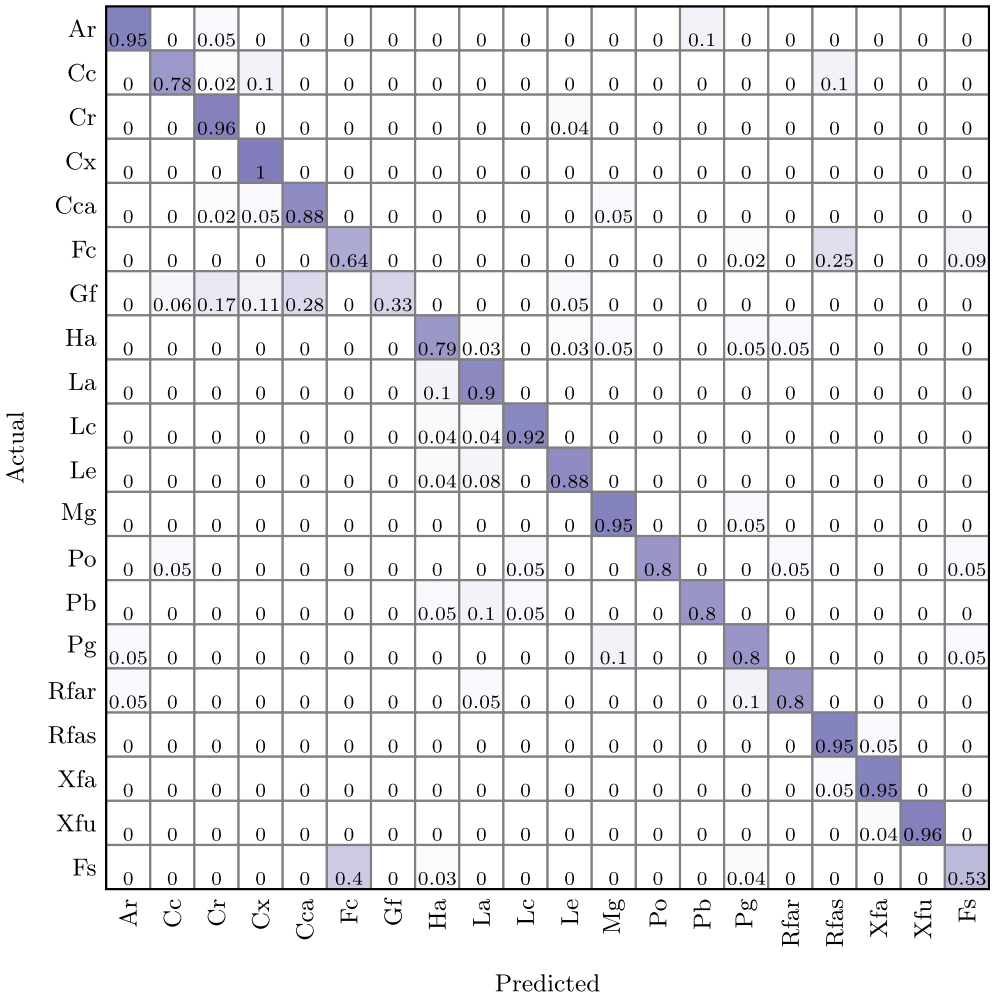

Fig. 13 – Confusion matrix obtained using the model based on Scattering networks.

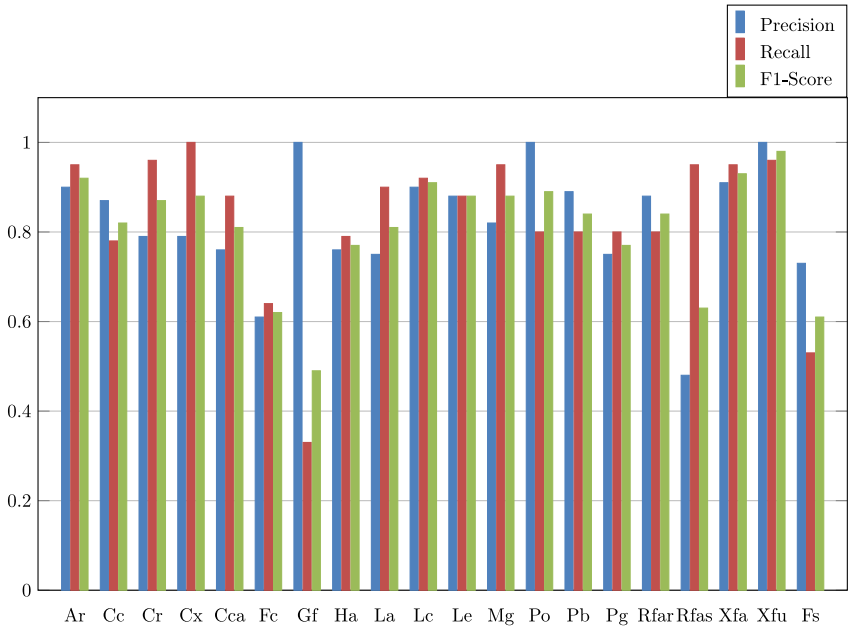

Fig. 14 – Distribution of precision, recall, and F1-score with respect to different species, using model based on scattering networks.

We believe that the present work, albeit preliminary, represents an advancement toward the effective practical application of automated lichens classification in fieldwork. To this aim, however, a considerable amount of further work still needs to be carried out. On the one hand, the set of considered species needs to be enlarged, and the adequacy of the proposed techniques must be assessed on the new classification problem. On the other, the computational effort at inference time required by the model has to be taken into account for the model to be deployed on mobile devices. Whether the application of some model reduction techniques such as, e.g., pruning (Zullich et al., 2021) will be necessary, is a matter of further investigation. Clearly, deploying the classification system on mobile devices would make possible to use citizen science for environmental monitoring. The automation of the training pipeline, to allow non-machine learning experts not only to employ an automated classification system, but also to train it or fine-tune it to newly acquired data is a further (challenging) aim that would allow to refine the system or to adapt it to similar tasks, based on new data.

We conclude the discussion by recalling that, to properly use lichens as bioindicators, it is necessary to count the number of different *taxa* per unit area. Thus, the estimation of the extent of the surface captured in an image is of fundamental importance. Since the present study is a preliminary one, such an issue was not taken into account. However, addressing it could be a relatively easy task, given that modern mobile devices are equipped with sensors capable of acquiring depth images, thus allowing a solution by exploiting three-dimensional data. Alternatively, the actual size of the captured area could be roughly estimated by a separate neural network trained on labelled examples of the exact lichen species combined with information on the size of lobes and on the distance of the subject to the camera. The use of lichens diversity for estimating the LBI requires the correct identification of the taxa occurring in a given portion of a tree trunk (Anpa, 2001). The identification of lichens in the field is often a challenging task, and can require the investigation of anatomic features, such as spores shape and size, to achieve an identification at the specific or infraspecific level. Human operators monitoring the LBI can collect samples for performing a more detailed identification in a laboratory, since it was evidenced (Brunialti et al., 2002) that the average score of correct identification by operators in the field can sometimes be below 50%. Arguably, an automated system will never be able to achieve the same results of human operators by the sole analysis of images taken in the field. However, even if not always capable of identifying all the taxa occurring in an image to the species level, a machine learning-based automated system can certainly provide an estimation of the lichen diversity based upon morphology alone. Plus, it will be capable to provide an output far quicker than any human operator. Thus, the application of a machine learning-based system for monitoring the morphological diversity of lichen taxa occurring on tree trunks could provide a first estimate of LBI in a very limited amount of time, without the need for trained operators, and even involving volunteers in citizen science activities. After a first estimate, authorities interested in a more detailed estimate can involve experts in those areas which are highlighted as critical in the first estimate. Such a combined approach could allow for a wider coverage of LBI and air quality investigation, while at the same time focusing efforts and fundings in critical areas only.

7. Conclusion

We have reported the first (to our knowledge) horizontal study aimed at achieving an automated image-based lichens classification, testing different approaches ranging from classical image descriptors to convolutional neural networks to a combination of descriptors and convolutional neural networks. The study considered 20 species and employed a patch-based classification approach to mitigate the scarcity of data. All the methods resulted in good accuracy, the best (~ 0.89) being the one obtained by the classical dense SIFT descriptor. We have also briefly discussed the possible impact of the proposed methodology and results on lichen biodiversity index assessments.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors are grateful to Prof. Mauro Tretiach for his useful suggestions. They are also indebted to the anonymous reviewers for their helpful comments, that definitely improved the quality of the manuscript.

REFERENCES

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., … Zheng, X. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. URL: https://www.tensorflow.org/software.available.from.tensorflow.org.

Aljahdali, S., Ansari, A., & Hundewale, N. (2012). Classification of image database using SVM with Gabor magnitude. In *2012 international conference on multimedia computing and systems (pp. 126–132). Tangiers, Morocco* https://doi.org/10.1109/ICMCS.2012.6320228

Ammann, K., Herzig, R., Liebendoerfer, L., & Urech, M. (1987). Multivariate correlation of deposition data of 8 different air pollutants to lichen data in a small town in Switzerland. *Advances in Aerobiology*, 51, 401–406.

Andreux, M., Angles, T., Exarchakis, G., Leonarduzzi, R., Rochette, G., Thiry, L., Zarka, J., Mallat, S., Andén, J., Belilovsky, E., Bruna, J., Lostanlen, V., Chaudhary, M., Hirn, M. J., Oyallon, E., Zhang, S., Cella, C., & Eickenberg, M. (2020). Kymatio: Scattering transforms in python. *Journal of Machine Learning Research*, 21, 1–6. URL: http://jmlr.org/papers/v21/19-047.html.

Anpa, I. (2001). *Indice di biodiversità lichenica: Manuale*. Roma: ANPA, Agenzia Nazionale per la Protezione dell'Ambiente.

Bini, G., Bonannini, M., & Ferrarese, R. (2001). *I.B.L. indice di biodiversità lichenica*. Manuale ANPA, Dipartimento di Stato dell'Ambiente, Controlli e Sistemi Informativi.

Bruna, J., & Mallat, S. (2013). Invariant scattering convolution networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 35*, 1872–1886. https://doi.org/10.1109/TPAMI.2012.230

Brunialti, G., Giordani, P., Isocrono, D., & Loppi, S. (2002). Evaluation of data quality in lichen biomonitoring studies: The Italian experience. *Environmental Monitoring and Assessment, 75*, 271–280.

Burges, C. J. (1998). A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery, 2*, 121–167.

Castello, M., Martellos, S., & Nimis, P. L. (2006). Victoria: An on-line information system on the lichens of Victoria Land (Continental Antarctica). *Polar Biology, 29*, 604–608.

Cislaghi, C., & Nimis, P. L. (1997). Lichens, air pollution and lung cancer. *Nature, 387*, 463–464.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition (pp. 248–255). Miami, FL, USA*. https://doi.org/10.1109/CVPR.2009.5206848

Dyrmann, M., Karstoft, H., & Midtiby, H. S. (2016). Plant species classification using deep convolutional neural network. *Biosystems Engineering, 151*, 72–80. https://doi.org/10.1016/j.biosystemseng.2016.08.024. URL: https://www.sciencedirect.com/science/article/pii/S1537511016301465.

Fei-Fei, L., & Perona, P. (2005). A bayesian hierarchical model for learning natural scene categories. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05) (pp. 524–531). San Diego, CA, USA: IEEE volume 2*.

Galanty, A., Danel, T., Węgrzyn, M., Podolak, I., & Podolak, I. (2021). Deep convolutional neural network for preliminary in-field classification of lichen species. *Biosystems Engineering, 204*, 15–25.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press. http://www.deeplearningbook.org.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *2016 IEEE conference on computer vision and pattern recognition (CVPR) (pp. 770–778). Las Vegas, NV, USA*. https://doi.org/10.1109/CVPR.2016.90

Iandola, F. N., Han, S., Moskewicz, M. W., Ashraf, K., Dally, W. J., & Keutzer, K. (2016). Squeezenet: Alexnet-level accuracy with 50x fewer parameters and < 0.5mb model size. *arXiv, 1602*, Article 07360.

Kanmani, P., & Rajiv Kannan, A. (2017). *Evaluation of morphological responses in Parmotrema Tinctorum lichen using back propagation neural networks collected from Servarayan hills of Tamilnadu, India*. International Research Journal of Pharmacy.

LeCun, Y., Kavukcuoglu, K., & Farabet, C. (2010). Convolutional networks and applications in vision. In *Proceedings of 2010 IEEE international symposium on circuits and systems* (pp. 253–256). IEEE.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision, 60*, 91–110.

Mallat, S. (2012). Group invariant scattering. *Communications on Pure and Applied Mathematics, 65*, 1331–1398.

Morrison, L. W. (2016). Observer error in vegetation surveys: A review. *Journal of Plant Ecology, 9*, 367–379.

Napoletano, P. (2017). Hand-crafted vs learned descriptors for color texture classification. In S. Bianco, R. Schettini, A. Trémeau, & S. Tominaga (Eds.), *Computational color imaging* (pp. 259–271). Cham: Springer International Publishing.

Nimis, P. L. (2016). *The lichens of Italy. A second annotated catalogue*. EUT Edizioni Università di Trieste.

Nimis, P., & Martellos, S. (2002). Italic, a database on Italian lichens. *Bibliotheca Lichenologica, 82*, 271–282.

Nimis, P. L., & Purvis, O. W. (2002). Monitoring lichens as indicators of pollution. In P. L. Nimis, C. Scheidegger, & P. A. Wolseley (Eds.), *Monitoring with lichens—monitoring lichens* (pp. 7–10). Springer.

Nimis, P. L., Scheidegger, C., & Wolseley, P. A. (2002). Monitoring with lichens—monitoring lichens. In *Monitoring with lichens—monitoring lichens* (pp. 1–4). Springer.

Ojala, T., Pietikainen, M., & Maenpaa, T. (2002). Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 24*, 971–987.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., & Chintala, S. (2019). *Pytorch: An imperative style, high-performance deep learning library* (pp. 8024–8035).

Preetha, K. S., Yuvaraj, N., Jenifa, G., Indhu, R., & Kanmani, P. (2021). Lichen element based autonomous air pollution monitoring around smart cities – a deep learning approach. *Turkish Journal of Computer and Mathematics Education, 12*, 151–161.

Rassem, T. H., & Khoo, B. E. (2011). Object class recognition using combination of color sift descriptors. In *2011 IEEE international conference on imaging systems and techniques (pp. 290–295). Batu Ferringhi, Malaysia: IEEE*.

Rawat, W., & Wang, Z. (2017). Deep convolutional neural networks for image classification: A comprehensive review. *Neural Computation, 29*, 2352–2449.

Saleem, M. H., Khanchi, S., Potgieter, J., & Arif, K. M. (2020). Image-based plant disease identification by deep learning meta-architectures. *Plants, 9*. https://doi.org/10.3390/plants9111451. URL: https://www.mdpi.com/2223-7747/9/11/1451.

Shorten, C., & Khoshgoftaar, T. (2019). A survey on image data augmentation for deep learning. *Journal of Big Data, 6*. https://doi.org/10.1186/s40537-019-0197-0

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv, 1409–1556*.

Sivic, & Zisserman. (2003). Video google: A text retrieval approach to object matching in videos. In *Proceedings ninth IEEE international conference on computer vision (pp. 1470–1477 Vol. 2). Nice, France*. https://doi.org/10.1109/ICCV.2003.1238663

Soon Jye, K., Sugumaran, M., Sorayya, M., Mogeeb, M., & Sarinder Kaur, D. (2017). Automated plant identification using artificial neural network and support vector machine. *Frontiers in Life Science, 10*(1), 98–107.

Szeliski, R. (2010). *Computer vision: Algorithms and applications* (1st ed.). London: Springer-Verlag.

Tan, C., Sun, F., Kong, T., Zhang, W., Yang, C., & Liu, C. (2018). A survey on deep transfer learning. In V. Kůrková, Y. Manolopoulos, B. Hammer, L. Iliadis, & I. Maglogiannis (Eds.), *Artificial neural networks and machine learning – ICANN 2018* (pp. 270–279). Cham: Springer International Publishing.

Zhang, L., Zhou, Z., & Li, H. (2012). Binary Gabor pattern: An efficient and robust descriptor for texture classification. In *2012 19th IEEE international conference on image processing (pp. 81–84). Orlando, FL, USA*. https://doi.org/10.1109/ICIP.2012.6466800

Zullich, M., Medvet, E., Pellegrino, F. A., & Ansuini, A. (2021). Speeding-up pruning for artificial neural networks: Introducing accelerated iterative magnitude pruning. In *2020 25th international conference on pattern recognition (ICPR) (pp. 3868–3875)*. https://doi.org/10.1109/ICPR48806.2021.9412705. Online.