Using Neural Encoder-Decoder Models With Continuous Outputs for Remote Sensing Image Captioning

RITA RAMOS\textsuperscript{1} AND BRUNO MARTINS\textsuperscript{2}, (Member, IEEE)
\textsuperscript{1}\textsuperscript{1}INESC-ID, Instituto Superior Técnico, University of Lisbon, 1649-004 Lisbon, Portugal
\textsuperscript{2}LUMLIS (Lisbon ELLIS Unit), INESC-ID, Instituto Superior Técnico, University of Lisbon, 1649-004 Lisbon, Portugal
Corresponding author: Rita Ramos (ritaparadaramos@tecnico.ulisboa.pt)

This work was supported in part by the COST Action Multi3Generation (Grant CA18231), and in part by the Fundação para a Ciência e Tecnologia (FCT), namely through the FCT Project named MIMU (Grant PTDC/CCI-CIF/32607/2017), through the FCT Ph.D. Scholarship with number 2020.06106.BD, and through the INESC-ID Multi-Annual Funding from the PIDDAC Program (Grant UIDB/50021/2020).

\section*{ABSTRACT}
Remote sensing image captioning involves generating a concise textual description for an input aerial image. The task has received significant attention, and several recent proposals are based on neural encoder-decoder models. Most previous methods are trained to generate discrete outputs corresponding to word tokens that match the reference sentences word-by-word, thereby optimizing the generation locally at token-level instead of globally at sentence-level. This paper explores an alternative generation method based on continuous outputs, which generates sequences of embedding vectors instead of directly predicting word tokens. We argue that continuous output models have the potential to better capture the global semantic similarity between captions and images, e.g. by facilitating the use of loss functions matching different views of the data. This includes comparing representations for individual tokens and for the entire captions, and also comparing captions against intermediate image representations. We experimentally compare discrete versus continuous output captioning methods over the UCM and RSICD datasets, which are extensively used in the area despite some issues which we also discuss. Results show that the alternative encoder-decoder framework with continuous outputs can indeed lead to better results on the two datasets, compared to the standard approach based on discrete outputs. The proposed approach is also competitive against the state-of-the-art model in the area.

\section*{INDEX TERMS}
Deep neural networks, natural language generation, remote sensing image captioning.

\section*{I. INTRODUCTION}
The idea of interacting with remote sensing imagery through natural language has been gaining increased interest [1]–[3]. Automatically annotating remote sensing images with short text descriptions (i.e., captions) can be an effective approach to describe the contents of large image repositories concerning specific areas of the Earth. The generated captions can be used to find and group similar images through textual queries, thus rendering large remote sensing image collections as both indexable and discoverable.

The predominant approach for remote sensing image captioning is based on encoder-decoder neural network models, combining convolutional, recurrent, and neural attention components. First, a convolutional encoder is used to build representations for the different regions of an input image. These representations are used by a recurrent decoder that generates the respective caption word-by-word, at each step using neural attention to weight the contribution of the different image regions, according to relevance for the current prediction. The decoder component is trained to predict word tokens at each step, using a softmax normalization layer to produce a probability distribution over the vocabulary. A decoding algorithm (e.g., greedy decoding, a beam search procedure, or sampling-based minimum Bayes-risk decoding) can then sample from the resulting probability distribution, in order to generate the output caption.

Recently, an alternative approach for language generation was suggested by Kumar and Tsvetkov [4] in the context of automated machine translation, replacing the conventional softmax layer with an embedding layer (i.e., replacing the generation of discrete word-based outputs with the generation...
The main contributions of our study are as follows: the image encoder, fine-tuned for remote sensing images). Previous approaches (e.g., using an EfficientNet model [5] captioning method that uses continuous outputs in the decoder, trained with relatively few examples). The predominant approach relying on discrete decoder outputs optimizes models at the token level, simply matching the generated versus the reference sentences word-by-word with the standard cross-entropy loss. We instead generate continuous word vector representations, computing similarity not just between individual token embeddings, but also optimizing at sequence-level against entire captions and image representations. Image captioning leveraging continuous outputs can be particularly interesting in the domain of remote sensing imagery, for which only limited amounts of training data are available (e.g., the combined loss-functions, at different granularities, have the potential to improve results, even with small models trained with relatively few examples).

In our work, we argue that there are other possible advantages to language generation methods leveraging continuous outputs. In particular, these approaches can also facilitate the use of novel loss functions to optimize semantic similarity at different granularities: token level, sentence level, and in terms of image-vs-text similarity. The predominant approach putting, which only used greedy decoding with a token-level loss.

We released our code1 to advance and encourage future research on the task of remote sensing image captioning. Unfortunately, most previous studies in the area have not been accompanied by open-source implementations. We believe this can be important to facilitate comparisons between methods, and the development of extensions.

In summary, we propose a novel remote sensing image captioning method that uses continuous outputs in the decoder component, also featuring other differences to previous approaches (e.g., using an EfficientNet model [5] as the image encoder, fine-tuned for remote sensing images). The main contributions of our study are as follows:

- We propose a novel encoder-decoder framework for remote sensing image captioning, using continuous outputs in the decoder, together with a strong image encoder pre-trained with in-domain data, that also predicts continuous representations through natural language supervision. We argue that the use of continuous outputs can facilitate the optimization in terms of semantic similarity towards the reference captions in the training data. This can contribute to improved results, even with small models following a relatively standard CNN+RNN architecture, and trained with datasets that are small and not particularly diverse.

- We assessed the improvements that can be obtained with in-domain fine-tuning of the image encoder. In comparison, most previous studies in remote sensing image captioning have used simple ResNet models pre-trained on ImageNet photos.

- We compared novel loss functions for model training with basis on continuous outputs, going beyond token-level optimization and integrating different terms. We also advanced a novel decoding strategy that explores continuous outputs, going beyond standard beam search by also evaluating the generated sentences according to similarity towards the input image.

Both these ideas extend the work of Kumar and Tsvetkov [4] for text generation with continuous outputs, which only used greedy decoding with a token-level loss.

- We discuss limitations of the existing datasets for evaluating remote sensing image captioning methods, arguing that good performance in this domain requires methods capable of learning from relatively small datasets without much diversity.

The rest of this document is organized as follows: Section II summarizes recent research on remote sensing image captioning. Section III presents the proposed approach, discussing the particularities of the encoder and decoder components. Section IV presents the experimental evaluation of the proposed approach, detailing the implementation, describing the datasets and analyzing their limitations, describing the evaluation metrics, and finally discussing the obtained results. The paper ends with Section V, presenting our conclusions together with directions for future work.

II. RELATED WORK

The interaction between natural language and image data is nowadays a popular topic within the machine learning and computer vision communities, assuming also a significant relevance in the context of remote sensing data (e.g., supporting a more general audience in the access to these data). In particular, this interaction is an essential component of tasks such image retrieval from text queries [6], visual question answering [1], [2], or image captioning, with this latter being the focus of the present study.

Early approaches for natural image captioning employed templates together with linguistic constraints, and similar approaches have also been applied to remote sensing imagery. For instance Shi et al. [7] described a method relying on Convolutional Neural Networks (CNNs) for analysing aerial images and detecting specific objects, afterwards using this information to fill templates. This approach does not rely on examples of images paired with captions, avoiding the need for extensive training data. However, if given access to this type of data, one can instead explore the use of language models conditioned on input image representations (e.g., neural models combining convolutional encoders to represent images, and recurrent decoders to generate text). This second type of approaches is currently the most predominant [8].

1 https://github.com/RitaRamos/remote-sensing-images-caption/tree/master/src
The first public datasets pairing remote sensing images to textual descriptions, referred to as UCM and Sydney and specifically designed for supporting experiments with captioning methods, were introduced by Qu et al. [9]. These authors also showed that standard neural encoder-decoder methods, proposed for general image captioning, can also be effective for remote sensing images. More recently, Lu et al. [10] published a larger and more diverse benchmark dataset, named RSICD. In addition, Lu et al. conducted extensive experiments on the three datasets (i.e., UCM, Sydney, and RSICD), assessing the performance of neural encoder-decoder models that also used attention mechanisms.

After the work from Lu et al. [10], most subsequent studies have also used neural encoder-decoder approaches, mostly differing in terms of the attention mechanisms that are considered for weighting the contribution of the different input parts. For instance Zhang et al. [11] incorporated low-level features and high-level attributes on the attention mechanism. In the proposed approach, the encoder-decoder model could separately attend to image regions from the last convolution layer of a CNN encoder, and to attribute features extracted from the final layer of the same CNN encoder. Yuan et al. [12] introduced scale-wise and spatial-wise multi-level attention mechanisms designed to attend to image features of different scales and different spatial positions. Similarly to Zhang et al. [11], these authors also tried to leverage the semantic information present in high-level image attributes, in this case using the results from multi-label classification within a graph convolutional network, to better capture the relationships between attributes. In subsequent work, Zhang et al. [13] proposed an attention mechanism that is explicitly optimized during training, with a loss that aligns image features and word embeddings. More recently, the same authors [14] proposed a Label-Attention Mechanism (LAM) that uses word embedding vectors associated to the labels predicted for the input image.

The current state-of-the-art results for remote sensing image captioning were reported by Li et al. [3], leveraging a novel multi-level attention mechanism that uses three attention structures: one focusing on the different image regions; another focusing on the previously generated words; and a separate one deciding whether to attend to the image information or the caption information. In addition, these authors have used and released modified versions of the three aforementioned datasets, correcting a variety of errors in the textual descriptions associated to the images.

Besides tailored attention mechanisms, other previous studies on remote sensing image captioning have explored alternative paths for improving the results of standard encoder-decoder neural models. These include studies (a) exploring multi-scale feature representations [15], [16]; (b) using novel loss functions [17] or training procedures based on reinforcement learning [18], improving on the standard cross-entropy loss [17]; (c) extending and combining the set of reference captions, associated to each image, through summarization [19] or retrieval [20] approaches; or (d) using decoder components based on the Transformer architecture [18].

Table 1 summarizes results from previous studies that have used the aforementioned three datasets, through standard metrics such as BLEU, METEOR or CIDER [21]. In our work, we also assessed model performance using UCM and RSICD, thus considering the larger dataset that is currently available in the area, as well as a smaller dataset that allows us to see how performance varies as a function of the available amount of training data. It should nonetheless be noted that the values in Table 1 are not all directly comparable, due to the fact that different studies used either the original or the updated versions of the datasets from Li et al. [3], or due to small differences in the computation of the evaluation metrics (e.g., different estimates for the IDF term weight component in metrics such as CIDEr).

III. THE PROPOSED APPROACH

This section describes the proposed approach, which explores continuous outputs for language generation in the context of remote sensing image captioning. Continuous word representations (i.e., embeddings) have been widely used as inputs to NLP models, although their use for language generation outputs was only recently proposed in the context of machine-translation [4], [22]. The main motivation behind the original introduction of continuous outputs was efficiency in the presence of large vocabularies, but we instead focus on other potential advantages. Specifically, instead of only using a token-level loss based on comparing embeddings, we show that continuous outputs can also facilitate the use of loss functions that optimize semantic similarity at different granularities: token level, sentence level, and in terms of image-vs-text similarity.

In more detail, we have that the traditional approach for language generation is based on discrete outputs, using a softmax normalization layer to produce, for each output position, a probability distribution over the tokens in a vocabulary (i.e., for assigning scores to discrete tokens). The alternative generation method based on continuous outputs instead employs a final layer that produces embedding vectors, replacing the generation of discrete tokens for the generation of word embeddings, as shown in Figure 1. Thus, continuous output generation models can be optimized in respect to semantic similarity, e.g. by minimizing the distance between output embeddings and target word embeddings, as opposed to optimizing exact word-by-word matches with the reference sentences through the cross-entropy loss. We further argue that continuous outputs for language generation can facilitate model training in ways that potentiate the use of novel loss functions, e.g. comparing the generated captions at the sequence-level (instead of, or in complement to, performing comparisons at the level of individual tokens) against the ground-truth captions, or comparing the generated captions against representations for the input images [23].

We also combine the aforementioned main idea (i.e., the use of continuous outputs for language generation, leveraging
TABLE 1. Overview on previous results for remote sensing image captioning with neural methods.

| Method                        | UCM                  | Sydney                           | KSTOC                            |
|-------------------------------|----------------------|----------------------------------|----------------------------------|
| Attention-Based (8-bit) [10]   | 0.7484               | 0.5250                           | 0.3886                           |
| Attention-Base (8-bit) [10]    | 0.8157               | 0.6182                           | 0.4256                           |
| IC Attention                  | 0.8135               | 0.6352                           | 0.4173                           |
| SM Attention                  | 0.8154               | 0.6458                           | 0.4240                           |
| ML Attention + Semantic [12]  | 0.8330               | 0.6623                           | 0.4371                           |
| VQA (13)                      | 0.8192               | 0.6587                           | 0.4380                           |
| SAT (LAM-TE) [14]             | 0.8208               | 0.7229                           | 0.4880                           |
| Adagrow (LAM-TE) [14]         | 0.8570               | 0.7430                           | 0.5100                           |
| Multi-Scale Capturing [16]    | 0.5940               | 0.4290                           | —                               |
| Denoising-Based Fusion [15]   | 0.8306               | 0.6345                           | 0.3296                           |
| SD-RSIC ResNet500 [19]        | 0.7430               | 0.5180                           | 0.3580                           |
| SD-RSIC DenseNet50 [19]       | 0.7470               | 0.5180                           | 0.3750                           |
| VRTMN [20]                    | 0.8028               | 0.6395                           | 0.4258                           |
| VRTMN-SCT [18]                | —                   | —                               | —                               |
| SVM-D BOW [22]                | 0.7635               | 0.5195                           | 0.3654                           |
| VGG16-50GB [23]               | 0.8319               | 0.6596                           | 0.4456                           |
| ResNet50 [25]                 | 0.8518               | 0.6976                           | 0.4571                           |
| ResNet101 [25]                | 0.8210               | 0.6700                           | 0.4775                           |
| Multi-level attention [3]     | 0.8864               | 0.7271                           | 0.5222                           |

FIGURE 1. Image captioning models based on discrete (left) or continuous (right) outputs. The traditional generation method employs a softmax normalization layer to score discrete tokens. The alternative generation approach with basis on continuous outputs uses an embedding layer and predicts word embeddings instead.

different and complementary loss terms), together with a novel image encoding procedure relying on continuous outputs. Usually, the encoder component in image captioning models is a Convolutional Neural Network (CNN) pre-trained to predict discrete classes through a large dataset such as ImageNet (i.e., the encoder is also pre-trained through the cross-entropy loss, using a final softmax normalization layer to predict a probability distribution over the possible classes for the input images). In our work, the encoder uses a final layer that also produces a continuous embedding, corresponding to a CNN model that is instead trained to predict an embedding for the words in a reference caption associated to the input image. This training procedure for the image encoder is in line with recent studies that efficiently learn image representations from language supervision [24].

The following sections describe each of the aforementioned aspects of the proposed approach.

A. THE IMAGE ENCODER COMPONENT

Most remote sensing image captioning studies use a CNN encoder pre-trained on the ImageNet dataset, although ImageNet only contains natural images (i.e., ground-level photos) with very different characteristics from remote sensing images [12]. Additionally, the ImageNet task involves classifying the main object of the image, whereas for remote sensing image captioning there are often multiple objects of interest, that need to be considered within a single image. This mismatch between the type of input images and task/domain can make the image representations, produced with standard encoders, less effective. In our work, we fine-tuned a EfficientNet model on the remote sensing image captioning datasets, considering a fine-tuning task based on natural language supervision related to predict embeddings from associated captions.

EfficientNet models have achieved state-of-the-art accuracy on tasks involving the analysis of natural images (i.e., on ImageNet, and also on transfer learning experiments), with less parameters than previous CNNs (e.g., models such as ResNet, DenseNet, etc.). In brief, EfficientNet models combine multiple Mobile inverted Bottleneck Convolution (MBConv) blocks, similar to those used in previous CNN architectures [25], [26], extended with a squeeze-and-excitation optimization component [27]. The number of blocks in a concrete EfficientNet architecture is decided with basis on a compound scaling method that attempts to balance the width (i.e., the number of filters in convolutional layers), depth (i.e., the number of layers), and resolution (i.e., the size of the input images) of the network architecture. Additional details are given in the original paper by Tan and Le [5] and, in our experiments, we used an EfficientNet-B5 model (i.e., a CNN model with approximately 30 million parameters) pre-trained over the ImageNet dataset.

To adapt the EfficientNet model for the task of remote sensing image captioning, we specifically considered a fine-tuning task related to predicting an embedding for a reference caption associated to the input image. First, we assign each image in the training data with a target embedding built from a reference caption, by randomly sampling from the multiple reference captions associated to each image and then averaging the corresponding GloVe word embeddings. Then, the last fully-connected layer from the EfficientNet model, initially pre-trained on ImageNet, is replaced by a different fully-connected layer with dimensionality equal to 300.
(i.e., the dimensionality of the GloVe embeddings [28] used in our experiments). The resulting model is fine-tuned with the smooth L1 loss, computed between the predicted caption embedding and the target caption embedding. The smooth L1 loss, also referred to as the Huber loss, produces a linear penalty when the absolute difference between predictions and targets is high, and a quadratic penalty when the difference is close to zero. The equation for comparing two values \( x \) and \( y \) is as follows and, when comparing embedding vectors, we can take the average over the different vector dimensions.

\[
\text{smoothL1}(x, y) = \begin{cases} 
\frac{1}{2} (x - y)^2 & \text{for } |x - y| \leq 1, \\
|x - y| - \frac{1}{2} & \text{otherwise}.
\end{cases}
\]

In the previous expressions, \( W \) is a learned parameter tensor, \( d \) is the dimensionality of the image features \( V \), \( a_t \) are the attention weights for each image region, and \( c_t \) is the attended image vector (i.e., the resulting visual context vector, corresponding to a weighted average of the image features \( v_i \) for the \( K \) different regions).

For each input image, the EfficientNet encoder extracts the current image features, and the initial hidden states of the LSTM are then initialized with a global mean-pooled image feature, projected to the same dimensionality of the LSTM hidden states. At each time-step, the LSTM decoder receives as input the GloVe embedding of the current word, concatenated with the visual attention context vector. It then predicts an embedding word vector as output.

The image captioning model is trained with a novel loss function that was specifically designed to explore generation with continuous outputs. This loss function combines 3 separate terms: (a) a token-level smooth L1 loss computed between predicted and target token embeddings; (b) a sentence-level smooth L1 loss computed between a representation for the generated caption (i.e., the mean of the predicted embedding vectors) and the ground-truth caption (i.e., the mean of the target word embeddings); (c) an image-level loss comparing the generated caption against the image representation [23], corresponding to the smooth L1 loss computed between the average of the predicted embeddings, and the image representation obtained from the last layer of the encoder CNN.

\[
L_1 = \text{smoothL1}(\hat{e}_t, e_t),
\]

\[
L_2 = \text{smoothL1} \left( \frac{1}{T} \sum_{t=1}^{T} \hat{e}_t, \frac{1}{T} \sum_{t=1}^{T'} e_t \right),
\]

\[
L_3 = \text{smoothL1} \left( \frac{1}{T} \sum_{t=1}^{T} \hat{e}_t, i \right),
\]

\[
\text{loss} = w_1 L_1 + w_2 L_2 + w_3 L_3.
\]

In the previous expressions, \( \hat{e}_t \) is the predicted embedding at time-step \( t \), \( e_t \) is the corresponding target embedding, and \( i \) is the image representation. Each of the 3 separate terms (i.e., \( L_1, L_2, \) and \( L_3 \)) has a weight parameter to be adjusted, respectively corresponding to \( w_1, w_2, \) and \( w_3 \) (see Section IV-A for training details).

At inference time, we can use the greedy nearest-neighbor decoding scheme proposed by Kumar and Tsvetkov [4] for continuous outputs. In this method, a word is generated from a
predicted embedding by selecting the word of the vocabulary that has the nearest embedding, according to the smooth L1 loss (i.e., the minimum value according to the function described in Equation 1). Besides greedy decoding, we also tested an adapted beam-search procedure that runs b parallel searches through the sequences of most likely words, leveraging continuous outputs as a way to score the entire generated captions according to similarity towards the input image. In our tailored decoding procedure, at each generation step, a beam (i.e., a sequence of words generated up to a given step) is evaluated with basis on (a) the similarity of the embedding evaluated by the model at the current step and the corresponding target word embedding, and (b) the similarity between the average of the embeddings for the words generated thus far and the current embedding, evaluated against the image representation.

\[
\text{score} = \lambda \cdot \text{smoothL1}(\hat{e}_T, w_T) + (1 - \lambda) \cdot \text{smoothL1}\left(\frac{1}{T} \sum_{i=1}^{T} w_{t, i}\right). \tag{9}
\]

In the previous expression, \(\hat{e}_t\) is the predicted word embedding at time-step \(T\), \(w_T\) is the embedding for the word being evaluated as a candidate for step \(T\), \(i\) is the image representation, and \(\lambda\) is a tuning parameter controlling the contribution of the similarity towards the image.

### IV. EXPERIMENTAL EVALUATION

This section presents the experimental evaluation of the proposed method. Section IV-A presents implementation details, while Sections IV-B and IV-C respectively describe the datasets (emphasizing potential limitations) and the evaluation metrics. Finally, Section IV-D presents and discusses the obtained results, comparing our complete method against ablated model versions.

#### A. IMPLEMENTATION DETAILS

For the image encoder, we adopt an EfficientNet-B5 model pre-trained on ImageNet, as available from an open-source Pytorch implementation.\(^2\) The EfficientNet model is fine-tuned on the task of predicting embeddings for captions, using 90% of the instances in the image captioning training datasets, and with the remaining 10% being left out for adjusting early stopping (i.e., to stop training after 5 consecutive epochs without improvement). We use the Adam optimizer with an initial learning rate of 1e-4, adjusting the parameters of all the layers (i.e., each EfficientNet layer is unfrozen).

On what regards the decoder, the LSTM hidden state has a dimensionality of 512, followed by a dropout layer of 0.5 that precedes the output layer. Both the LSTM decoder and the EfficientNet encoder produce outputs with a dimensionality of 300, corresponding to the size of the pre-trained GloVe embeddings\(^2\) that are used as targets. The attention context vector has also 300 dimensions.

For training the encoder-decoder model for image captioning, we use Adam with a learning rate of 4e-4, only adjusting the parameters of the decoder component (i.e., the weights of the encoder are fixed). We use a teacher forcing strategy for model training, in which ground truth word embeddings are used as input to the LSTM, instead of using the model outputs from prior time steps as input. Early stopping and learning rate update criteria are defined with basis on a validation loss. Specifically, the learning rate is decayed after 5 consecutive epochs without improvement (with a shrink factor of 0.8), and the training is stopped if there is no improvement after 12 consecutive epochs. The different loss weighting terms (i.e., \(w_1, w_2\) and \(w_3\)) are learned with GradNorm\(^3\), i.e., an algorithm to automatically balance the contribution of different components/tasks within a loss function, that works on the basis of dynamically tuning the gradient magnitudes. We also use Adam for the GradNorm optimizer, with a learning rate of 0.025.

A baseline with the standard discrete output approach was also implemented, for comparison with our continuous output model. We tested this baseline with our fine-tuned encoder, and also with a default encoder pre-trained on the ImageNet dataset. For the baseline decoder, we use the same LSTM+attention approach, except for the final embedding output layer (i.e., in the baseline, the final fully-connected layer uses instead a softmax normalization, and training is made with the standard cross-entropy loss).

#### B. DATASETS AND THEIR LIMITATIONS

Our experiments relied on two public datasets that are commonly used in the area\(^4\), namely the UCM and the RSICD remote sensing image captioning datasets. Examples for the scenes from each dataset, together with the corresponding object categories and captions, are shown in Figure 2.

The UCM dataset is based on the UC Merced land use dataset\(^5\) for scene classification. Qu et al.\(^6\) manually added textual descriptions to each of the images, thus creating the first publicly available collection of remote sensing images paired to textual descriptions, supporting research in image captioning. This dataset contains 2100 images and 10500 captions, with 5 captions per image. Each image has 256×256 pixels, with a ground-level resolution of 0.3048m per pixel. Each image also belongs to one of the following 21 categories: agricultural, airplane, baseball diamond, beach, buildings, chaparral, dense residential, forest, freeway, golf course, harbor, intersection, medium density residential, mobile home park, overpass, parking lot, river, runway, sparse residential, storage tanks, and tennis courts\(^6\).

Along with the UCM captioning dataset, Qu et al.\(^6\) also proposed the Sidney dataset, based on another pre-existing image collection\(^7\). This dataset is much smaller than the UCM captioning dataset (which is itself already small), with only 613 images, each associated with 5 reference captions. The images have the size of 500×500 pixels, with a ground-level resolution of 0.5m per pixel. Each image belongs to

\(^{2}\)https://github.com/lukemelas/EfficientNet-PyTorch

\(^{3}\)https://github.com/lukemelas/EfficientNet-PyTorch

\(^{4}\)https://github.com/lukemelas/EfficientNet-PyTorch

\(^{5}\)https://github.com/lukemelas/EfficientNet-PyTorch

\(^{6}\)https://github.com/lukemelas/EfficientNet-PyTorch

\(^{7}\)https://github.com/lukemelas/EfficientNet-PyTorch
FIGURE 2. Examples of different images, together with the corresponding categories, for each of the two datasets used in our tests. On the right, the first image is shown together with its five reference captions.

one of the following 7 categories: airport, industrial, meadow, ocean, residential, river, and runway. Given the small size of the Sidney dataset, we decided not to use it in our experiments, although several previous studies have also considered its use – see the results on Table 1.

Lu et al. [10] claimed that the aforementioned two datasets are simply too small, at the same time also containing reference captions with a low diversity. These facts pose limitations to the proper evaluation of remote sensing image captioning models. As an alternative, the authors created the RSICD dataset, which is currently the largest publicly available dataset in the area. RSICD contains 10921 images obtained from Google Earth, Baidu Map, MapABC, and Tianditu, with 30 different categories: airport, bridge, beach, baseball field, open land, commercial, center, church, desert, dense residential, forest, farmland, industrial, mountain, medium residential, meadow, port, pond, parking, park, playground, river, railway station, resort, storage tanks, stadium, sparse residential, square, school, and viaduct. The images have 224 × 224 pixels, with different ground-level resolutions. Each image was assigned to 5 reference captions, resulting in a total of 54605 captions. A total of 24333 captions were manually written by volunteers, and the remaining captions were obtained by randomly duplicating the captions associated to each corresponding image.

The original versions of all the three public datasets for remote sensing image captioning, and especially the RSICD dataset, contained errors in the descriptions, as identified by Li et al. [3]. These authors introduced various revisions and error corrections (e.g., fixing typographical and grammatical errors), and they have released the improved versions. We therefore used the updated versions of the UCM and RSICD datasets in our experiments.

The datasets are also originally partitioning into 80% of the images for training, 10% for validation, and the remaining 10% for testing. Moreover, the captions are pre-tokenized into words, and the full sets of words occurring in each training data split are considered as the vocabulary (i.e., we did not limit the vocabulary by only keeping the most frequent tokens, as this is not necessary for good performance with the generation of continuous outputs [4]). For facilitating comparisons against previous studies, we used the original splits made available for the datasets.

Despite the error corrections introduced by Li et al. [3], our analysis of the UCM and RSICD datasets still revealed some problems in terms of correct English writing. Moreover, we note that (a) RSICD is still significantly smaller (and less diverse) than popular natural image captioning datasets (i.e., datasets of ground-level photos such as Flickr30K or MS-COCO), and that (b) improvements should be made in terms of the data splits.

We have for instance made an analysis to the training, validation, and testing splits of RSICD, discovering that word usage across the captions in the validation split is very different from that of the training and test splits, as seen in Figure 3. This can introduce problems when relying on the validation split to assess the models and choose optimal hyper-parameters (e.g., choosing stopping criteria to prevent over-fitting). The aforementioned differences can perhaps explain why most previous studies have not attempted to use early stopping based on the validation performance, instead choosing a fixed number of epochs for training.

3https://github.com/120343/modified
The diversity of the captions within RSICD is also of some concern. Although this dataset contains a high number of captions, there are several images featuring the exact same descriptions (see Figure 4), due to the way through which the dataset was built. The word vocabulary employed in the captions is also relatively small. Compared to Flickr8K (i.e., a small, although highly accredited, dataset of 8000 natural images, each paired to 5 captions), the vocabulary is much smaller (i.e., 7378 words in Flickr8K, versus 2080 words in RSICD, despite the larger number of images).

Future work in the area should address the construction of better evaluation datasets, given the aforementioned limitations. It is likely the case that conclusions made through tests with the aforementioned datasets, regarding architectural choices for captioning methods with a better performance, fail to generalize to other scenarios and concrete applications (e.g., we noticed that captioning models leveraging continuous outputs, which performed better than discrete models on our tests over RSICD – see Section IV-D – often generate many repetitions when trained on natural image datasets such as Flickr30K or MS-COCO).

C. EVALUATION METRICS

To evaluate the proposed image captioning approach, we used standard metrics in the literature for evaluating text generation methods [21], including BLEU (i.e., BLEU-1, BLEU-2, BLEU-3, and BLEU-4), ROUGE, METEOR, CIDEr and SPICE. The use of standard metrics facilitates the comparison against previous research in the area – see Table 1 for an overview on previous methods.

In brief, BLEU calculates precision between n-gram occurrences in the generated and reference captions. ROUGE considers recall instead of precision. METEOR goes beyond exact word comparisons by considering different matching levels, including synonyms. CIDEr [35] uses term frequency times inverse document frequency (TF-IDF) to compute how well the generated caption matches the consensus of the multiple reference captions. Finally, SPICE [36] measures similarity through graph representations, by parsing the generated and reference captions into semantic scene graphs.

Given the recent criticism over the use of BLEU, ROUGE, or METEOR, for evaluating text generation, we also use the recently proposed BERTScore metric [37], which has a higher correlation with human quality judgments. BERTScore computes a similarity score for each token in a candidate caption against each token in a reference caption, relying on contextual embeddings provided by BERT instead of exact token matches.

All the aforementioned metrics, with the exception of BERTScore, were calculated through the implementation in the MS-COCO caption evaluation package.5

D. EXPERIMENTAL RESULTS

Through experiments over the RSICD dataset, we first attempted to verify the effectiveness of the proposed encoder. We compared a baseline encoder-decoder model with neural attention (i.e., a model using EfficientNet pre-trained on ImageNet, together with a standard softmax decoder) against a similar model using the proposed encoder fine-tuned with in-domain data (i.e., EfficientNet fine-tuned on the remote sensing image captioning data). The results are presented at the top of Table 2. The fine-tuned encoder achieved better results, clearly outperforming the standard encoder. As expected, using a CNN encoder pre-trained on ImageNet does not result in effective image representations for our task. Fine-tuning the encoder with in-domain data is an important aspect for remote sensing image captioning.

Next, we focused on assessing the use of continuous outputs for language generation on the RSICD dataset. Table 2 (middle) shows the results obtained by our captioning model, corresponding to the fine-tuned encoder together with the continuous output decoder. We looked at the contribution of the different loss terms proposed in Section III-B. The continuous output model trained with the $L_1$ loss (i.e., optimizing the similarity of each predicted word embedding against the target embeddings) achieves a better performance on all the metrics, when compared to the conventional model based on discrete outputs, trained with the cross-entropy loss. This result demonstrates the effectiveness of optimizing semantic similarity, as opposed to exact word-by-word reference matches. Adding a sentence-level loss term (i.e., $L_1 + L_2$) further improves the performance for some of the metrics (e.g., in terms of BLEU scores), and the same is also true when adding the image-level loss term (i.e., $L_1 + L_2 + L_3$). The best results are obtained from balancing the contribution of each loss term, by learning the weights that should be considered ($L$). GradNorm [32] was used to extract static weights for the loss terms (i.e., by training the model once with GradNorm, and then using the discovered weights to re-train the model). The discovered weights correspond to

---

1. BLEU (https://github.com/tylin/coco-caption)
2. ROUGE (https://github.com/Tiiiger/rouge)
3. METEOR (https://github.com/Tiiiger/bert_score)
4. BERTScore (https://github.com/Tiiiger/bert_score)
5. MS-COCO caption evaluation package (https://github.com/tlylin/coco-caption)
TABLE 2. Results on the RSICD dataset. Top: The attention-based encoder-decoder model leveraging discrete outputs, without and with the fine-tuned encoder. Middle: The proposed encoder-decoder model based on continuous outputs, i.e., the fine-tuned encoder together with the continuous output decoder, finally combined with beam-search decoding. Bottom: The current state-of-the-art model. The best performance is shown in bold and the second-best is underlined.

| Methods                        | BLEU 1 | BLEU 2 | BLEU 3 | METEOR | CIDER  | ROUGE_L | SPICE | BERTScore |
|--------------------------------|--------|--------|--------|--------|--------|---------|-------|-----------|
| Baseline (w/o fine-tuned encoder) | 0.6355 | 0.5962 | 0.4135 | 0.3430 | 0.3100 | 1.8228  | 0.5639 | 0.3796    | 0.5896    |
| Discrete Model                 | 0.7495 | 0.6227 | 0.5294 | 0.4544 | 0.3771 | 2.5047  | 0.6619 | 0.4823    | 0.6865    |
| Continuous Model: L1           | 0.7535 | 0.6414 | 0.5499 | 0.4735 | 0.3820 | 2.6952  | 0.6753 | 0.4913    | 0.6966    |
| Continuous Model: L1 + L2      | 0.7711 | 0.6550 | 0.5610 | 0.4852 | 0.3726 | 2.6182  | 0.6684 | 0.4854    | 0.6902    |
| Continuous Model: L1 + L2 + L3  | 0.7676 | 0.6581 | 0.5692 | 0.4961 | 0.3745 | 2.6603  | 0.6712 | 0.4828    | 0.6948    |
| **L(w1 = 1.18, w2 = 1.01, w3 = 0.81)** | 0.7836 | 0.6751 | 0.5849 | 0.5107 | 0.3738 | 2.7474  | 0.6783 | 0.4854    | 0.7026    |
| **L w/ Beam Search**           | 0.7846 | 0.6794 | 0.5915 | 0.5190 | 0.3714 | 2.7777  | 0.6810 | 0.4891    | 0.7073    |
| Previous State-of-the-Art [3]  | 0.8058 | 0.6778 | 0.5866 | 0.5163 | 0.4718 | 2.7716  | 0.7257 | 0.4786    | -         |

TABLE 3. Results on the UCM dataset. Top: The encoder-decoder neural-model leveraging discrete outputs, without and with the fine-tuned encoder. Middle: The two best image captioning models based on continuous outputs. Bottom: The current state-of-the-art model. Best performance is shown in bold and the second-best is underlined.

| Methods                        | BLEU 1 | BLEU 2 | BLEU 3 | METEOR | CIDER  | ROUGE_L | SPICE | BERTScore |
|--------------------------------|--------|--------|--------|--------|--------|---------|-------|-----------|
| Baseline (w/o fine-tuned encoder) | 0.6977 | 0.6167 | 0.5562 | 0.5055 | 0.3333 | 2.3957  | 0.6360 | 0.3878    | 0.6528    |
| Discrete Model                 | 0.8138 | 0.7551 | 0.6951 | 0.6326 | 0.3963 | 3.1186  | 0.7470 | 0.4517    | 0.7098    |
| Continuous Model: L1           | 0.8377 | 0.7739 | 0.7178 | 0.6631 | 0.4444 | 3.4239  | 0.8026 | 0.4857    | 0.7967    |
| **L(w1 = 1.18, w2 = 0.91, w3 = 0.91)** | 0.8451 | 0.7731 | 0.7135 | 0.6571 | 0.4282 | 3.2031  | 0.7861 | 0.4834    | 0.8051    |
| **L w/ Beam Search**           | 0.8510 | 0.7810 | 0.7226 | 0.6666 | 0.4299 | 3.2008  | 0.7864 | 0.4538    | 0.8059    |
| Previous State-of-the-Art [3]  | 0.8864 | 0.8233 | 0.7735 | 0.7271 | 0.5222 | 3.3074  | 0.8441 | 0.5021    | -         |

the values of 1.18, 1.01, and 0.81, respectively for the $w_1$, $w_2$, and $w_3$ parameters.

The obtained results confirm that the proposed continuous output model outperforms the conventional approach based on discrete outputs, in terms of all the metrics. This includes the metrics that better correlate with human judgements, namely for CIDER with an improvement of 0.2427, SPICE with an improvement of 0.0031, and BERTScore with an improvement of 0.0161.

Besides looking at the different loss terms, we also compared a default greedy decoding approach, against the tailored beam-search procedure that was advanced at the end of Section III-B. In this case, we use a beam of 5, with 0.03 for the value of the $\lambda$ parameter in Equation 9, a minimum length of 6 words for the generated captions, and preventing the repetition of the same word more than twice. The results show slight improvements with the use of beam-search decoding, over most of the metrics.

Table 2 (bottom) also shows that the proposed approach is competitive against the current state-of-art, namely the multi-level attention model of Li et al. [3]. Our model achieves slightly higher scores in terms of several metrics (e.g., BLEU-4 or SPICE), and the ideas advanced in our work can also be combined with more advanced neural attention mechanisms, such as those from Li et al. [3].

In terms of qualitative results, some captions generated by the different models are shown in Appendix A. In these examples, the first two models based on discrete outputs (without and with the fine-tuned encoder, respectively) either omitted or included incorrect information, whereas the proposed captioning model (i.e., trained with our loss function $L$) was able to provide textual descriptions that better match the semantic content of the input images. Indeed, previous studies [11], [18] on the area have reported problems in terms of generating content that is not in the images, due to the presence of highly frequent words and n-grams during training. There is, for instance, a tendency in generating words related to trees (e.g., generating expressions like several green trees) for parking images on the test set, independently of whether there are or not trees in the image [11], as can be seen in the first example. Our model was the only one to avoid this error, in most cases correctly generating information that was present in the images. In the second image, the first two models based on discrete outputs again produced the error of mentioning several trees, whereas our model was able to identify the main aspects of the image by describing the building and referring the river. In fact, the proposed model generated the most similar description to the ground-truth references (i.e., a large white building is near a river). We have also observed cases where the approach based on discrete outputs completely failed to describe the contents of the images, while our model was able to succeed, as it can be seen in the third example. Our model referred the square, which is the most relevant element of the image, whereas the approaches based on discrete outputs mis-recognized a football field or a resort. The last image shows an example for which our beam-search decoding procedure helped the proposed captioning model to avoid repeating the word white. Overall, as also suggested by the quantitative results, the qualitative examples shown in Appendix A further confirm that our model is better at capturing the semantic contents of the input images.

On what regards experiments with the UCM dataset, which is much smaller than RSICD, the results also confirmed the effectiveness of fine-tuning the encoder with in-domain data (see Table 3, on top). The proposed encoder, fine-tuned on
TABLE 4. Examples of captions generated by the different models. The red color is used for emphasising incorrect predictions.

| Image | Generated Captions |
|-------|--------------------|
| ![Image](image1.jpg) | **Baseline:** Many cars are parked in a parking lot near several green trees.  
**Discrete Model:** Many cars are parked in a parking lot with several green trees.  
**Continuous Model $L_1$:** Many cars are parked in a parking lot with several green trees.  
**Continuous Model $L_1 + L_2$:** Many cars are parked in a parking lot near many green trees.  
**Continuous Model $L_1 + L_2 + L_3$:** Many cars are parked in a parking lot near many green trees.  
**Continuous Model $L$:** Many cars are parked in a parking lot.  
**Continuous Model $L$ w/ Beam Search:** Many cars are parked in a parking lot. |
| ![Image](image2.jpg) | **Baseline:** A large number of trees were planted around the factory.  
**Discrete Model:** A white building is near a road with several green trees.  
**Continuous Model $L_1$:** A white building is near a road and some green trees.  
**Continuous Model $L_1 + L_2$:** A green trees are near a circle building.  
**Continuous Model $L_1 + L_2 + L_3$:** A large building is near a road.  
**Continuous Model $L$:** A white building is near a river.  
**Continuous Model $L$ w/ Beam Search:** A white building is near a river. |
| ![Image](image3.jpg) | **Baseline:** Many buildings and some green trees are around a playground with a football field in it.  
**Discrete Model:** Many green trees and several buildings are in a resort.  
**Continuous Model $L_1$:** Many green trees are around a square with a playground.  
**Continuous Model $L_1 + L_2$:** Many green trees are around a square.  
**Continuous Model $L_1 + L_2 + L_3$:** Many green trees are around a square.  
**Continuous Model $L$:** Many green trees are around a square.  
**Continuous Model $L$ w/ Beam Search:** Many green trees are around a square. |
| ![Image](image4.jpg) | **Baseline:** Many green trees and a parking lot are around a square with green meadows  
**Discrete Model:** A white center building is surrounded by some green trees and several buildings.  
**Continuous Model $L_1$:** Some green trees are around a large white building.  
**Continuous Model $L_1 + L_2$:** Some green trees are around a white field.  
**Continuous Model $L_1 + L_2 + L_3$:** Some green trees are around a large white building.  
**Continuous Model $L$:** Many green trees are around a white building.  
**Continuous Model $L$ w/ Beam Search:** Many green trees are around a large white building. |

the UCM dataset, outperforms by a large margin the baseline with the conventional encoder pre-trained on ImageNet.

Additionally, we evaluated generation with continuous output representations also on the UCM dataset. Table 3 (middle) presents the obtained results for our best two captioning models, namely one trained with the $L_1$ term and the another with the complete $L$ loss, containing the weights discovered through GradNorm. Both these continuous-output models are notably better than the ones using discrete outputs, regarding all the evaluation metrics. The difference between continuous
and discrete approaches is more pronounced in the UCM dataset than in the RSICD dataset, noticing for instance an improvement of 0.0953 in terms of BERTScore. The tailored beam search procedure could again further improve results, in this case with hyper-parameters corresponding to the values of 3 for the number of beams, $\lambda$ equal to 0.03, a minimum length of 4 words, and preventing the repetition of the same word more than twice.

The proposed approach also achieved comparable results to the current state-of-art model [3] on the UCM dataset, e.g. with the model that used the $L_1$ loss alone achieving the highest CIDER score (see the bottom row of Table 3).

V. CONCLUSION AND FUTURE WORK

This paper presented a novel encoder-decoder model for remote sensing image captioning. We specifically explored the use of continuous output representations for language generation, replacing the generation of discrete tokens in favour of word embeddings. A new loss was proposed, optimizing semantic similarity also at the sequence-level, contrasting to the standard word-level cross-entropy loss. Our encoder component also relies on continuous outputs, and we fine-tuned this part of the model to predict the mean word embedding vectors of a corresponding image caption, rather than just using a CNN model pre-trained to predict discrete image classes on the ImageNet dataset.

Experimental results confirmed the effectiveness of the proposed encoder and decoder components. Fine-tuning the encoder on remote sensing images lead to significant performance improvements. In comparison with the predominant generation method, the alternative based on continuous outputs could better capture the global semantic similarity between captions and images. The overall results are also comparable with the current state-of-art, and we notice that the ideas advanced in the paper can also be combined with other recent advancements in the area.

Despite the interesting results, there are also many possible ideas for future work. In particular, a better attention mechanism should be considered, taking inspiration on recent alternative approaches for remote sensing image captioning. Our model uses a simple scaled dot-product attention, whereas various previous studies have proposed attention methods better tailored for remote sensing images (e.g., assessing visual information at different scales), including the multi-level attention model that is the current state-of-the-art [3].

The proposed loss function uses averaged embedding vectors to impose global sequence-level guidance, comparing representations for the entire ground-truth versus the generated captions through averages of token embeddings. Future work can perhaps consider alternative formulations for this sentence-level component of the loss function, namely with approaches based on optimal transport [38] between sequences of embedding vectors. The word-level component of the proposed loss function, based on a smooth $L_1$ loss between ground-truth and predicted embeddings, can perhaps also be improved by considering adaptations of loss functions proposed for structured prediction [39]. Moreover, instead of considering teacher forcing for model training, the use of continuous outputs can also facilitate the design of approaches that reduce the exposure bias [40] (e.g., we can perhaps interpolate between ground-truth and generated embeddings, instead of using a standard scheduled sampling procedure).

Finally, the convolutional and recurrent components that are currently being used as the encoder and decoder, respectively, can in the future perhaps also be replaced by new components based on the Transformer architecture, given that these models are currently achieving state-of-the-art results on a variety of language processing and computer vision tasks [41]–[45]. Still, it should be noted that Transformer models can be more difficult to train. They typically require large amounts of training data, whereas the existing datasets for remote sensing image captioning are comparatively small and not particularly diverse - see the discussion in Section IV-B. Moreover, most previous Transformers for image captioning rely on image regions extracted with a R-CNN model, whereas for our case we would ideally like to process the raw images directly [45], without requiring a separate model for detecting relevant regions. These and other issues (i.e., the need for combining static embeddings for our continuous outputs, together with some form of positional encoding within an auto-regressive Transformer decoder) make the combination of our main idea with Transformers non-trivial, although deserving of additional research.

ACKNOWLEDGMENT

The authors would like to thank NVIDIA Corporation with the donation of the two Titan Xp GPUs used in their experiments.

APPENDIX

QUALITATIVE RESULTS

Table 4 presents examples for the captions that can be generated with the models that were considered in our tests, for four different images from the RSICD dataset.

REFERENCES

[1] S. Lobry, D. Marcos, J. Murray, and D. Tuia, “RSVQA: Visual question answering for remote sensing data,” IEEE Trans. Geosci. Remote Sens., vol. 58, no. 12, pp. 8555–8566, Dec. 2020.
[2] X. Zheng, B. Wang, X. Du, and X. Lu, “Mutual attention inception network for remote sensing visual question answering,” IEEE Trans. Geosci. Remote Sens., vol. 60, pp. 1–14, 2022.
[3] Y. Li, S. Fang, L. Jiao, R. Liu, and R. Shang, “A multi-level attention model for remote sensing image captions,” Remote Sens., vol. 12, no. 6, p. 939, Mar. 2020.
[4] S. Kumar and Y. Tsivitkov, “Von Mises–Fisher loss for training sequence to sequence models with continuous outputs,” 2018, arXiv:1812.04616.
[5] M. Tan and Q. Le, “EfficientNet: Rethinking model scaling for convolutional neural networks,” in Proc. Int. Conf. Mach. Learn., 2019, pp. 6105–6114.
[6] Q. Cheng, Y. Zhou, P. Fu, Y. Xu, and L. Zhang, “A deep semantic alignment network for the cross-modal image-text retrieval in remote sensing,” IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens., vol. 14, pp. 4284–4297, 2021.
W. Huang, Q. Wang, and X. Li, “Denoising-based multiscale feature fusion,” IEEE Trans. Geosci. Remote Sens., vol. 55, no. 6, pp. 3632–3634, Jun. 2017.

M. Stefanini, M. Cornia, L. Baraldi, S. Ciscianelli, G. Fiameni, and R. Cucchiara, “From show to tell: A survey on deep learning-based image captioning,” 2021, arXiv:2107.06912.

B. Qu, X. Li, D. Tao, and X. Lu, “Deep semantic understanding of high resolution remote sensing image,” in Proc. IEEE Int. Conf. Compat., Inf. Telecommun. Syst. (CITS), Jul. 2016, pp. 1–5.

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.

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, Mar. 2019.

Z. Yuan, X. Li, and Q. Wang, “Exploring multi-level attention and semantic relationship for remote sensing image captioning,” IEEE Access, vol. 8, pp. 2608–2620, 2020.

Z. Zhang, W. Zhang, W. Diao, M. Yan, X. Gao, and X. Sun, “VAA: Visual aligning attention model for remote sensing image captioning,” IEEE Access, vol. 7, pp. 137355–137364, 2019.

Z. Zhang, W. Diao, W. Zhang, M. Yan, X. Gao, and X. Sun, “LAM: Remote sensing image captioning with label-attention mechanism,” Remote Sens., no. 11, vol. 20, p. 2349, Oct. 2019.

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. 2021.

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.

X. Li, X. Zhang, W. Huang, and Q. Wang, “Fraction cross entropy loss for remote sensing image captioning,” IEEE Trans. Geosci. Remote Sens., vol. 59, no. 6, pp. 5246–5257, Jun. 2021.

X. Shen, B. Liu, Y. Zhou, J. Zhao, and M. Liu, “Remote sensing image captioning via variational autoencoder and reinforcement learning,” Knowl.-Based Syst., vol. 203, Sep. 2020, Art. no. 105920.

G. Sumbul, S. Nayak, and B. Demir, “SD-RSIC: Summarization driven deep remote sensing image captioning,” 2020, arXiv:2006.08432.

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.

A. Celikyilmaz, E. Clark, and J. Gao, “Evaluation of text generation: A survey,” 2020, arXiv:2006.14799.

C. Y. Park and Y. Tsvetkov, “Learning to generate word- and phrase-embeddings for efficient phrase-based neural machine translation,” in Proc. 3rd Workshop Neural Genet. Transl., 2019, pp. 241–248.

M. Nikolaus, M. Abdou, M. Lamm, R. Aralikatte, and D. Elliott, “Compositional generalization in image captioning,” in Proc. 23rd Conf. Comput. Natural Lang. Learn. (CoNLL), 2019, pp. 1–12.

A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, “Learning transferable visual models from natural language supervision,” 2021, arXiv:2103.00020.

M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, “MobileNetV2: Inverted residuals and linear bottlenecks,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 4510–4520.

M. Tan, B. Chen, R. Pang, V. Vasudevan, M. Sandler, A. Howard, and Q. V. Le, “MnasNet: Platform-aware neural architecture search for mobile,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 2820–2828.

J. Hu, L. Shen, and G. Sun, “Squeeze-and-excitation networks,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 7132–7141.

J. Pennington, R. Socher, and C. Manning, “Glove: Global vectors for word representation,” in Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP), 2014, pp. 1532–1534.

A. Buslaev, V. I. Iglivotik, E. Khvedchenya, A. Parinov, M. Druzhinin, and A. A. Kalinin, “Albumentations: Fast and flexible image augmentations,” Information, vol. 11, no. 2, p. 125, 2020.

S. Hochreiter and J. Schmidhuber, “Long short-term memory,” Neural Comput., vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

RITA RAMOS was born in Lisbon, Portugal, in 1994. She received the M.Sc. degree in computer science from the Instituto Superior Técnico, University of Lisbon, in 2019, where she is currently pursuing the Ph.D. degree in computer science.

In 2019, she also joined the Human Language Technologies Laboratory, INESC-ID. Her research interests include multimodal machine learning and natural language generation.

BRUNO MARTINS (Member, IEEE) was born in Lisbon, Portugal, in 1979. He received the Ph.D. degree in computer science from the University of Lisbon, in 2009.

He is currently an Associate Professor with the Instituto Superior Técnico, University of Lisbon, a Researcher with the Human Language Technologies Laboratory, INESC-ID, and a member of the Lisbon ELLIS Unit (LUMLIS). He works on problems related to the general areas of information retrieval, text mining, and the geographical information sciences. He has been involved in several research projects related to geospatial aspects in information access and retrieval, and he has accumulated a significant expertise in addressing challenges at the intersection of information retrieval and the geographical information sciences (i.e., in an area that is often referred to as geographical information retrieval).