A Novel SVM-Based Decoder for Remote Sensing Image Captioning

Genc Hoxha, *Graduate Student Member, IEEE*, and Farid Melgani, *Fellow, IEEE*

***Abstract*— Most of the remote sensing image captioning (IC) models are based on encoder–decoder frameworks where a convolutional neural network (CNN) encodes the image information and a recurrent neural network (RNN) decodes the image information into a sentence description. In order to achieve good accuracies, encoder–decoder frameworks relying on RNNs typically require a huge amount of annotated samples. Furthermore, they demand high and expensive computational power in order to have reasonable training and testing time. In this article, we aim to address these issues by introducing a novel decoder that is based on support vector machines (SVMs). In particular, instead of RNNs, we propose a novel network of SVMs to decode the image information into a sentence description. The proposed IC system is particularly interesting when just a limited amount of training samples is available. Experiments conducted on four different IC datasets confirm the promising capability of the proposed IC system to generate descriptions that are highly correlated with the image content. The proposed IC system is characterized by short training and inference times compared to other state-of-the-art models.**

***Index Terms*— Convolutional neural networks (CNNs), image captioning (IC), recurrent neural networks (RNNs), support vector machines (SVMs), unmanned aerial vehicles (UAVs).**

I. Introduction

THE fast development of remote sensing (RS) technology has enabled the acquisition of high spatial resolution images that are crucial for earth monitoring and observation. This has led to abundant information and to new challenges regarding the study and interpretability of RS images.

The automatic interpretation of RS images has been mainly concentrated on techniques, such as image classification/segmentation and object recognition. The aim of such techniques is to represent images with a set of spatialized semantic land-cover classes (labels). Due to their intrinsic nature, these techniques do not capture the attributes and the relationships that exist between different land-cover classes. It is worth mentioning that the attributes and the relationships between different land-cover classes are part of the high-level semantic information of an RS scene. Recently, in order to have a better understanding of an RS scene, image captioning (IC) has attracted the attention of the community. IC is a difficult but fundamental task in artificial intelligence (AI) that aims to generate a textual description (i.e., sentence or caption) of the content of an image. It requires both the knowledge of computer vision and natural language processing (NLP) fields in order to better understand the content of an image and express this knowledge through a sentence description. Unlike previous techniques, IC not only provides the labels of different land-cover classes and their locations but is also able to capture and express the relationships that exist between different land-cover classes with a sentence. IC can be very important in several RS applications such as image retrieval [1].

Inspired by the IC works developed in the computer vision community [2]–[10], IC can be divided into three main categories: 1) template-based; 2) retrieval-based; and 3) encoder–decoder IC frameworks. Template-based IC systems are composed of fixed sentence templates. First, object detection algorithms are exploited in order to detect objects and actions, and then, the fixed templates are filled with the detected objects. The only example of the template-based IC method in the RS community is the work of Shi and Zou [11] where a fully convolutional network model is used to detect the objects of an image and a language model based on fixed templates is used to generate the descriptions. The sentences generated by template-based IC are, in general, correct from a grammatical and content viewpoint. However, they are heavily hand-designed, and because of the fixed templates, the generated descriptions tend to be less natural compared to human descriptions.

The second type of IC is retrieval-based IC. In this methodology, the generation of a sentence is treated as a retrieval problem. First, given a target image, retrieval-based IC methods search and retrieve from an archive the most similar images (to the target image) along with their descriptions. Then, to the target image, one or more descriptions of its most similar images are assigned. As an example, Wang *et al.* [12] mapped images and descriptions in the same semantic space in order to learn a distance metric to quantify the similarity between images and descriptions. At inference time, to a target image, five descriptions that have the smallest distance with the considered image are assigned. Retrieval-based IC systems do not have the capability to generate novel descriptions, and they assume that there is always a related image–text pair in the archive for a target image. This assumption might not always be true leading to descriptions that are uncorrelated to the image content.

Manuscript received March 28, 2021; revised June 21, 2021; accepted August 11, 2021. Date of publication August 25, 2021; date of current version January 26, 2022. *(Corresponding author: Farid Melgani.)*

The authors are with the Department of Information Engineering and Computer Science, University of Trento, 38123 Trento, Italy (e-mail: genc.hoxha@unitn.it; melgani@disi.unitn.it).

Digital Object Identifier 10.1109/TGRS.2021.3105004

The most widely used ICs in the RS community are encoder–decoder frameworks [13]–[21] that are inspired by the progress in deep learning and machine translation [22]–[24]. Encoder–decoder IC methods exploit pretrained convolutional neural networks (CNNs) to encode the visual feature of an image and sequential models, such as recurrent neural networks (RNNs) or long short-term memory (LSTM) [25], in order to translate the encoded visual features in a sentence description. Encoder–decoder frameworks can be divided as: 1) simple encoder–decoder and 2) attention-based encoder–decoder. Simple encoder–decoder represents an image with a fixed-length vector, and the same feature vector is used as input to the decoder to generate the description one word at a time. Encoder–decoders that exploit attention mechanisms focus their attention on different parts of an image and extract different feature vectors for each part. In this case, the decoder uses feature vectors extracted by those portions of an image that are more related to the generated words.

The first work in the RS community that uses a simple encoder–decoder framework is exploited in [13] where different CNNs [26]–[28] are used to encode the visual features of images and RNN, or LSTM [25] is used to generate the captions. Zhang *et al.* [14] detected the main objects from an RS scene and forwarded the detected objects into an RNN model to generate the descriptions. Handcrafted features [29]–[31], in addition to deep features and attention mechanism [8], [24], are explored in [15] to generate sentences. An attribute attention mechanism is introduced in [16]. A multiscale cropping and training mechanism is introduced in [17] for data augmentation to alleviate the problem of overfitting. To deal with the large-scale variation present in RS images, a multiscale feature fusion combined with a denoising mechanism is introduced in [18], and two different multiscale methods based on feature pyramid networks [32] are presented in [33]. Lu *et al.* [19] introduced an active attention mechanism where the sound of the name of different objects uttered by humans is used as guiding information to generate descriptions, and a retrieval topic recurrent memory is introduced in [20] where sentence topics are used to guide the description generation process. Sumbul *et al.* [34] introduced a summarization-driven RS IC system where a pretrained pointer generator [35] is used to summarize the ground-truth captions with the aim of keeping only the relevant information, which is then integrated into the IC system through an attention mechanism to generate coherent descriptions. A combination of a simple encoder–decoder IC framework and a retrieval-based IC framework is explored in [21] to alleviate the misrecognition problem of encoder–decoder IC frameworks. Li *et al.* [36] introduced a truncation cross-entropy loss with the aim of alleviating the overfitting problem, whereas Zhao *et al.* [37] proposed a structured attention mechanism that is able to exploit structured spatial relationships that are widely present in an RS scene in contrast to previous unstructured attention methods.

In general, encoder–decoder frameworks generate novel sentences that are very similar (from a syntax and a lexical viewpoint) to the sentences produced by humans. However, encoder–decoder frameworks that use RNNs as decoder are affected by various issues. For example, the performance of encoder–decoder frameworks based on RNNs depends on the number of annotated training samples (the larger the training set, the lower the risk of overfitting). Indeed, in the computer vision community, the datasets that are used to train and test IC systems in the form of encoder–decoder frameworks are characterized by a very large amount of annotated samples. An example of such a dataset is the MS COCO dataset [38] that is composed of more than 300 000 images where each image is annotated with five descriptions. In contrast to the computer vision community, in the RS community, datasets used to perform IC are typically small since creating big datasets is not always possible as it is an expensive process in terms of time and resources. Another issue is the high number of hyperparameters that are needed to be carefully chosen in order to have good performance. Furthermore, deep learning methodologies demand expensive computational power units, such as graphics processing units (GPUs), to have reasonable training and testing times. It is worth mentioning that the more complex the system, the more acute the aforementioned issues.

In order to cope with the aforementioned problems, in this article, we propose a novel decoder that is based on a network of support vector machines (SVMs) [39] for those situations in which it is possible to only have a limited number of training samples. SVMs are well-known classifiers in the RS community [40]. They are based on the margin maximization principle that renders them less sensitive to overfitting compared to deep learning methodologies [40]. In this work, a network of SVMs is used as a decoder instead of RNNs or LSTMs to alleviate the problem of overfitting and speed up training and inference times. Another advantage of SVMs is the low number of hyperparameters that need to be chosen to yield an accurate system. The proposed IC framework is shown in Fig. 1. A CNN extracts image features and represents them with a fixed-length feature vector, and a network of k SVM multiclass classifiers in cascade translates the feature vector into a sentence description. The last SVM multiclass classifier is rendered recurrent to model the dependence on the previous words while generating the new words of a sentence. Note that this work is part of simple encoder–decoder networks that do not explore any kind of attention mechanism.

Overall, the main contributions of our work can be summarized as follows.

1) A novel decoder architecture based on SVM is introduced for the first time in the framework of IC. It is suitable for situations in which only a few training samples are available to alleviate the problem of overfitting.
2) The proposed framework achieves better results compared to simple encoder–decoder frameworks in terms of accuracy and shows comparable and, sometimes, better results compared with the more sophisticated encoder–decoders that exploit attention mechanisms.
3) The proposed method is characterized by extremely short training and inference times.

The rest of this article is organized as follows. Section II introduces the proposed IC system. Section III shows the

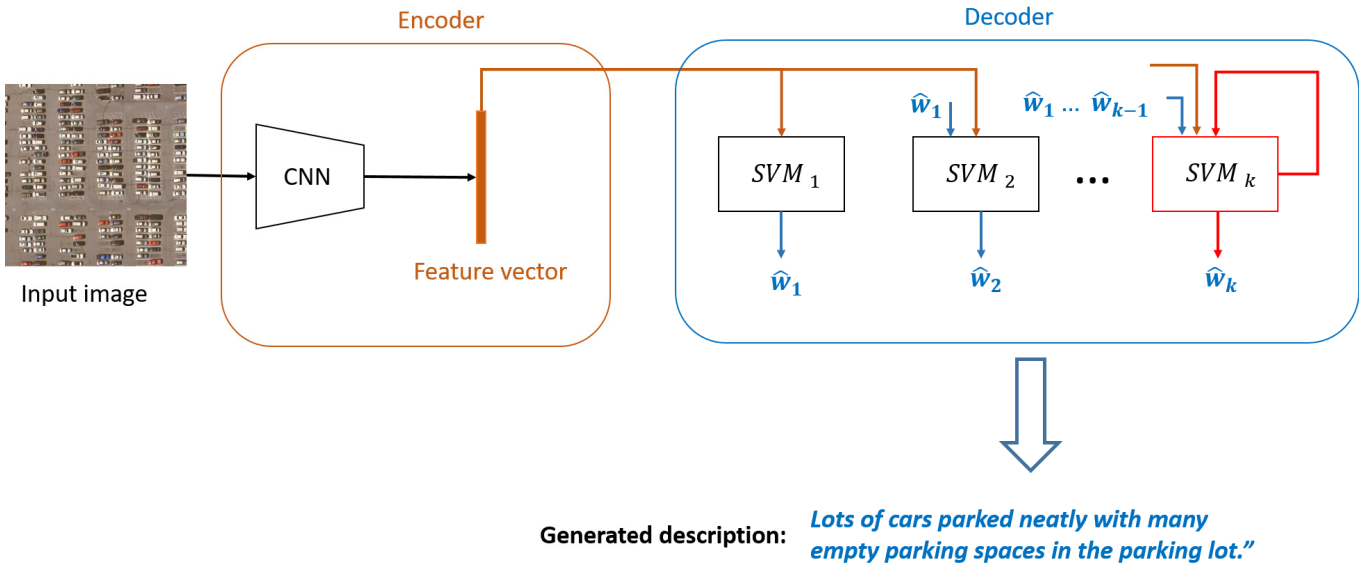


Fig. 1. Overview of the proposed captioning method. The proposed method consists of two parts: an encoder, which maps the images into feature space, and a decoder composed of a network of K SVM multiclass classifiers that generate the captions. The kth classifier (highlighted in red) is rendered recurrent. The prediction process stops when a particular words indicating the end of the sequence is predicted.

experimental results on four different RS IC datasets. Finally, Section IV draws the conclusion of this work.

II. Proposed IC System

Let $X = \{X_1, X_2, \dots, X_M\}$ be a training set consisting of M images and X_i be the ith image. Let us assume that each image X_i is annotated with one or more sentence descriptions (or captions). Let $S_i = \{s_{i,j}\}_{j=1}^{J}$ be the set of sentences associated with the image X_i and $s_{i,j}$ be the jth sentence description in the set. Each sentence $s_{i,j}$ can be formulated as a set of ordered words $s_{i,j} = \{w_{ij,l}\}_{l=1}^{L}$, where $w_{i,j,l}$ is the lth word of the sentence $s_{i,j}$ and L is the maximum length of $s_{i,j}$. As any encoder–decoder IC framework, our proposed IC system is composed of two steps: 1) image representation and 2) sentence generation. The first step aims to represent the input image with discriminative features, while the second one is focused on the translation of the features into a sentence description. In this work, we have used a pretrained CNN to deal with the first part and a network of k SVM multiclass classifiers to deal with the language part. In particular, the kth SVM is rendered recurrent to model the dependence of the previously predicted words while generating the successive words of the sentence description.

A. Image Representation

The first step of an IC system is to represent the images with discriminative features. To this end, we rely on CNNs since they have shown to be able to overcome the need for handcrafted features [41]. To be in the same line as in most previous encoder–decoder IC systems in the RS community, in our work, we exploit the VGG16 [27] CNN architecture pretrained on ImageNet [42]. The image features are obtained passing each image X_i through the pretrained CNN architecture (omitting the last fully connected layer) as follows:

$$f_i = \text{VGG16}(X_i) \tag{1}$$

where f_i is the feature vector associated with image X_i.

B. SVM Decoders

The main difference between the proposed IC system and the previous works is the sentence generation part or decoding stage. While most of the IC systems use sequential models, such as RNN and LSTM as a decoder, in this work for the first time, we develop a network of k SVMs in cascade to decode the features into a sentence, as shown in Fig. 1. More precisely, given an image X_i and one of its sentence descriptions $s_{i,j}$, the first SVM, namely, SVM_1, learns the mapping between the feature vector f_i of the considered image X_i and the first word $w_{i,j,1}$ of the sentence $s_{i,j}$ represented by the following formulation:

$$w_{i,j,1} = \text{SVM}_1(f_i). \tag{2}$$

The second SVM multiclass classifier (SVM_2), in turn, learns the mapping between, on the one hand, the feature vector f_i and the first word $w_{i,j,1}$ of sentence $s_{i,j}$ and, on the other hand, the second word $w_{i,j,2}$, as shown in the following equation:

$$w_{i,j,2} = \text{SVM}_2(f_i, w_{i,j,1}). \tag{3}$$

Following the same logic, the subsequent $k-1$ SVM multiclass classifiers (SVM_{k-1}) will learn the mapping between, on the one hand, the image features f_i and the previous $k-2$ words $w_{i,j,l}$ (with $l = 1, 2, \dots, k-2$) and, on the other hand, the subsequent word $w_{i,j,k-1}$

$$w_{i,j,k-1} = \text{SVM}_{k-1}(f_i, w_{i,j,1}, \dots, w_{i,jk-2}). \tag{4}$$

The last multiclass SVM classifier, namely, SVM_k, is a particular classifier as it is rendered recurrent. In a recurrent manner, this classifier learns the mapping between the image features f_i and $k-1$ previous words, on the one hand, and the subsequent $L-k$ words, on the other hand, where L is the length of the considered sentence. To each sentence, two special words w_0 "startseq" and w_{L+1} "endseq" indicating the start and the end of a sentence are added, respectively. Each word $w_{i,j,l}$ of the sentence $s_{i,j}$ is encoded using one-hot encoding with dimension V, which is the vocabulary size. In order to represent the previous words at a given point l in the sentence, we encode the part of the sentence up to l in two ways: 1) by concatenating the word vectors or 2) by relying on a bag of words (BoW) representation.

1) Sentence Encoding With Word Concatenation: The word concatenation allows preserving the sequential order of the words. The kth SVM multiclass classifier recurrently learns the mapping between the image features and a fixed size of $k-1$ previous words, on the one hand, and the subsequent $L-k$ words, on the other hand. Image features and the $k-1$ previous word vectors are concatenated together. More precisely, at each step (iteration), we have a fixed window size shift of $k-1$ words while learning the mapping of the subsequent words $w_{i,j,l}$. The recurrent SVM multiclass classifier is able to capture temporal sequences in an explicit way up to order k, but the input vector may be large. This, however, is well handled by SVM, which tolerates high-dimensional inputs.

2) Sentence Encoding Using BoW: BoW is an encoding technique used in the NLP field where each sentence is represented as a vector of a fixed length of vocabulary size V and each entry of the vector represents the number of times that each word appears in the considered sentence. In order to have the BoW representation of part of a sentence, we simply sum up the one-hot vector representations of the words composing the considered part of the sentence. An advantage of this encoding is that, in order to learn the mapping of a subsequent word, we exploit all the previous words of the sentence (and not just a subset) while keeping unchanged the size of the generated code (V). A drawback is that the word order is lost. As in the previous sentence encoding strategy, image features and word vectors are concatenated together.

C. SVM Training and Inference

For simplicity, let us consider a binary classification. Let us assume to have a training set consisting of M vectors from d-dimensional feature space $x_i \in R^d (i = 1, 2, \ldots, M)$ where each training sample is associated with a positive or negative class $y_i \in \{1, -1\}$. The SVM consists in mapping the data into a higher dimensional feature space, i.e., $\Phi(x) \in R^{d'}$ $(d' \gg d)$, with the aim of finding a hyperplane that separates the two classes by minimizing the following cost function:

$$\Psi(\omega, \xi) = \frac{1}{2}\|\omega\|^2 + C\sum_{i=1}^{M}\xi_i \tag{5}$$

which is a combination of two criteria, margin maximization and error minimization. ξ_i is the so-called slack variable, and C is a regularization parameter. The cost minimization function $\Psi(\omega, \xi)$ is subjected to the following constraints:

$$y_i(\bar{\omega} \cdot \Phi(x_i)) + b \geq 1 - \xi_i, \quad i = 1, 2, \ldots, M \tag{6}$$

and

$$\xi_i \geq 0, \quad i = 1, 2, \ldots, M. \tag{7}$$

The optimization problem can be transformed into a dual formulation and kernelized, leading to quadratic programming (QP) solution [39]. In the end, the following discriminant function is obtained:

$$h(x) = \sum_{i \in S} \alpha_i y_i K(x_i, x) + \bar{b} \tag{8}$$

where $K(\cdot, \cdot)$ is a kernel function and S a subset of indices $i = 1, 2, \ldots, M$. The binary classification case can be easily extended to multiclass classification following one versus one or one versus all strategies [40].

During the test phase, the features of the test images are given as input to the network of k SVM classifiers to generate the sentence descriptions of the images one word at a time. The sentence generation process of a test image X_t is described with the following equations:

$$\widehat{w_{t,1}} = \text{SVM}_1(f_i) \tag{9}$$

$$\widehat{w_{t,2}} = \text{SVM}_2(f_i, \widehat{w_{i,1}}) \tag{10}$$

$$\vdots$$

$$\widehat{w_{t,k}} = \text{SVM}_k(f_i, \widehat{w_{t,1}}, \ldots, \widehat{w_{t,k-1}}). \tag{11}$$

The recurrent SVM (SVM_k), depending on the sentence encoding (see Fig. 2), will recurrently predict the subsequent words on the basis of the image features and the previous words. The sentence generation process stops when predicting the special word "endseq" indicating the end of the sentences.

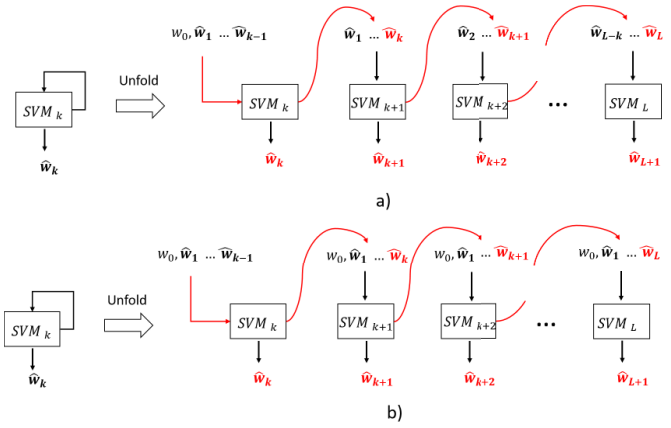


Fig. 2. Recurrent SVM multiclass classifier (a) with word concatenation (SVM-D CONC) and (b) with BoW (SVM-D BOW). w_0 and w_{L+1} are special tokens indicating the start and the end of a sentence, respectively.

III. Experiments

A. Dataset Descriptions and Evaluation Metrics

1) Datasets: To validate the proposed RS IC system, we conducted experiments on four different datasets: UAV,

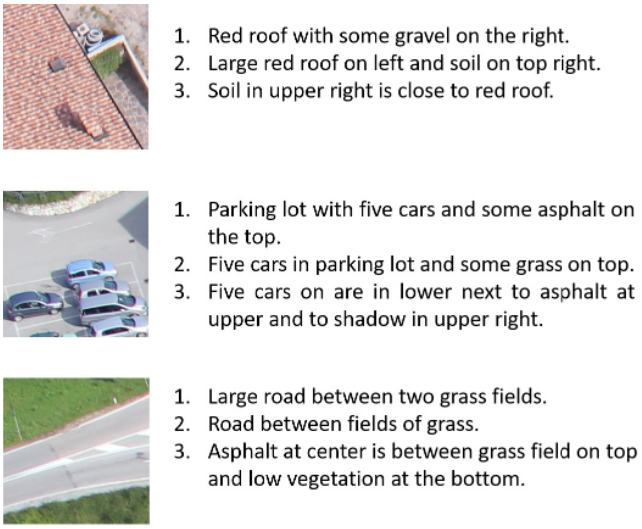


Fig. 3. Example of three images from the UAV dataset along with their three descriptions.

UCM-captions, Sydney-captions, and Remote Sensing Image Captioning datasets (RSICDs). The first three datasets are characterized by a small number of annotated images and are more suitable for our IC system. The datasets are described in the following.

a) UAV dataset: It was acquired by an unmanned aerial vehicle (UAV) with EOS 550D camera near the city of Civezzano (Italy) on October 17, 2012. It is composed of ten RGB images of size 5184×3456 with a spatial resolution of 2 cm from which six images are used for training, one image for validation, and three images for testing. From these, crops of size 256×256 are generated. In particular, 1746 crops are extracted from training from validation images and 882 from testing images, and they are used for training, validation, and testing, respectively. To each crop, three descriptions written by different annotators are assigned. Fig. 3 shows three examples from the dataset along with their descriptions.

b) UCM caption dataset: It is based on the UC Merced Land Use dataset [43] and proposed in [13]. It contains 2100 images of size 256×256 characterized by a spatial resolution of 30.48 cm. Each image is annotated with five different sentences.

c) Sydney caption dataset: It originates from the Sydney dataset [44] and is proposed in [13]. The dataset is composed of 613 images characterized by a spatial resolution of 50 cm. Each image is annotated with five different descriptions.

d) RSICD: This dataset is the largest in the RS community used for IC and is provided in [15]. It is composed of more than 10 000 images of various resolutions and fixed size 224×224. Each image is annotated with five descriptions.

2) Evaluation Metrics: The typical metrics used to evaluate the performances of an RS IC system are BiLingual Evaluation Understudy (BLEU) [45], Recall-Oriented Understudy for Gisting Evaluation (ROUGE_L) [46], Metric for Evaluation of Translation with Explicit Ordering (METEOR) [47], and Consensus-based Image description Evaluation (CIDEr) [48]. They measure how close the outputs of an IC system (generated description) are to the descriptions provided by human experts (reference descriptions).

B. Experimental Setup

The experiments were performed maintaining the default splitting for the three publically available datasets (UCM, Sydney, and RSICD): 80%, 10%, and 10% for training, validation, and test, respectively, whereas, for the UAV dataset, the splitting is 60% for training, 10% validation, and 30% test. The vocabulary sizes V for each dataset are 127, 338, 216, and 3323 words for UAV, UCM, Sydney, and RSCID, respectively. As it was previously discussed, the peculiarity of this work is a decoding process based on K SVM multiclass classifiers. In our experiments, we adopted $K = 4$, that is, we have four SVM multiclass classifiers in total where the last one is recurrent. We believe this value captures satisfactorily within-sentence correlation, while model complexity keeps contained. It is also noteworthy that, in the literature, the BLEU metric typically does not go beyond $n = 4$ because within-sentence correlation drops significantly. In order to provide an in-depth analysis, we conduct an additional thorough experimental study, which is reported in Section II-E.

We rely on a linear SVM multiclass classifier that has only one free parameter that is the value of the regularization parameter C. The best values found on the validation set are $C = 10^{-3}$, 10^{-1}, 10^{-2}, and 10^{-2} for SVM_1, SVM_2, SVM_3, and SVM_4, respectively, for all datasets. The SVM implementation is based on LIBLINEAR [49] and is implemented in python. The experiments were conducted on an Intel Xeon CPU E5-1620 v3 @ 3.50-GHz machine.

The image features are obtained using VGG-16 as a backbone pretrained on ImageNet. VGG-16 produces a fixed feature vector of 4096 dimensions. Each SVM takes as input the image features and the previously predicted words (if present) to predict the subsequent words. All image and word features are scaled to be in the range [0, 1] using minimax scalar characterized by the following formula:

$$x' = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \tag{12}$$

where x is the original feature value, $x_{\min}$ and $x_{\max}$ are the minimum and maximum feature values, respectively, and x' is the normalized feature value. Note that the recurrent SVM at each step (iteration) takes as input both image features and previously predicted words to recurrently predict the subsequent ones. Depending on the sentence encoding process, we have a different dependence on the previously predicted words. In the case of sentence encoding with word concatenation, the prediction of a new word depends on a fixed window size of $k = 4$ previously predicted words. This strategy allows maintaining the word order. Conversely, the sentence encoding is performed using BoW, the prediction of a new word depends on all previous words (from sentence start), but the word order is not preserved. The iteration process of the recurrent SVM ends when a special token indicating the end of the sentence is predicted, or the maximum possible length of a sentence is reached. The maximum length of a sentence is chosen on the

basis of the longest sentence present in the training set, and it is different for each dataset.

C. Description of Reference Methods

To show the effectiveness of the proposed method, we compare it with some state-of-the-art methods. The majority of these works are based on the encoder–decoder frameworks. Among them, we can find encoder–decoder frameworks without and with attention mechanisms. It is worth noting that our SVM-based IC system belongs to the category of encoder–decoder methods, which does not exploit any kind of attention mechanism. Besides these two major categories, it is noteworthy to mention a retrieval method proposed in [12] and called CSMLF. In this method, images and sentences are mapped into the same latent semantic space, and a distance metric is developed to measure the similarity between images and sentences. In the following, we describe briefly the methods used for comparison.

1) Simple Encoder–Decoder Frameworks:

a) VLAD + RNN and VLAD + LSTM introduced in [15]: These methodologies are based on handcrafted features where the encoder is the well-known vector of locally aggregated descriptors (VLAD) [31], and the decoder is the simple RNN and LSTM [25], respectively. LSTM is a more sophisticated version of the simple RNN.

b) mRNN, mGRU, and mLSTM introduced in [13]: These methodologies use deep features where the pretrained VGG 16 CNN architecture is employed as an encoder to represent the image with a fixed-length feature vector and as decoder the simple RNN, LSTM, and GRU, respectively. The gated recurrent unit (GRU) [50] is a variant of the simple RNN.

c) mGRU-embed word: It is the same multimodal framework used in [13] in which images are encoded using the pretrained VGG16 architecture to obtain the image features represented by a fixed length vector, and GRU is used as a decoder to generate the descriptions. The difference is that, instead of training the word vectors from scratch, the authors exploit the pretrained word vectors obtained by the global vector (GLOVE) model [51].

d) Merge GRU-D [52]: This method uses VGG-16 to encode the image features into a fixed length feature vector and GRU to generate the descriptions. Different from the previous methods, GRU deals only with the sentence part. The image features are concatenated with the GRU output in a subsequent layer to condition the sentence generation with the image information.

2) Attention-Based Encoder–Decoder Frameworks:

a) Soft attention and hard attention: These are two attention-based methods [8], [24] introduced in the RS community in [15]. The image features are obtained using VGG16 CNN architecture. Unlike the simple encoder–decoder framework that exploits the penultimate fully connected layer to represent an image, here, convolutional layers are trained to produce convolutional maps of different parts of the image. The different parts of the images are weighted differently by the LSTM decoder to decide where to focus the attention while generating the words composing the sentence. In the hard attention model, a sampling strategy is used to decide the focus of the attention.

TABLE I
PERFORMANCE COMPARISON OF THE MACHINES USED FOR THE EXPERIMENTS

Method	Ours (CPU E5-1620)	[19] (CPU E5-2650)
CPU Speed	$4\ x\ 3.6\ GHz$	$\mathbf{8\ x\ 2\ GHz}$
CPU Threads	8	**16**
PassMark	9154	**10953**
Maximum memory size	$375GB$	$\mathbf{750GB}$

b) ConvCap: It is an attention-based model that is based on CNNs as encoder and decoder. In particular, VGG-16 is used to encode the image, and the CNN architecture designed by Aneja *et al.* [53] is employed as a decoder to generate the descriptions.

c) Retrieval topic recurrent memory network (RTRMN) [20]: It is an attention-based method based on ResNet-101 CNN architecture as encoder and a memory network as a decoder. Topic words are extracted and retrieved from the reference descriptions and are used to guide the decoder in generating the descriptions. "RTRMN" semantic and "RTRMN" statistical are two variants of the RTRMN that is based on the semantic topic words and on the statistical topic words, respectively.

d) Sound Active Attention (SAA) [19]: It is another attention-based method that exploits sound information to guide the decoder for generating the description of an image. It uses as encoder VGG-16 and sound GRU to encode the image information and the sound information, respectively, as well as a separate GRU as a decoder to generate the descriptions.

e) SD-RSIC [34]: It is an attention-based method that exploits the summary of the ground-truth captions to guide the decoder to generate the description of an image. It uses different pretrained CNN and LSTM to encode the image information and generate the descriptions, respectively. To be coherent with the other IC systems, we will consider only the results where VGG-16 is used as an encoder.

D. Experimental Results on Different Datasets

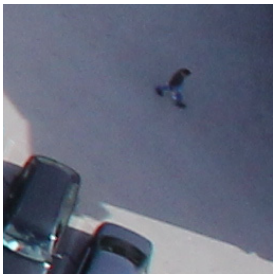
In this section, we discuss the experimental results achieved on four different datasets. The comparison with previously mentioned reference methods is done only for the three publicly available datasets. Regarding our UAV dataset, we have reported only the results achieved by our IC systems and merge GRU-D [52] that we implemented following the instructions in [52]. More details about the implementation of the merge GRU-D method are provided in Section III-E. The SVM-D CONC and SVM-D BOW are the two proposed IC systems that are based on the word concatenation and BoW model to encode the sentences, respectively. The results of most of the methods in terms of accuracy, training, and test times are taken from the experiments performed in [19]. The experiments provided by Lu *et al.* [19] are conducted on Ubuntu 14.04.5 LTS with 48 Intel CPU E5-2650 v4 @ 2.20, which

TABLE II

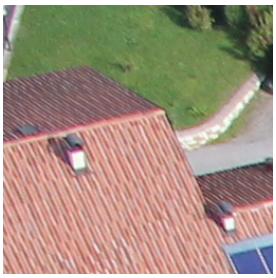
EVALUATION SCORES (%) AND TIMES NEEDED FOR TRAINING (MINUTES) AND TESTING (SECONDS) ON THE UAV DATASET. THE BEST RESULTS ARE REPRESENTED IN BOLD, WHILE THE SECOND BEST RESULTS ARE REPRESENTED IN ITALIC

Method	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR	ROUGE-L	CIDEr-D	Training time (minutes)	Test time (seconds)
Merge GRU-D [52]	*66.03*	55.08	44.87	35.37	31.31	66.76	368.36	4.9	20.20
SVM-D BOW	**68.84**	**58.05**	**48.33**	*39.22*	**32.81**	**69.63**	**391.31**	*3.8*	**1.73**
SVM-D CONC	65.13	*56.53*	*48.15*	**39.69**	*32.17*	*69.31*	*389.45*	**3.3**	*1.87*

1. **GT**: There is a vineyard field.
2. **Merge GRU-D:** There is vineyard field.
3. **SVM-D BOW:** There is a vineyard field.
4. **SVM-D CONC:** There is a vineyard field.



1. **GT:** Two cars on the bottom left and a person on the top.
2. **Merge GRU-D:** Black car on bottom left and road on bottom.
3. **SVM-D BOW:** There are two cars on the left and shadow on the bottom.
4. **SVM-D CONC:** There are two cars on the bottom left.



1. **GT:** Red roof on bottom and grass field on top.
2. **Merge GRU-D :** Red roof on bottom and red roof on bottom.
3. **SVM-D BOW:** Red roof at bottom is close to grass field.
4. **SVM-D CONC:** Red roof at bottom right is close to grass field at bottom.

1. **GT:** Road between grass field.
2. **Merge GRU-D:** Road on top and low vegetation on bottom.
3. **SVM-D BOW:** Road between grass.
4. **SVM-D CONC:** Road between grass field.

Fig. 4. Captioning examples of test images from the UAV dataset. The first description corresponds to one of the ground-truth descriptions, while the second, third, and fourth descriptions are generated by Merge GRU-D, SVM-D BOW, and SVM-D CONC models, respectively. The words in red indicate errors in the generated descriptions.

has a better performance compared to our machine. See Table I for more details. It is worth noting that, even though we have trained the merge GRU-D for 50 epochs, on each dataset (see Section III-E), the reported results in terms of accuracy, training, and testing times are based on the model that achieves the highest validation accuracy. The epoch number in which the highest validation accuracy is reached differs from dataset to dataset, and it is much less than 50. In particular, the epoch number with the highest validation accuracy is 13, 18, 10, and 3 for UAV, Sydney, UCM, and RSICD datasets, respectively.

1) Results on UAV Caption Dataset: Table II reports the quantitative results of our two IC systems and merge GRU-D [52] in terms of different metrics, and training and test times. We can see that both our IC systems show better results compared with merge GRU-D [52]. In particular, we can notice that SVM-D BOW achieves the highest results in almost all the metrics. In terms of training and testing times, we can see that our two proposed decoder is much faster compared to merge GRU-D [52], in particular, the testing time. Fig. 4 depicts four examples of images where the first description of each image is one of the reference descriptions, whereas the second, third, and fourth ones are generated by GRU-D [52], SVM-D BOW, and SVM-D CONC, respectively. We can notice that all the generated descriptions of our two models, even though with some errors, are in line with the image semantic content. In particular, the descriptions of the first and fourth images (Fig. 4) contain all the semantic information of the image and are very similar to the reference descriptions, whereas the descriptions of the second and third images (see Fig. 4) are affected by some errors, or they miss some semantic information. The descriptions generated by the merge GRU-D [52] seem to be more affected by errors. In fact, except from the first image (see Fig. 4) where the description is very accurate, the rest of the descriptions contain some errors related to the position and orientation of the objects. In particular, the generated descriptions seem to be biased toward the word "bottom." In the last column of Table II, the training and inference times of the models are reported. Our two models are faster than merge GRU-D [52], in particular, the test time. Note that the test time represents the time to generate all the descriptions of the images found on the test set.

2) Results on Sydney Caption Dataset: In this dataset, we compare the results of our IC systems with 13 different state-of-the-art IC systems that include retrieval-based, encoder–decoder-based, and attention-based encoder–decoder IC frameworks.

In Table III, the results of each method are reported, where the best and the second best results are in bold and italic, respectively, whereas the "−" symbol indicates that the corresponding metrics are not available for the

TABLE III

EVALUATION SCORES (%) AND TIMES NEEDED FOR TRAINING (MINUTES) AND TESTING (SECONDS) ON THE SYDNEY CAPTION DATASET. "−" INDICATES THAT THE CORRESPONDING METRIC ARE NOT AVAILABLE FOR THAT MODEL. THE BEST RESULTS ARE REPRESENTED IN BOLD, WHILE THE SECOND BEST RESULTS ARE REPRESENTED IN ITALIC

Method	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR	ROUGE-L	CIDEr-D	Training time (minutes)	Test time (seconds)
VLAD+RNN [15]	56.58	45.14	38.07	32.79	26.72	52.71	93.72	-	-
VLAD +LSTM [15]	49.13	34.12	27.60	23.14	19.30	42.01	91.64	-	-
mRNN [13]	51.30	37.50	20.40	19.30	18.50	-	161.00	-	-
mLSTM [13]	54.60	39.50	22.30	21.20	20.50	-	186.00	-	-
mGRU [13]	69.64	60.92	52.39	44.21	31.12	59.17	171.55	-	-
mGRU embedword [13]	68.85	60.03	51.81	44.29	30.36	57.47	168.94	-	-
Merge GRU-D [52]	73.07	63.37	56.41	49.87	33.09	63.34	193.93	4.2	2.45
CSMLF [12]	59.98	45.83	38.69	34.33	24.75	50.18	75.55	-	-
ConvCap [53]	74.72	65.12	57.25	50.12	34.76	66.74	214.84	-	-
Soft-attention [15]	73.22	66.74	**62.23**	**58.20**	**39.42**	*71.27*	**249.93**	-	-
Hard-attention [15]	*75.91*	66.10	58.89	52.58	*38.98*	**71.89**	218.19	-	-
SAA [19]	68.82	60.73	52.94	45.39	30.49	58.20	170.52	-	-
SD-RSIC [34]	72.4	62.1	53.2	45.1	34.2	63.6	139.5	-	-
SVM-D BOW	**77.87**	**68.35**	*60.23*	53.05	37.97	69.92	227.22	*3.37*	*0.39*
SVM-D CONC	75.47	*67.11*	59.70	*53.08*	36.43	67.46	*222.22*	**2.26**	**0.23**

TABLE IV

EVALUATION SCORES (%) AND TIMES NEEDED FOR TRAINING (MINUTES) AND TESTING (SECONDS) ON THE UCM CAPTION DATASET. "−" INDICATES THAT THE CORRESPONDING METRIC ARE NOT AVAILABLE FOR THAT MODEL. THE BEST RESULTS ARE REPRESENTED IN BOLD, WHILE THE SECOND BEST RESULTS ARE REPRESENTED IN ITALIC

Method	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR	ROUGE-L	CIDEr-D	Training time (minutes)	Test time (seconds)
VLAD+RNN [15]	63.11	51.93	46.06	42.09	29.71	58.78	200.66	121.43	2.57
VLAD +LSTM [15]	70.16	60.85	54.96	50.30	34.64	65.20	231.31	291.18	2.89
mRNN [13]	60.10	50.70	32.80	20.80	19.30	-	214.00	18.64	6.39
mLSTM [13]	63.50	53.20	37.50	21.30	20.30	-	222.50	23.58	5.83
mGRU [13]	42.56	29.99	22.91	17.98	19.41	37.97	124.82	34.00	32.43
mGRU embedword [13]	75.74	69.83	64.51	59.98	36.85	66.74	279.24	31.75	29.76
Merge GRU-D [52]	75.74	67.16	60.63	55.29	37.81	69.11	274.85	14	7.2
CSMLF [12]	36.71	14.85	7.63	5.05	9.44	29.86	13.51	-	-
ConvCap [53]	70.34	56.47	46.24	38.57	28.31	59.62	190.15	1567.32	56.21
Soft-attention [15]	74.54	65.45	58.55	52.50	38.86	72.37	261.24	1251.51	140.02
Hard-attention [15]	**81.57**	73.12	67.02	61.82	**42.63**	*76.98*	*299.47*	1310.45	14.79
SAA [19]	79.62	**74.01**	**69.09**	**64.77**	38.59	69.42	294.51	38.08	37.01
SD-RSIC [34]	74.8	66.4	59.8	53.8	39.0	69.5	213.2	-	-
RTRMN (semantic) [20]	55.26	45.15	39.62	35.87	25.98	55.38	180.25	-	-
RTRMN (statistical) [20]	*80.28*	*73.22*	*68.21*	*63.93*	*42.58*	**77.26**	**312.70**	-	-
SVM-D BOW	76.35	66.64	58.69	51.95	36.54	68.01	271.42	*10.80*	**1.73**
SVM-D CONC	76.53	69.47	64.17	59.42	37.02	68.77	292.28	**9.80**	*1.90*

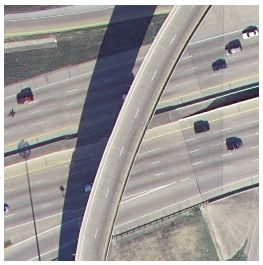
considered models. In particular, our two proposed IC systems not only can outperform all the simple encoder–decoder frameworks with a good margin in all the metrics but also show comparable or better results compared with attention-based IC systems. It is noteworthy that the results in terms of BLEU-1 and BLEU-2 achieved by SVM-D BOW are the best.

3) Results on UCM Caption Dataset: Here, we compare the results of our IC systems with 15 different state-of-the-art IC methods that comprise retrieval-based, encoder–decoder-based, and attention-based encoder–decoder IC frameworks. In Table IV, the results of each method are reported. The proposed IC systems can outperform the simple encoder–decoder and CSMLF methods in all the metrics. Furthermore, we can see that the results achieved by our IC systems, in particular, by SVM-D CONC, are also higher compared with some methods that exploit attention mechanisms, such as ConvCap [53], soft attention [15], and RTRMN (statistical) [20]. In the last two columns of Table IV, the training and inference times of each method are reported. We can see that our method is the fastest one. In particular, our IC systems are two to ten times faster in terms of training time compared to simple encoder–decoder frameworks and four to 170 times faster compared with more complicated IC systems that exploit attention mechanisms. We can notice that the same ratios are similar for the inference time.

Fig. 5 depicts four examples of images from the UCM dataset where the first description of each image is one of the reference descriptions, whereas the second, third, and fourth descriptions are generated by GRU-D [52], SVM-D BOW, and SVM-D CONC, respectively. From a visual inspection, we can see that all the descriptions generated by all the models are highly correlated with the images' content. In particular, one can notice that the SVM-D CONC seems to produce

1. **GT**: Some buildings with grey roofs are pressed together.
2. **Merge GRU-D:** There are some houses pressed together.
3. **SVM-D BOW:** There are some buildings with grey roofs.
4. **SVM-D CONC:** There are some buildings with grey roofs pressed together.

1. **GT:** An intersection with some houses and plants in the corners.
2. **Merge GRU-D:** An intersection with some houses and plants at the corners.
3. **SVM-D BOW:** An intersection with some houses and plants at the corners.
4. **SVM-D CONC:** An intersection with some houses and plants at the corners



1. **GT:** A road go across another two vertically with lots of cars on the road.
2. **Merge GRU-D:** An overpass with a road go across another two roads diagonally.
3. **SVM-D BOW:** An overpass with a road go across another roads diagonally with some cars on the roads
4. **SVM-D CONC:** An overpass with a road go across another two roads diagonally with some cars on the roads.

1. **GT:** A house with bushes surrounded is in the sparse residential area.
2. **Merge GRU-D:** A villa with plants surrounded and a road go through this area.
3. **SVM-D BOW:** A house with plants surrounded in the sparse residential area.
4. **SVM-D CONC:** A house with plants surrounded in the sparse residential area.

Fig. 5. Captioning examples of test images from the UCM dataset. The first description corresponds to one of the ground-truth descriptions, while the second, third, and fourth descriptions are generated by Merge GRU-D, SVM-D BOW, and SVM-D CONC models, respectively. The words in red indicate errors in the generated descriptions.

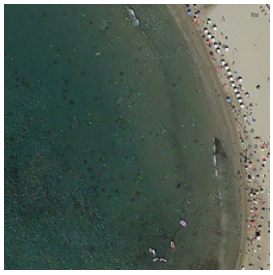
TABLE V

EVALUATION SCORES (%) AND TIMES NEEDED FOR TRAINING (MINUTES) AND TESTING (SECONDS) ON THE RSICD CAPTION DATASET. "–" INDICATES THAT THE CORRESPONDING METRIC ARE NOT AVAILABLE FOR THAT MODEL. THE BEST RESULTS ARE REPRESENTED IN BOLD, WHILE THE SECOND BEST RESULTS ARE REPRESENTED IN ITALIC

Method	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR	ROUGE-L	CIDEr-D	Training time (minutes)	Test time (seconds)
VLAD+RNN [15]	49.38	30.91	22.09	16.77	19.96	42.42	103.92	-	-
VLAD +LSTM [15]	50.04	31.95	23.19	17.78	20.46	43.34	118.01	-	-
mRNN [13]	45.58	28.25	18.09	12.13	15.69	31.26	19.15	-	-
mLSTM [13]	50.57	32.42	23.19	17.46	17.84	35.02	31.61	-	-
mGRU [13]	42.56	29.99	22.91	17.98	19.41	37.97	124.82	-	-
mGRU embedword [13]	60.94	46.24	36.80	29.81	26.14	48.20	159.54	-	-
Merge GRU-D [52]	60.30	42.48	32.03	25.20	22.94	4383	65.90	98.33	90.51
CSMLF [12]	51.06	29.11	19.03	13.52	16.93	37.89	33.88	-	-
ConvCap [53]	63.36	51.03	41.74	34.52	33.25	57.70	166.48	-	-
Soft-attention [15]	*67.53*	**53.08**	*43.33*	*36.17*	**32.55**	**61.09**	**196.43**	-	-
Hard-attention [15]	66.69	*51.82*	41.64	34.07	*32.01*	*60.84*	179.25	-	-
SAA [19]	**67.60**	44.33	**44.33**	**36.45**	31.09	55.36	*193.96*	-	-
SD-RSIC [34]	64.5	47.1	36.4	29.4	24.9	51.9	77.5	-	-
RTRMN (semantic) [20]	62.01	46.23	36.44	29.71	28.29	55.39	151.46	-	-
RTRMN (statistical) [20]	61.02	45.14	35.35	28.59	27.51	54.52	148.20	-	-
SVM-D BOW	61.12	42.77	31.53	24.11	23.03	45.88	68.25	*41.25*	**11.36**
SVM-D CONC	59.99	43.47	33.55	26.89	22.99	45.57	68.54	**35.82**	*23.32*

more complete descriptions compared to SVM-D BOW or to merge GRU-D [52] (Fig. 5). SVM-D CONC is able to detect and describe the fact that the buildings are in close contact in the first image in Fig. 5 or that there are two roads in the third image in Fig. 5, whereas the generated descriptions from SVM-D BOW do not capture this information. Indeed, we can see that this analysis is clearly reflected in Table IV where we have a similar BLEU-1 score for both the systems, while, regarding BLEU-2, BLEU-3, and BLEU-4, SVM-D CONC shows better results compared to SVM-D BOW. SVM-D CONC is also able to generate more complete descriptions than merge GRU-D [52] as it can be seen from the third image in Fig. 5 where merge GRU-D misses the cars on the road.

4) Results on RSICD Caption Dataset: The RSICD dataset is the biggest one in the RS community. It is about five times larger than the three small datasets presented so far. It is worth noting that our two SVM-based decoder IC systems are explicitly developed for those situations in which only small datasets are present. The RSICD dataset is not part of those situations considering the number of images (more than 10 000) and the number of descriptions (more than 50 000) [15]. Furthermore, this dataset has a vocabulary size of dimension 3323 words that can be translated into 3323 unique classes that becomes rather complex for an SVM multiclass classifier to deal with. For this reason, we have significantly reduced the vocabulary size of the dataset by taking only the



1. **GT**: Many people are in a yellow beach near a green ocean.
2. **Merge GRU-D:** Many green trees are in a piece of a yellow desert.
3. **SVM-D BOW:** Many people are in a piece of a green ocean.
4. **SVM-D CONC:** Many people are in a piece of a green ocean

1. **GT:** There is a playground in front of a white house.
2. **Merge GRU-D:** There are some trees around the stadium.
3. **SVM-D BOW:** A playground is surrounded by many buildings.
4. **SVM-D CONC:** A playground with a football field in it.

1. **GT:** Several planes are near a large building in an airport with several runways.
2. **Merge GRU-D:** A plane is near a terminal.
3. **SVM-D BOW:** Many planes are parked near an airport.
4. **SVM-D CONC:** Many planes are parked near a terminal in an airport.

5. **GT:** A red church is near some cars and several other buildings.
6. **Merge GRU-D**: Many buildings are in a school with a road.
7. **SVM-D BOW:** Many red buildings in a.
8. **SVM-D CONC:** Many red buildings in the middle of the land.

Fig. 6. Captioning examples of test images from the RSICD dataset. The first description corresponds to one of the ground-truth descriptions, while the second, third, and fourth descriptions are generated by Merge GRU-D, SVM-D BOW, and SVM-D CONC models, respectively. The words in red indicate errors in the generated descriptions.

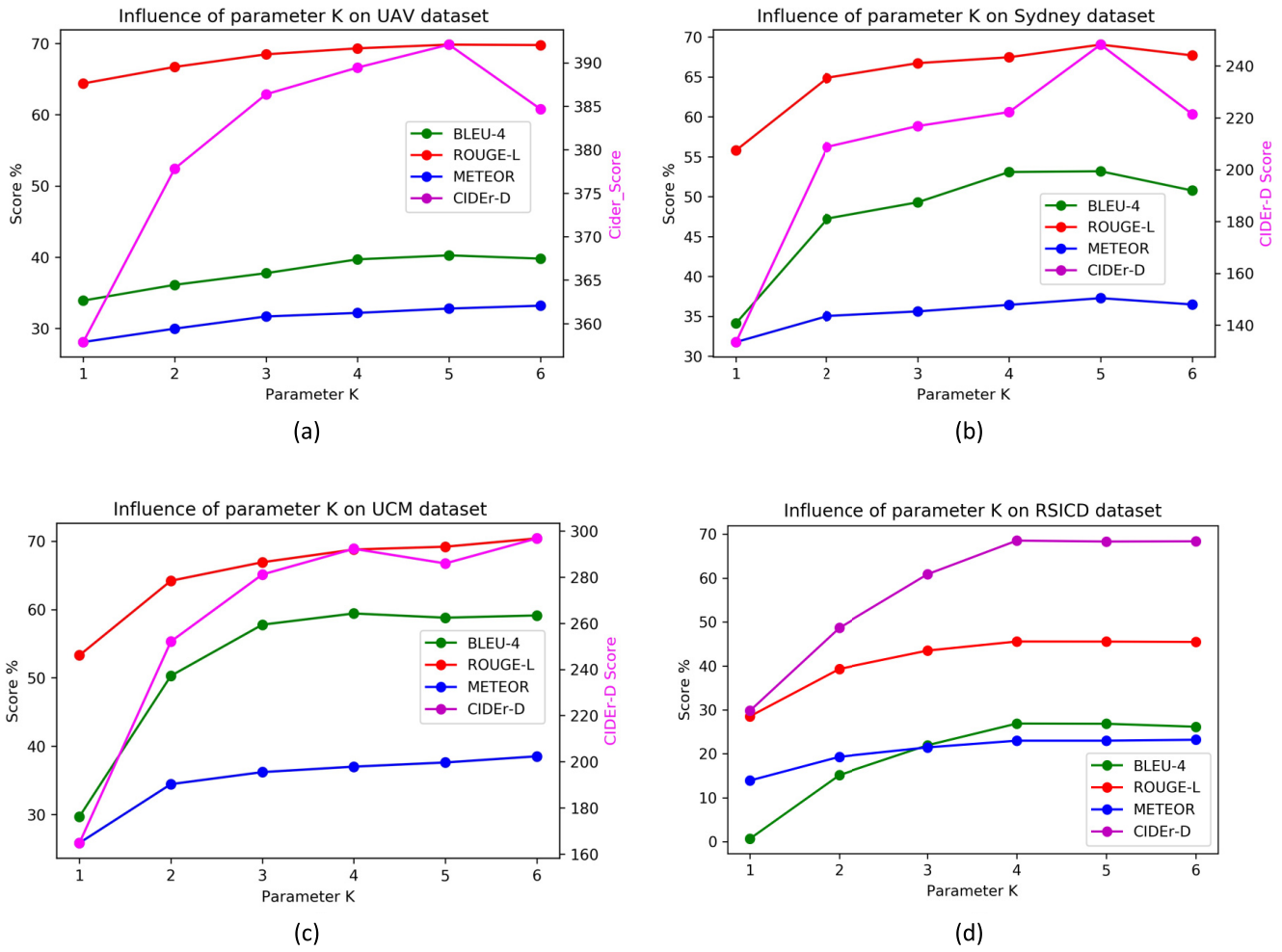


Fig. 7. Effect of parameter K of SVM-D CONC in all the four explored datasets. (a) UAV, (b) Sydney, (c) UCM, and (d) RSICD.

TABLE VI
EVALUATION SCORES (%) AND STANDARD DEVIATION IN FUNCTION OF AMOUNT OF TRAINING SAMPLES (%) USED TO TRAIN MERGE GRU-D [49] AND SVM-D CONC (OURS)

Dataset	Method	TRAINING %	Mean and standard deviation BLEU-1	Mean and standard deviation BLEU-2	Mean and standard deviation BLEU-3	Mean and standard deviation BLEU-4	Mean and standard deviation METEOR	Mean and standard deviation ROUGE-L	Mean and standard deviation CIDEr-D
UAV	Merge GRU-D	25	60.01±3.20	49.19±3.54	39.81±3.45	31.68±2.98	28.91±1.68	61.30±4.44	231.74±140.06
	SVM-D CONC	25	59.77 ±0.92	51.83 ±0.67	44.04 ±0.78	37.09 ±1.26	29.28 ±0.53	65.85 ±0.53	365.25 ± 5.7
	Merge GRU-D	50	59.27±3.46	49.05±3.22	39.99±3.13	32.06±2.85	28.78±1.47	60.89±4.21	299.48±66.07
	SVM-D CONC	50	62.33±1.5	53.91 ±1.19	45.70 ±1.05	37.80 ±0.93	30.55 ±0.67	67.56 ±0.69	378.23 ±5.07
	Merge GRU-D	100	**66.03**	55.08	44.87	35.37	31.31	66.76	368.36
	SVM-D CONC	100	65.13	**56.53**	**48.15**	**39.69**	**32.17**	**69.31**	**389.45**
Sydney	Merge GRU-D	25	69.78±2.03	59.91±1.73	52.75±1.65	46.65±1.76	33.41±1.23	62.84±2.15	195.27±15.25
	SVM-D CONC	25	73.88±0.83	64.71±0.87	57.32±0.76	50.90±0.64	35.81±1.01	65.81±1.56	213.02±12.37
	Merge GRU-D	50	70.97±1.63	61.4±1.25	53.72±1.34	47.12±1.69	34.79±1.50	64.26±1.47	197.10±9.6
	SVM-D CONC	50	75.09±2.21	66.03±2.42	58.15±2.73	51.18±2.96	36.01±0.90	67.08±1.29	215.93±13.49
	Merge GRU-D	100	73.07	63.37	56.41	49.87	33.09	63.34	193.93
	SVM-D CONC	100	**75.47**	**67.11**	**59.70**	**53.08**	**36.43**	**67.46**	**222.22**
UCM	Merge GRU-D	25	69.12±1.72	59.66±2.23	52.66±2.69	46.92±3.00	32.56±1.19	63.51±2.01	233.21±14.58
	SVM-D CONC	25	71.50±2.64	62.93±3.07	56.72±3.54	51.33±3.92	33.08±1.95	63.49±2.93	253.26±21.54
	Merge GRU-D	50	72.73±0.80	63.51±0.91	56.92±1.03	51.39±1.11	35.12±1.11	66.24±1.10	251.86±7.20
	SVM-D CONC	50	75.40±1.02	67.49±1.25	61.73±1.53	56.64±1.86	36.37±0.55	67.50±1.03	281.82±10.70
	Merge GRU-D	100	75.74	67.16	60.63	55.29	37.81	69.11	274.85
	SVM-D CONC	100	**76.53**	**69.47**	**64.17**	**59.42**	**37.02**	**68.77**	**292.28**
RSICD	Merge GRU-D	25	54.46±1.33	36.96±1.03	25.29±3.02	19.95±0.89	23.99±8.83	39.60±0.93	51.68±1.60
	SVM-D CONC	25	58.15±0.18	41.13±0.27	31.24±0.38	24.69±0.40	21.74±0.07	43.63±0.38	61.16±0.90
	Merge GRU-D	50	56.25±1.28	38.57±1.09	28.10±0.92	21.28±0.87	21.13±0.71	41.35±1.18	56.04±2.77
	SVM-D CONC	50	59.57±0.65	42.52±0.72	32.56±0.79	25.91±0.84	22.47±0.39	44.76±0.56	65.20±1.08
	Merge GRU-D	100	**60.30**	42.48	32.03	25.20	22.94	43.83	65.90
	SVM-D CONC	100	**59.99**	**43.47**	**33.55**	**26.89**	**22.99**	**45.57**	**68.54**

most 430 frequent words in our experiments. This reduction of the vocabulary size resulted in better results and also in acceptable training and inference times. We have used the same configuration also with the merge GRU-D.

In Table V, the results of each state-of-the-art method and our two SVM-D IC solutions are reported. We can see that the proposed methods outperform the simple encoder–decoder frameworks (except for mGRU embedword) and CSMLF. However, the results are lower compared with more sophisticated systems that exploit attention mechanisms. The results of SVM-D BOW and SVM-D CONC are very similar, and we can note the same behavior as on the other datasets, where, in terms of BLEU-1 and BLEU-2, SVM-D BOW shows better results, while, in terms of BLEU-3 and BLEU-4, SVM-D CONC appears the best. Furthermore, they are characterized by short training and inference times. Training and inference time results of other methods are not provided in [19]. Considering the training and inference times of other methods in the UCM dataset, we can expect that the same ratio would be applied also for the RSCID dataset. Fig. 6 depicts four examples of images from the RSCID dataset where the first description of each image is one of the reference descriptions,

whereas the second, third, and fourth are generated by merge GRU-D, SVM-D BOW, and SVM-D CONC. It comes out that the generated descriptions of our two methods are more in line with the image content compared to merge GRU-D. In particular, the description generated by merge GRU-D of the first image (Fig. 6) completely misses the semantic content of the scene.

E. Impact of Parameter K and Number of Training Samples

Since, in the previous experiments, SVM-D CONC has performed slightly better compared to SVM-D BOW on all the datasets (see Tables II–V), we will analyze further the proposed decoding approach by running a set of additional experiments on the SVM-D CONC decoder. In particular, we will focus on two aspects. The first one is the importance of the K parameter, which controls the number of SVMs to construct the cascaded decoder. The second aspect is related to the impact of the number of training samples on its generalization capability.

For the sake of comparison, we implemented merge GRU-D [52]. Its encoder is the VGG-16 pretrained on ImageNet (same as ours), and the decoder is GRU. The decoder deals only with the language part, and the image features are introduced in a subsequent layer by concatenating them with the GRU output. As in [52], the image features are mapped in an embedding space whose dimension is 128, which is the same as the word embedding layer and the GRU hidden state. The Adam optimizer [54] with the default parameters and the cross-entropy loss function is used to train the network in 50 epochs [52]. The learning rate is reduced by 10% if there are two epochs without any improvement on the validation set performance. The batch size is set to 128.

The parameter K determines the number of SVMs in the cascade that composes our SVM decoders. In particular, in SVM-D CONC, it determines the window size of the previously considered words while predicting the successive ones. We varied the parameter K from 1 to 6. Fig. 7 depicts the results obtained for each dataset. As expected, setting $K = 1$ has a drastic effect on the accuracy. The reason behind this is the fact that the generation of the subsequent words is limited to the previous word reducing to the minimum the word correlation within a sentence. On the other hand, increasing K extends the window size of the previous words that are considered to predict the subsequent ones leading to better exploitation of a context within a sentence and, in general, as a consequence to better accuracy. We also can note that, when the parameter K goes beyond 4, the improvement ceases to be significant and, in some cases, start to drop.

To assess further the generalization capability of our method, we run experiments by reducing further the number of training samples. In particular, we randomly generate five subsets of training samples composed of a half (50%) and another five subsets consisting of a quarter (25%) of the original set of training samples. The training of both SVM-D CONC and GRU-D [52] is performed on each dataset. Table VI reports the average metrics and the related standard deviation (on the five runs). In particular, our method shows more accuracy and stability compared to merge GRU-D with the exception of the Sydney dataset where 50% of training samples are considered. Furthermore, one can also observe that the drop in accuracy of our method is lower compared to the merge GRU-D in almost all the metrics. This confirms, as expected from the intrinsic generalization properties of SVM, that our method is less sensitive to the size of the training set.

IV. Conclusion

In this work, we have presented a novel RS IC system that is based on a network of SVM multiclass classifiers. The proposed IC system is part of the well-known encoder–decoder family and uses CNNs to represent the image with a set of discriminative features and a network of SVMs as the decoder (instead of RNNs) to generate the image descriptions. In particular, the last SVM multiclass classifier is rendered recurrent to model the dependence of the past words while generating the subsequent words of a sentence. In particular, the dependence on the previous words is modeled using a fixed window size of previous words (SVM-D CONC) or considering all the past words (SVM-D BOW). The former has the advantage of preserving the word order but within a fixed window size, while the latter has no constraint on window size but does not preserve the word order. The proposed system is particularly interesting in those situations characterized by the availability of only a few training samples. It exhibits very short training and inference times. Moreover, it requires the setting of just one hyperparameter, namely, the regularization parameter C. The experiments carried out on four different RS captioning datasets confirm the effectiveness of the proposed IC system, especially on small datasets. As future work, we think it worth exploring more sophisticated NLP strategies that can capture better the word dependencies without resorting to a predefined fixed window size while preserving the word order.

References

[1] G. Hoxha, F. Melgani, and B. Demir, "Toward remote sensing image retrieval under a deep image captioning perspective," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 13, pp. 4462–4475, 2020, doi: 10.1109/JSTARS.2020.3013818.

[2] Y. Yang, C. L. Teo, H. Daum, and Y. Aloimonos, "Corpus-guided sentence generation of natural images," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2011, pp. 444–454.

[3] G. Kulkarni *et al.*, "Baby talk: Understanding and generating simple image descriptions," in *Proc. CVPR*, 2011, pp. 1601–1608, doi: 10.1109/CVPR.2011.5995466.

[4] S. Li, G. Kulkarni, T. L. Berg, A. C. Berg, and Y. Choi, "Composing simple image descriptions using web-scale N-grams," in *Proc. 15th Conf. Comput. Natural Lang. Learn.*, Portland, OR, USA, Jun. 2011, pp. 220–228. Accessed: Oct. 23, 2019. [Online]. Available: https://www.aclweb.org/anthology/W11-0326

[5] J. Mao, W. Xu, Y. Yang, J. Wang, and A. L. Yuille, "Explain images with multimodal recurrent neural networks," 2014, *arXiv:1410.1090*. [Online]. Available: http://arxiv.org/abs/1410.1090

[6] A. Karpathy and L. Fei-Fei, "Deep visual-semantic alignments for generating image descriptions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3128–3137, doi: 10.1109/CVPR.2015.7298932.

[7] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and tell: A neural image caption generator," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3156–3164, doi: 10.1109/CVPR.2015.7298935.

[8] K. Xu *et al.*, "Show, attend and tell: Neural image caption generation with visual attention," 2015, *arXiv:1502.03044*. [Online]. Available: http://arxiv.org/abs/1502.03044

[9] L. Gao, K. Fan, J. Song, X. Liu, X. Xu, and H. T. Shen, "Deliberate attention networks for image captioning," in *Proc. AAAI*, Jul. 2019, vol. 33, no. 1, pp. 8320–8327, doi: 10.1609/aaai.v33i01.33018320.

[10] L. Gao, X. Li, J. Song, and H. T. Shen, "Hierarchical LSTMs with adaptive attention for visual captioning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 5, pp. 1112–1131, May 2020, doi: 10.1109/TPAMI.2019.2894139.

[11] Z. Shi and Z. Zou, "Can a machine generate humanlike language descriptions for a remote sensing image?" *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 6, pp. 3623–3634, Jun. 2017, doi: 10.1109/TGRS.2017.2677464.

[12] B. Wang, X. Lu, X. Zheng, and X. Li, "Semantic descriptions of high-resolution remote sensing images," *IEEE Geosci. Remote Sens. Lett.*, vol. 16, no. 8, pp. 1274–1278, Aug. 2019, doi: 10.1109/LGRS.2019.2893772.

[13] B. Qu, X. Li, D. Tao, and X. Lu, "Deep semantic understanding of high resolution remote sensing image," in *Proc. Int. Conf. Comput., Inf. Telecommun. Syst. (CITS)*, Jul. 2016, pp. 1–5, doi: 10.1109/CITS.2016.7546397.

[14] X. Zhang, X. Li, J. An, L. Gao, B. Hou, and C. Li, "Natural language description of remote sensing images based on deep learning," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, Jul. 2017, pp. 4798–4801, doi: 10.1109/IGARSS.2017.8128075.

[15] X. Lu, B. Wang, X. Zheng, and X. Li, "Exploring models and data for remote sensing image caption generation," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 4, pp. 2183–2195, Apr. 2018, doi: 10.1109/TGRS.2017.2776321.

[16] X. Zhang, X. Wang, X. Tang, H. Zhou, and C. Li, "Description generation for remote sensing images using attribute attention mechanism," *Remote Sens.*, vol. 11, no. 6, p. 612, 2019, doi: 10.3390/rs11060612.

[17] X. Zhang, Q. Wang, S. Chen, and X. Li, "Multi-scale cropping mechanism for remote sensing image captioning," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, Jul. 2019, pp. 10039–10042, doi: 10.1109/IGARSS.2019.8900503.

[18] W. Huang, Q. Wang, and X. Li, "Denoising-based multiscale feature fusion for remote sensing image captioning," *IEEE Geosci. Remote Sens. Lett.*, vol. 18, no. 3, pp. 436–440, Mar. 2020, doi: 10.1109/LGRS.2020.2980933.

[19] X. Lu, B. Wang, and X. Zheng, "Sound active attention framework for remote sensing image captioning," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 3, pp. 1985–2000, Mar. 2020, doi: 10.1109/TGRS.2019.2951636.

[20] B. Wang, X. Zheng, B. Qu, and X. Lu, "Retrieval topic recurrent memory network for remote sensing image captioning," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 13, pp. 256–270, 2020, doi: 10.1109/JSTARS.2019.2959208.

[21] G. Hoxha, F. Melgani, and J. Slaghenauffi, "A new CNN-RNN framework for remote sensing image captioning," in *Proc. Medit. Middle-East Geosci. Remote Sens. Symp. (M2GARSS)*, Mar. 2020, pp. 1–4, doi: 10.1109/M2GARSS47143.2020.9105191.

[22] K. Cho *et al.*, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," 2014, *arXiv:1406.1078*. [Online]. Available: http://arxiv.org/abs/1406.1078

[23] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in Neural Information Processing Systems*, vol. 27, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds. New York, NY, USA: Curran Associates, 2014, pp. 3104–3112. Accessed: Oct. 29, 2019. [Online]. Available: http://papers.nips.cc/paper/5346-sequence-to-sequence-learning-with-neural-networks.pdf

[24] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," 2014, *arXiv:1409.0473*. [Online]. Available: http://arxiv.org/abs/1409.0473

[25] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997, doi: 10.1162/neco.1997.9.8.1735.

[26] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, vol. 25, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. New York, NY, USA: Curran Associates, 2012, pp. 1097–1105. Accessed: May 16, 2019. [Online]. Available: http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf

[27] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*. [Online]. Available: http://arxiv.org/abs/1409.1556

[28] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proc. IEEE erence Comput. Vis. Pattern Recognit. (CVPR)*, Boston, MA, USA, Jun. 2015, pp. 1–9, doi: 10.1109/CVPR.2015.7298594.

[29] J. Sivic and A. Zisserman, "Video Google: A text retrieval approach to object matching in videos," in *Proc. 9th IEEE Int. Conf. Comput. Vis.*, vol. 2, Oct. 2003, pp. 1470–1477, doi: 10.1109/ICCV.2003.1238663.

[30] F. Perronnin, Y. Liu, J. Sanchez, and H. Poirier, "Large-scale image retrieval with compressed Fisher vectors," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 3384–3391.

[31] H. Jégou, F. Perronnin, M. Douze, J. Sánchez, P. Pérez, and C. Schmid, "Aggregating local image descriptors into compact codes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 9, pp. 1704–1716, Sep. 2012.

[32] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 936–944, doi: 10.1109/CVPR.2017.106.

[33] X. Ma, R. Zhao, and Z. Shi, "Multiscale methods for optical remote-sensing image captioning," *IEEE Geosci. Remote Sens. Lett.*, early access, Jul. 29, 2020, doi: 10.1109/LGRS.2020.3009243.

[34] G. Sumbul, S. Nayak, and B. Demir, "SD-RSIC: Summarization-driven deep remote sensing image captioning," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 8, pp. 6922–6934, Aug. 2020, doi: 10.1109/TGRS.2020.3031111.

[35] A. See, P. J. Liu, and C. D. Manning, "Get to the point: Summarization with pointer-generator networks," in *Proc. 55th Annu. Meeting Assoc. Comput. Linguistics*, vol. 1, Vancouver, BC, Canada, Jul. 2017, pp. 1073–1083, doi: 10.18653/v1/P17-1099.

[36] X. Li, X. Zhang, W. Huang, and Q. Wang, "Truncation cross entropy loss for remote sensing image captioning," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 6, pp. 5246–5257, Jun. 2020, doi: 10.1109/TGRS.2020.3010106.

[37] R. Zhao, Z. Shi, and Z. Zou, "High-resolution remote sensing image captioning based on structured attention," *IEEE Trans. Geosci. Remote Sens.*, early access, Apr. 12, 2021, doi: 10.1109/TGRS.2021.3070383.

[38] T.-Y. Lin *et al.*, "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 740–755, doi: 10.1007/978-3-319-10602-1_48.

[39] V. Vapnik and V. Vapnik, *Statistical Learning Theory*. New York, NY, USA: Wiley, 1998, pp. 156–160.

[40] F. Melgani and L. Bruzzone, "Classification of hyperspectral remote sensing images with support vector machines," *IEEE Trans. Geosci. Remote Sens.*, vol. 42, no. 8, pp. 1778–1790, Aug. 2004, doi: 10.1109/TGRS.2004.831865.

[41] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "CNN features Off-the-shelf: An astounding baseline for recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2014, pp. 806–813. [Online]. Available: https://www.cv-foundation.org/openaccess/content_cvpr_workshops_2014/W15/html/Razavian_CNN_Features_Off-the-Shelf_2014_CVPR_paper.html

[42] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255, doi: 10.1109/CVPR.2009.5206848.

[43] Y. Yang and S. Newsam, "Bag-of-visual-words and spatial extensions for land-use classification," in *Proc. 18th SIGSPATIAL Int. Conf. Adv. Geographic Inf. Syst.*, San Jose, CA, USA, 2010, p. 270, doi: 10.1145/1869790.1869829.

[44] F. Zhang, B. Du, and L. Zhang, "Saliency-guided unsupervised feature learning for scene classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 4, pp. 2175–2184, Apr. 2015, doi: 10.1109/TGRS.2014.2357078.

[45] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "BLEU: A method for automatic evaluation of machine translation," in *Proc. 40th Annu. Meeting Assoc. Comput. Linguistics*, Stroudsburg, PA, USA, 2002, pp. 311–318, doi: 10.3115/1073083.1073135.

[46] C.-Y. Lin, "ROUGE: A package for automatic evaluation of summaries," in *Proc. Text Summarization Branches Out*, Barcelona, Spain, Jul. 2004, pp. 74–81. Accessed: Jul. 20, 2020. [Online]. Available: https://www.aclweb.org/anthology/W04-1013

[47] S. Banerjee and A. Lavie, "METEOR: An automatic metric for MT evaluation with improved correlation with human judgments," in *Proc. ACL Workshop Intrinsic Extrinsic Eval. Measures Mach. Transl. Summarization*, Ann Arbor, MI, USA, Jun. 2005, pp. 65–72. Accessed: Jul. 20, 2020. [Online]. Available: https://www.aclweb.org/anthology/W05-0909

[48] R. Vedantam, C. L. Zitnick, and D. Parikh, "CIDEr: Consensus-based image description evaluation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Boston, MA, USA, Jun. 2015, pp. 4566–4575, doi: 10.1109/CVPR.2015.7299087.

[49] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "LIBLINEAR: A library for large linear classification," *J. Mach. Learn. Res.*, vol. 9, pp. 1871–1874, Jun. 2008.

[50] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," 2014, *arXiv:1412.3555*. [Online]. Available: http://arxiv.org/abs/1412.3555

[51] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, Doha, Qatar, Oct. 2014, pp. 1532–1543. Accessed: Jan. 7, 2019. [Online]. Available: http://www.aclweb.org/anthology/D14-1162

[52] M. Tanti, A. Gatt, and K. P. Camilleri, "Where to put the image in an image caption generator," *Natural Lang. Eng.*, vol. 24, no. 3, pp. 467–489, May 2018, doi: 10.1017/S1351324918000098.

[53] J. Aneja, A. Deshpande, and A. G. Schwing, "Convolutional image captioning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, UT, USA, Jun. 2018, pp. 5561–5570, doi: 10.1109/CVPR.2018.00583.

[54] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. ICLR*, 2015, pp. 1–15.

Genc Hoxha (Graduate Student Member, IEEE) received the B.S. degree in electronics and telecommunications engineering and the M.Sc. degree in telecommunications engineering from the University of Trento, Trento, Italy, in 2014 and 2018, respectively, where he is currently pursuing the Ph.D. degree in signal processing and pattern recognition with the ICT Doctoral School.

His research interests include machine learning and image processing with applications to remote sensing image analysis.

Farid Melgani (Fellow, IEEE) received the State Engineer degree in electronics from the University of Batna, Batna, Algeria, in 1994, the M.Sc. degree in electrical engineering from the University of Baghdad, Baghdad, Iraq, in 1999, and the Ph.D. degree in electronic and computer engineering from the University of Genoa, Genoa, Italy, in 2003.

He is currently a Full Professor of telecommunications with the Department of Information Engineering and Computer Science, University of Trento, Trento, Italy, where he teaches pattern recognition, machine learning, and digital transmission. He is also the Head of the Signal Processing and Recognition Laboratory and the Coordinator of the Doctoral School in Industrial Innovation, University of Trento. He is a coauthor of more than 240 scientific publications. His research interests are in the areas of remote sensing, signal/image processing, pattern recognition, machine learning, and computer vision.

Dr. Melgani is also an Associate Editor of the IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING, *International Journal of Remote Sensing*, and IEEE JOURNAL ON MINIATURIZATION FOR AIR AND SPACE SYSTEMS.