Multilabel Annotation of Multispectral Remote Sensing Images using Error-Correcting Output Codes and Most Ambiguous Examples

Anamaria Radoi, *Member, IEEE*, and Mihai Datcu, *Fellow, IEEE*

***Abstract*—This paper presents a novel framework for multilabel classification of multispectral remote sensing images using error-correcting output codes. Starting with a set of primary class labels, the proposed framework consists in transforming the multiclass problem into multiple binary learning subtasks. The distributed output representations of these binary learners are then transformed into primary class labels. In order to train robust binary classifiers on a reduced annotated dataset, the learning process is iterative and involves determining most ambiguous examples, which are included in the training set at each iteration. As part of the semantic image recognition process, two categories of high-level image representations are proposed for the feature extraction part. First, deep convolutional neural networks are used to form high-level representations of the images. Second, we test our classification framework with a bag-of-visual words model based on the scale invariant feature transform, used in combination with color descriptors. In the first case, we propose the usage of pretrained state-of-the-art deep learning models that cancel the need to estimate model parameters of complex architectures, whereas, in the second case, a dictionary of visual words must be determined from the training set. Experiments are conducted on GeoEye-1 and Sentinel-2 images and the results show the effectiveness of the proposed approach toward a multilabel classification, when compared to other methods.**

***Index Terms*—Error-correcting output codes (ECOCs), multilabel image classification, pretrained convolutional neural networks, support vector machines (SVMs).**

I. Introduction

THE technological development of the recently launched satellites allowed the acquisition of large archives of high-resolution (HR) and very HR images. In general, the semantic annotation of these images is a difficult task to accomplish because of the massive volume of unlabeled data and to the fact that the labeling process is expensive. In this context, automatic procedures for image classification and information retrieval are required.

Image classification can be approached using supervised and unsupervised methods, respectively. Unsupervised classification methods group the data into clusters of samples with similar characteristics and they do not require labeled data. However, this leads to a lack of correspondence between the retrieved clusters and their semantic meaning. In contrary, supervised classification methods start with a set of labeled samples (called training set), which provides a precise correspondence between the samples and their labels. This information is further used to derive models for the classification system that are afterward applied on a *test set*.

Among the supervised classification methods used in the remote sensing domain, methods based on support vector machines (SVMs) have shown good performance results [1]–[3]. In the majority of situations, the classification of remote sensing images involves the discrimination between multiple classes. This problem is usually tackled by combining multiple binary classifiers. SVMs are examples of classifiers whose algorithm cannot deal with multiple classes directly, but they have to resort to techniques that decompose the multiclass problem into several binary classification problems. The most popular multiclass classification techniques are one-versus-rest (OVR) and one-against-one (OAO). The OVR strategy consists in conducting one binary classification per class that discriminates between the samples from one class against the samples from the rest of the classes. The OAO strategy performs pairwise comparisons between classes, yielding to a number of $\mathrm{nc}(\mathrm{nc}-1)/2$ binary classifiers if nc is the number of classes. Numerous experiments have shown that the OAO method is more suitable for practical use than the OVR method [4]. One of the reasons for this behavior is the fact that in the case of the OVR approach, the binary problems are unbalanced and this becomes even more problematic for an increasing number of classes. In this sense, an optimal combination of the classes can represent a possible solution.

Recent advances have shown that convolutional neural networks (CNNs) achieve high accuracy values for remote sensing image classification [5]. Very deep CNNs, such as VGG [6], are difficult to train and, as the network gets deeper, the performance of the classification may get saturated and may even start to degrade rapidly due to vanishing gradients. As a solution to this problem, He *et al.* proposed a novel neural network architecture, called residual network (ResNet), that has achieved state-of-the-art performance in image classification, object detection, and

Manuscript received September 14, 2018; revised February 3, 2019 and March 24, 2019; accepted May 5, 2019. Date of publication July 1, 2019; date of current version July 30, 2019. This work was supported by the German Aerospace Center DLR. *(Corresponding author: Anamaria Radoi.)*

A. Radoi is with the Faculty of Electronics, Telecommunications and Information Technology, University Politehnica of Bucharest, 061071 Bucharest, Romania (e-mail: anamaria.radoi@upb.ro).

M. Datcu is with the Remote Sensing Technology Institute, German Aerospace Center, 82234 Oberpfaffenhofen, Germany, and also with the Research Center for Spatial Information CEOSpaceTech, UPB, 061071 Bucharest, Romania (e-mail: mihai.datcu@dlr.de).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JSTARS.2019.2916838

semantic segmentation tasks [7]. Inspired from the VGG architecture, ResNet introduces "shortcut connections" that perform identity mapping and add the result to the outputs of several stacked layers. However, the main disadvantage of these architectures is the need for learning a large number of parameters, and, as a direct consequence, the training set has to be large.

An alternative approach for solving the multiclass learning problem is to use Error-Correcting Output Codes (ECOCs) [8] that reduce the multilabel classification to several binary classification problems. There are three main steps that the ECOCs approach to multiclass learning must follow. First, following some predefined rules, primary class labels are combined into several binary metaclasses that are groups of primary class labels. Second, binary classifiers are constructed with the scope of reducing the uncertainty about the correct class of the input. Third, the outputs of the binary classifiers are combined to determine the primary class to which each sample pertains. The ECOC approach can make use of algebraic error-correcting codes or the codes can be designed by the user such that they satisfy some constraints [8]. The classification scheme presented by Dietterich and Bakiri [8] considered associating codewords (from codes with error correction properties) directly to the classes so that the misclassifications can be corrected. ECOC has proved good performance in applications such as speech recognition [8] and document classification [9]. In [10], it is shown that the worst-case training error of the ECOC approach is better than the OAO approach. One reason for this is the fact that most of the binary classifiers encompassed in the multiclass classification scheme decide if a sample pertains to a particular group of classes or not rather than deciding if it pertains to a single class or not (OAO approach). This results in a more balanced partition between the positive and negative examples used for learning the models of the binary classifiers.

In order to yield satisfactory classification accuracy, the size of the training set has to be significant and this implies a large number of annotated samples that have to be provided to the classifier in order to build the model. When dealing with the classification of remote sensing images, the labeled data are scarce. In addition, the quality of the labeled data has a strong influence over the classification results. For this reason, the training samples have to be adequately selected in order to obtain a correct classification even for ambiguous examples. In the last few years, active learning techniques have been developed as possible solutions to the aforementioned issues [3], [11]–[13]. However, in the case of ECOC-based framework, the user would be required to provide information regarding metaclasses and not directly about the original labels as in the case of usual active learning techniques. Therefore, using active learning methods for training several binary classifiers of the ECOC-based framework is difficult. For this reason, we adopt another approach that considers learning from most ambiguous training examples.

In this paper, we propose a novel general solution for the multilabel classification of remote sensing images using an ECOC-based framework composed of multiple binary classifiers for which we define the concept of *metaclasses*. From an information theoretic point of view, each correct classification made by the binary classifiers decreases the uncertainty over the original class labels. Therefore, the binary classifiers need to achieve a high accuracy rate. Considering SVM binary classifiers, this can be obtained through a slight modification of the training procedure that allows determining the most ambiguous training examples for each pair of metaclasses. Starting with a small and suboptimal part of the training set, the actual training set is iteratively enlarged with new samples from the unused part of the training set. The procedure is based on a criterion (e.g., the distance with respect to the separation hyperplane [11], [14], [15]) that facilitates the choice of the most representative samples to iteratively update the parameters of the discriminative models. In this regard, we study the positive impact that the modified training procedure has over the training of binary SVM classifiers.

Assuming that the classification algorithm is established, another critical step in image classification is feature extraction. One of the widely used methods for feature extraction is the bag-of-visual words (BOVWs) model that, in many applications, attains remarkable performances [13], [16]. The BOVWs model makes use of a dictionary of visual words that is determined in an unsupervised manner. This additional step of building a dictionary, although a bit time-consuming, is performed only during the training step. Another option for the feature extraction module is to use a state-of-the-art deep neural network architecture from which we retain only the part corresponding to feature extraction. The train procedure of the deep neural network can be performed directly on the dataset being examined or on an already existing large annotated dataset (i.e., which could have been developed for a completely different image classification task). The first solution is not very practical since it would require training a new deep neural network for each dataset and, as a consequence, huge annotated datasets would be needed. Therefore, we adopt the second solution.

The main contributions of this paper can be summarized as follows: 1) a novel ECOC-based framework for the multilabel classification of remote sensing images; 2) an iterative strategy to train robust ECOC's binary classifiers by determining most ambiguous examples when multilabel classification is involved; 3) a reduction in the computational cost obtained with the iterative strategy for learning ECOC's binary classifiers on small annotated datasets; and 4) a high-level image description method based on a data-independent deep CNN architecture (i.e., modified ResNet) that was pretrained on an existing large annotated dataset designed for other classification tasks.

The rest of the paper is structured as follows. Section II presents an overview of the proposed approach toward the multilabel classification task, whereas Sections III to V present an in-depth analysis of the constituent steps of the classification procedure. Section VI illustrates several experimental results and Section VII draws the final conclusions.

II. Preliminaries and Proposed Approach

A. Problem Formulation

Let us consider a collection of unlabeled remote sensing data $\mathcal{X} = \{\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_N}\}$ that needs to be classified. A small training set represents the starting point in solving the multilabel classification task, whose aim is to label the remaining part of the remote sensing data collection $\mathcal{X}$.

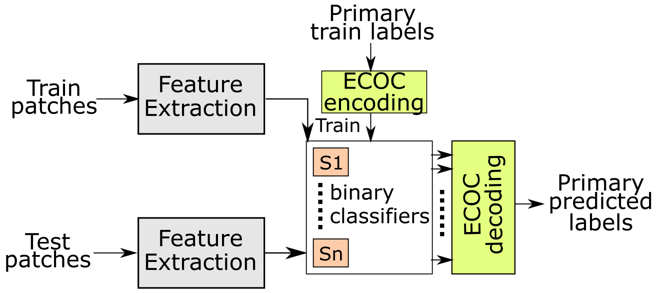


Fig. 1. Scheme of the proposed approach for multilabel classification of remote sensing images.

B. Proposed Approach for Multilabel Classification

The proposed classification framework is composed of two main parts: 1) feature extraction and 2) ECOC-based multilabel classification using the SVM binary classifiers optimized for most ambiguous examples. The whole classification procedure, including the learning phase, is depicted in Fig. 1.

The solution proposed to solve the multilabel classification task is based on ECOC, which decomposes the multilabel problem into several binary-label problems. The approach follows three main steps:

1) establish the code used to encode the class labels (or, class indexes assigned to the class labels);
2) binary classifications;
3) decode the class label from the binary vector formed by the outputs of the binary classifications;

In order to initiate the classification procedure, the first step is to establish the ECOC encoding table, whose role is to provide a one-to-one correspondence between the set of primary class labels and the set of binary codewords. The one-to-one correspondence can be defined using well-known algebraic error-correction codes, random codes, or designed codes.

The encoding table generates the inclusion of the input examples into new classes called, in the following, *metaclasses*. More precisely, the metaclasses and their corresponding metalabels are partitions of the initial classes according to the encoding table. The role of the binary classifiers in the second step is to decide whether an instance is part of a metaclass or not. Each inclusion of the input examples into metaclasses leads, in fact, to a reduction in the ambiguity regarding the corresponding class index/label.

At decoding, all the metalabels provided by the binary classifiers are collected by the decoder into vectors of binary elements. These sequences of 0s and 1s are then corrected and decoded following pre-established decoding rules that depend on the choice of error-correcting output codes. In fact, the decoding is the last step that leads to the annotation of the input examples with the primary class labels based on the decoder's decision.

The binary classifiers used in this paper are SVM classifiers that are trained in an iterative manner. The learning phase is performed over a reduced training set volume and most ambiguous items help to improve the performance of the classifiers. This learning approach, denoted as SVM-MA, leads to an improved robustness level and decreased learning time if compared to learning over larger annotated sets.

TABLE I
EXAMPLE OF CODE

Class index	Codeword		
	Bit 1	Bit 2	Bit 3
class 1	0	0	0
class 2	1	1	1
class 3	1	1	0
class 4	0	1	1

As for the feature extraction module, we consider two categories of high-level image representations. The first category consists in fusing color statistics and BOVW high-level features (denoted as BOVWC). The BOVW model involves computing histograms over the occurrences of visual words from an *a-priori* determined dictionary, whereas the computation of high-level descriptors relies on the extraction of other low-level features [17]. A common choice for the low-level feature extraction is the scale invariant feature transform (SIFT), which computes distinctive local features that are invariant to scale and rotation, and are robust with respect to distortions, noise, or changes in illumination [18]. The second category of high-level image representations is derived using a pretrained deep neural network architecture, designed for a different classification task (e.g., ImageNet challenge [19]). The main advantage of using a pretrained deep neural network architecture is that feature extraction is performed in an unsupervised manner and there is no cost for training the architecture. Moreover, the feature extraction module based on pretrained architectures is data independent. In this paper, we consider using a deep pretrained ResNet architecture that achieved state-of-the-art performance in many visual recognition tasks [7].

III. ECOC-BASED FRAMEWORK FOR MULTILABEL CLASSIFICATION

A. Metaclasses Definition

Let us assume that class indexes are encoded following the table shown in Table I. In this case, each class label is encoded into codewords of three bits. Therefore, in order to determine a class label, we first need to determine the three bits by applying three binary classifications. The metalabels 0_i and 1_i, with i being the ith binary classifier, are defined by reading on the columns the assignment of 0 s and 1 s. More precisely, on the first column, for the first classifier ($i = 1$), we consider that metaclass 1_1 contains class 1 and class 4, whereas metaclass 0_1 contains class 2 and class 3, as shown in Fig. 2. In the same manner, the remaining metaclasses are defined for the rest of the binary classifiers.

B. Establishing the Encoding Scheme

Code definition represents the starting point when building a multilabel classification scheme based on ECOC. There is a multitude of binary codes that can be formed, but a multilabel

TABLE II
EXAMPLE OF DESIGNED CODE

Class	Codeword				
	Man-made structures	Repetitive textured surfaces	Green predominant	Increased land-use variability	Urban only
high-density population	1	0	0	1	1
forest	0	1	1	0	0
water	0	1	0	0	0

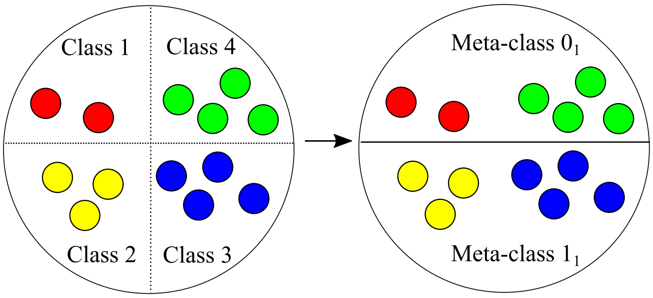


Fig. 2. Transformation of classes into metaclasses following the example shown in Table I for the first bit in the codewords.

classification task imposes some restrictions. First, the codewords are required to be of same length (n bits). This requirement comes natural because each instance is passed through n binary classifiers, each providing a binary output. The n-bits vector representation is then translated to a class index/label. Second, in order to be able to decode the vector of metalabels into the initial class index/labels, each class index/label has to be uniquely mapped into a codeword in the encoding table.

Taking into account these restrictions, three types of codes, along with their encoding and decoding schemes, are discussed in the following part of the section, namely, designed codes, random codes, and error-correcting codes.

C. Designed Codes

One possibility to assign codewords to each class is to design a distributed code where each bit of the codeword represents a characteristic of the class. In [8], the handwritten digit recognition is performed by assigning codewords whose bits express particular geometrical characteristics of each digit (e.g., contains vertical lines, horizontal lines, diagonal lines, closed curves, and so on). This approach cannot be directly applied to the classification of remote sensing images, which are characterized by complex structural components [20]. In addition, the increased spatial resolution of current sensors implies the existence of an intractable number of possible geometrical shapes inside the compound structures. Therefore, in the case of remote sensing classification, a viable solution is to include characteristics that focus on the texture and color of the images. An example is shown in Table II, where 1 represents the existence and 0 the nonexistence of the designated characteristic. The 0 and 1 bits on each column of the encoding table can be regarded as positive and negative answers to questions regarding the properties of each class. The high-density population class is specific for large cities where there are many man-made structures and the land-use varies (e.g., there are parks, buildings, residential area, industrial area, and shops). In contrary, the classes forest and water are characterized by repetitive textured surfaces. Therefore, in the case of designed codes, each metaclass indicates the presence or absence of a common property across a subset formed by original classes.

1) *Encoding:* The encoding step means assigning to each class index a codeword from the defined code.
2) *Decoding:* The decoding can be performed using minimum distance decoding, which computes the Hamming distances between the binary vector representing a new instance and all the existing codewords. We recall that the Hamming distance between two binary vectors, $\mathbf{a} = [a_0, a_1, \ldots, a_{n-1}]$ and $\mathbf{b} = [b_0, b_1, \ldots, b_{n-1}]$, is given by

$$d_H(\mathbf{a}, \mathbf{b}) = \sum_{i=0}^{n-1} a_i \oplus b_i \tag{1}$$

where $\oplus$ is addition modulo 2.

D. Random Codes

Random codes are constructed by choosing each bit of the codewords uniformly at random from $\{0, 1\}$. Once the code assignation is established, the encoding and decoding steps are performed as in the case of using designed codes. However, as specified above, the generated code has to fulfill a constraint, namely, to assign unique codewords to each class index. A solution to increase the probability of having different encodings for different class indexes is to use longer codewords, which reduces the probability of generating two identical strings of bits, but increases the number of binary classifiers that have to be trained.

E. Error-Correcting Codes—A Coding Theory Perspective

Multilabel classification can be regarded as a communication theory problem in which the index of a class is transmitted over a "noisy channel." This approach follows the Shannon–Weaver model of communication presented in the Shannon's famous paper [21], which considers that the communication channel between a source and a receiver is affected by noise. In order to diminish the effect of noise, error-correcting codes are built and their role is to allow the transmission of information in a reliable manner.

In the multilabel classification, the classification errors have multiple causes. For example, because of particular information extraction algorithms, the information may be distorted and the receiver gets the class information corrupted. In addition, the classification model may fit very well details or noise in the training data, but the model may fail to classify new data (i.e., overfitting). Another possible source of errors may be the poorly selected training data, which has a negative impact on the performance of the learned model.

The mission of the error correcting code is therefore to correct corrupted labels. Ideally, at the end-user, we would like to have a perfect decoder that corrects the errors occurring during classification and decodes exactly the message sent by the source image. In practice, not all types of error configurations can be corrected. The number of errors that a decoder can correct depends on the code capabilities.

The majority of the existing error-control systems rely on block coding that introduce controlled amounts of redundancy into a transmitted sequence, providing the decoder the ability to detect and possibly correct a limited number of errors. The source delivers a k-bits length message and the encoder transforms the message into a *codeword* $\mathbf{c}$ of $n > k$ bits by appending control bits to the original message.

Linear block codes are widely used in practice for several reasons. First, the encoding and decoding procedures are facilitated by the linearity property of the codes. Second, the processing time is smaller than in the case of other codes, e.g., convolutional codes. Different types of linear codes can be used to design ECOC-based multilabel classifiers, e.g., linear cyclic codes and Bose–Chaudhuri–Hocquenghem (BCH) codes. These two codes have special properties. Linear cyclic codes are characterized by the fact that any permutation of a codeword is also a codeword, whereas BCH are powerful error-control codes used in many communication-related applications [22]–[25].

1) Encoding: A block error control code $\mathcal{C}$ consists of a set of M codewords $\{C_0, C_1, \dots, C_{M-1}\}$ of length n, each representing a distinct message that can be transmitted by the source. The codewords $\mathbf{c} = [c_0, c_1, \dots, c_{n-1}] \in \mathcal{C}$ are transmitted over the noisy channel. The decoder receives the distorted codeword and tries to reconstruct the message in a reliable manner.

The ability of a code to detect and correct errors is strictly linked to the concept of minimum Hamming distance, computed between all distinct pairs of codewords in $\mathcal{C}$. A code with minimum distance $d_{\min}$ can detect $d_{\min} - 1$ errors and can correct up to $\lceil (d_{\min} - 1)/2 \rceil$ errors, where $\lceil x \rceil$ is the upper bound integer of x [23].

Linear block codes are error-correction codes with special mathematical properties (i.e., the linearity of the codes). The codeword of n bits contains the k bits of the message (called *information bits*) and $n - k$ *control bits*, where each control bit is a linear combination of the information bits. In this paper, we consider that the k bits of information come from the binary representation of the class index using the most significant bit first rule. For example, the class index 2 represented on four bits is 0010.

Each linear code has a corresponding parity check matrix $\mathbf{H}$ with the following property [22]

$$\mathbf{H}\mathbf{c}^T = 0 \tag{2}$$

for all codewords $\mathbf{c}$ in code $\mathcal{C}$. The parity check matrix $\mathbf{H}$ is fixed and characterizes each code in part. Relation (2) provides the encoding rules, namely, the relations between the control bits and the information bits. Being linear codes and taking into account that all the elements (including the elements of the parity check matrix) are 0 or 1, all the computations involve linear operations, more precisely, additions modulo 2.

Linear cyclic codes are special linear codes that can be defined using a generator polynomial $g(X)$, which is chosen from the set of divisors of polynomial $X^n - 1$. We say that the cyclic codes are ideals in the algebra of polynomials modulo $X^n - 1$. One of the properties of the generator polynomials is that it divides the polynomial associated to each codeword

$$g(X) | c(X) \tag{3}$$

where $c(X) = c_0 + c_1 X + \dots + c_{n-1} X^{n-1}$ is the polynomial associated to codeword $\mathbf{c} \in \mathcal{C}$. Due to this property, in the case of cyclic codes, if $c(X)$ is a codeword, then $X^q c(X)$, with $q \geq 0$ is also a codeword. From this, any right circular shift of a codeword is still a codeword.

An equally important polynomial is the parity check polynomial, denoted by $h(X)$

$$h(X) = \frac{X^n - 1}{g(X)} \tag{4}$$

which gives the corresponding parity check matrix

$$\mathbf{H} = \begin{bmatrix} 0 & 0 & \dots & 0 & h_k & h_{k-1} & \dots & h_1 & h_0 \\ 0 & 0 & \dots & h_k & h_{k-1} & h_{k-2} & \dots & h_0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ h_k & h_{k-1} & \dots & h_1 & h_0 & 0 & \dots & 0 & 0 \end{bmatrix}. \tag{5}$$

The degree of the generator polynomial $m = \text{degree}(g(X))$ gives the number of control bits, whereas the degree of the parity check polynomial $k = \text{degree}(h(X))$ represents the number of information bits.

2) Syndrome Decoding: In the case of error-correcting linear codes, syndrome decoding is considered to achieve good performance in terms of computational time and correction properties of the code [22]–[25].

Assuming that codeword $\mathbf{c} \in \mathcal{C}$ encodes a certain class label and that $\mathbf{e} = [e_0, e_1, \dots, e_{n-1}]$ is the error vector (i.e., $e_i = 1$ if an error occurs on the ith position and $e_i = 0$, otherwise), the vector of binary classifiers' outputs $\mathbf{r}$ is given by

$$\mathbf{r} = \mathbf{c} + \mathbf{e}. \tag{6}$$

A syndrome vector can be computed for each possible vector $\mathbf{r}$

$$S(\mathbf{r}) = \mathbf{H}\mathbf{r}^T. \tag{7}$$

Using (2) and (6), yields that the syndrome corresponding to $\mathbf{r}$ is equal to the syndrome corresponding to the error vector $\mathbf{e}$

$$S(\mathbf{r}) = \mathbf{H}\mathbf{e}^T = S(\mathbf{e}). \tag{8}$$

In order to perform error correction, there must be a one-to-one correspondence between each configuration of errors (i.e., errors made by one or more classifiers) and the syndromes. Therefore, a lookup table of all the possible pairs of form $(\mathbf{e}, S(\mathbf{e}))$ can be computed and stored. When a vector $\mathbf{r}$ is received, the corresponding syndrome is computed. The error vector can be retrieved by looking at the corresponding line in the lookup table. This tells the error-correcting system where the classification errors occurred and allows the correction of the received vector. Finally, if the syndrome equals 0, then the decoder decides that no error occurred and maps the received codeword to the index of the corresponding class.

Using the syndrome decoding algorithm saves storage and computational time. The lookup table contains only error-syndrome pairs, resulting in a fast correction of errors and decoding. For these reasons, in this paper, this decoding strategy is used to transform the vector of binary classifiers' outputs $\mathbf{r}$ back into codewords, and, then, into corresponding class indexes.

IV. Binary Classifiers in the ECOC-Based Framework

The strategy based on error-correcting codes combines the outputs of several binary classifiers to produce a label that represents a semantic class. The role of each binary classifier is to decide upon the membership of an instance to a particular metaclass. In what follows, we present two ways of building binary classifiers. The first option consists in training the well-known SVM binary classifiers, whereas the second one aims at improving the SVM binary classifiers by determining the most representative examples, for each pair of metalabels, which lead to an optimal hyperplane with respect to a given criterion.

A. SVM-Based Classification

SVM classifiers are originally designed for binary classification [26]. The SVM problem can be stated as follows: Given a set of labeled data $\mathcal{T} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_l, y_l)\}$, where $\mathbf{x}_j$ is the data sample and $y_j \in \{-1, 1\}$ is its corresponding label, the optimal hyperplane P is determined by optimizing the following expression:

$$\max_{\mathbf{w},b,\epsilon} \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{j=1}^{l} \epsilon_j$$
$$\text{subject to } y_j(\mathbf{w}^T\mathbf{x}_j + b) \geq 1 - \epsilon_j, \text{ with } \epsilon_j \geq 0. \tag{9}$$

In the above expression, the hyperplane P is defined as

$$\mathbf{w}^T\mathbf{x} + b = 0 \tag{10}$$

where $\mathbf{x}$ is a sample point in the feature space and $C > 0$ is a regularization parameter. Please note that, above, label $y_j = -1$ corresponds to metalabel 0_i, whereas label $y_j = 1$ corresponds to meta-label 1_i for the ith binary classifier.

B. Training SVM-MA With Most Ambiguous Examples

Training accurate binary classifiers requires an important number of annotated examples. We aim to build robust binary classifiers with a small training set. In order to achieve this goal, the proposed learning procedure involves the evaluation of the informativeness of the examples. At first iteration, we start training with a small annotated training set $\mathcal{T}_0$, selected uniformly at random from the available dataset. In addition, we consider having access to another small set $\mathcal{L}$ of randomly chosen examples. We perform several iterations and, at each iteration n_i, we augment the training dataset with the most informative examples from $\mathcal{L}$, which were not included in the previous training set, along with their corresponding metalabels.

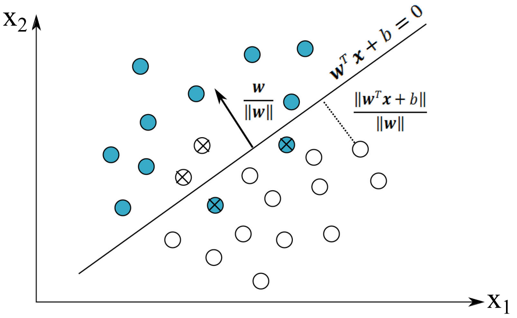

x_2
x_1
$\mathbf{w}^T\mathbf{x} + b = 0$
$\frac{w}{\|w\|}$
$\frac{\|w^T x + b\|}{\|w\|}$

Fig. 3. Two-class SVM linear classification. The ambiguous examples are placed close to the boundary line because they are the ones for which the classifier's uncertainty is highest (i.e., crossed points in the figure are classified incorrectly).

The majority of the errors produced by an SVM classifier appear when the examples are close to the separation hyperplane, as shown in Fig. 3. If these ambiguous examples receive correct metalabels and a new train set is formed by adding them to the previous train set, an optimal separation hyperplane is better identified and the classification error is diminished [27]. Taking advantage of the geometrical characteristics of linear SVMs, the selection of the closest examples with respect to the separation hyperplane reduces to simple computations of distances from a point in space, $\mathbf{x}$, to a hyperplane P

$$d(\mathbf{x}, P) = \frac{|\mathbf{w}^T\mathbf{x} + b|}{\|\mathbf{w}\|}. \tag{11}$$

The distances with respect to the hyperplane are rearranged in ascending order with the scope of prioritizing the most ambiguous examples which most likely received wrong labels. This selection criteria was initially introduced in [14] and [15] for text classification or retrieval of top-k most relevant images. In this paper, this selection criteria is applied in an iterative manner. At each iteration, new examples that were wrongly classified using the hyperplane determined at previous iteration, receive correct metalabels and they are added to the train set. Next, a new SVM separation hyperplane is determined based on the newly formed training dataset. The algorithm is detailed in Fig. 4.

The final decision regarding the membership of an instance $\mathbf{x}$ from a metaclass 0_i or 1_i is taken using the SVM model parameters determined before, namely

$$y = \text{sign}(\mathbf{w}^T\mathbf{x} + b) = \begin{cases} 1 & \text{if } \mathbf{w}^T\mathbf{x} + b > 0 \\ -1 & \text{otherwise} \end{cases} \tag{12}$$

and mapping the result to a bit following the rule: $-1 \rightarrow 0$, $1 \rightarrow 1$.

V. Feature Extraction

The ECOC-based classification framework is tested on two category of features: 1) BOVW model-based descriptors (BOVWC features) combined with color statistics; and 2) high-level image representations derived from pretrained deep

```
Input: Initial training set $\mathcal{T}_0$, set $\mathcal{L}$, number of iterations
    $N_{iter}$, number of corrected meta-labels per iteration $N_{corr}$
Output: SVM model parameters ($\mathbf{w}$, $b$)
Train initial SVM classifier on set $\mathcal{T} = \mathcal{T}_0$
for $n_i = 0$ to $N_{iter} - 1$ do
    for each instance $\mathbf{x} \in \mathcal{L}$ do
        Perform classification for instance $\mathbf{x}$
        Compute the distance from $\mathbf{x}$ to the SVM separation
            hyperplane
    end for
    Sort distances in ascending order
    Discover first $N_{corr}$ meta-labels that are wrong and
        correct them
    Add the corrected meta-labels and corresponding data
        examples to the training set $\mathcal{T}$
    Remove same data examples from $\mathcal{L}$
    Update SVM model parameters ($\mathbf{w}$, $b$)
end for
```

Fig. 4. Proposed training algorithm of binary classifiers with most ambiguous examples.

neural networks. We start by briefly presenting BOVWC features and, then, discuss over the advantages of using deep CNNs such as ResNet for feature extraction.

A. Representations Based on BOVW Models

In this setup, feature extraction consists in fusing color statistics with high-level BOVW model-based representations of the patches. This representation is denoted as *BOVWC*. The BOVW approach implies building a dictionary of D visual words, which are, in general, determined by applying unsupervised clustering methods (e.g., K-means) on low-level features extracted from the training set [17]. The low-level feature vectors are extracted using a dense variant of the SIFT introduced by Lowe in [18]. This makes the approach slightly different than the original SIFT extraction method, which first determines particular keypoints in the image [18]. A SIFT descriptor is essentially a histogram over surrounding subregions and orientations of image gradients that characterize a particular location.

Assuming that the dictionary is already determined, each SIFT descriptor is mapped to the closest visual word in l_2-norm. Afterward, the high-level features are just histograms registering the occurrences of visual words contained by the dictionary. The described feature extraction method is robust in the sense that is invariant with respect to scale, orientation, changes in illumination and modifications of objects' positions inside patches [17], [18].

B. Pretrained Deep Neural Networks for Feature Extraction

In this paper, we propose the usage of a pretrained deep neural network architecture, for generating a set of high-level representations with a fixed architecture, same set of weights, and no class-dependency across all datasets. This approach avoids a major drawback that BOVW model-based representations encounter, namely, the need to determine a dictionary of visual words that are specific to a particular dataset. A similar approach was investigated in [28], where an Overfeat model, derived for the ImageNet classification task [29], is used to generate image representations, which are then inserted in a trainable custom CNN to produce the desired semantic labels. However, learning the custom CNN architecture requires a considerable amount of training data. In this paper, we avoid this problem by using an iterative learning procedure, with limited amount of annotated data, to train the ECOC-based multilabel classifier.

The deep neural network architecture chosen for the feature extractor is ResNet50, an architecture that achieved state-of-the-art performance in image classification, object detection, and semantic segmentation tasks [7]. The original ResNet50 architecture contains, apart from two pooling layers, 49 convolutional layers and one fully connected layer, that amount to 25.6 million of parameters. This architecture is trained on the ImageNet challenge dataset [19] consisting of 1.2 million HR training images and 1000 corresponding labels designed for visual object recognition. The main advantages of using a pretrained architecture are as follows: 1) the training of the deep neural network is performed only once (i.e., using an open and large annotated dataset); 2) once the parameters are determined, the architecture is used as a general purpose feature extractor for any classification task; and 3) the feature extraction technique is data independent. In order to build a feature extractor from the original ResNet architecture, we exclude the last fully connected layer, which was performing the actual classification task in the ImageNet challenge. We call the resulting features, ResNet features.

VI. Experiments

A. Datasets

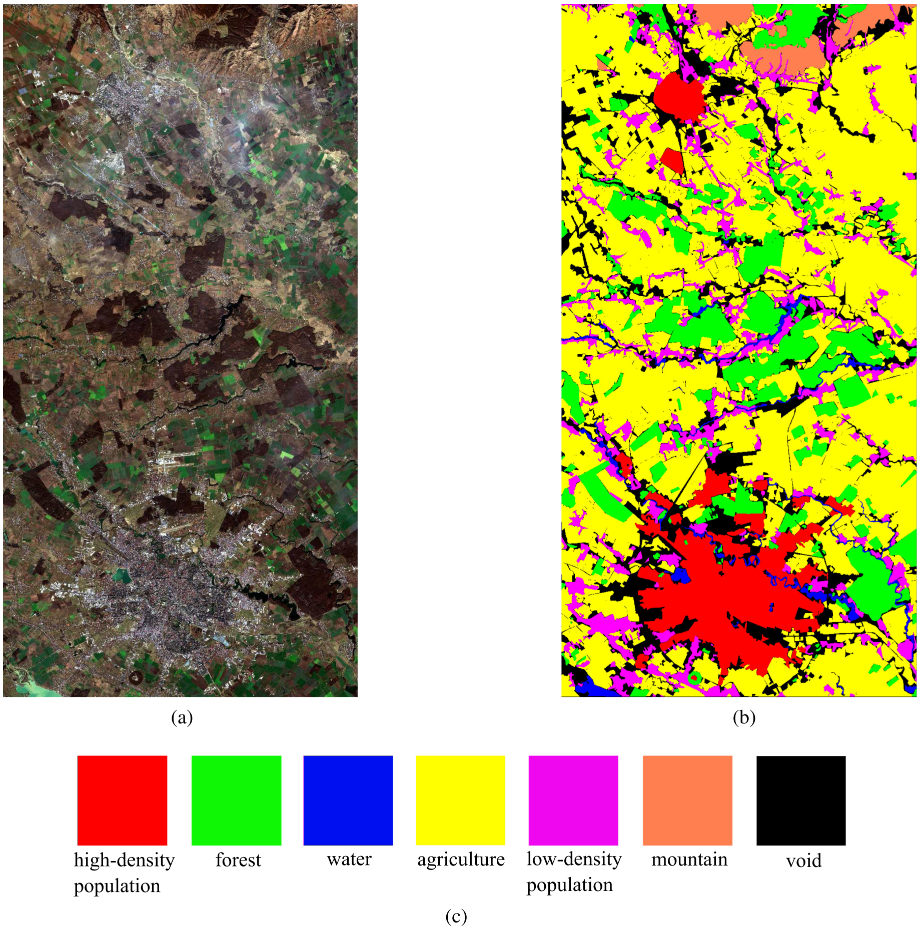
We tested the proposed algorithms on two scenes with multiple complex classes. The first one is a Sentinel-2 MSI image (10-m spatial resolution) acquired over Bucharest, Romania, and another smaller city, on December 23th, 2015. The second one is a GeoEye-1 image (1.65-m spatial resolution) acquired over Hobart, Tasmania, Australia, on February 5th, 2009. The first scene contains 3999×7802 pixels, whereas the second one 3759×3188 pixels. The number of spectral bands considered is four in both cases (red, green, blue, and near-infrared).

The reference ground truth for the Sentinel-2 scene contains six semantic classes, whereas the ground truth for the GeoEye-1 scene contains nine classes, as it can be observed in Figs. 5 and 6.

B. Experimental Results

The scenes are divided into patches of $p \times p$ pixels. Considering the spatial resolutions of the two scenes and an appropriate spatial coverage for the classes of interest, the chosen patch size is 30×30 pixels for Sentinel-2 and 40×40 pixels for GeoEye-1, which corresponds to a spatial coverage of 300×300 m^2 and 66×66 m^2, respectively.

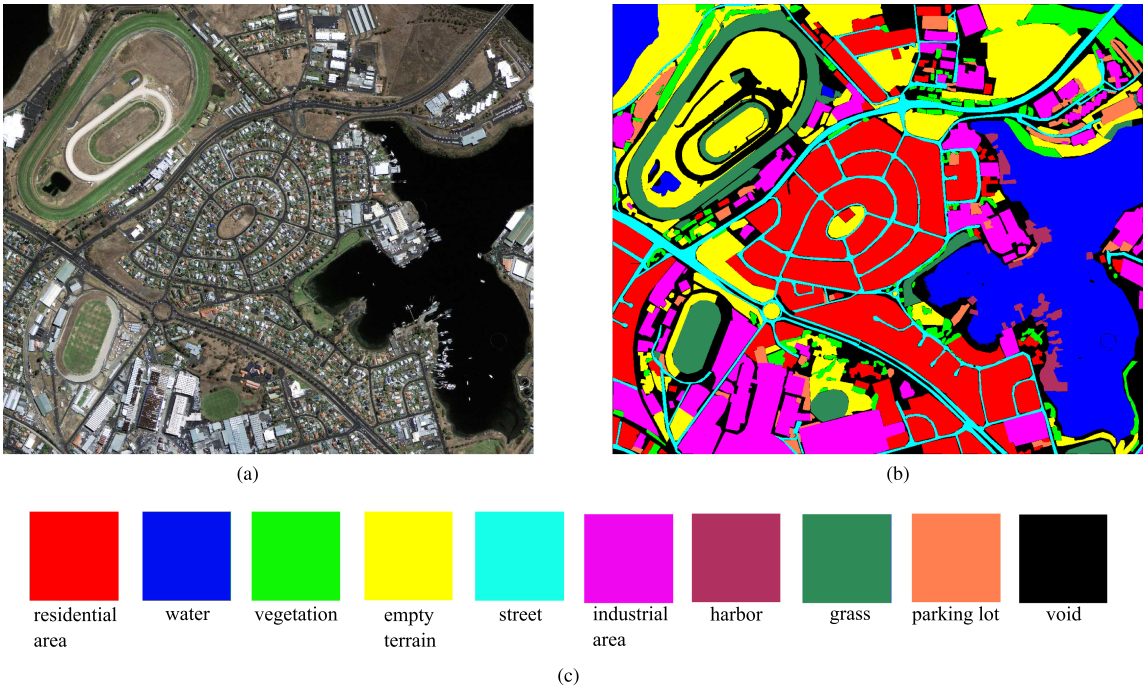
Two types of features are extracted for each image patch. First, we consider BOVWC features comprised of statistical descriptors (e.g., the mean and dispersion values per each band



(a) (b)

(c)

Fig. 5. Sentinel-2 scene and its corresponding ground truth. (a) Image. (b) Ground truth. (c) Legend.



(a) (b)

(c)

Fig. 6. GeoEye-1 and its corresponding ground truth. (a) Image. (b) Ground truth. (c) Legend.

TABLE III
DESIGNED CODE FOR SENTINEL-2 IMAGE

Class	Codeword				
	M	R	G	V	U
high-density population	1	0	0	1	1
forest	0	1	1	0	0
water	0	1	0	0	0
agriculture	0	0	1	0	0
low-density population	1	0	0	1	0
mountain	0	0	0	1	0

TABLE IV
DESIGNED CODE FOR GEOEYE-1 IMAGE

Class	Codeword					
	M	R	G	V	H	B
residential area	1	0	0	1	1	0
water	0	1	0	0	0	1
vegetation	0	0	1	0	0	0
empty terrain	0	1	0	0	0	0
street	0	0	0	0	0	0
industrial area	1	0	0	0	0	0
harbor	1	0	0	1	0	1
grass	0	1	1	0	0	0
parking lot	1	0	0	1	0	0

TABLE V
LEGEND FOR DESIGNED CODE

Abbreviation	Meaning
M	man-made structures
R	repetitive textured surfaces
G	green predominant
V	increased land-use variability
U	urban only
H	large number of houses
B	blue predominant

of the multispectral domain) and BOVW model-based representations. The BOVW features are histograms over a learned dictionary of D visual words. The SIFT low-level descriptors are extracted over a dense grid of locations and each location is characterized by 4×4 neighboring subregions and 8 orientation levels, yielding 128-dimensional feature vectors per each pixel. In the second case, the features are extracted using a pretrained ResNet50 deep architecture, from which we removed the last fully connected layer. The length of the ResNet feature vectors is 2048. The following experiments are performed for both types of features described above.

1) Code Selection: The experiments were performed using all three types of codes detailed in Section III-B, namely, designed, random, and linear cyclic codes.

The encoding tables used for designed codes are shown in Table III for the Sentinel-2 scene and in Table IV for the GeoEye-1 scene, whereas the corresponding legend of the considered properties is presented in Table V. The properties of the images that are taken into account for code construction are correlated to the spatial resolution of the images and to the capability of the codewords to provide a discriminative behavior across classes. We report, in the case of BOVWC features, an overall accuracy level of 90.88% for the Sentinel-2 scene and 78.21% for the GeoEye-1 scene, respectively, whereas, in the case of ResNet features, an overall accuracy level of 93.45% for the Sentinel-2 scene and 90.53% for the GeoEye-1 scene, respectively. Although, a smaller number of binary classifiers are used for the designed-based encoding, the obvious disadvantage of using this type of encoding tables is the subjectivity in determining the shared properties across different land-cover classes.

In the case of random codes, the probability of repeating the same codeword in a code should be as small as possible. For this reason, using a small codeword length is not useful. However, using too many bits to represent each semantic class would yield a large number of binary classifiers to be trained. We vary the codeword lengths between 5 and 21 bits and measured the overall accuracy in each case. For these codes, we report best overall accuracies of ±1% around the accuracies obtained for designed codes.

In the case of error-correcting codes, the six semantic labels of the Sentinel-2 scene can be represented on $k = 3$ bits of information, whereas the nine semantic labels of the GeoEye-1 scene is represented on $k = 4$ bits of information. A mention is to be made here: no special rule was followed when assigning the class indexes (e.g., 1, 2, 3, and so on) to the class semantic labels (e.g., forest, water, agriculture and so on). Following the results that we obtained in [30], we consider using only linear cyclic codes because of their superior performance and less restrictions imposed to the generator polynomials if compared to BCH codes. As in the case of random codes, we vary the codeword length between 5 and 21 bits and, for each case, we measure the performance level.

The performance results measured for different codeword lengths n are shown in Fig. 7. In the case of random codes, the overall accuracy value increases with n, but it oscillates around the maximum value after reaching it. It is interesting to observe that the maximum overall accuracy is achieved quite fast. The missing bars on the graph correspond to the cyclic codes (n, k) for which no generator polynomial $g(X)$ of degree $m = n - k$ can be formed from the divisors of $X^n - 1$. In the case of linear cyclic codes, the best choices for codeword lengths are $n = 7$ bits for the Sentinel-2 scene and $n = 15$ bits for the GeoEye-1 scene. The first code (7, 3) is able to correct 1 bit-error, whereas the second one (15, 4) can correct configurations of up to 4 bit-errors. In the case of GeoEye-1 scene, we observe also a peak of the overall accuracy for the well-known Hamming code (7, 4), which also corrects 1 bit-error. Another remark is related to the fact that one needs more bits to check possible errors when the number of information bits representing the semantic labels is larger. It is worth mentioning that no constraints were imposed when selecting the generator polynomial for a given (n, k) pair. In fact, multiple experiments carried in the same conditions revealed that similar results are obtained for different generator polynomials and fixed (n, k) pair. Compared to random codes, the linear cyclic codes achieve a greater performance with a smaller number of binary classifiers and are more stable. In what follows, we use the linear cyclic codes (7, 3) for the Sentinel-2 scene and $(15, 4)$ for the GeoEye-1 scene.

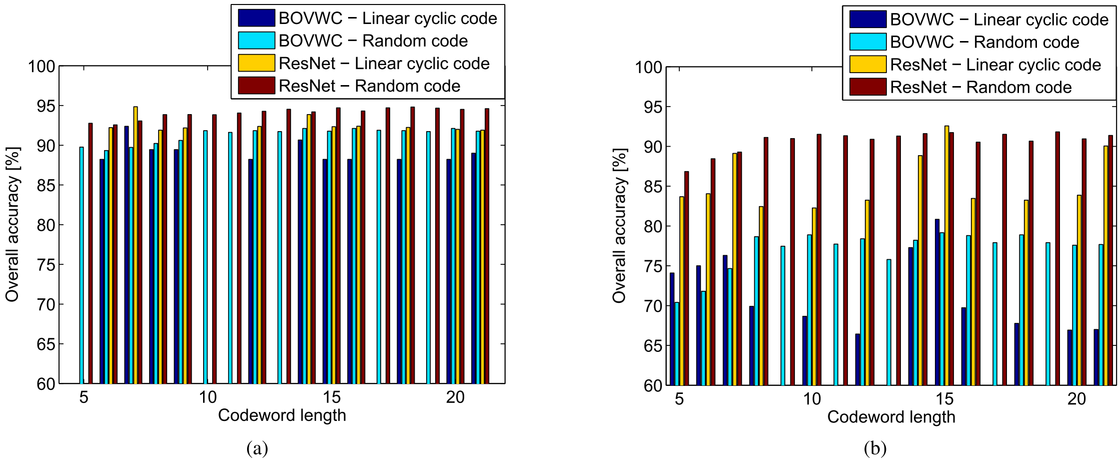


Fig. 7. Choice of the codeword length n for random and linear cyclic codes. Two categories of features are considered, namely BOVWC and ResNet features. (a) Sentinel-2. (b) GeoEye-1.

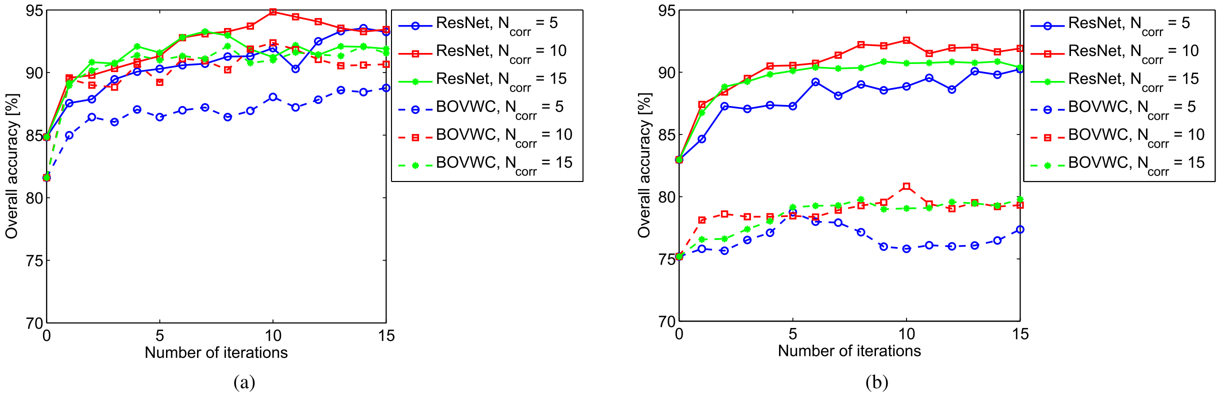


Fig. 8. Variation of the classification accuracy with the number of iterations, N_{iter}, for different number of corrected meta-labels per iteration, N_{corr} and for both categories of features considered (i.e., BOVWC and ResNet features). (a) Sentinel-2. (b) GeoEye-1.

2) Iterative Training of SVM-MA: In this paper, we use SVM as binary classifiers to discriminate between metalabels and propose an iterative training procedure to learn the corresponding parameters. We consider that the initial training set is 5% of each dataset, with examples that are randomly selected from all classes. In addition, 20% of each dataset (randomly selected) is used for the iterative learning procedure, whereas the rest is used solely for testing. The initial training set is increased at each iteration with N_{corr} most informative examples, whose predicted metalabels need to be corrected. In this experiment, we vary the number of iterations N_{iter} between 0 (i.e., corresponds to training an SVM classifier with a limited amount of data and no iterations) and 15, and consider the number of corrected meta-labels $N_{\text{corr}} \in \{5, 10, 15\}$. As shown in Fig. 8, the accuracy increases with the number of iterations and rapidly attains a maximum value for $N_{\text{iter}} = 10$ and $N_{\text{corr}} = 10$. We observe, in the case of BOVWC features, an increase in accuracy of 7% (GeoEye-1) to 11% (Sentinel-2), whereas, in the case of ResNet features, an increase of 10% is achieved in a small number of iterations.

3) Learning SVM Binary Classifiers With a Large Training Set: In this experiment, we use a large training set (25% of the data), the same for all n binary classifiers, and no iterations. From our experiments, the SVM binary classifiers trained with a larger training set do not perform as well as SVM-MA ones. In some cases, they might even fail in discriminating well between different classes. For example, in the case of GeoEye-1 with BOVWC features, the accuracy is almost 75%. In contrary, when using SVM-MA, the position of the hyperplanes is gradually modified based on the most informative examples with respect to the distance criterion. Moreover, selecting most ambiguous examples leads to customly defining training sets for each binary classifier in part. This yields a better discrimination across different meta-labels. In addition, the time spent to train the binary SVM classifiers with a large training set is considerable, e.g., in the case of Sentinel-2 scene, the same number of classifiers of ECOC are trained in 96 s with SVM-MA technique (ten iterations), whereas, it takes almost 10 min to train them with a larger training set and no iterations.

4) Feature Extraction: The feature extraction is an important step in any classification framework. In this experiment, we tested four types of features: 1) color descriptors (first- and second-order statistics per patch); 2) BOVW; 3) color

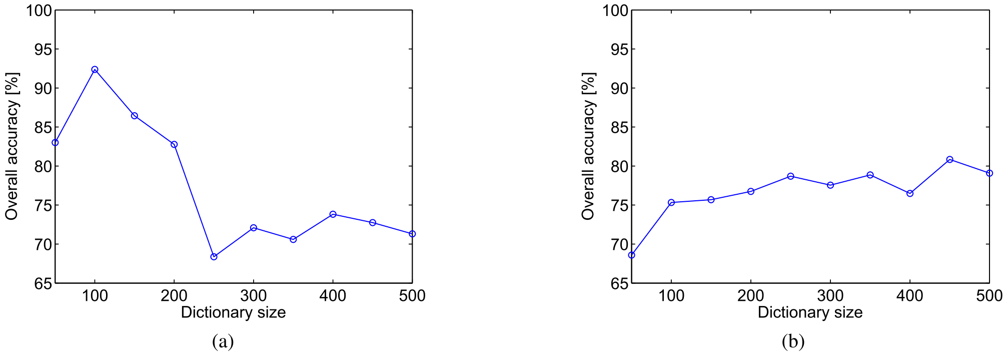


Fig. 9. Choice of the dictionary size for the BOVW model of representation. (a) Sentinel-2. (b) GeoEye-1.

TABLE VI
FEATURE CHOICE

Feature	Overall accuracy [%]	
	Sentinel-2	GeoEye-1
Color statistics	80.93	73.77
BOVW	82.16	79.99
BOVW & color statistics (BOVWC)	92.38	80.84
ResNet features	**94.85**	**92.57**

descriptors and BOVW (BOVWC); and 4) features extracted using a pretrained ResNet architecture.

Although the water, grass, forest, and empty terrain are easy to be identified through their color, not the same happens when we want to distinguish man-made structures. For this reason, complementing the color information with BOVW representations of local descriptors proves beneficial in terms of recognizing land-cover classes. This aspect can be observed from the results shown in Table VI. In order to determine the best BOVW model, we performed experiments for various dictionary sizes $D \in \{50, 100, 150, \ldots, 500\}$. As shown in Fig. 9, Sentinel-2 patches can be represented with a smaller number of visual words ($D = 100$) compared to the GeoEye-1 scene ($D = 450$). This is mainly due to the greater level of detail of the GeoEye-1 scene if compared to the Sentinel-2 scene (i.e., the spatial resolution of GeoEye-1 is 1.65 m, whereas the spatial resolution for Sentinel-2 is 10 m).

As it can be depicted in Table VI, the performances achieved when using deep CNNs (in our case ResNet features) for feature extraction are better than the ones obtained in the case of BOVWC representations. In the case of the GeoEye-1 scene, the increase in accuracy is important, almost 12%, achieving over 92% overall accuracy for the ResNet-based features. In the case of the Sentinel-2 scene, the accuracy is just a bit higher than for BOVWC (i.e., only with 2.47%). This difference is explained by the higher spatial resolution of the GeoEye-1 sensor and by the fact that the original ResNet architecture was pretrained on the ImageNet dataset and can be used to detect even small objects.

5) Other Performance Measures and Comparisons With Other Methods: For best cases in terms of overall classification accuracy (OA), the classification performance was also evaluated in terms of per-class accuracies (PC) and Kappa index ($\mathcal{K}$). These measures are computed starting from the confusion matrix C, which has the number of predicted labels on the rows and the ground truth on the columns

$$\mathrm{OA} = \sum_{i=1}^{\mathrm{nc}} \frac{C_{ii}}{N} \tag{13}$$

$$\mathrm{PC}_j = \frac{C_{jj}}{C_{+j}} \quad \forall j \in \{1, \ldots, nc\} \tag{14}$$

$$\mathcal{K} = \frac{\frac{1}{N}\sum_{i=1}^{\mathrm{nc}} C_{ii} - \frac{1}{N^2}\sum_{i=1}^{\mathrm{nc}} C_{i+}C_{+i}}{1 - \frac{1}{N^2}\sum_{i=1}^{\mathrm{nc}} C_{i+}C_{+i}} \tag{15}$$

where N represents the total number of classified instances, nc is the number of classes, C_{ij} is the number of instances in ground truth class j and classified as class i, and the values C_{i+} and C_{+j} are computed as

$$C_{i+} = \sum_{j=1}^{\mathrm{nc}} C_{ij} \tag{16}$$

$$C_{+j} = \sum_{i=1}^{\mathrm{nc}} C_{ij}. \tag{17}$$

The corresponding results are shown in Tables VII and VIII.

Looking at the per-class accuracies, the best performance is obtained for the agriculture, forest, and high-density population classes in the case of Sentinel-2 scene and for the residential area, water, and empty terrain classes in the case of GeoEye-1 scene. These results are somehow expected taking into account the spatial coverage of these classes in the ground survey and the fact that the BOVWC and ResNet representations describe precisely the local appearance of each patch, which is beneficial for classes characterized by a certain structural pattern. Moreover, compared to BOVWC, ResNet features prove to be more discriminant in the case of low-density population class (Sentinel-2 scene) and industrial area, grass, empty terrain, and vegetation classes (GeoEye-1 scene).

In the same tables, we compare the proposed ECOC-based classification procedure with other methods, i.e., OAO-based SVM multiclass (SVM OAO), semisupervised classification with active queries (SSC AQ) [3], and original ResNet classifier (i.e., with the last fully connected layer included to perform the actual classification) [7] trained directly on the GeoEye-1

TABLE VII
Per-Class Accuracy Rates (PC), Overall Accuracy (OA), and Kappa Index ($\mathcal{K}$) of Classifications on Sentinel-2 Scene

Class	BOVWC features					ResNet features					Original ResNet classifier
	ECOC-SVM-MA			SVM OAO	SSC AQ	ECOC-SVM-MA			SVM OAO	SSC AQ	
	Designed	Random	Linear cyclic			Designed	Random	Linear cyclic			
high-density population	85.67	76.08	90.27	62.13	45.81	87.98	88.39	88.39	72.10	92.48	87.81
forest	92.64	91.52	92.07	69.10	25.95	94.09	95.95	94.45	80.21	85.92	72.86
water	14.76	29.61	10.68	16.18	0	40.58	47.10	36.43	0	17.57	49.42
agriculture	93.92	97.55	98.07	84.02	88.16	97.76	96.71	98.44	96.60	95.85	96.76
low-density population	62.63	63.64	37.37	45.55	25.70	84.83	79.68	80.11	61.21	57.37	32.92
mountain	12.63	6.42	9.87	11.99	0	48.67	56.89	61.69	15.66	5.00	11.92
OA	90.88	91.89	92.38	75.10	67.84	93.45	93.06	**94.85**	84.37	88.40	85.16
$\mathcal{K}$	0.70	0.72	0.74	0.55	0.36	0.88	0.87	**0.89**	0.73	0.78	0.72

TABLE VIII
Per-Class Accuracy Rates (PC), Overall Accuracy (OA), and Kappa Index ($\mathcal{K}$) of Classifications on GeoEye-1 scene

Class	BOVWC features					ResNet features					Original ResNet classifier
	ECOC-SVM-MA			SVM OAO	SSC AQ	ECOC-SVM-MA			SVM OAO	SSC AQ	
	Designed	Random	Linear cyclic			Designed	Random	Linear cyclic			
residential area	87.08	92.19	88.16	87.53	99.73	93.78	93.64	95.69	93.00	93.09	85.06
water	98.28	96.55	98.86	94.19	85.07	99.62	99.24	99.81	97.99	89.94	99.69
vegetation	69.84	66.67	54.84	50.41	0	85.94	81.25	81.25	40.87	0	80.16
empty terrain	76.87	78.19	80.97	81.46	71.02	90.97	92.51	94.05	90.22	88.22	93.97
street	42.00	48.00	75.51	35.79	0	64.00	82.00	76.00	35.56	23.33	41.05
industrial area	59.75	53.14	66.24	59.70	8.33	83.75	82.50	83.39	60.07	65.10	64.14
harbor	11.76	10.00	33.33	0	0	41.18	5.10	47.06	0	0	0
grass	45.81	41.29	58.82	34.69	13.62	86.45	91.61	94.84	82.08	47.67	97.95
parking lot	48.65	13.51	57.89	10.00	0	45.95	40.54	40.54	12.12	16.66	42.85
OA	78.21	77.62	80.84	74.46	64.15	90.53	91.82	**92.57**	82.40	77.32	85.74
$\mathcal{K}$	0.72	0.71	0.75	0.70	0.52	0.88	0.88	**0.91**	0.78	0.71	0.82

TABLE IX
Number of Binary Classifiers Needed for Multiclass Annotation (Best Cases for ResNet Features and BOVWC Features in the Parenthesis)

Class	Number of classes	Type of code			SVM OAO
		Designed	Random	Linear cylic	
Sentinel-2	6 classes	5	19 (16)	7	21
GeoEye-1	9 classes	6	18 (15)	15	45

and Sentinel-2 data, respectively. The ECOC-based classification procedure proposed in this paper considers the most ambiguous examples to improve the performance achieved by the binary classifiers. A different semisupervised approach is SSC AQ [3]. The SSC AQ algorithm aims at building an hierarchical clustering tree and determining the most coherent examples to be included in the active queries with the goal of reducing the global estimated classification error. In order to train the SVM OAO and ResNet classifiers, we use 25% of the dataset as training sets. SSC AQ is trained with an initial set of labeled examples that amounts to 5% of the datasets (i.e., same amount as for initial train set of ECOC-SVM-MA) and 100 active queries. In the case of OAO-based SVM methods, we tested linear and radial basis function kernels, but the later ones reported a decrease in accuracy of up to 40%.

The first observation is related to the improvement brought by the ECOC-based classification framework over the SVM-OAO. This is explained by the fact that ECOC produces more balanced training sets (positive and negative) than in the case of SVM-OAO. The improvement is also maintained even when the binary SVM classifiers are not learned in an iterative manner—in this case, the overall accuracy is 81.60% (BOVWC)/84.85% (ResNet features) for Sentinel-2 and 75.19% (BOVWC)/82.97% (ResNet features) for GeoEye-1, which are higher than the results reported for SVM-OAO. Second, SSC AQ performs quite well for classes that appear most frequently, but fails when the number of examples in a class is small. Third, we observe that the ECOC-based classifier outperforms the ResNet classifier. This is somehow expected because training a deep neural network with millions of parameters requires a large set of annotated data (e.g., ImageNet contains 1.2 million HR training images). In fact, this is the main drawback that many deep neural networks encounter.

6) Final Remarks: An OAO-based SVM classifier needs t1o train $\mathrm{nc}(\mathrm{nc}-1)/2$ binary classifiers to discriminate between nc semantic classes. The proposed approach employs a smaller number of classifiers, which is determined by the length of the codewords that are used to encode the semantic labels. The numbers of classifiers are synthesized in Table IX.

As expected, a smaller number of binary classifiers is a major advantage for fast training and testing the ECOC-based classification framework. Once the ResNet or BOVWC features computed, we measured the computational time spent for training all the classifiers. In the case of the original ResNet classifier, we measured the time spent for learning the parameters through Stochastic Gradient Descent with backpropagation.

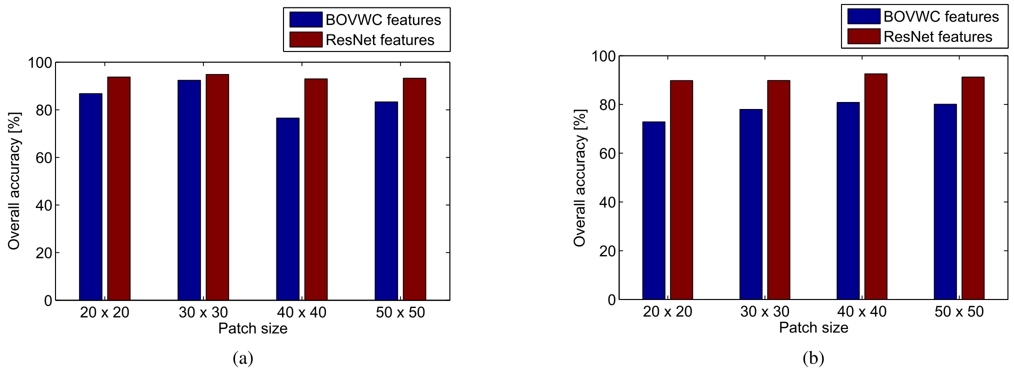


Fig. 10. Performance evaluation for different patch sizes. (a) Sentinel-2. (b) GeoEye-1.

TABLE X
AVERAGE COMPUTATIONAL TIME [s]

Classifier	Dataset	
	Sentinel-2	GeoEye-1
ECOC-SVM-MA	96	48
SVM-OAO	160	30
SSC AQ	207	23
ResNet classifier	2280	900

The average results are provided in Table X for Sentinel-2 and GeoEye-1 datasets. We mention that the computational times are measured on Intel Xeon E5-1680v3, 8 cores @3.2 GHz, equipped with NVIDIA QUADRO M4000 GPU with 8 GB RAM. As expected, in the ECOC-based framework, the computational time is considerably smaller than the time required for training the entire ResNet classification architecture.

We also performed classifications for other windows sizes $p \in \{10, 20, 30, 40\}$ and the results are shown in Fig. 10. These results confirm the proper selection of the patch size at the beginning of our experiments.

VII. CONCLUSION

In this paper, we proposed a multilabel classification framework based on block-coding schemes and learning over most ambiguous examples. The multilabel classification problem is decomposed into multiple binary subclassifications following a set of encoding rules established before the training. The classification framework represents a solution for the multi-label annotation of remote sensing images for which manually annotated training samples are hard to obtain. In this sense, the system is conceived to work in the context of a limited amount of annotated data.

The multilabel classification using ECOC encoding and decoding steps has a threefold benefit. First, the reorganization of the classes into metaclasses results into more balanced training sets and thus to a better training of the binary classifiers. Second, if the code has error-correction capabilities, part of the misclassifications caused by the binary classifier can be corrected. Third, the number of binary classifiers to be trained is smaller than in the SVM OAO case.

Experiments using different encoding schemes (i.e., linear cyclic, designed, and random codes) prove the generality of the proposed approach in the sense that any other encoding strategies can be plugged in the existing framework. Designed codes are tailored to represent several attributes of the analyzed classes, whereas random codes lead to an arbitrary organization of classes into metaclasses. One of the main challenges of the designed code approach is to define the discriminant characteristics of the analyzed classes such that each class is uniquely mapped into a codeword. In this context, the metaclasses for a designed code have to be defined by an expert before starting the training of the whole classification system. This makes designed codes rather difficult to use.

Furthermore, the choice of the binary classifiers can also be modified, but the advantage of using SVMs resides in the special geometrical properties that help us establish the examples that might bring the most useful information during training (i.e., SVM-MA method). This has a direct positive impact on the performance of the whole system. In this regard, the role of the iterations is to correct the behavior of binary classifiers. Even if the classification models might fail at the first round of iterations due to ill-chosen initial training set, the models are corrected in the next rounds. An interesting aspect is the fact that the number of corrected metalabels per iteration is rather small.

Finally, we tested our ECOC-based classification framework for two categories of high-level features, namely, features extracted with a pretrained deep CNN architecture (i.e., ResNet features) and BOVW features combined with color information. In the case of Sentinel-2 data, similar performance scores are obtained, whereas, in the case of GeoEye-1 data, the increase in performance is considerable when ResNet features are used. Nevertheless, ResNet feature extractor comes with the advantage of using the same pretrained architecture for all datasets, with no need to learn a specific dictionary for each dataset.

REFERENCES

[1] G. M. Foody and A. Mathur, “A relative evaluation of multiclass image classification by support vector machines,” *IEEE Trans. Geosci. Remote Sens.*, vol. 42, no. 6, pp. 1335–1343, Jun. 2004.

[2] P. Blanchart and M. Datcu, “A semi-supervised algorithm for auto-annotation and unknown structures discovery in satellite image databases,” *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 3, no. 4, pp. 698–717, Dec. 2010.

[3] J. Munoz-Mari, D. Tuia, and G. Camps-Valls, “Semisupervised classification of remote sensing images with active queries,” *IEEE Trans. Geosci. Remote Sens.*, vol. 50, no. 10, pp. 3751–3763, Oct. 2012.

[4] C.-W. Hsu and C.-J. Lin, “A comparison of methods for multiclass support vector machines,” *IEEE Trans. Neural Netw.*, vol. 13, no. 2, pp. 415–425, Mar. 2002.

[5] E. Maggiori, Y. Tarabalka, G. Charpiat, and P. Alliez, “Convolutional neural networks for large-scale remote-sensing image classification,” *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 2, pp. 645–657, Feb. 2017.

[6] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” in *Proc. 3rd Int. Conf. Learn. Representations*, San Diego, CA, USA, May 7–9, 2015, [Online]. Available: http://arxiv.org/abs/1409.1556

[7] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2016, pp. 770–778.

[8] T. G. Dietterich and G. Bakiri, “Solving multiclass learning problems via error-correcting output codes,” *J. Artif. Intell. Res.*, vol. 2, no. 1, pp. 263–286, 1995.

[9] A. Berger, “Error-correcting output coding for text classification,” in *Proc. IJCAI’99: Workshop Mach. Learn. Inf. Filtering*, 1999.

[10] V. Guruswami and A. Sahai, “Multiclass learning, boosting, and error-correcting codes,” in *Proc. 12th Annu. Conf. Comput. Learn. Theory*, 1999, pp. 145–155.

[11] M. Ferecatu and N. Boujemaa, “Interactive remote-sensing image retrieval using active relevance feedback,” *IEEE Trans. Geosci. Remote Sens.*, vol. 45, no. 4, pp. 818–826, Apr. 2007.

[12] D. Tuia, F. Ratle, F. Pacifici, M. F. Kanevski, and W. J. Emery, “Active learning methods for remote sensing image classification,” *IEEE Trans. Geosci. Remote Sens.*, vol. 47, no. 7, pp. 2218–2232, Jul. 2009.

[13] B. Demir and L. Bruzzone, “A novel active learning method in relevance feedback for content-based remote sensing image retrieval,” *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 5, pp. 2323–2334, May 2015.

[14] S. Tong and E. Chang, “Support vector machine active learning for image retrieval,” in *Proc. 9th ACM Int. Conf. Multimedia*, 2001, pp. 107–118.

[15] S. Tong and D. Koller, “Support vector machine active learning with applications to text classification,” *J. Mach. Learn. Res.*, vol. 2, pp. 45–66, 2002.

[16] O. E. Dai, B. Demir, B. Sankur, and L. Bruzzone, “A novel system for content-based retrieval of single and multi-label high-dimensional remote sensing images,” *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 11, no. 7, pp. 2473–2490, Jul. 2018.

[17] G. Csurka, C. R. Dance, L. Fan, J. Willamowski, and C. Bray, “Visual categorization with Bags of Keypoints,” in *Proc. Workshop Statist. Learn. Comput. Vis.*, 2004, pp. 1–22.

[18] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.

[19] O. Russakovsky *et al.*, “ImageNet large scale visual recognition challenge,” *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, 2015.

[20] L. Gueguen, “Classifying compound structures in satellite images: A compressed representation for fast queries,” *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 4, pp. 1803–1818, Apr. 2015.

[21] C. E. Shannon, “A mathematical theory of communication,” *Bell Syst. Tech. J.*, vol. 27, no. 3, pp. 379–423, 1948.

[22] E. R. Berlekamp, *Algebraic Coding Theory - Revised Edition*. River Edge, NJ, USA: World Sci. Publishing Co., Inc., 2015.

[23] S. B. Wicker, *Error Control Systems for Digital Communication and Storage*. Upper Saddle River, NJ, USA: Prentice-Hall, 1995.

[24] J. H. V. Lint, *Introduction to Coding Theory*, 3rd ed. Secaucus, NJ, USA: Springer-Verlag, 1998.

[25] E. R. Berlekamp, “The technology of error-correcting codes,” *Proc. IEEE*, vol. 68, no. 5, pp. 564–593, May 1980.

[26] C.-C. Chang and C.-J. Lin, “Libsvm: A library for support vector machines,” *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, pp. 27:1–27:27, 2011.

[27] J. Wang, P. Neskovic, and L. N. Cooper, “Training data selection for support vector machines,” in *Proc. Adv. Natural Comput.*, 2005, vol. 3610, pp. 554–564.

[28] D. Marmanis, M. Datcu, T. Esch, and U. Stilla, “Deep learning Earth observation classification using imagenet pretrained networks,” *IEEE Geosci. Remote Sens. Lett.*, vol. 13, no. 1, pp. 105–109, Jan. 2016.

[29] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, “Overfeat: Integrated recognition, localization and detection using convolutional networks,” in *Proc. Int. Conf. Learn. Representations*, Dec. 2013. [Online]. Available: http://arxiv.org/abs/1312.6229

[30] A. Radoi and M. Datcu, “Bag-of-visual words and error-correcting output codes for multilabel classification of remote sensing images,” in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2018, pp. 6955–6958.

Anamaria Radoi (M’13) received the B.Sc. degree in electronics and telecommunications from the University Politehnica of Bucharest (UPB), Bucharest, Romania, in 2010, the M.Sc. degree in communication systems from Ecole Polytechnique Federale de Lausanne (EPFL), Lausanne, Switzerland, in 2012, and the Ph.D. degree in electronics and telecommunications from the UPB, in 2015.

She is an Assistant Professor with the Department of Applied Electronics and Information Engineering, UPB. Before joining UPB, she served as a Research Assistant with EPFL and pursued two research internships with the German Aerospace Center, Oberpfaffenhofen, Germany. Her main research interests include signal and image processing, machine learning, pattern recognition, change detection, multitemporal analysis, and information and coding theory.

Mihai Datcu (SM’04–F’13) received the M.S. and Ph.D. degrees in electronics and telecommunications from the University Politechnica Bucharest (UPB), Bucharest, Romania, in 1978 and 1986, respectively, and the title “Habilitation diriger des recherches” in computer science from Louis Pasteur University, Strasbourg, France, in 1999.

Since 1981, he has been a Professor with the Department of Applied Electronics and Information Engineering, Faculty of Electronics, Telecommunications and Information Technology, UPB, working in image processing and electronic speckle interferometry. Since 1993, he has been a Scientist with the German Aerospace Center (DLR), Wessling, Germany. He is developing algorithms for model-based information retrieval from high-complexity signals and methods for scene understanding from very high-resolution synthetic aperture radar (SAR) and interferometric SAR data. He is currently a Senior Scientist and Image Analysis Research Group Leader with the Remote Sensing Technology Institute, DLR. Since 2011, he has also been leading the Immersive Visual Information Mining Research Laboratory, Munich Aerospace Faculty, Munich, Germany, and he is the Director of the Research Center for Spatial Information, UPB. He has held Visiting Professor appointments with the University of Oviedo, Oviedo, Spain; Louis Pasteur University; the International Space University, Strasbourg, France; the University of Siegen, Siegen, Germany; the University of Innsbruck, Innsbruck, Austria; the University of Alcala, Alcala, Spain; the University Tor Vergata, Rome, Italy; the Universidad Pontificia de Salamanca, campus de Madrid, Spain; the University of Camerino, Camerino, Italy; and the Swiss Center for Scientific Computing, Manno, Switzerland. From 1992 to 2002, he had a longer Invited Professor assignment with the Swiss Federal Institute of Technology, ETH Zurich. In 2001, he had initiated and led the Competence Centre on Information Extraction and Image Understanding for Earth Observation at ParisTech, Paris Institute of Technology, Telecom Paris, a collaboration of DLR with the French Space Agency (CNES). He has been a Professor Holder of the DLR-CNES Chair with ParisTech, Paris Institute of Technology, Telecom Paris. He initiated the European frame of projects for Image Information Mining and is involved in research programs for information extraction, data mining and knowledge discovery, and data understanding with the European Space Agency, NASA, and in a variety of national and European projects. He has authored more than 450 scientific publications, among them about 80 journal papers, and a book on number theory. His research interests include information theoretical aspects, semantic representations in advanced communication systems, Bayesian inference, information and complexity theory, stochastic processes, model-based scene understanding, image information mining for applications in information retrieval and understanding of high-resolution SAR, and optical observations.

Dr. Datcu is a member of the ESA Big Data from Space Working Group. He and his team have developed and are currently developing the operational IIM processor in the Payload Ground Segment systems for the German missions TerraSAR-X, TanDEM-X, and the ESA Sentinel 1 and 2. He has served as a Co-Organizer of international conferences and workshops, and as a Guest Editor of a special issue on IIM of the IEEE and other journals. He received the Best Paper Award in 2006, the IEEE Geoscience and Remote Sensing Society Prize in 2008, the National Order of Merit with the rank of Knight, for outstanding international research results, awarded by the President of Romania, and in 1987 the Romanian Academy Prize Traian Vuia for the development of SAADI image analysis system and activity in image processing. He is also a member of the European Big Data from Space Working Group.